

**IROS'13**

**PPNIV'13**

**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# **IROS13 5<sup>th</sup> International workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**Full Day Workshop**  
November 3rd, 2013 Tokyo, Japan

<http://ppniv13.irccyn.ec-nantes.fr/>

## **Organizers**

**Pr Philippe Bonnifait (HEUDIASYC, France),  
Pr Christian Laugier (INRIA, France),  
Pr Philippe Martinet (IRCCYN, France),  
Pr Urbano Nunes (ISR, Portugal)  
Pr Christoph Stiller (KIT, Germany)**

## **Contact**

Professor Philippe Martinet  
Ecole Centrale de Nantes,  
IRCCYN-CNRS Laboratory  
1 rue de la Noë, BP 92101, 44321 Nantes Cedex - FRANCE  
Phone: +33 240 376 975, Sec : +33 240 376 900, Fax : +33 240 376 6934  
Email: [Philippe.Martinet@irccyn.ec-nantes.fr](mailto:Philippe.Martinet@irccyn.ec-nantes.fr)  
Home page: <http://www.irccyn.ec-nantes.fr/~martinet>

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



IROIS'13

PPNIV'13

5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2013 IEEE/RSJ International Conference on Intelligent Robots and Systems

## Foreword

The purpose of this workshop is to discuss topics related to the challenging problems of autonomous navigation and of driving assistance in open and dynamic environments. Technologies related to application fields such as unmanned outdoor vehicles or intelligent road vehicles will be considered from both the theoretical and technological point of views. Several research questions located on the cutting edge of the state of the art will be addressed. Among the many application areas that robotics is addressing, transportation of people and goods seem to be a domain that will dramatically benefit from intelligent automation. Fully automatic driving is emerging as the approach to dramatically improve efficiency while at the same time leading to the goal of zero fatalities. This workshop will address robotics technologies, which are at the very core of this major shift in the automobile paradigm. Technologies related to this area, such as autonomous outdoor vehicles, achievements, challenges and open questions would be presented. Main topics include: Road scene understanding, Lane detection and lane keeping, Pedestrian and vehicle detection, Detection, tracking and classification, Feature extraction and feature selection, Cooperative techniques, Collision prediction and avoidance, Advanced driver assistance systems, Environment perception, vehicle localization and autonomous navigation, Real-time perception and sensor fusion, SLAM in dynamic environments, Mapping and maps for navigation, Real-time motion planning in dynamic environments, 3D Modeling and reconstruction, Human-Robot Interaction, Behavior modeling and learning, Robust sensor-based 3D reconstruction, Modeling and Control of mobile robot, Multi-agent based architectures, Cooperative unmanned vehicles (not restricted to ground transportation), Multi autonomous vehicles studies, models, techniques and simulations.

Previously, several workshops were organized in the near same field. The 1st edition [PPNIV'07](#) of this workshop was held in Roma during ICRA'07 (around 60 attendees), the second [PPNIV'08](#) was in Nice during IROS'08 (more than 90 registered people), the third [PPNIV'09](#) was in Saint-Louis (around 70 attendees) during IROS'09, and the fourth edition [PPNIV'12](#) was in Vilamoura (over 95 attendees) during IROS'12. In parallel, we have also organized [SNOODE'07](#) in San Diego during IROS'07 (around 80 attendees), [SNOODE'09](#) in Kobe during ICRA'09 (around 70 attendees), and [RITS'10](#) in Anchorage during ICRA'10 (around 35 attendees), and the last one [PNAVHE11](#) in San Francisco during the last IROS11 (around 50 attendees).

This workshop is composed with 4 invited talks and 18 selected papers (8 selected for oral presentation and 10 selected for interactive session. Five sessions have been organized:

- Session I: Localization & mapping
- Session II: Perception
- Session III: Interactive session
- Session IV: Navigation, Control, Planning
- Session V: Situation awareness & Risk Assessment

Intended Audience concerns researchers and PhD students interested in mobile robotics, motion and action planning, robust perception, sensor fusion, SLAM, autonomous vehicles, human-robot interaction, and intelligent transportation systems. Some peoples from the mobile robot industry and car industry are also welcome.

This workshop is made in relation with IEEE RAS: RAS Technical Committee on “Autonomous Ground Vehicles and Intelligent Transportation Systems” (<http://tab.ieee-ras.org/>).

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



## Session I

### Localization & Mapping

- **Keynote speaker: Martin Adams (Universidad de Chile, Santiago, Chile)**  
**Title: New Concepts in Robotic Mapping: PHD Filter SLAM**
- **Title: Large-Scale Dense 3D Reconstruction from Stereo Imagery**  
**Authors: Pablo F. Alcantarilla, Chris Beall, Frank Dellaert**
- **Title: Generation of Accurate Lane-Level Maps from Coarse Prior Maps and Lidar**  
**Authors: Avdhut Joshi, Michael R. James**

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



2013 IEEE/RSJ International Conference on Intelligent Robots and Systems

## Session I

Keynote speaker: **Martin Adams**  
(Universidad de Chile, Santiago, Chile)

### **New Concepts in Robotic Mapping: PHD Filter SLAM**

**Abstract :** Applications for autonomous robots have long been identified in challenging environments including built-up areas, mines, disaster scenes, underwater and in the air. Robust solutions to autonomous navigation remain a key enabling issue behind any realistic success in these areas. Arguably, the most successful robot navigation algorithms to-date, have been derived from a probabilistic perspective, which takes into account vehicle motion and terrain uncertainty as well as sensor noise. Over the past decades, a great deal of interest in the estimation of an autonomous robot's location state, and that of its surroundings, known as Simultaneous Localisation And Map building (SLAM), has been evident. This presentation will explain recent advances in the representations of robotic measurements and the map itself, and their consequences on the robustness of SLAM. Fundamentally, the concept of a set based measurement and map state representation allows all of the measurement information, spatial and detection, to be incorporated into joint Bayesian SLAM frameworks. Representing measurements and the map state as sets, rather than the traditionally adopted vectors, is not merely a triviality of notation. It will be demonstrated that a set based framework circumvents the necessity for any fragile data association and map management heuristics, which are necessary, and often the cause of failure, in vector based solutions. Implementation details of the Bayesian set based estimator - the Probability Hypothesis Density (PHD) Filter, and its application to SLAM will be the focus of the presentation. Experimental results, demonstrating SLAM with laser, radar and vision based sensors in urban and marine environments will be demonstrated. Comparisons of PHD Filter based SLAM and state of the art vector based implementations will demonstrate the robustness of the former to the realistic situations of sensor false alarms, missed detections and clutter.

**Biography:** Martin Adams

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



**CMRSP**  
Center for Multidisciplinary Research on Signal Processing

**amtc**  
ADVANCED MINING TECHNOLOGY CENTER

# New Concepts in Robotic Mapping: PHD Filter SLAM

Martin Adams

Dept. Electrical Engineering, CMRSP, AMTC  
University of Chile (martin@ing.uchile.cl)



# Presentation Outline

## 1. What's in a Measurement:

- Landmark Existence and Spatial Uncertainty
- Why Radar?

## 2. Simultaneous Localisation & Map Building (SLAM).

- A Random Finite Set (RFS) Approach.
- PHD SLAM – Implementation.

## 3. Comparison of Vector Based SLAM (MH-FastSLAM) and PHD-SLAM – Results.

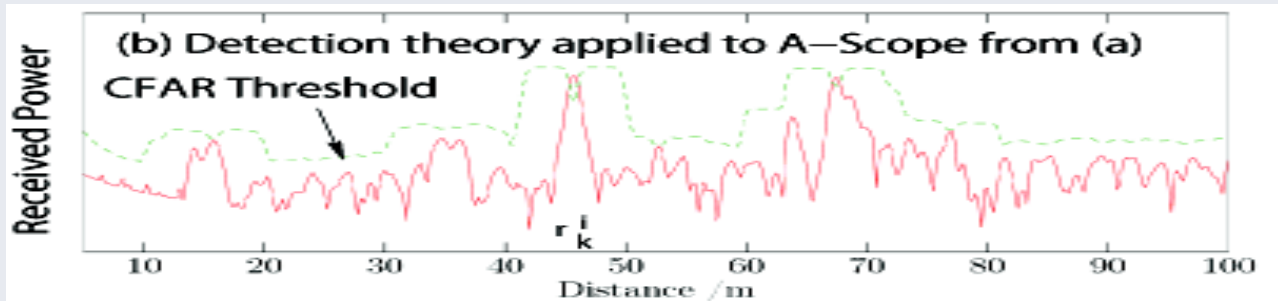
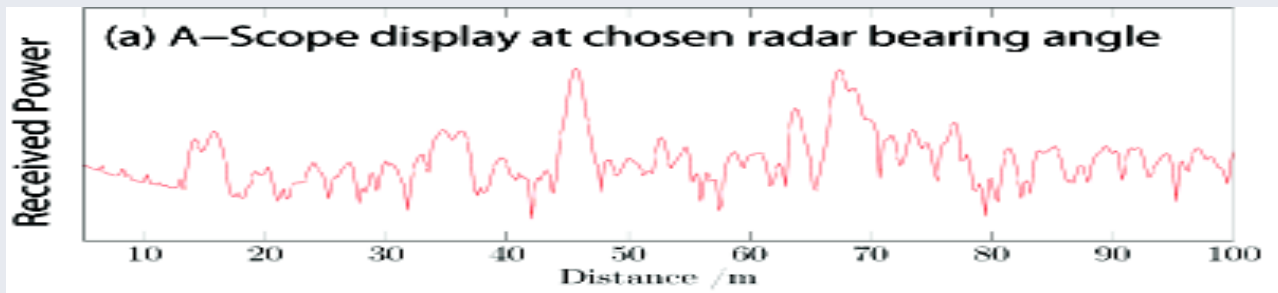
# Sensing the Environment

## Clearpath Robotic Skid Steer Platform

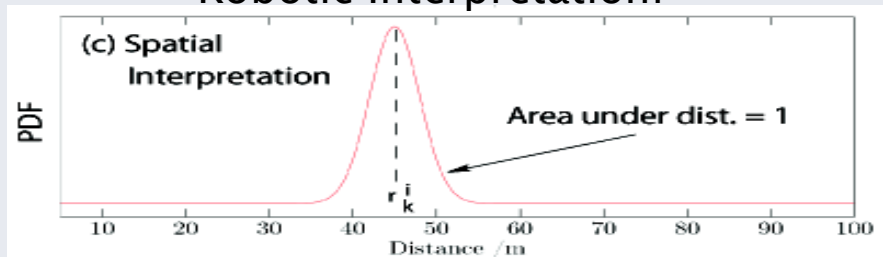


- Acumine Radar 360 deg. scanning unit, 94GHz FMCW
- Sick LD-LRS1000 Scanning LRF
- Microsoft Kinect camera system

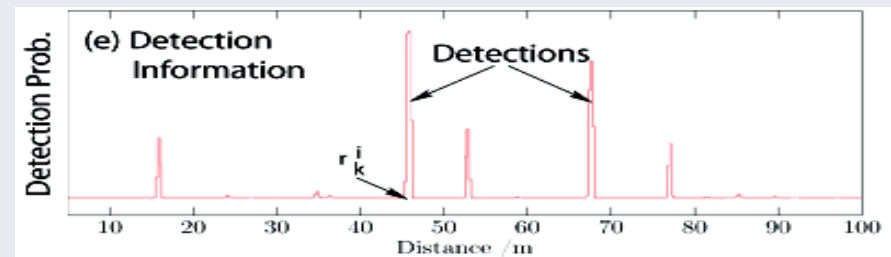
# What's in a Measurement?



Robotic Interpretation:



Radar Interpretation:



Result:

- Detection decision at range  $r_k^i$
- A-priori range uncertainty assumed/known  $(\sigma_k^2)^i$
- Subtly assumes unity detection probability  $P_D = 1$

- Multiple detection hypotheses  $H_1(r(q))$
- Associated probabilities of detection  $P_D$
- Associated probability of false alarm  $P_{fa}$

# What's in a Measurement?

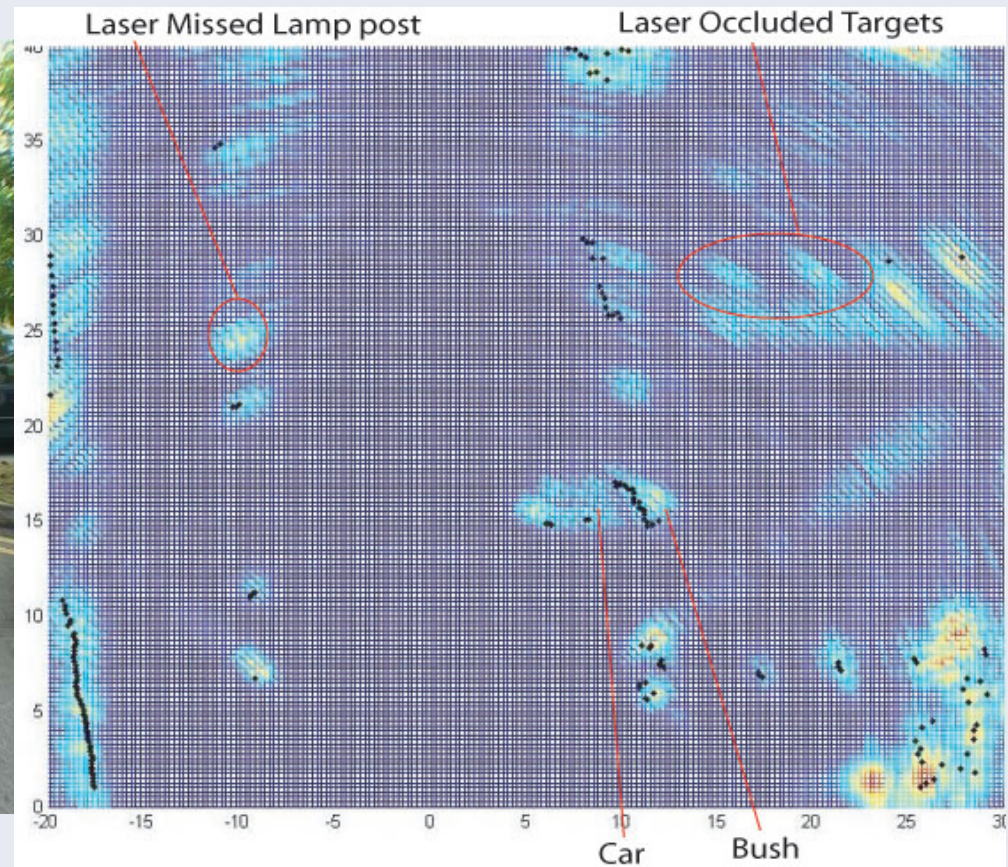
- In reality – Probability of Detection less than unity, but may not be known.
- However, landmark/feature measurements in SLAM result from a feature detection algorithm.
- Principled algorithms provide estimates of  $P_D$  and  $P_{fa}$ , or they can be estimated a-priori (e.g. RANSAC).
- **Ideal scenario: Represent all detection hypotheses in terms of their:**

$r_k^i, (\sigma_k^2)^i, P_D(r_k^i)$  and  $P_{fa}$  (i.e. range, spatial uncertainty, detection uncertainty and false alarm probability).



# Radar Based Projects: A\*Star – Radar vs. Ladar

- ✓ **Wider beam width**
- ✓ **Foliage penetration**



# Representing Maps

*Q: Why do we even care about error in the number of landmarks?*

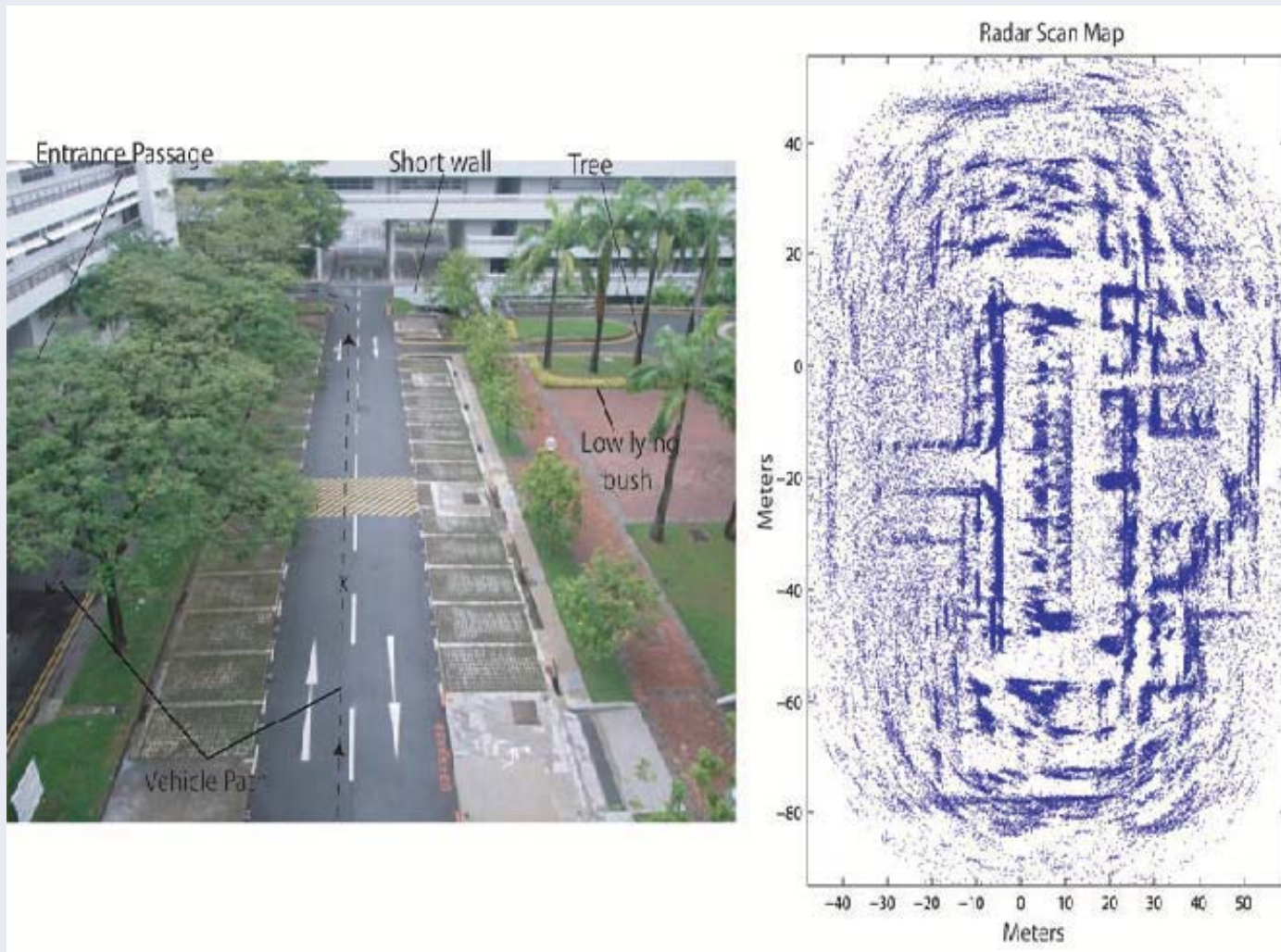
A:



Catastrophic consequences in applications such as search & rescue, obstacle avoidance, UAV mission...



# Importance of $P_{fa}$ – False Alarms



Radar detections registered to ground truth location.



# Radar Based Projects: A\*Star – Radar vs. Ladar

Video: Raw\_Data\_Display.avi

# Presentation Outline

## 1. What's in a Measurement:

- Landmark Existence and Spatial Uncertainty
- Why Radar?

## 2. Simultaneous Localisation & Map Building (SLAM).

- A Random Finite Set (RFS) Approach.
- PHD SLAM – Implementation.

## 3. Comparison of Vector Based SLAM (MH-FastSLAM) and PHD-SLAM – Results.

# SLAM Fundamentals

- In an unknown environment – robot & feature positions *must* be estimated simultaneously – SLAM.
- SLAM is a probabilistic algorithm

$$p(x_t, m \mid z_{1:t}, u_{1:t})$$

$x_t$  = State of the robot at time  $t$

$m$  = Map of the environment

$z_{1:t}$  = Sensor inputs from time 1 to  $t$

$u_{1:t}$  = Control inputs from time 1 to  $t$

- Update distribution estimate with Bayes theorem.

# SLAM: Approximate Particle Solutions – FastSLAM

A Factorised Solution to SLAM (FastSLAM):

Define joint vehicle *trajectory* & map vector state:  $\zeta_{0:k} = [X_{0:k}, M_k]$

The posterior distribution on this modified, joint state could then be factorized as,

$$p_{k|k}(\zeta_{0:k} | Z_{0:k}, U_{0:k-1}, X_0) = p_{k|k}(X_{0:k} | Z_{0:k}, U_{0:k-1}, X_0) \prod_{l=1}^m p_{k|k}(m_l | X_{0:k}, Z_{0:k}, U_{0:k-1}, X_0). \quad (4.47)$$

Particles represent trajectory distribution:

$$p_{k|k}(X_{0:k} | Z_{0:k}, U_{0:k-1}, X_0) \approx \sum_{i=1}^N w_k^{(i)} \delta(X_{0:k} - \hat{X}_{0:k|k}^{*(i)}).$$

each with their own EKF map estimate.

# SLAM: Approximate Particle Solutions – FastSLAM

Finding the particle weights:

Each particle receives weight related to how well the measurements (sensor scan), recorded from the true pose, when superimposed onto each particle, match the expected measurements.

This is:  $w_k^{(i)} \propto g_k(Z_k | X_{0:k}, Z_{0:k-1})$

Requires usual (fragile) feature association and management routines.

Particle resampling then takes place, based on the particle weights.

# SLAM: Approximate Particle Solutions – FastSLAM

Finding the particle weights:

Each particle receives weight related to how well the measurements (sensor scan), recorded from the true pose, when superimposed onto each particle, match the expected measurements.

This is:  $w_k^{(i)} \propto g_k(Z_k | X_{0:k}, Z_{0:k-1})$

Requires usual (fragile) feature association and management routines.

Particle resampling then takes place, based on the particle weights.

Highest weight particle chosen as estimated trajectory & its map as estimated map (MAP estimate).

# SLAM: Multi-Hypothesis (MH) FastSLAM

Multi-Hypothesis FastSLAM:

For each trajectory particle, multiple feature to detection associations are possible.



# SLAM: Multi-Hypothesis (MH) FastSLAM

Multi-Hypothesis FastSLAM:

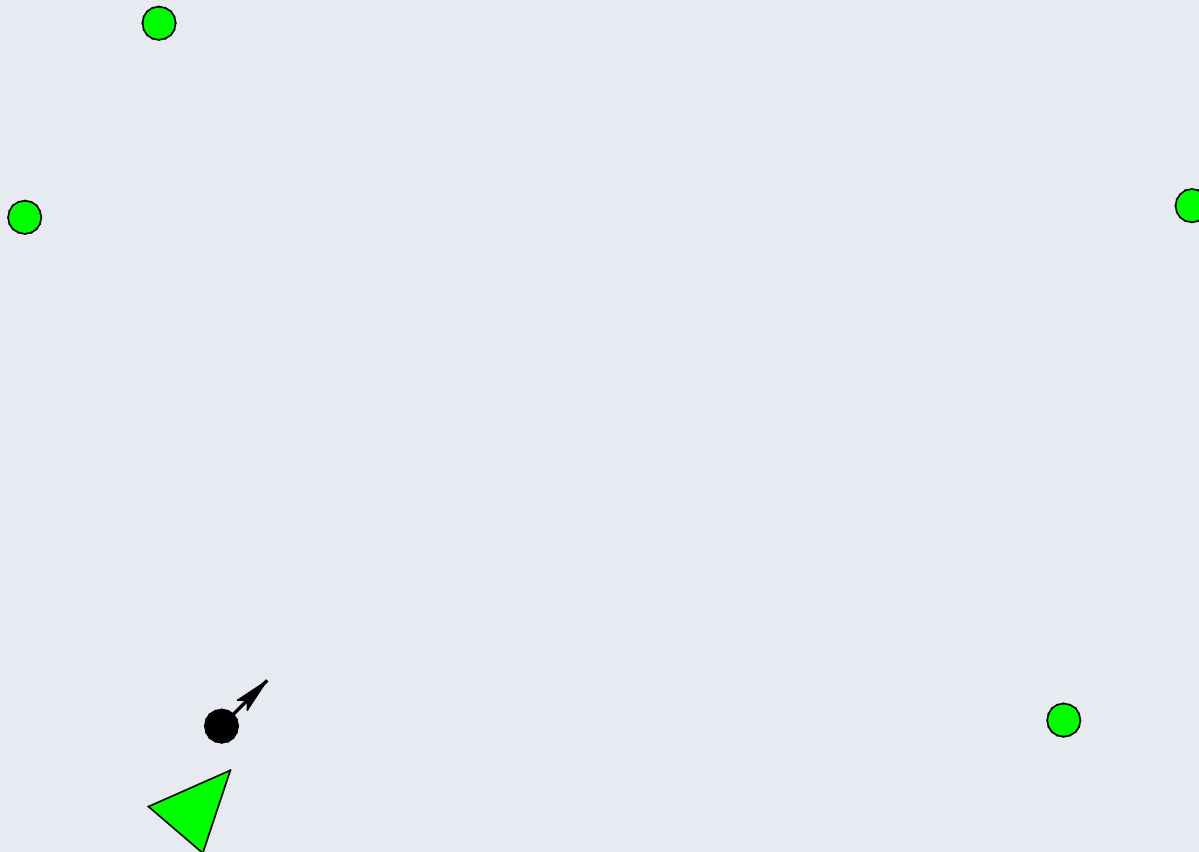
For each trajectory particle, multiple feature to detection associations are possible.

For each possible association, an intermediate particle is defined.

For each of these particles, the measurement likelihoods are Calculated, and a corresponding weight determined.

# SLAM: Multi-Hypothesis (MH) FastSLAM

Ground-truth robot and feature positions & single particle representation:



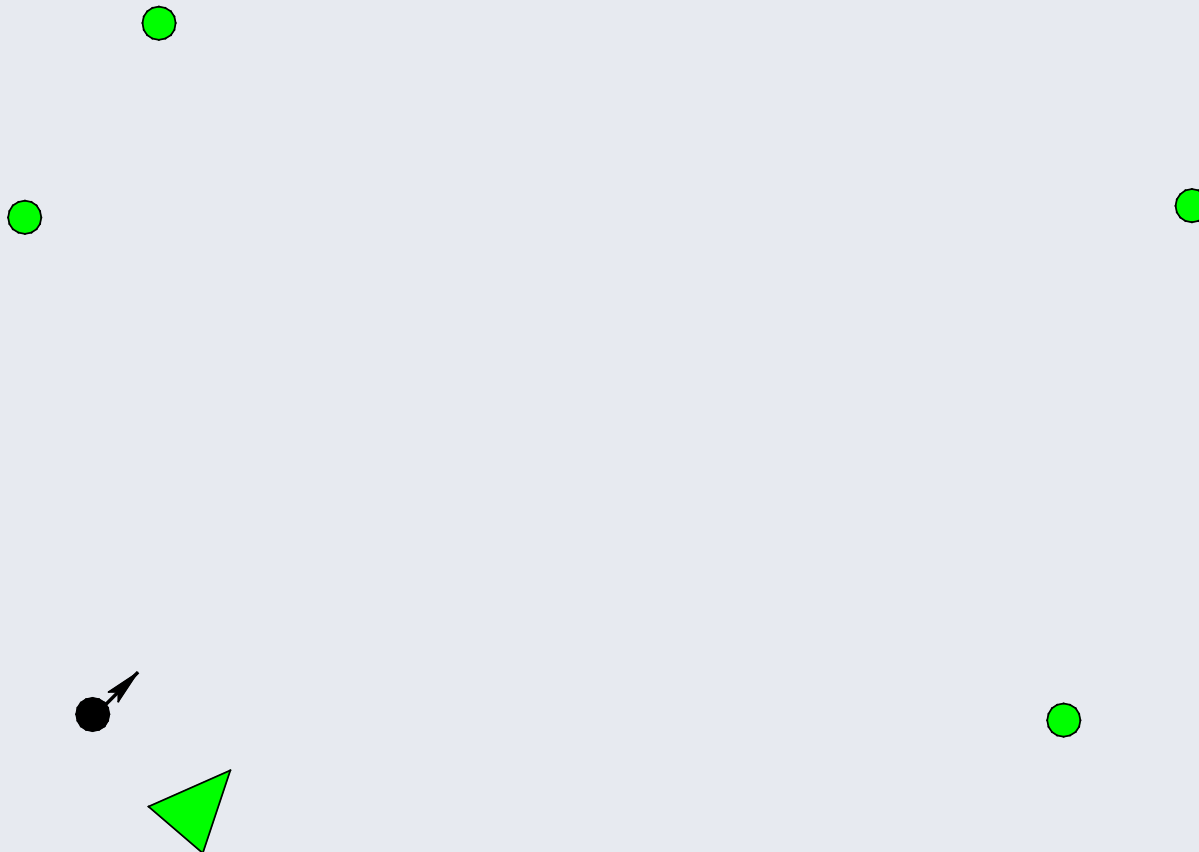
# SLAM: Multi-Hypothesis (MH) FastSLAM

Ground-truth robot and feature positions & single particle representation:



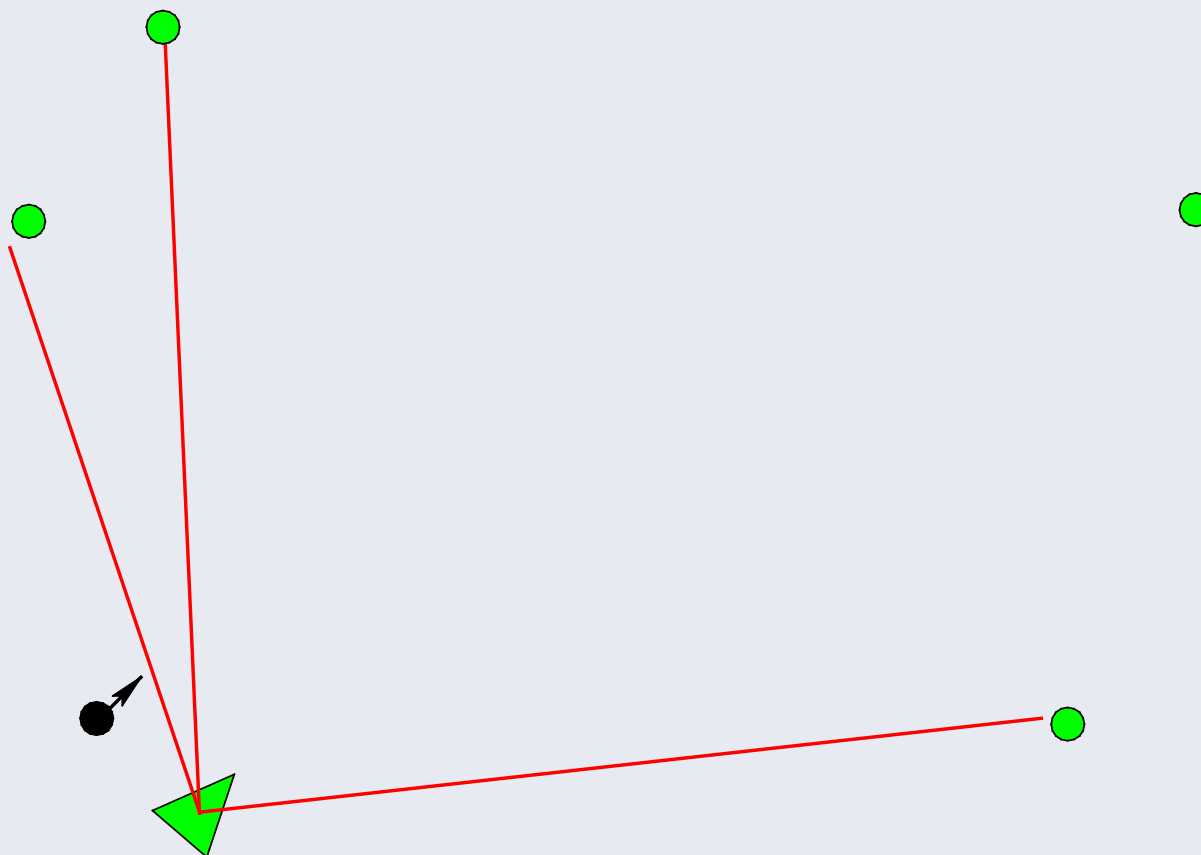
# SLAM: Multi-Hypothesis (MH) FastSLAM

Ground-truth robot and feature positions & single particle representation:



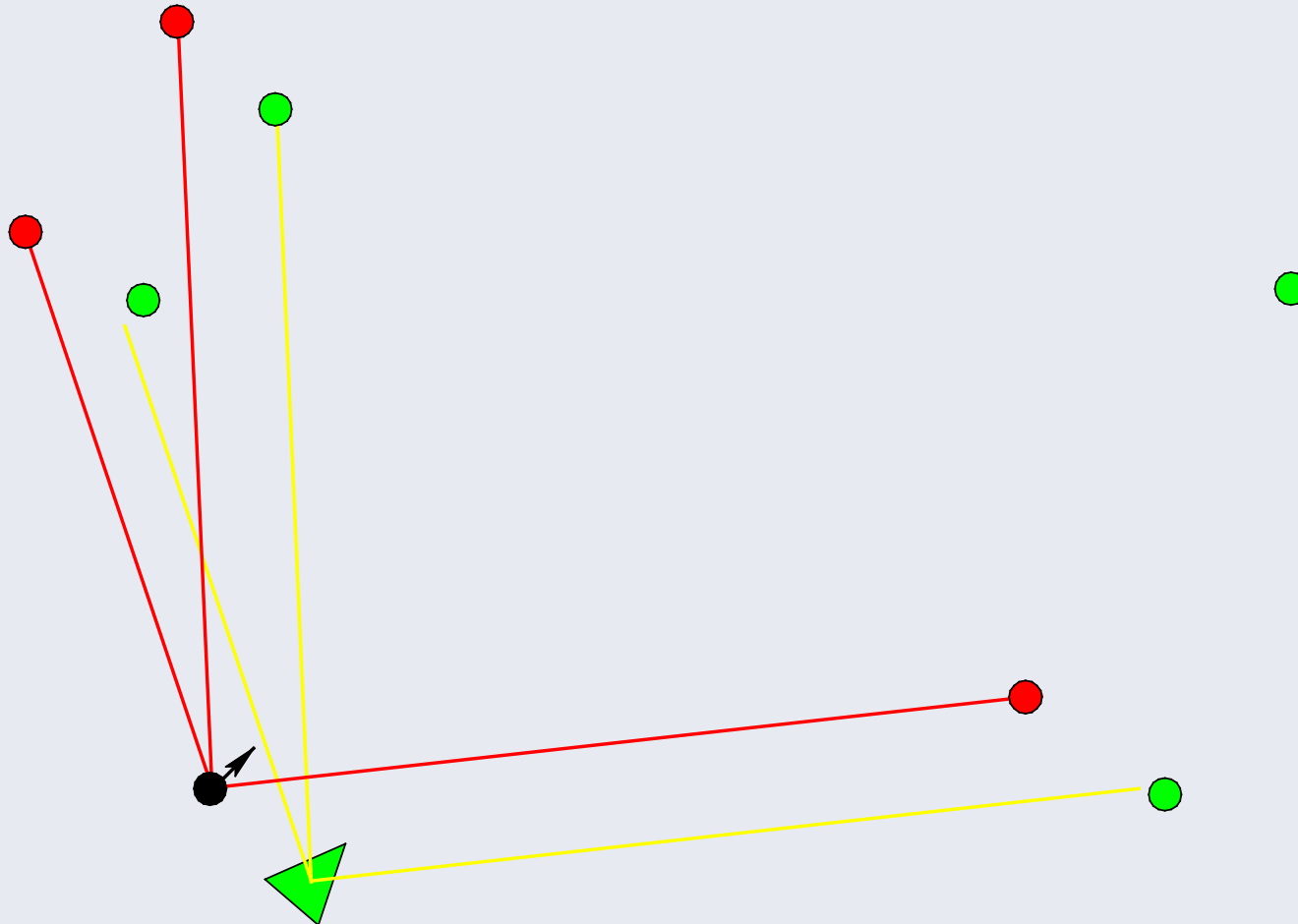
# SLAM: Multi-Hypothesis (MH) FastSLAM

Record sensor scan (range, bearing) – from ACTUAL pose.



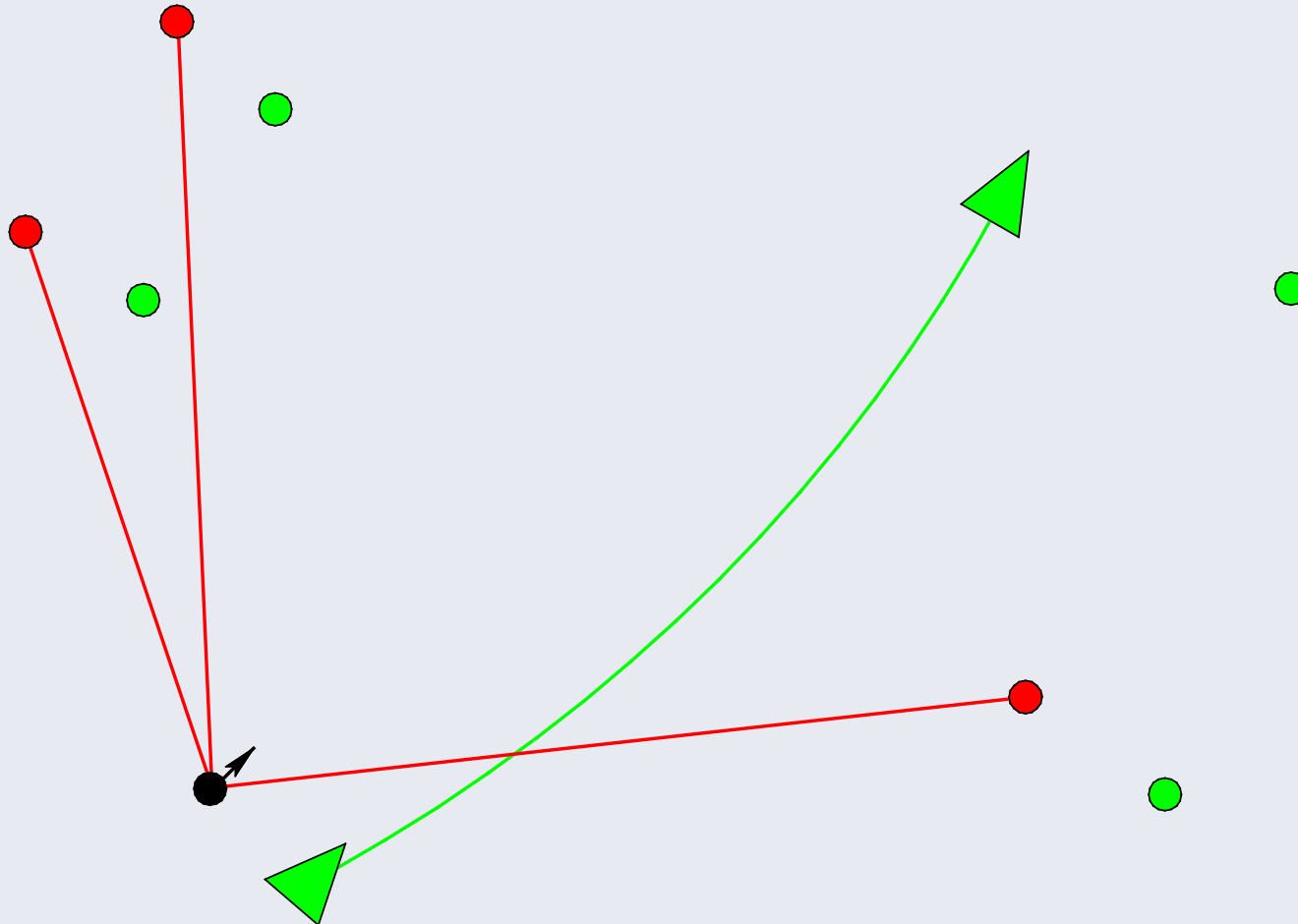
# SLAM: Multi-Hypothesis (MH) FastSLAM

Superimpose recorded scan onto particle.



# SLAM: Multi-Hypothesis (MH) FastSLAM

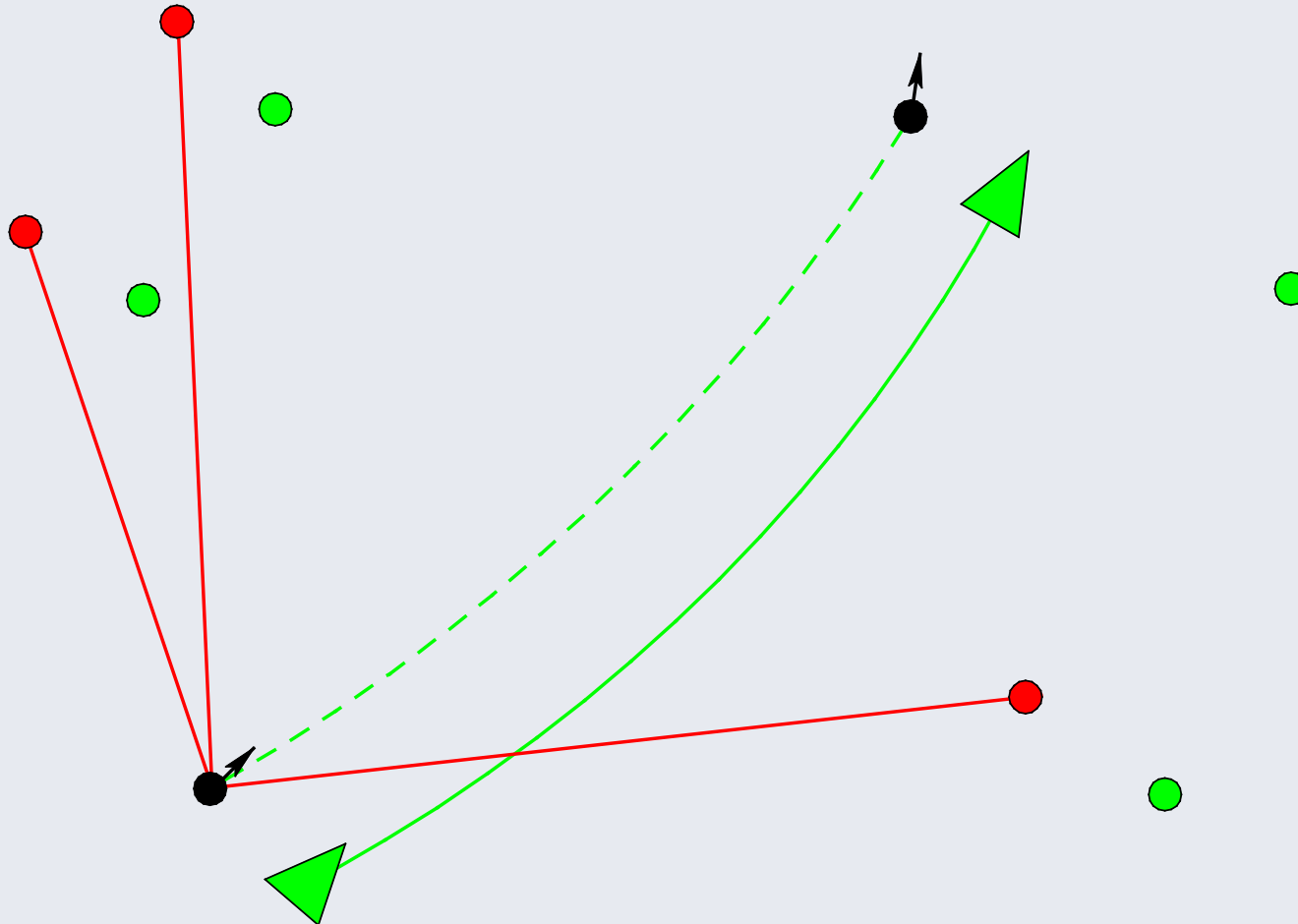
Move robot via computer input steering and velocity commands.





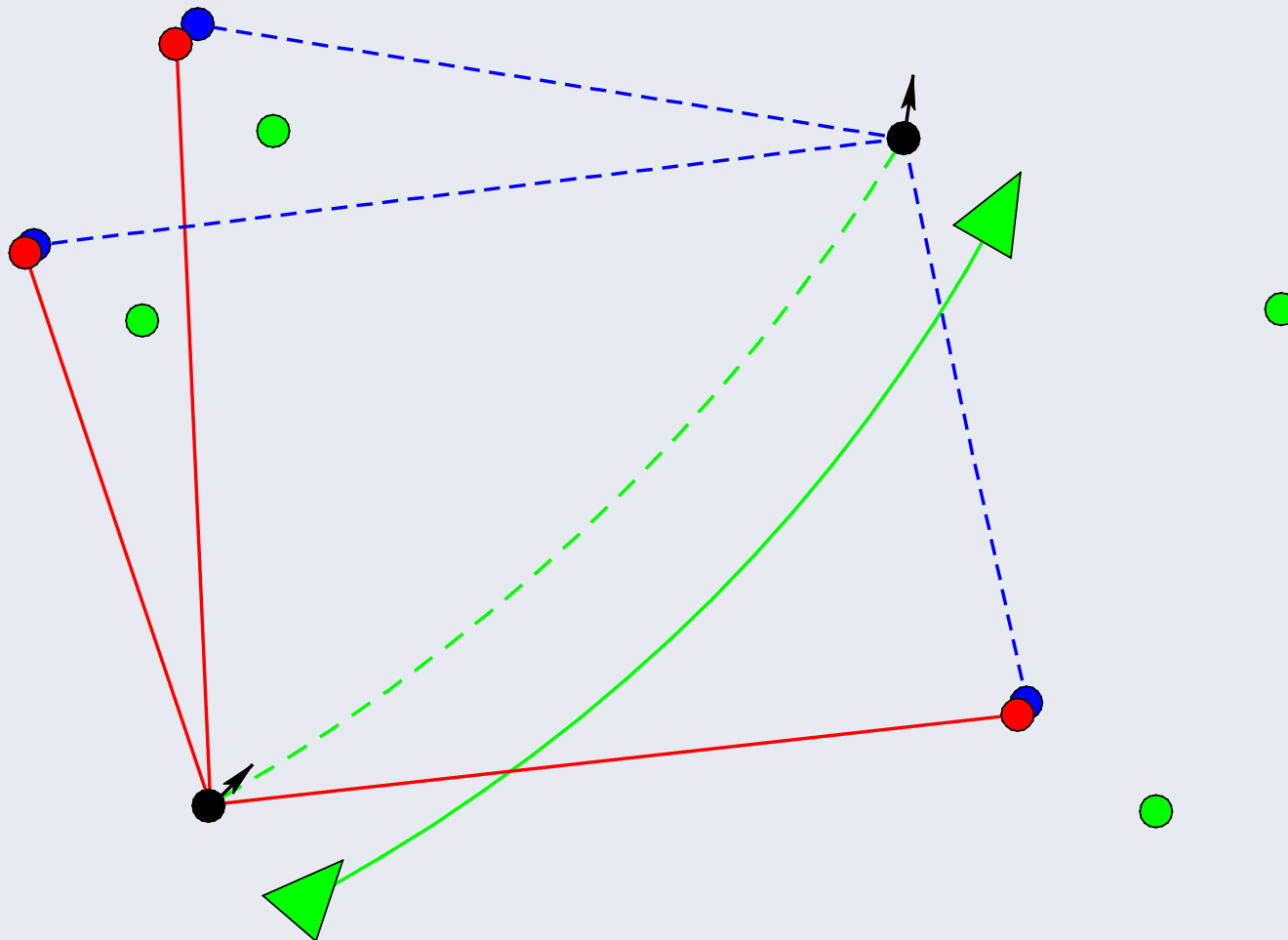
# SLAM: Multi-Hypothesis (MH) FastSLAM

Apply motion model and sampled noise to particle.



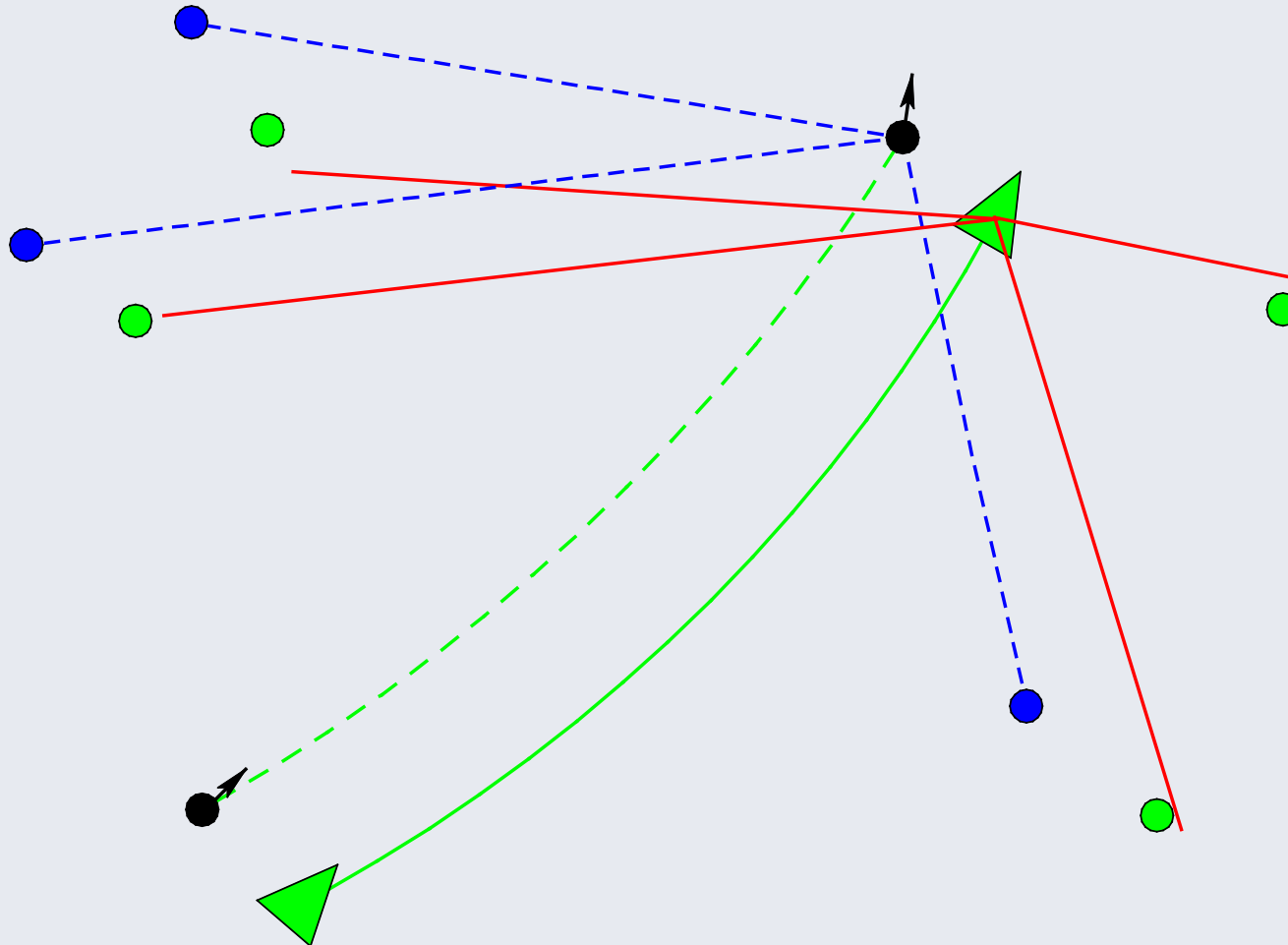
# SLAM: Multi-Hypothesis (MH) FastSLAM

Predict new measurements.



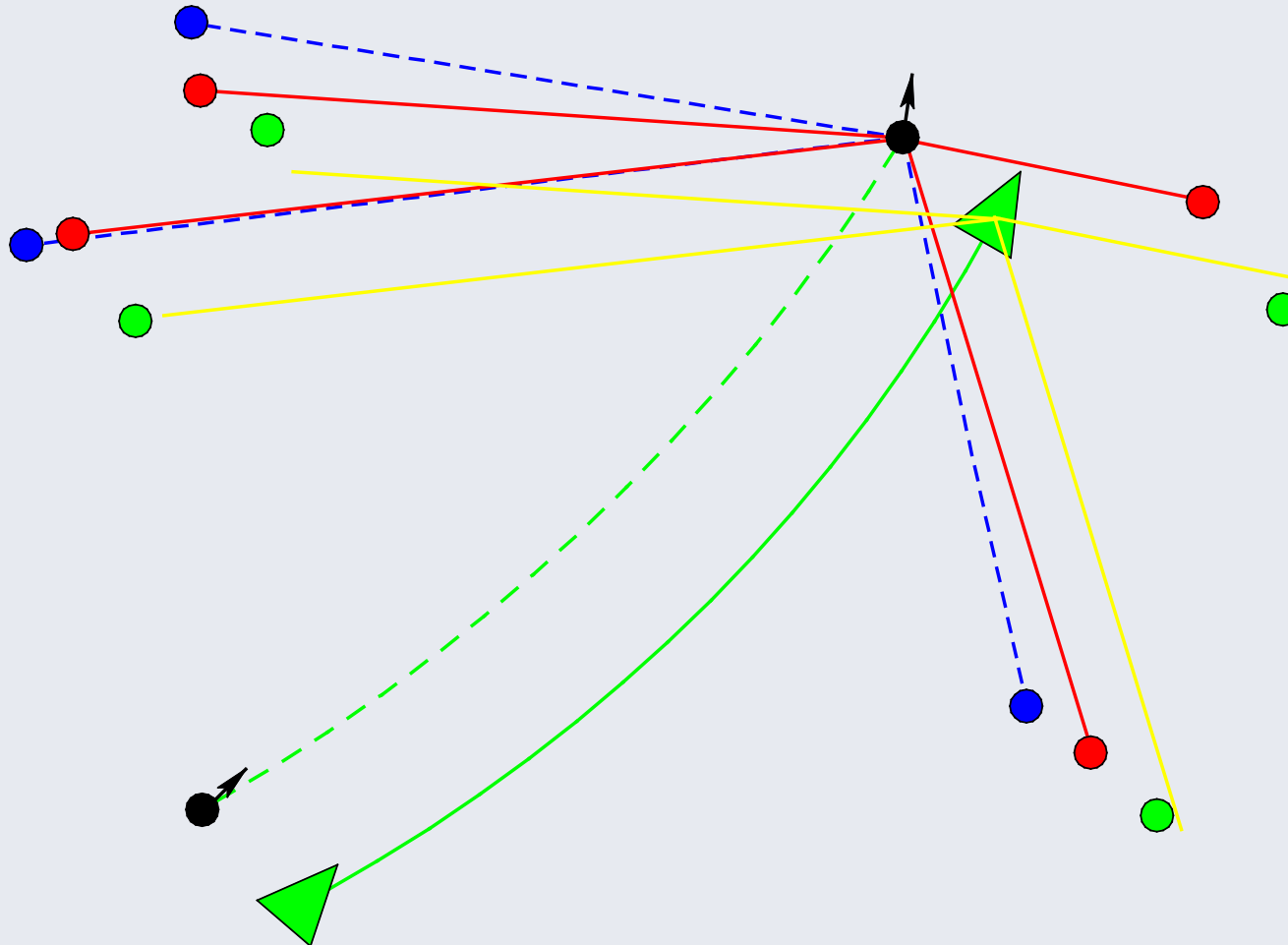
# SLAM: Multi-Hypothesis (MH) FastSLAM

Record new scan (range, bearing) – From ACTUAL new pose.



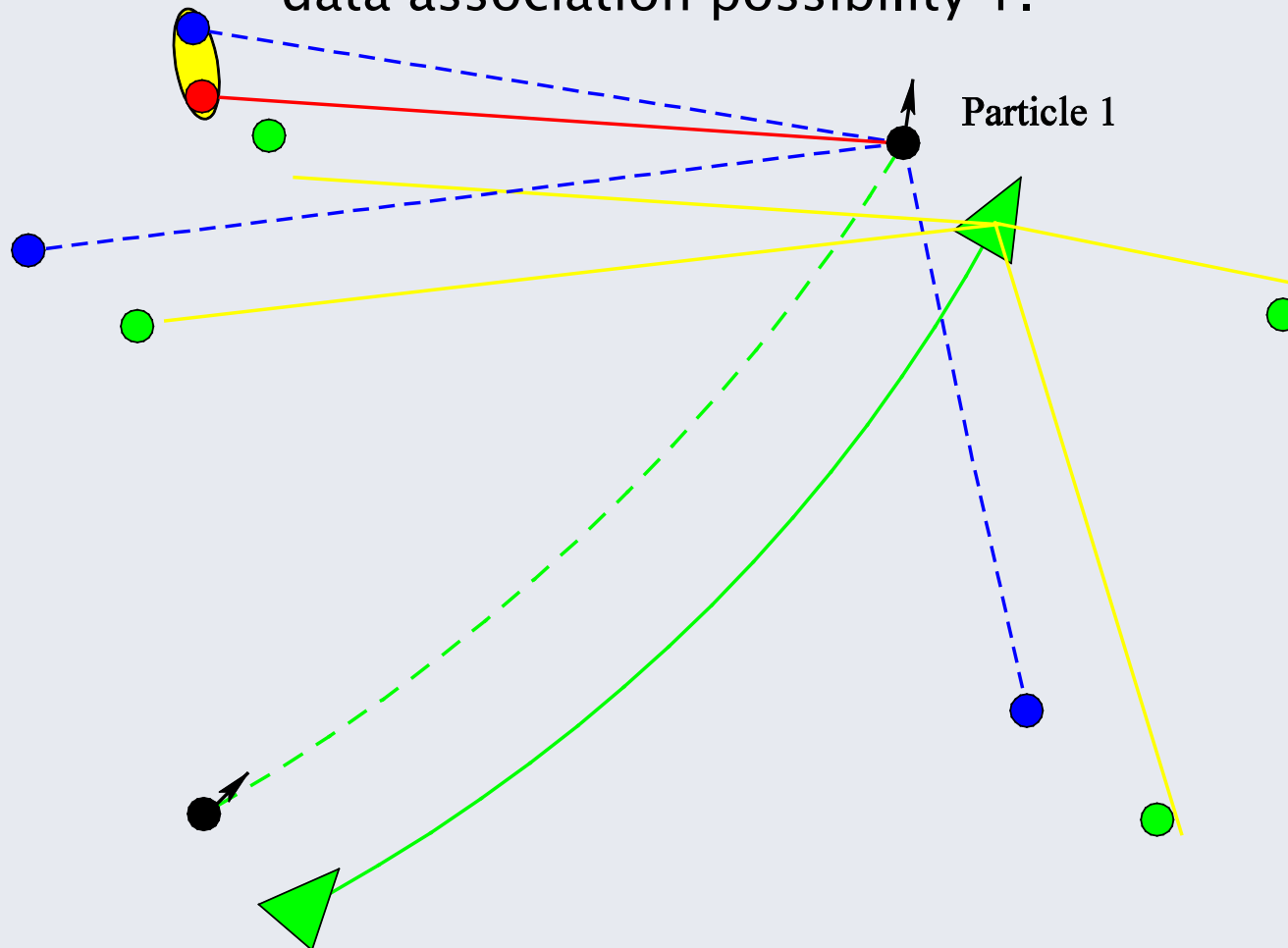
# SLAM: Multi-Hypothesis (MH) FastSLAM

Superimpose new scan onto new particle pose.



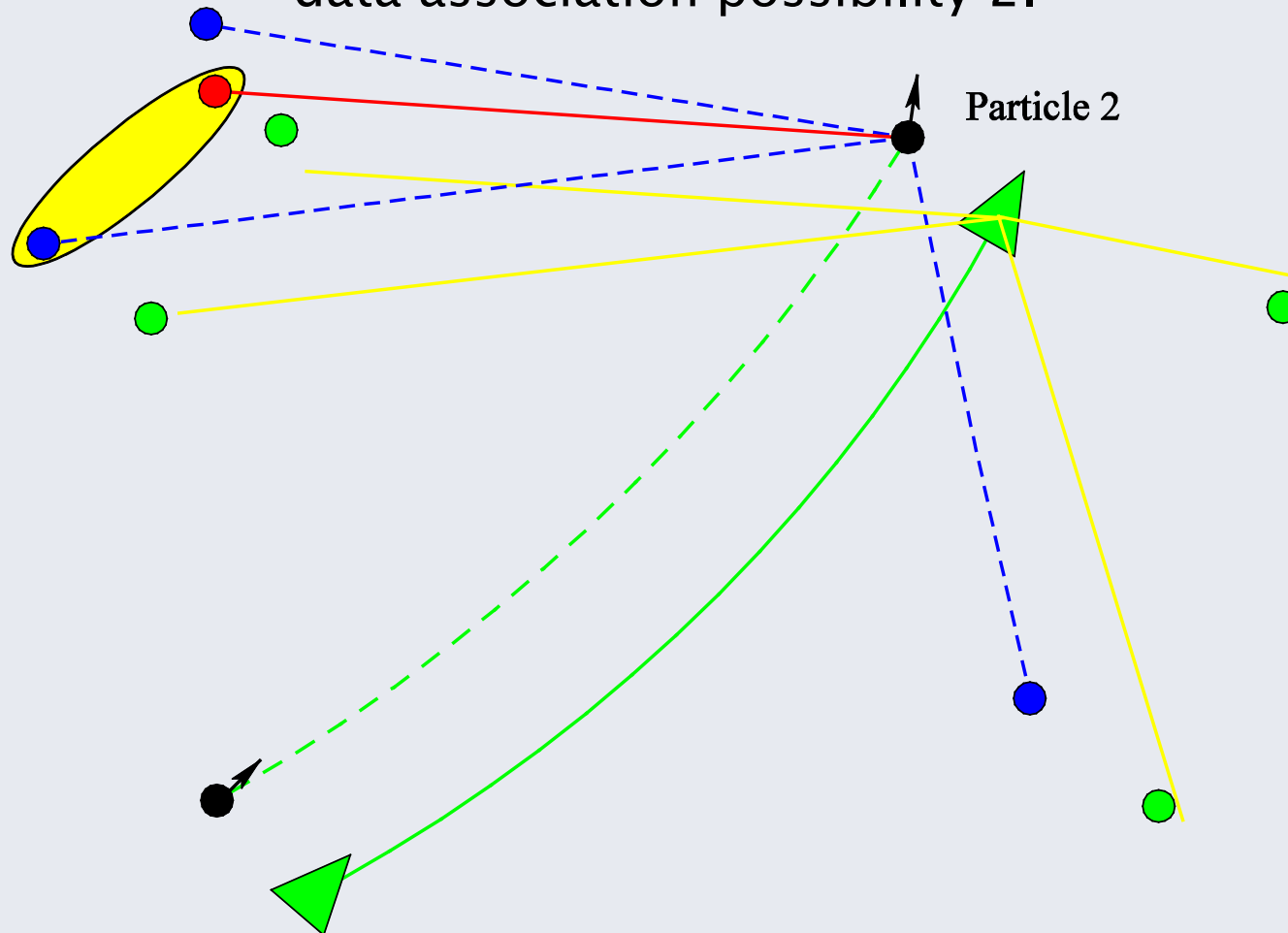
# SLAM: Multi-Hypothesis (MH) FastSLAM

Generate new particle, at same pose, which carries data association possibility 1.



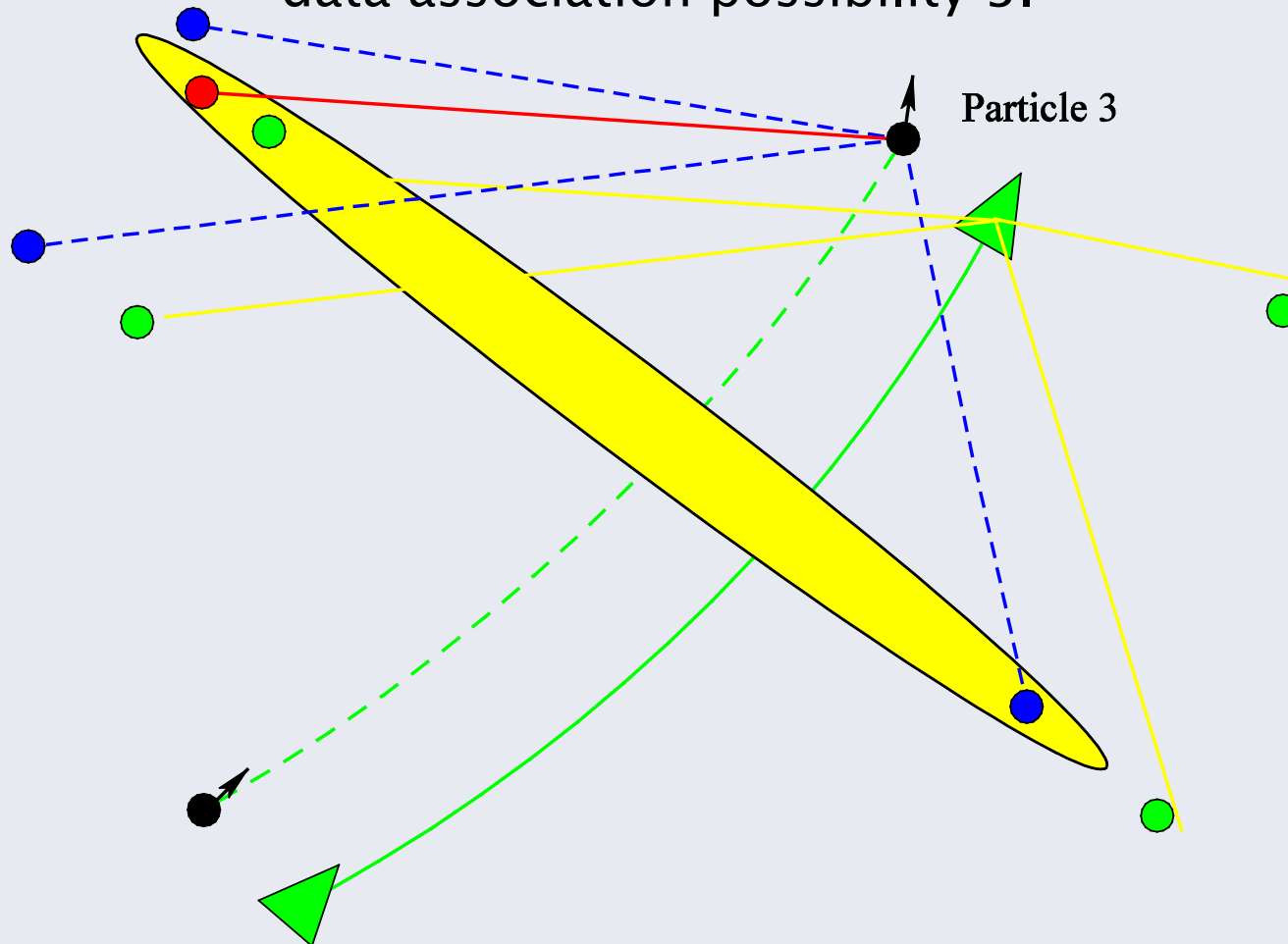
# SLAM: Multi-Hypothesis (MH) FastSLAM

Generate new particle, at same pose, which carries data association possibility 2.



# SLAM: Multi-Hypothesis (MH) FastSLAM

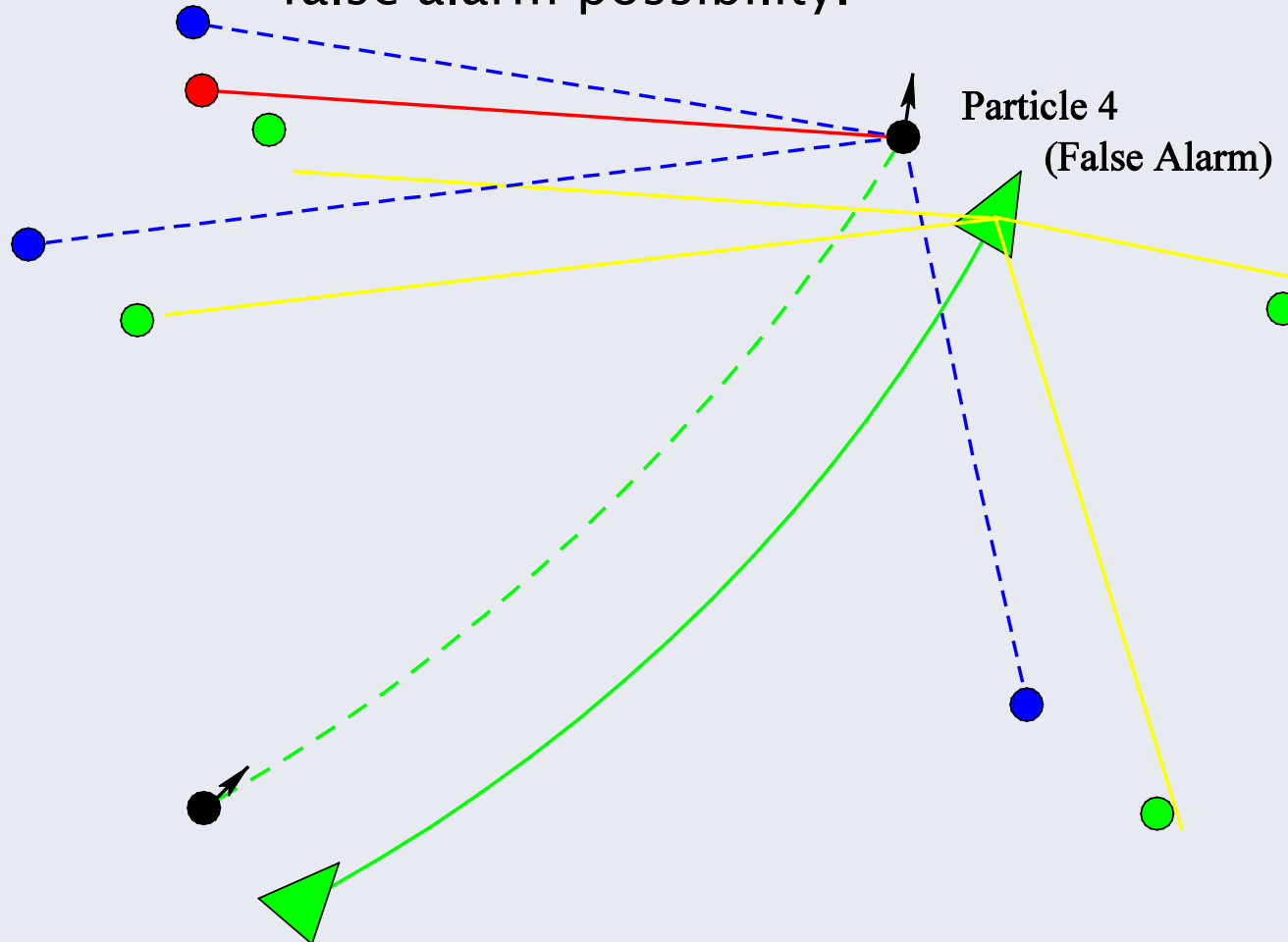
Generate new particle, at same pose, which carries data association possibility 3.





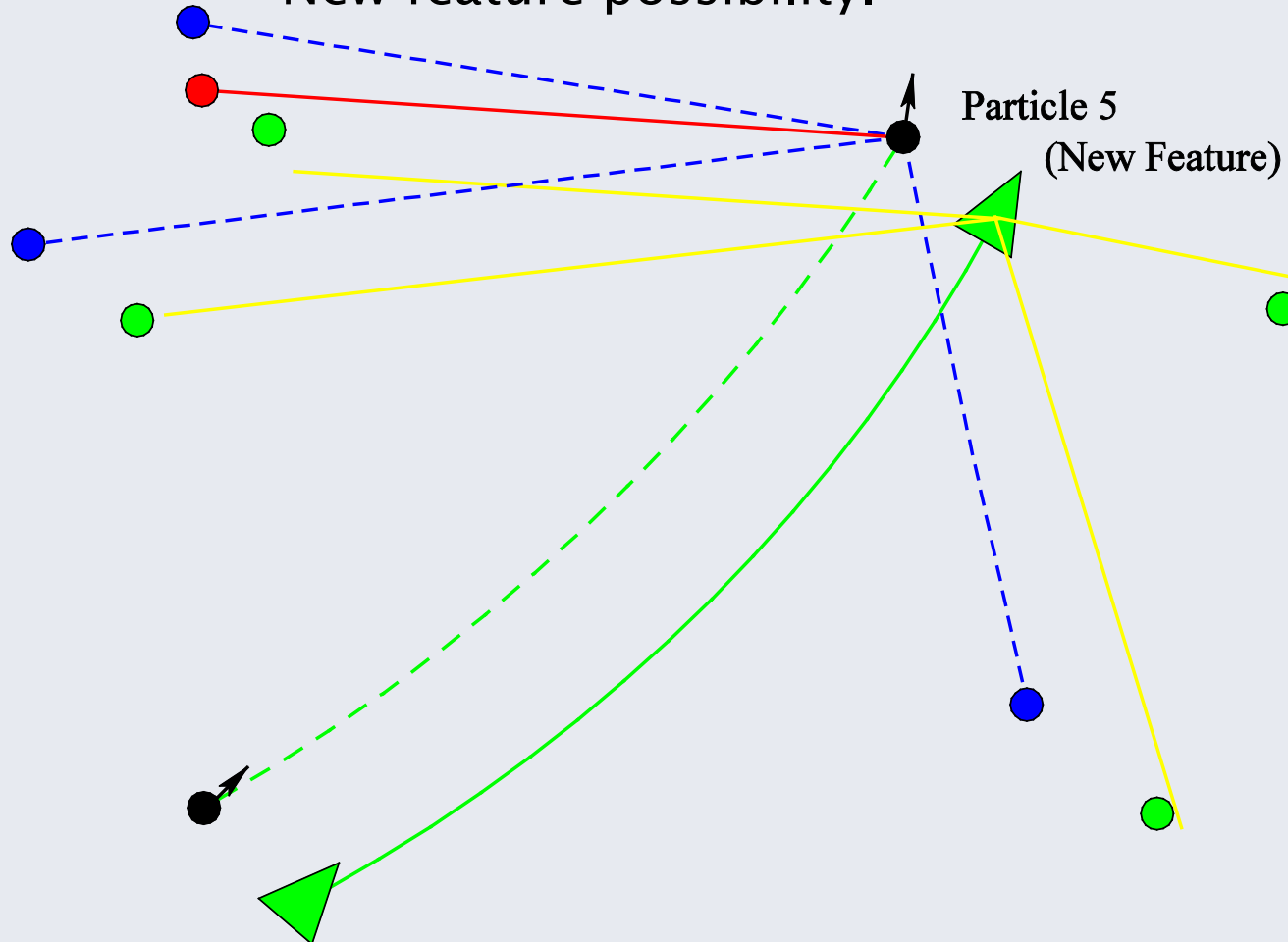
# SLAM: Multi-Hypothesis (MH) FastSLAM

Generate new particle, at same pose, which carries false alarm possibility.



# SLAM: Multi-Hypothesis (MH) FastSLAM

Generate new particle, at same pose, which carries New feature possibility.



# SLAM: Multi-Hypothesis (MH) FastSLAM

## Multi-Hypothesis FastSLAM:

For each trajectory particle, multiple feature to detection associations are possible.

For each possible association, an intermediate particle is defined.

For each of these particles, the measurement likelihoods are Calculated, and a corresponding weight determined.

Resampling, based on the weights is carried out, yielding the same Initial particle number.

# SLAM: Multi-Hypothesis (MH) FastSLAM

Multi-Hypothesis FastSLAM:

For each trajectory particle, multiple feature to detection associations are possible.

For each possible association, an intermediate particle is defined.

For each of these particles, the measurement likelihoods are Calculated, and a corresponding weight determined.

Resampling, based on the weights is carried out, yielding the same Initial particle number.

**A general vector based SLAM method, allowing MH feature Association.**

# SLAM: Multi-Hypothesis (MH) FastSLAM

Multi-Hypothesis FastSLAM:

For each trajectory particle, multiple feature to detection associations are possible.

For each possible association, an intermediate particle is defined.

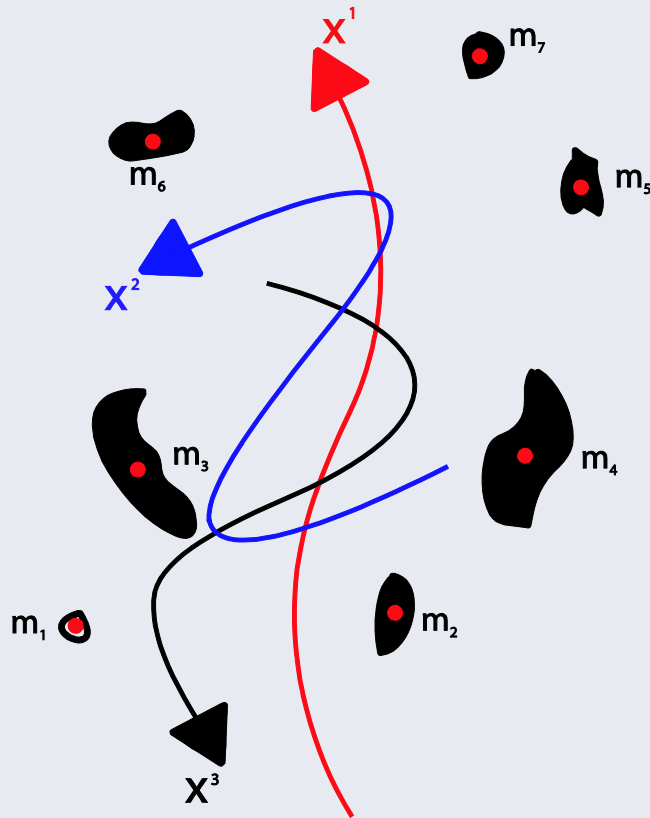
For each of these particles, the measurement likelihoods are Calculated, and a corresponding weight determined.

Resampling, based on the weights is carried out, yielding the same Initial particle number.

A general vector based SLAM method, allowing MH feature Association.

However, not clear how to include detection probabilities, and MH tracking has not been proved Bayes optimal.

# A Random Finite Set (RFS) Approach [Mullane, Vo, Adams '09]



Given  $\mathbf{X}^1$  :

$$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{m}_5, \mathbf{m}_6, \mathbf{m}_7]$$

Given  $\mathbf{X}^2$  :

$$\mathbf{M} = [\mathbf{m}_4, \mathbf{m}_3, \mathbf{m}_2, \mathbf{m}_1, \mathbf{m}_5, \mathbf{m}_7, \mathbf{m}_6]$$

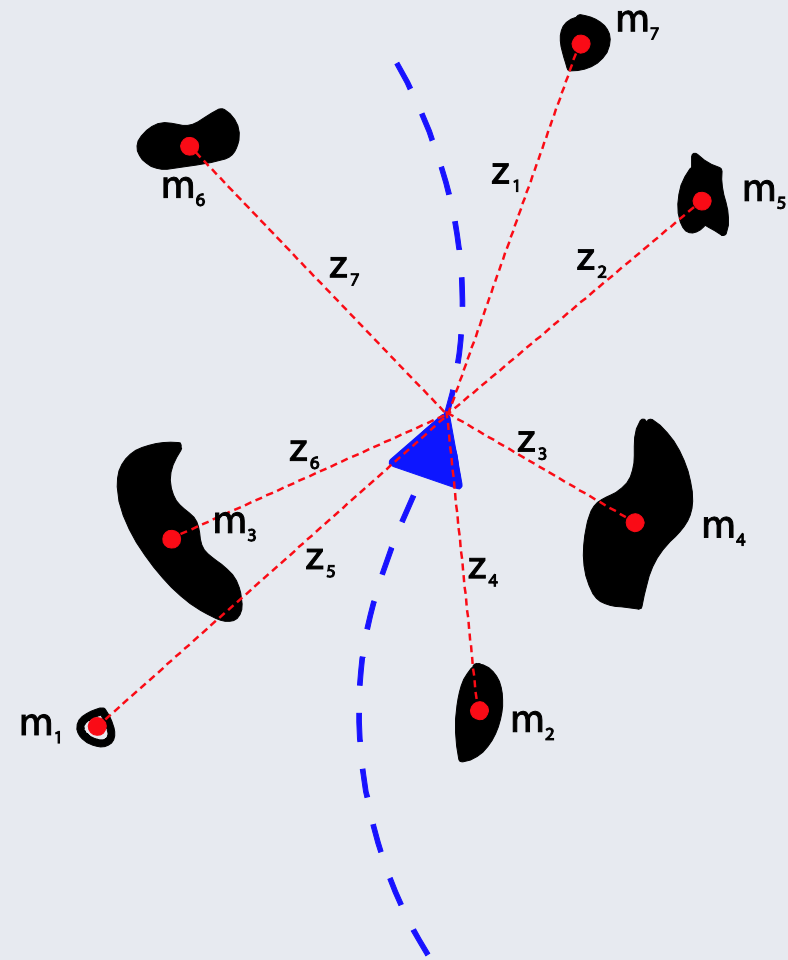
Given  $\mathbf{X}^3$  :

$$\mathbf{M} = [\mathbf{m}_6, \mathbf{m}_7, \mathbf{m}_5, \mathbf{m}_4, \mathbf{m}_3, \mathbf{m}_2, \mathbf{m}_1]$$

- Estimated map vector **depends on vehicle trajectory** ?

- RFS makes more sense as order of features *cannot/should not* be significant [Mullane, Adams 2009].

# A Random Finite Set (RFS) Approach



Untangle:

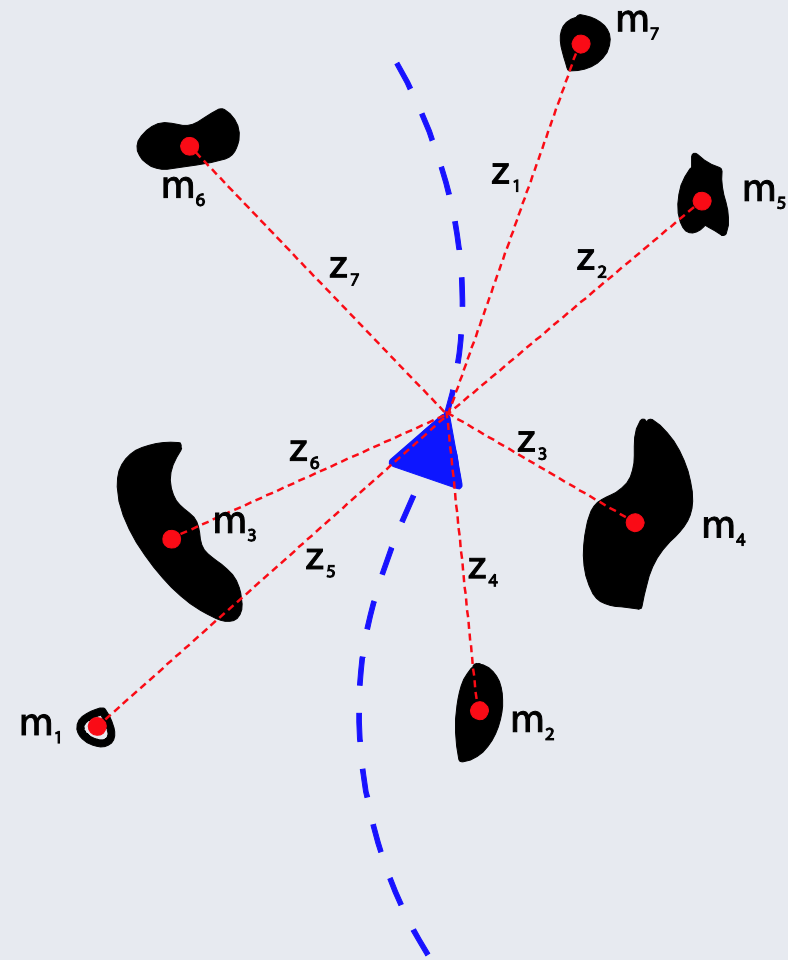
$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{z}_7]$$

?

$$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{m}_5, \mathbf{m}_6, \mathbf{m}_7]$$



# A Random Finite Set (RFS) Approach

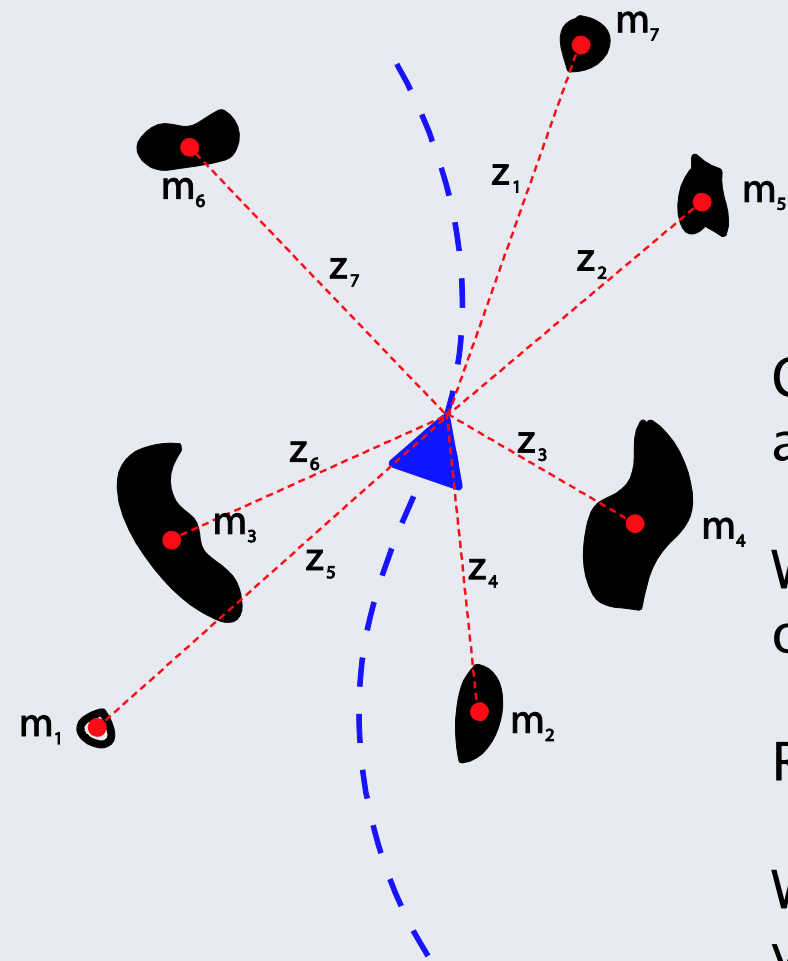


Untangle:

$$\mathbf{Z} = [z_1, z_2, z_3, z_4, z_5, z_6, z_7]$$

$$\mathbf{M} = [m_1, m_2, m_3, m_4, m_5, m_6, m_7]$$

# A Random Finite Set (RFS) Approach



Untangle:

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{z}_7]$$

$$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4, \mathbf{m}_5, \mathbf{m}_6, \mathbf{m}_7]$$

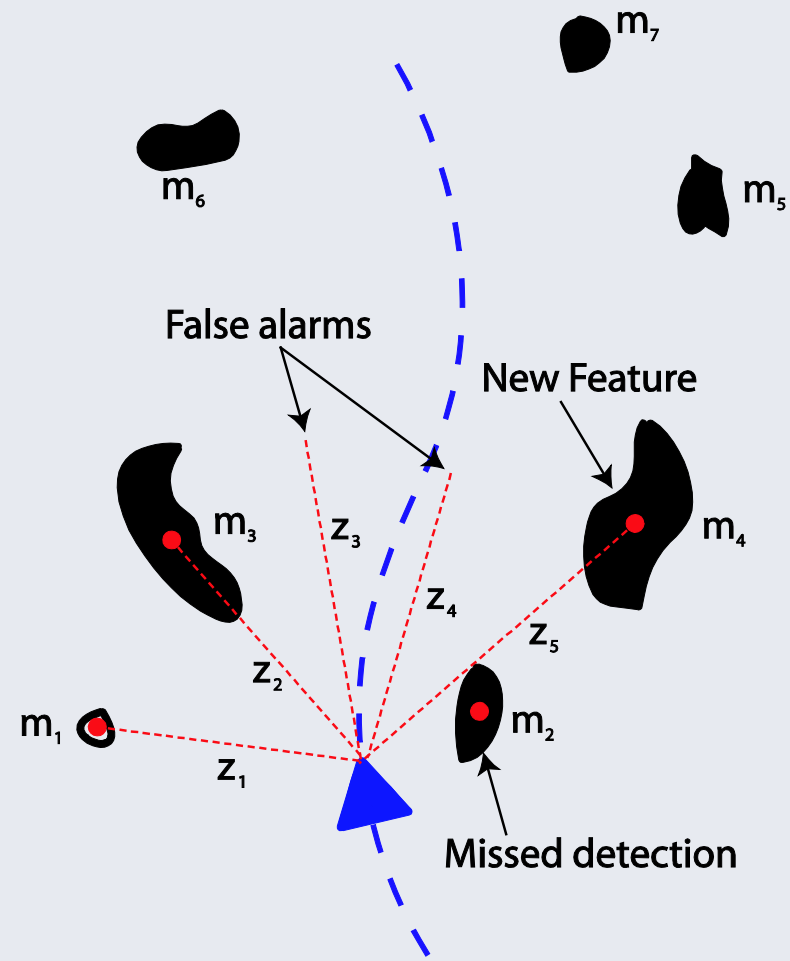
Current vector formulations *require* data association (DA) prior to Bayesian update:

Why? Features & measurements rigidly ordered in vector-valued map state.

RFS approach *does not require* DA.

Why? Features & measurements are finite valued sets. No distinct order assumed.

# A Random Finite Set (RFS) Approach



How to relate measurements and states of different dimensions?

# What is a RFS Measurement?

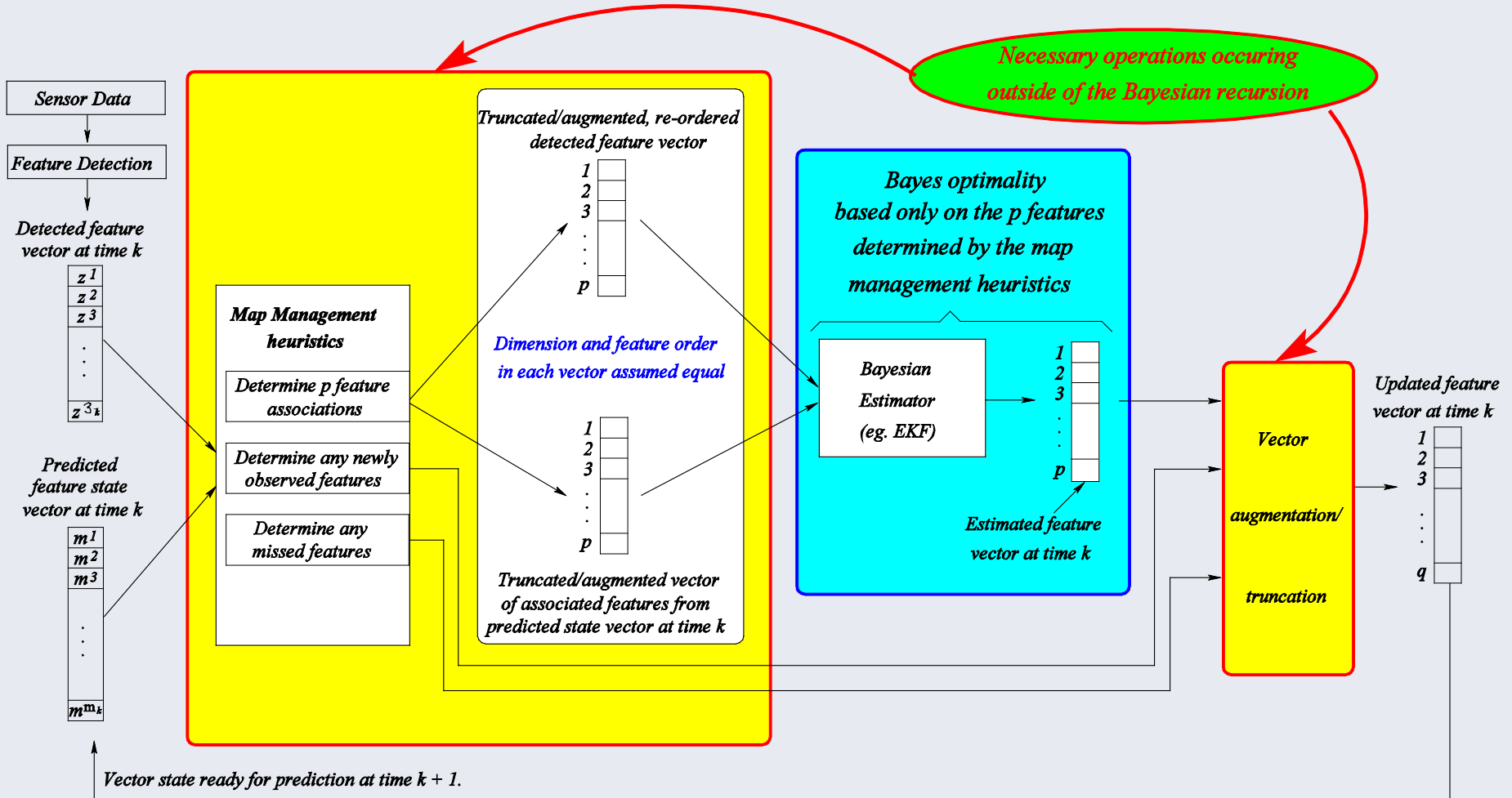
$$\mathcal{Z} = \{z^1, \dots, z^{\delta}\} = \{[r^1 \ \theta^1]^T, \dots, [r^{\delta} \ \theta^{\delta}]^T\} \quad (1)$$

Hence, at any instant, a sensor can be considered to collect a finite set  $\mathcal{Z} = \{z^1, \dots, z^{\delta}\}$  of measurements  $z^1, \dots, z^{\delta}$  from a measurement space  $\mathcal{Z}_0$  as follows:

$$\begin{aligned} \mathcal{Z} &= \emptyset && \text{(no features detected)} \\ \mathcal{Z} &= \{z^1\} && \text{(one feature } z^1 \text{ detected)} \\ \mathcal{Z} &= \{z^1, z^2\} && \text{(two features } z^1, z^2 \text{ detected)} \\ &\vdots && \vdots \\ \mathcal{Z} &= \{z^1, \dots, z^{\delta}\} && \text{(\delta features } z^1, \dots, z^{\delta} \text{ detected)} \end{aligned} \quad (2)$$

[Ronald Mahler, Lockheed Tactical Systems]

# RFSs versus Vectors for SLAM

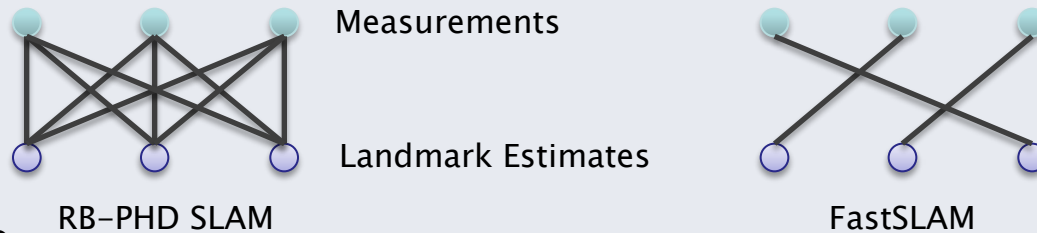


## Vector Based Mapping and SLAM



# RFS SLAM as a Generalization of RV SLAM

- RB-PHD-SLAM vs. FastSLAM
- Data association
  - In RB-PHD SLAM, all possible data associations considered by each particle
  - In FastSLAM, each particle considers one data association hypothesis



- Map update
  - In RB-PHD SLAM, every landmark estimate updated with every measurement, creating a set of weighted Gaussians based on measurement likelihood.
  - In FastSLAM, every landmark is updated with its associated measurement.

$$\left\{ \mathcal{N} \left( \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i \right), \mathcal{N} \left( \boldsymbol{\mu}_k^{i,j}, \boldsymbol{\Sigma}_k^{i,j} \right) \right\}, \quad \forall i \in \{1 \dots N_k^-\}, \forall j \in \{1 \dots |\mathcal{Z}_k|\}$$

- The FastSLAM map is a subset of the RB-PHD SLAM map.
- Importance Weighting

# RFS SLAM as a Generalization of RV SLAM

- Importance Weighting

- FastSLAM:

$$\begin{aligned}\omega_k^{[l]} &\equiv \omega_{k-1}^{[l]} p\left(\mathcal{Z}_k | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}\right) \\ &= \omega_{k-1}^{[l]} \prod_{\substack{j=1 \\ \theta(j) \neq 0}}^n \int p\left(\mathbf{z}^j | \mathbf{m}^{\theta(j)}, \mathbf{x}_{0:k}^{[l]}\right) p\left(\mathbf{m}^{\theta(j)} | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathbf{m}^{\theta(j)}\end{aligned}$$

- RB-PHD SLAM:

$$\omega_k^{[l]} = \omega_{k-1}^{[l]} \int p\left(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}^{[l]}\right) p\left(\mathcal{M}_k | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathcal{M}_k$$

$$\begin{aligned}&\int p\left(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}^{[l]}\right) p\left(\mathcal{M}_k | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathcal{M}_k \\ &= p\left(\mathcal{Z}_k | \emptyset, \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) p\left(\emptyset | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) + \\ &\quad \int p\left(\mathcal{Z}_k | \mathbf{m}^1, \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) p\left(\mathbf{m}^1 | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathbf{m}^1 + \\ &\quad \iint p\left(\mathcal{Z}_k | \mathbf{m}^1, \mathbf{m}^2, \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) p\left(\mathbf{m}^1, \mathbf{m}^2 | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathbf{m}^1 d\mathbf{m}^2 + \dots + \\ &\quad \int \dots \int p\left(\mathcal{Z}_k | \mathbf{m}^1, \dots, \mathbf{m}^m, \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) p\left(\mathbf{m}^1, \dots, \mathbf{m}^m | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathbf{m}^1 \dots d\mathbf{m}^m + \dots\end{aligned}$$

- Assume known map size

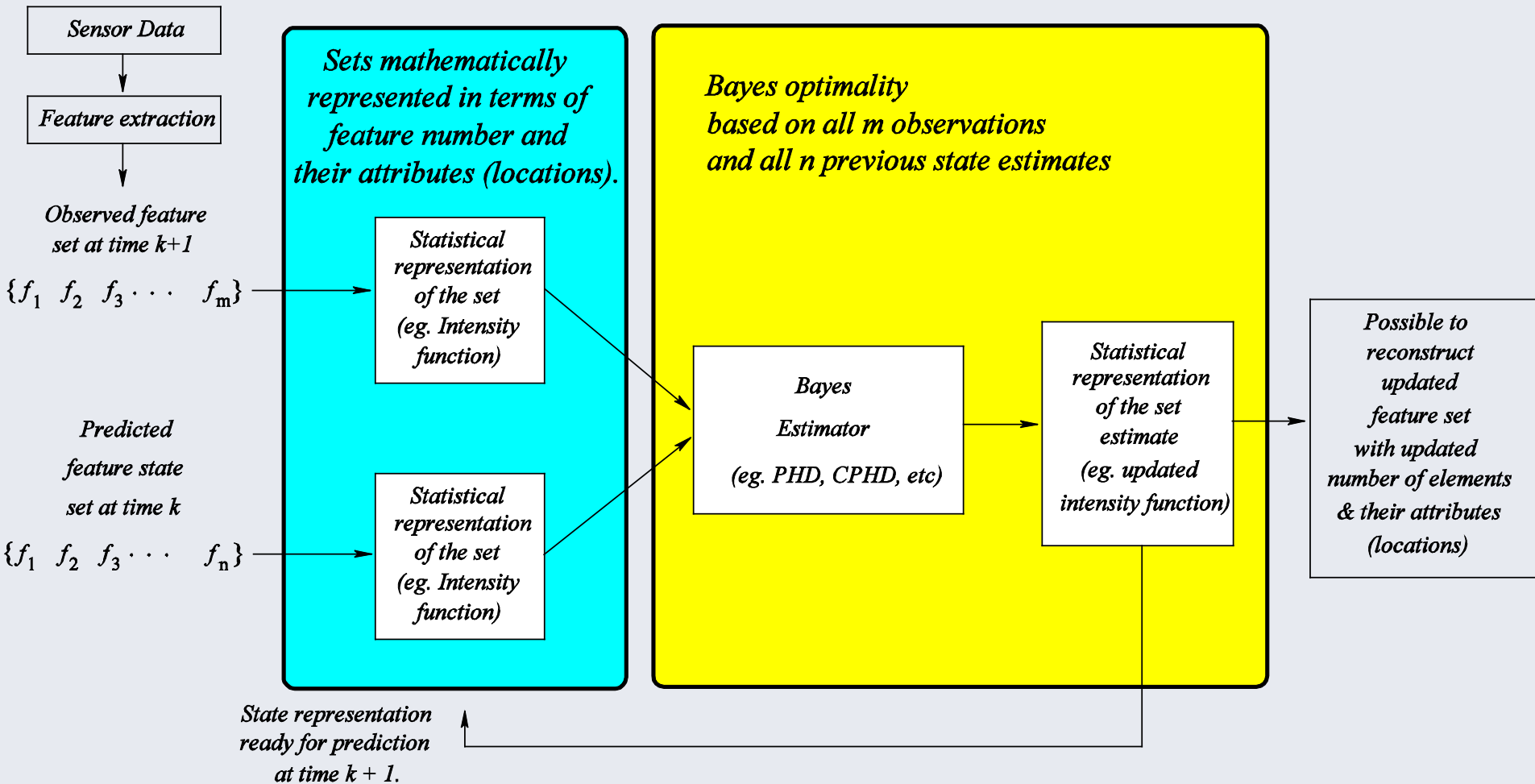
- Assume probability of detection = 1 for associated landmarks, and 0 for unassociated ones

- Then:

$$\begin{aligned}&\int p\left(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}^{[l]}\right) p\left(\mathcal{M}_k | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathcal{M}_k \\ &= \prod_{\substack{j=1 \\ \theta(j) \neq 0}}^n \int p\left(\mathbf{z}_k^j | \mathbf{m}^{\theta(j)}, \mathbf{x}_{0:k}^{[l]}\right) p\left(\mathbf{m}^{\theta(j)} | \mathbf{x}_{0:k}^{[l]}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right) d\mathbf{m}^{\theta(j)}\end{aligned}$$



# RFSs versus Vectors for SLAM



## RFS Based Mapping and SLAM

# How to do RFS SLAM – Intensity Function

From Point Process Theory:

A Random Finite Set can be approximated by its first order moment – *The Intensity function*  $\nu_k$  [Mahler 2003, Vo 2006].

# RFS SLAM – Intensity Function

From Point Process Theory:

A Random Finite Set can be approximated by its first order moment – *The Intensity function*  $\nu_k$  [Mahler 2003, Vo 2006].

$\nu_k$  has the following properties:

# RFS SLAM – Intensity Function

From Point Process Theory:

A Random Finite Set can be approximated by its first order moment – *The Intensity function*  $\nu_k$  [Mahler 2003, Vo 2006].

$\nu_k$  has the following properties:

1. Its integral, over the set, gives the *estimated number* of elements within the set.

# RFS SLAM – Intensity Function

From Point Process Theory:

A Random Finite Set can be approximated by its first order moment – *The Intensity function*  $\nu_k$  [Mahler 2003, Vo 2006].

$\nu_k$  has the following properties:

1. Its integral, over the set, gives the *estimated number* of elements within the set.
2. The locations of its maxima correspond to the *estimated values* of the set members.

# RFS SLAM – Intensity Function

From Point Process Theory:

A Random Finite Set can be approximated by its first order moment – *The Intensity function*  $\nu_k$  [Mahler 2003, Vo 2006].

$\nu_k$  has the following properties:

1. Its integral, over the set, gives the *estimated number* of elements within the set.
2. The locations of its maxima correspond to the *estimated values* of the set members.

Intensity function can be propagated through the *Probability Hypothesis Density* (PHD) filter.

## Example: 1D Intensity Function (PHD)

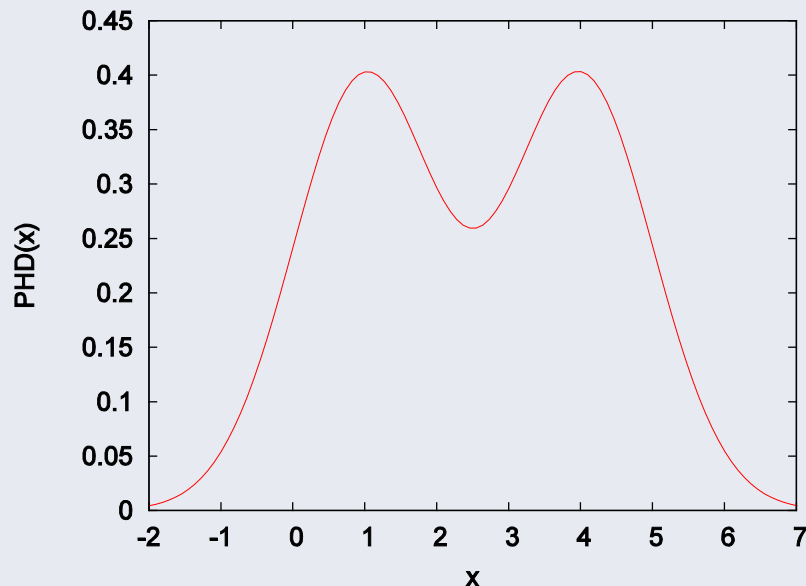
E.g. 2 Features located at  $x=1$  and  $x=4$  with spatial variance:  $\sigma^2 = 1$   
i.e. Feature set  $\{1, 4\}$  [Mahler 2007].

Suitable Gaussian Mixture PHD: 
$$\text{PHD}(x) = \frac{1}{\sqrt{2\pi}\sigma} \left[ \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-4)^2}{2\sigma^2}\right) \right]$$

# Example: 1D Intensity Function (PHD)

E.g. 2 Features located at  $x=1$  and  $x=4$  with spatial variance:  $\sigma^2 = 1$   
i.e. Feature set  $\{1, 4\}$  [Mahler 2007].

Suitable Gaussian Mixture PHD: 
$$\text{PHD}(x) = \frac{1}{\sqrt{2\pi\sigma}} \left[ \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-4)^2}{2\sigma^2}\right) \right]$$

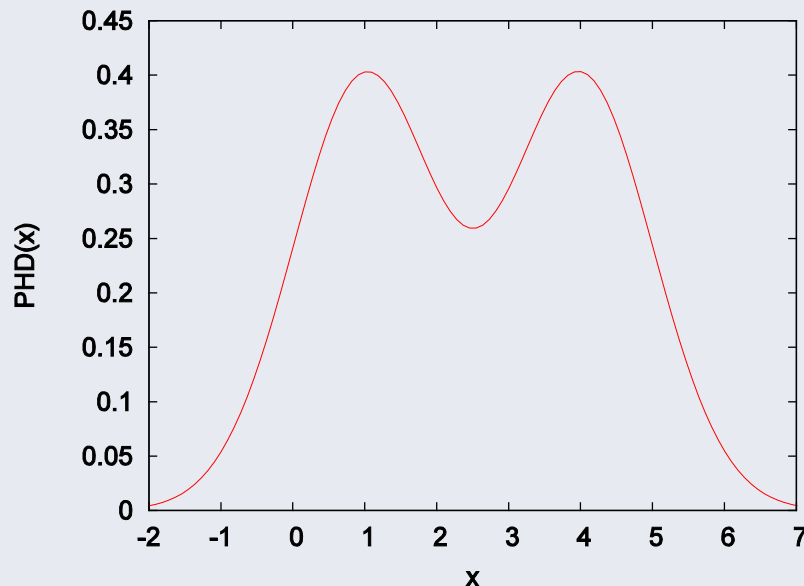




# Example: 1D Intensity Function (PHD)

E.g. 2 Features located at  $x=1$  and  $x=4$  with spatial variance:  $\sigma^2 = 1$   
 i.e. Feature set  $\{1, 4\}$  [Mahler 2007].

Suitable Gaussian Mixture PHD: 
$$\text{PHD}(x) = \frac{1}{\sqrt{2\pi\sigma}} \left[ \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-4)^2}{2\sigma^2}\right) \right]$$



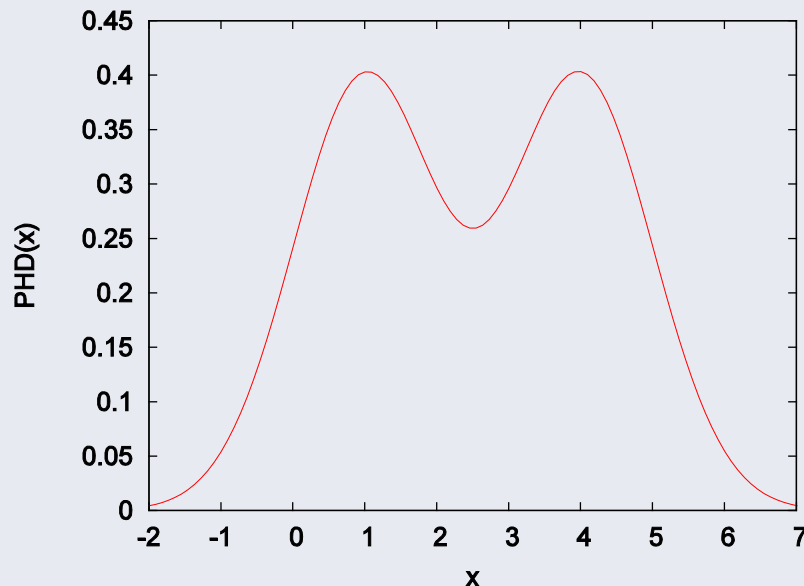
Note: Maxima of PHD occur near  $x=1$  and  $x=4$  and

$$\int \text{PHD}(x) dx = 1 + 1 = 2 = \text{No. of targets!}$$

# Example: 1D Intensity Function (PHD)

E.g. 2 Features located at  $x=1$  and  $x=4$  with spatial variance:  $\sigma^2 = 1$   
 i.e. Feature set  $\{1, 4\}$  [Mahler 2007].

Suitable Gaussian Mixture PHD: 
$$\text{PHD}(x) = \frac{1}{\sqrt{2\pi\sigma}} \left[ \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-4)^2}{2\sigma^2}\right) \right]$$



**Important Point:**

**A PHD is NOT a PDF, since in general it does not integrate to unity!**

Note: Maxima of PHD occur near  $x=1$  and  $x=4$  and

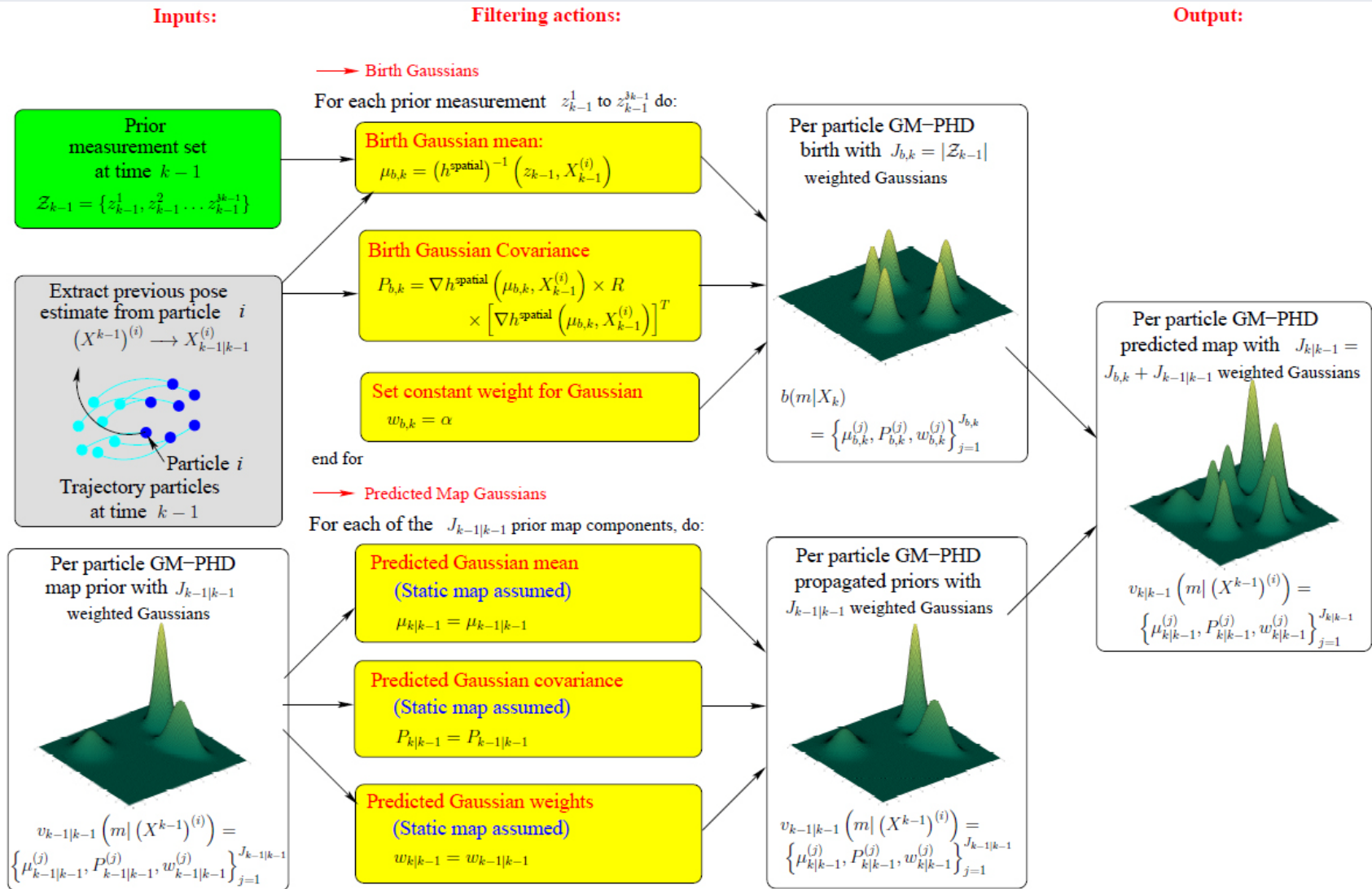
$$\int \text{PHD}(x) dx = 1 + 1 = 2 = \text{No. of targets!}$$

# Implementing PHD SLAM – PHD Predictor

PHD Predictor Equation:

$$\underbrace{v_{k|k-1}(m|X^k)}_{\text{Predicted PHD}} = \underbrace{v_{k-1|k-1}(m|X^{k-1})}_{\text{Prior PHD}} + \underbrace{b(m|X_k)}_{\text{Birth PHD}}$$

# Implementing PHD SLAM – PHD Predictor



# Implementing PHD SLAM – PHD Corrector

PHD Corrector Equation:

$$\underbrace{v_{k|k}(m|X^k)} = \underbrace{v_{k|k-1}(m|X^k)(1 - P_D(m|X_k))}$$

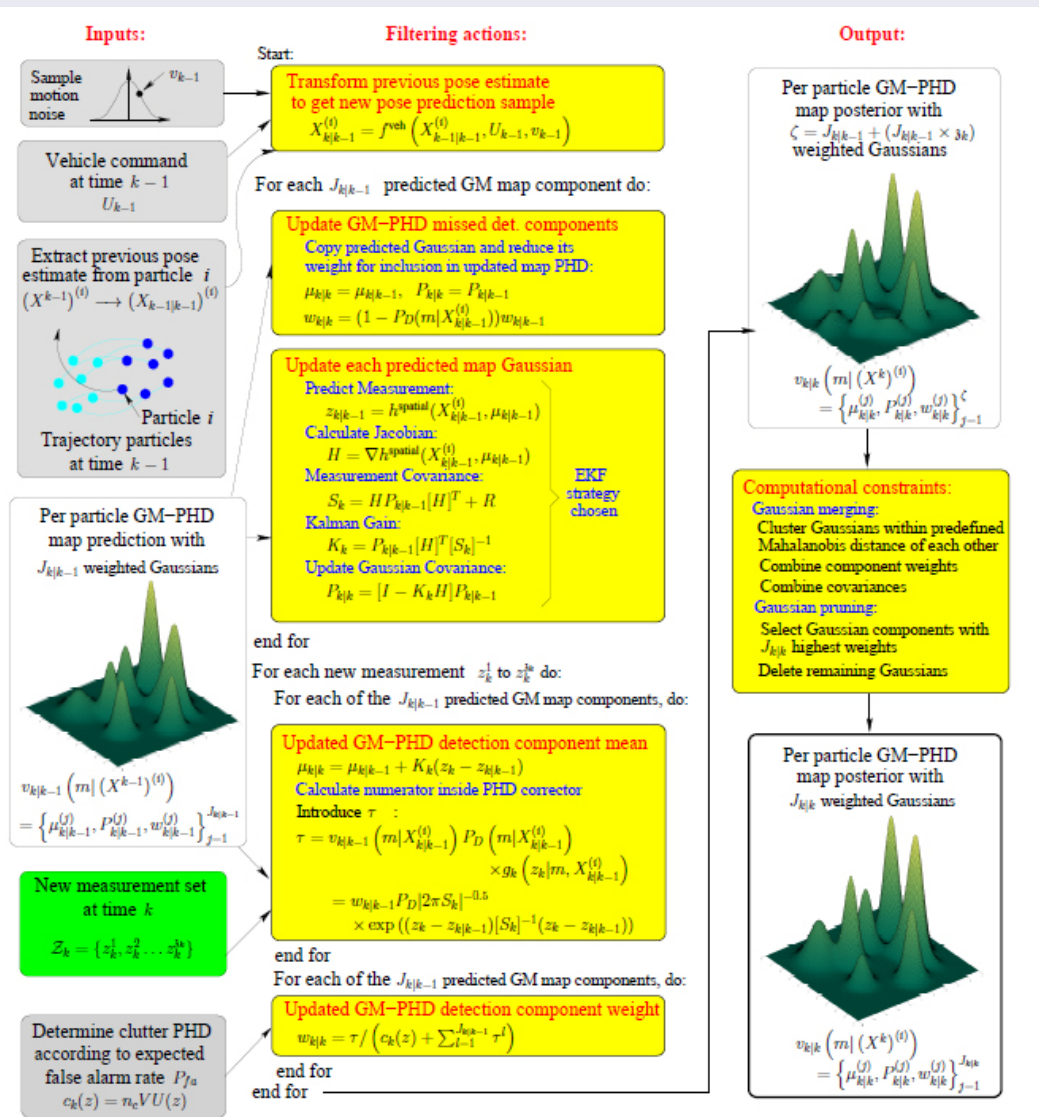
Posterior  
PHD

All predicted features weighted  
by their probs. missed detection

$$+ \underbrace{v_{k|k-1}(m|X^k) \sum_{z \in \mathcal{Z}_k} \frac{\Lambda(m|X_k)}{c_k(z|X_k) + \int_{\xi \in \mathcal{M}_k} \Lambda(\xi|X_k) v_{k|k-1}(\xi|X^k) d\xi}}$$

All predicted features, updated by the spatial locations of all  
the new measurements, and their probabilities of detection

# Implementing PHD SLAM – PHD Corrector



# Implementing PHD SLAM – Particle updates

RB PHD SLAM:

$$\left\{ \eta_{k-1}^{(i)}, (X^{k-1})^{(i)}, \underbrace{\left\{ \mu_{k-1|k-1}^{(j)}, P_{k-1|k-1}^{(j)}, w_{k-1|k-1}^{(j)} \right\}_{j=1}^{J_{k-1|k-1}^{(i)}}}_{\text{Prior GM-PHD}} \right\}_{i=1}^N$$

Prior GM-PHD  
 $v_{k-1|k-1}^{(i)}(m|(X^{k-1})^{(i)})$

$$\longrightarrow \left\{ \eta_k^{(i)}, (X^k)^{(i)}, \underbrace{\left\{ \mu_{k|k}^{(j)}, P_{k|k}^{(j)}, w_{k|k}^{(j)} \right\}_{j=1}^{J_{k|k}^{(i)}}}_{\text{Posterior GM-PHD}} \right\}_{i=1}^N$$

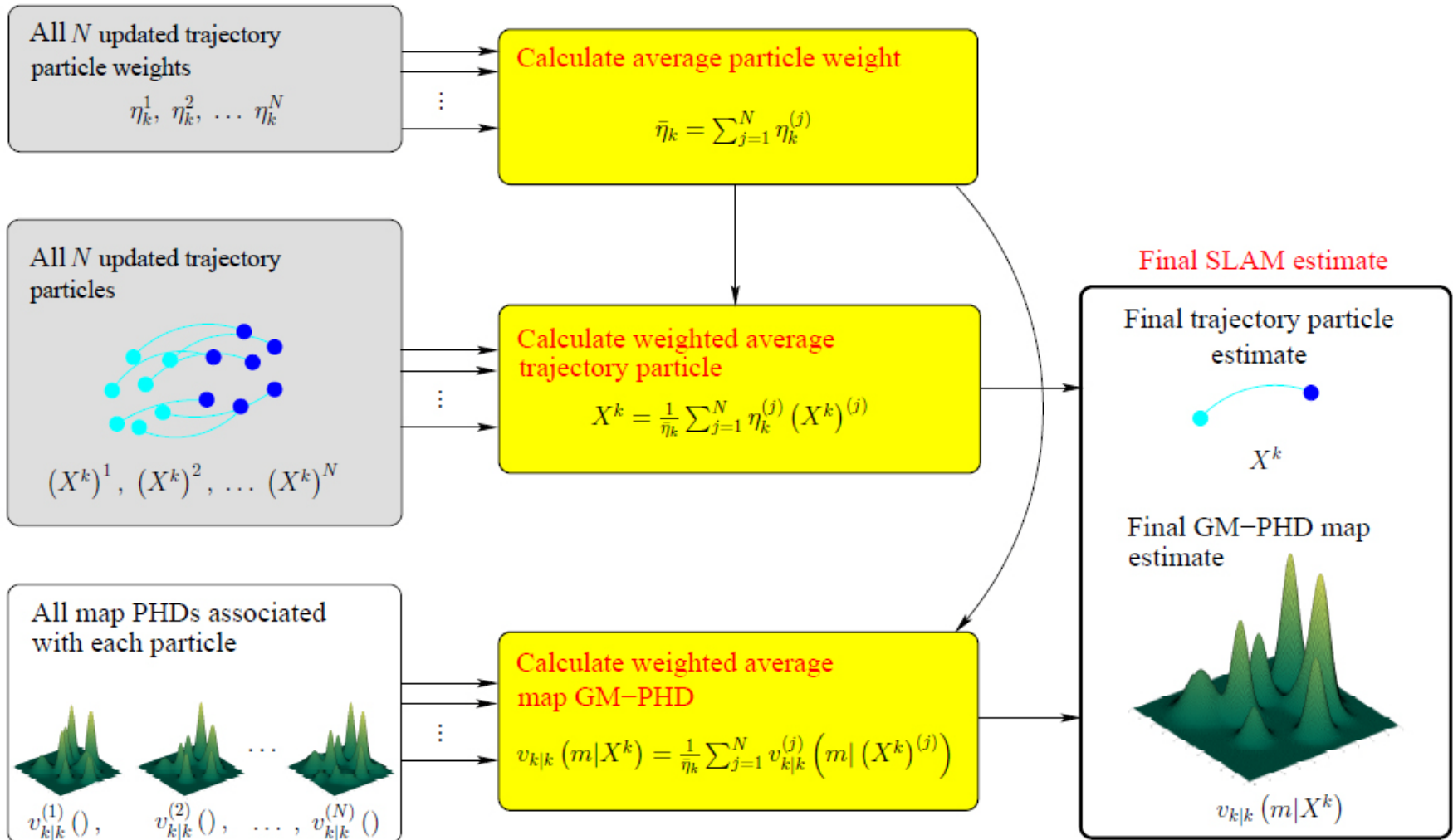
Posterior GM-PHD  
 $v_{k|k}^{(i)}(m|(X^k)^{(i)})$

# Implementing PHD SLAM – SLAM EAP Map

## Inputs:

## Filtering actions:

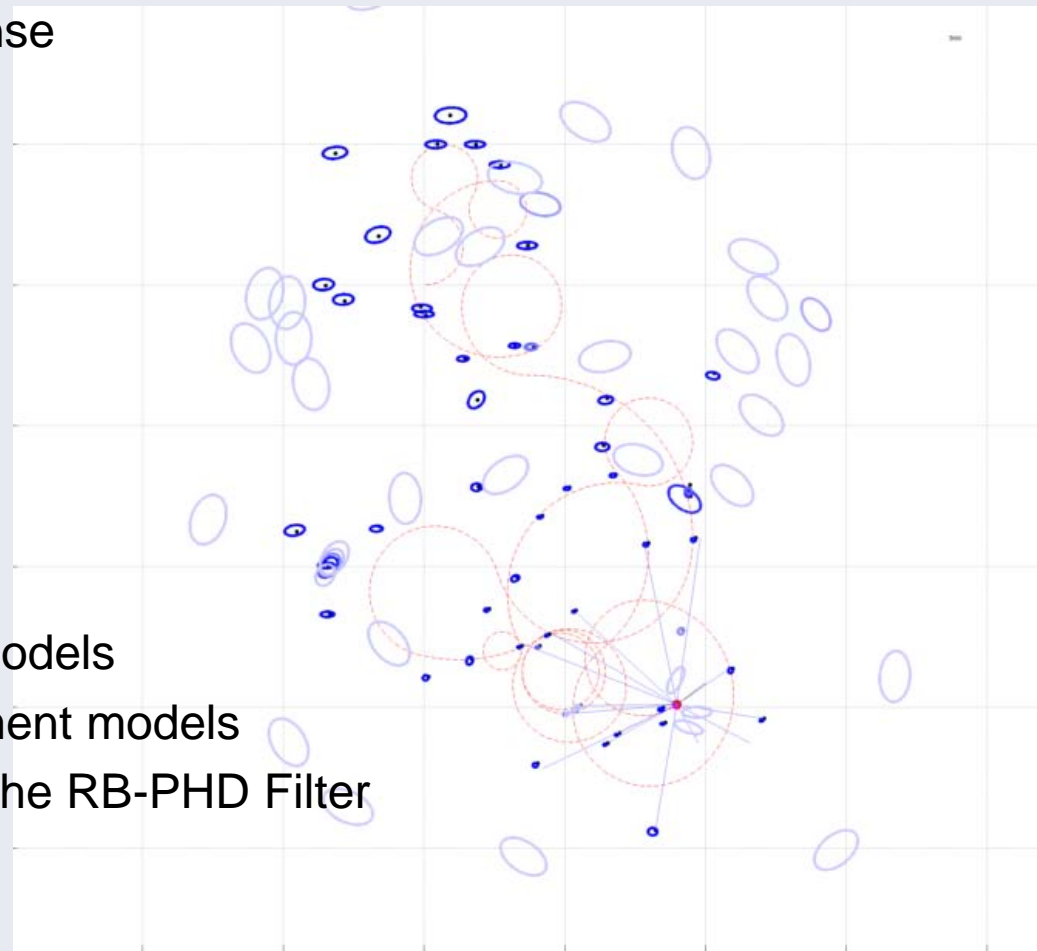
## Output:





# C++ Library for RFS SLAM

- Open source, with BSD-3 License
- Dependencies:
  - Boost::math\_c99 1.48
  - Boost::timer 1.48
  - Boost::system 1.48
  - Boost::thread 1.48
  - Eigen3
- Tested on Ubuntu 13.04
- Template library
  - Define your own process models
  - Define your own measurement models
- Includes an implementation of the RB-PHD Filter
- Includes a 2-d SLAM example
- Well documented
- Will be updated with new published research
- Download at: <https://github.com/kykleung/RFS-SLAM>



# Presentation Outline

## 1. What's in a Measurement:

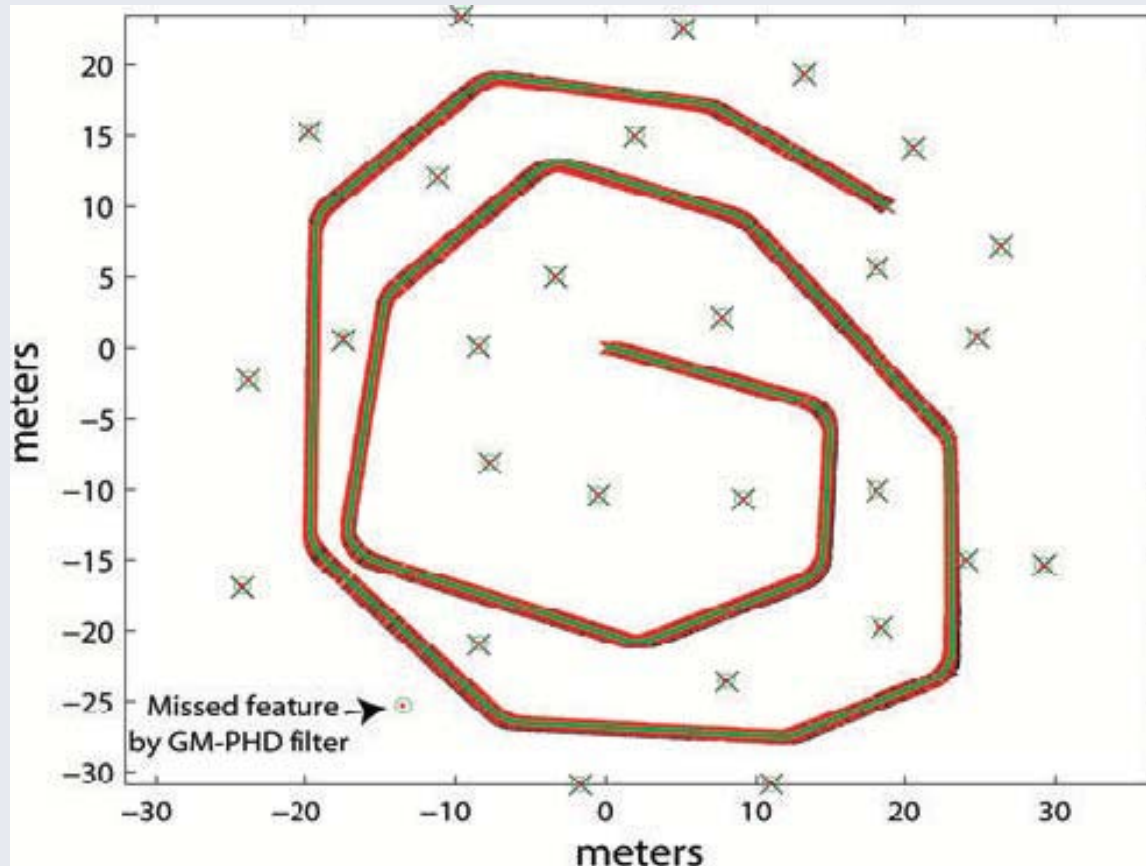
- Landmark Existence and Spatial Uncertainty
- Why Radar?

## 2. Simultaneous Localisation & Map Building (SLAM).

- A Random Finite Set (RFS) Approach.
- PHD SLAM – Implementation.

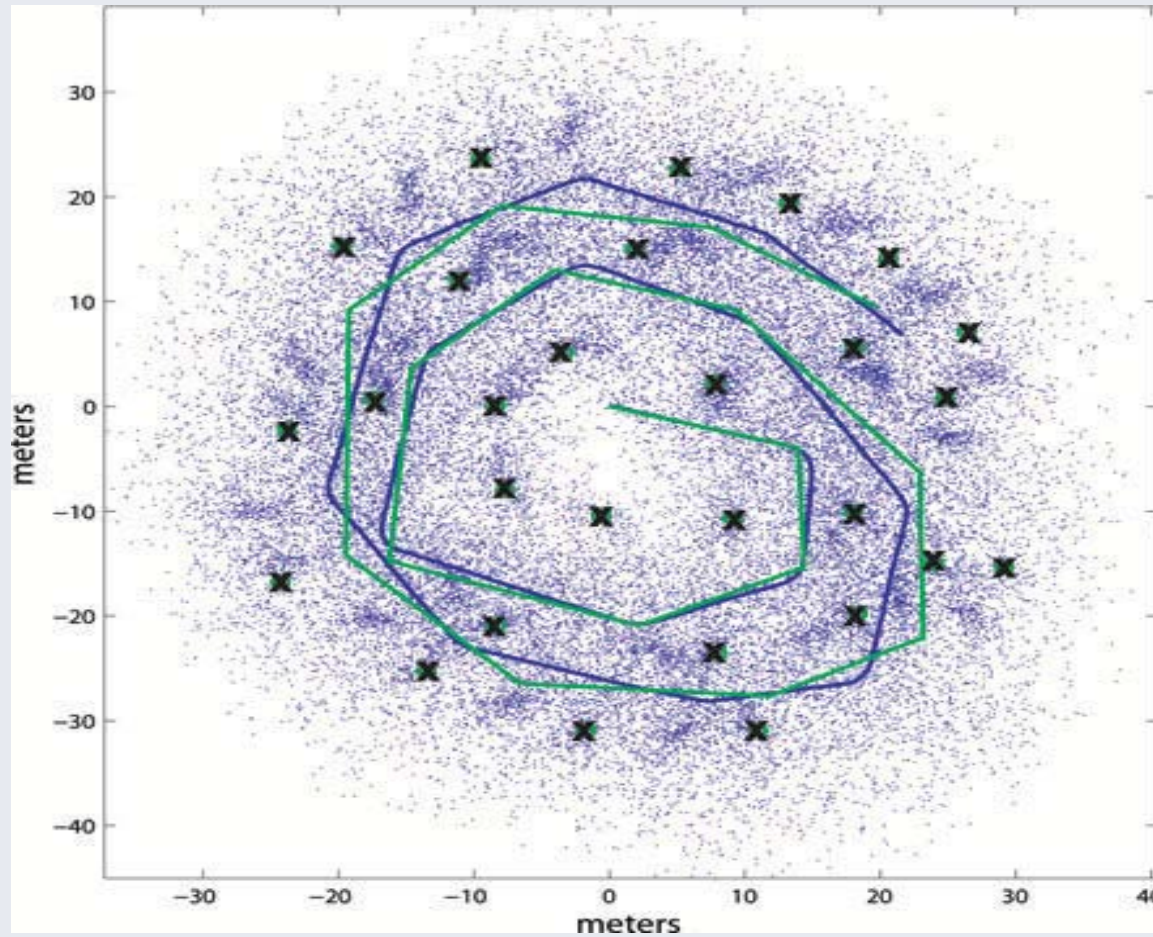
## 3. Comparison of Vector Based SLAM (MH-FastSLAM) and PHD-SLAM – Results.

# RFS Versus Vector Based SLAM



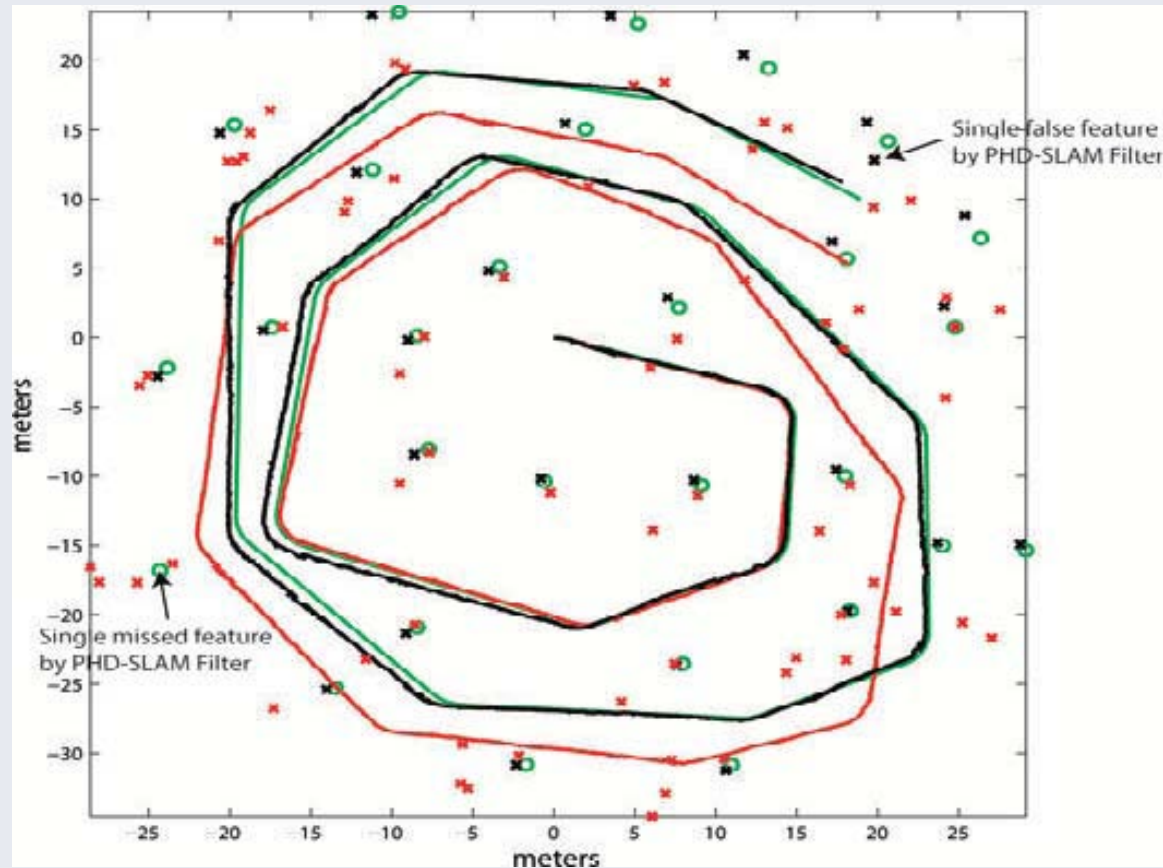
Comparative results for the proposed GM-PHD SLAM filter (black) and that of **FastSLAM (red)**, compared to **ground truth (green)**.

# RFS Versus Vector Based SLAM



The raw dataset at a clutter density of  $0.03 \text{ m}^{-2}$ .

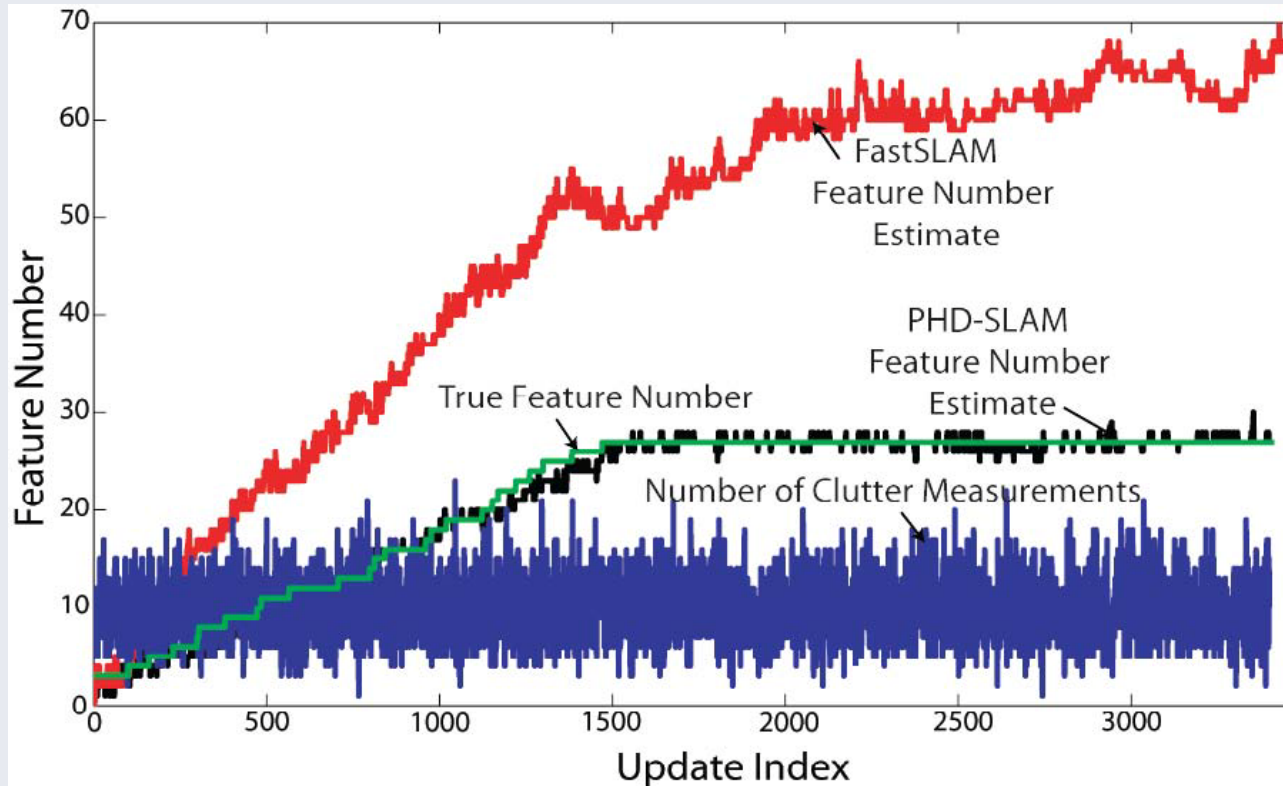
# RFS Versus Vector Based SLAM



The estimated trajectories of the GM-PHD SLAM filter (black) and that of **FastSLAM** (red). Estimated feature locations (crosses) are also shown with the **true features** (green circles).

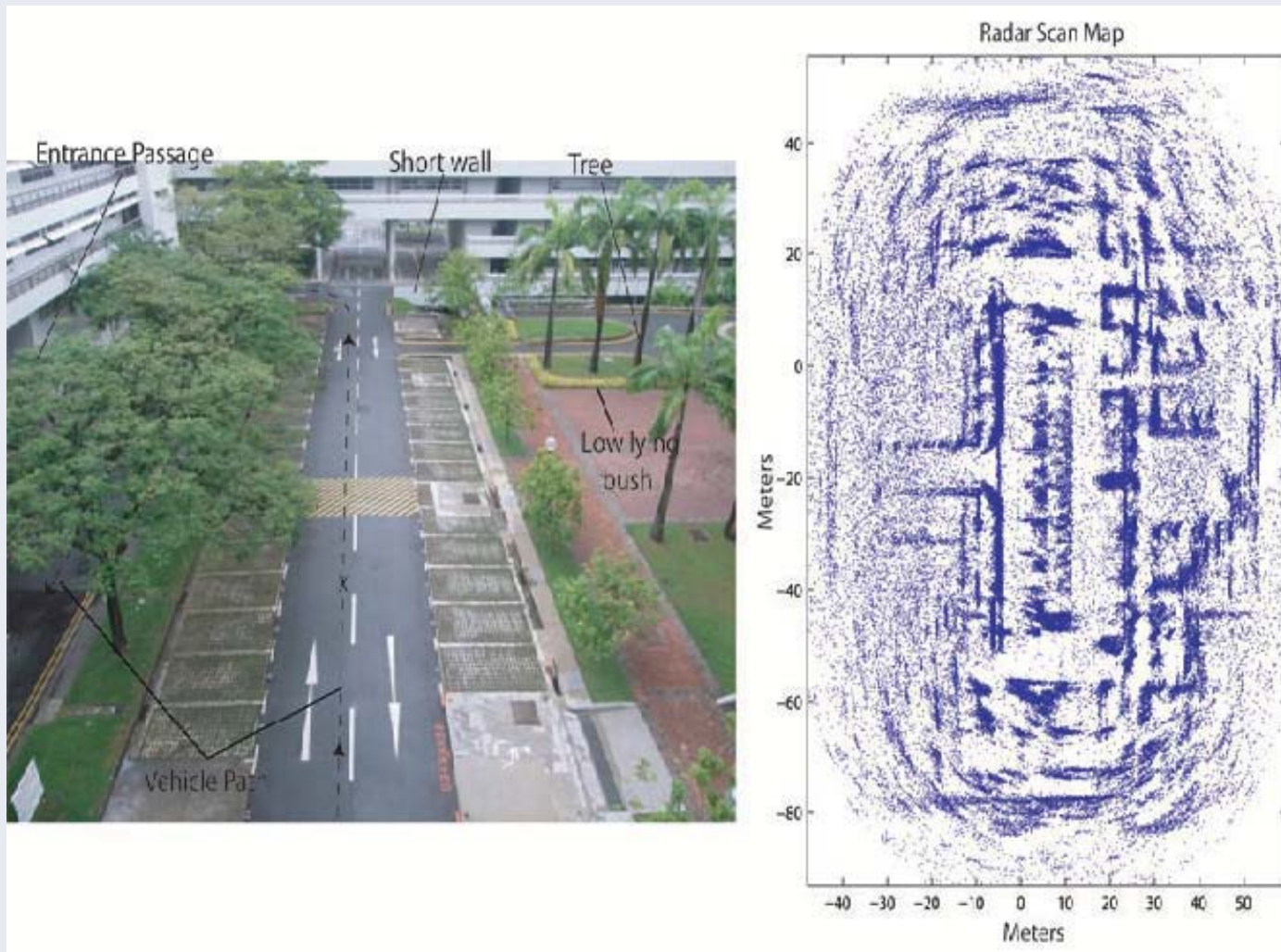


# RFS Versus Vector Based SLAM



Feature number estimates.

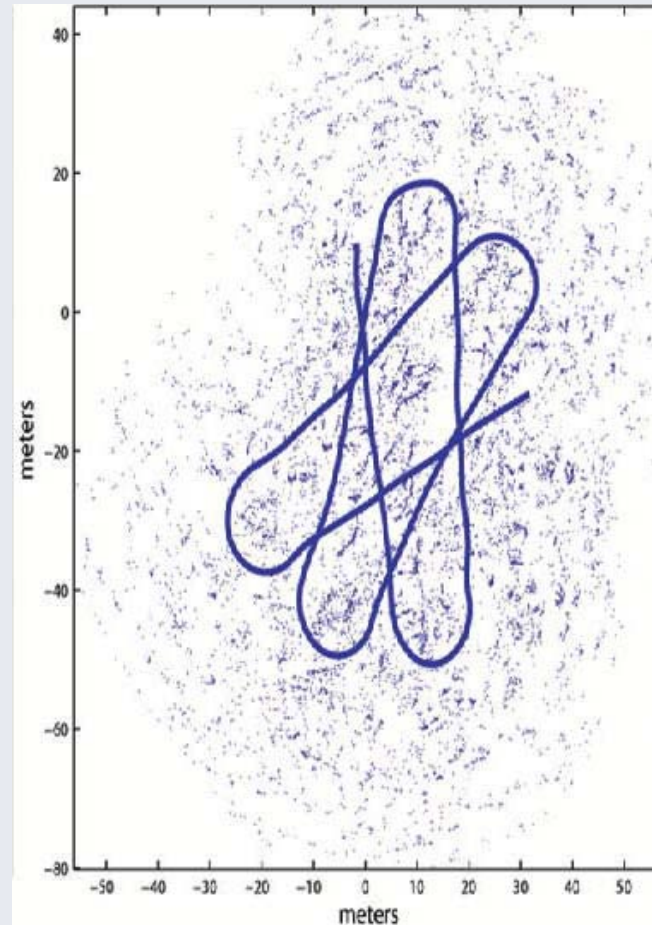
# RFS Versus Vector Based SLAM



Sample data registered from radar.

# RFS Versus Vector Based SLAM

SLAM input: Odometry path + radar data

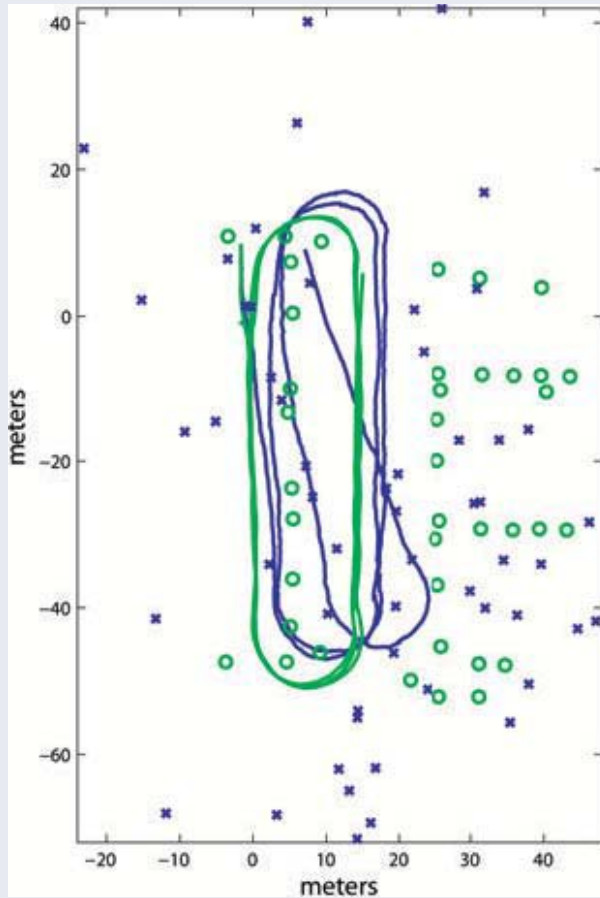


Extracted point feature measurements registered to odometry.

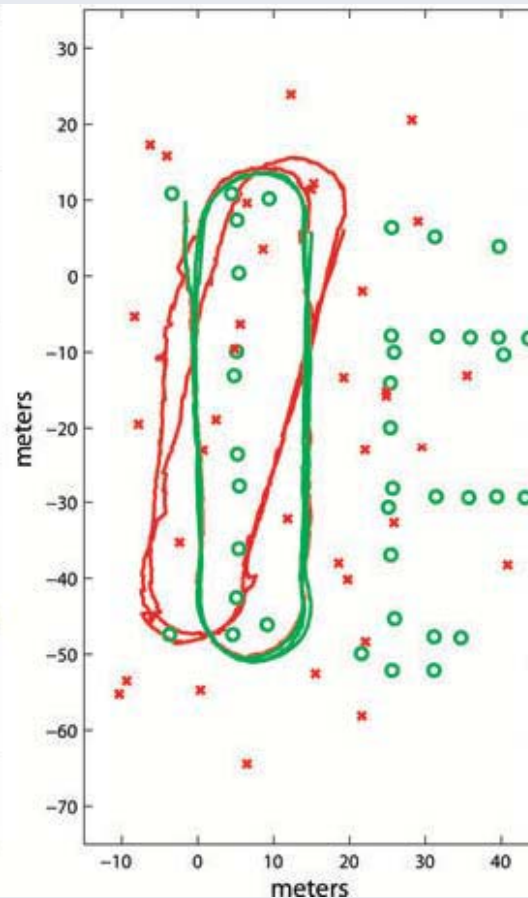


# RFS Versus Vector Based SLAM

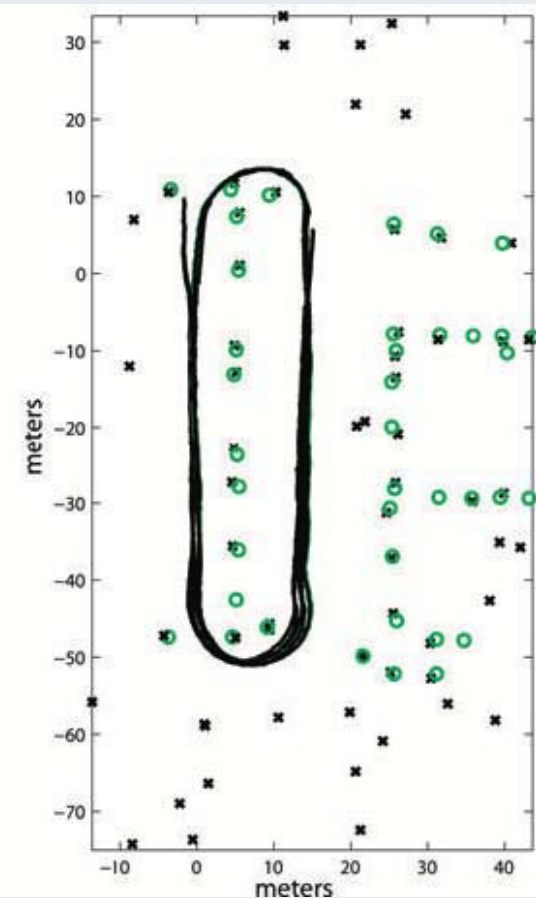
NN-EKF



FastSLAM



PHD-SLAM



EKF, FastSLAM and PHD-SLAM with Radar data.

# Singapore – MIT Alliance: CENSAM Project

- Environmental monitoring of coastal waters.
- Navigation and map info. necessary above/below water surface.
- Fusion of sea surface radar, sub-sea sonar data for combined surface/sub-sea mapping.



Autonomous Kayak Surface Vehicle with Radar

# Singapore – MIT Alliance: CENSAM Project



# Singapore – MIT Alliance: CENSAM Project

Coastal Mapping, Surveillance, HARTS / AIS verification

Mobile platform can remove blind spots from land-based radar.

Video: [CoastalModelling.avi](#)

Video: [CoastalandAIS.avi](#)

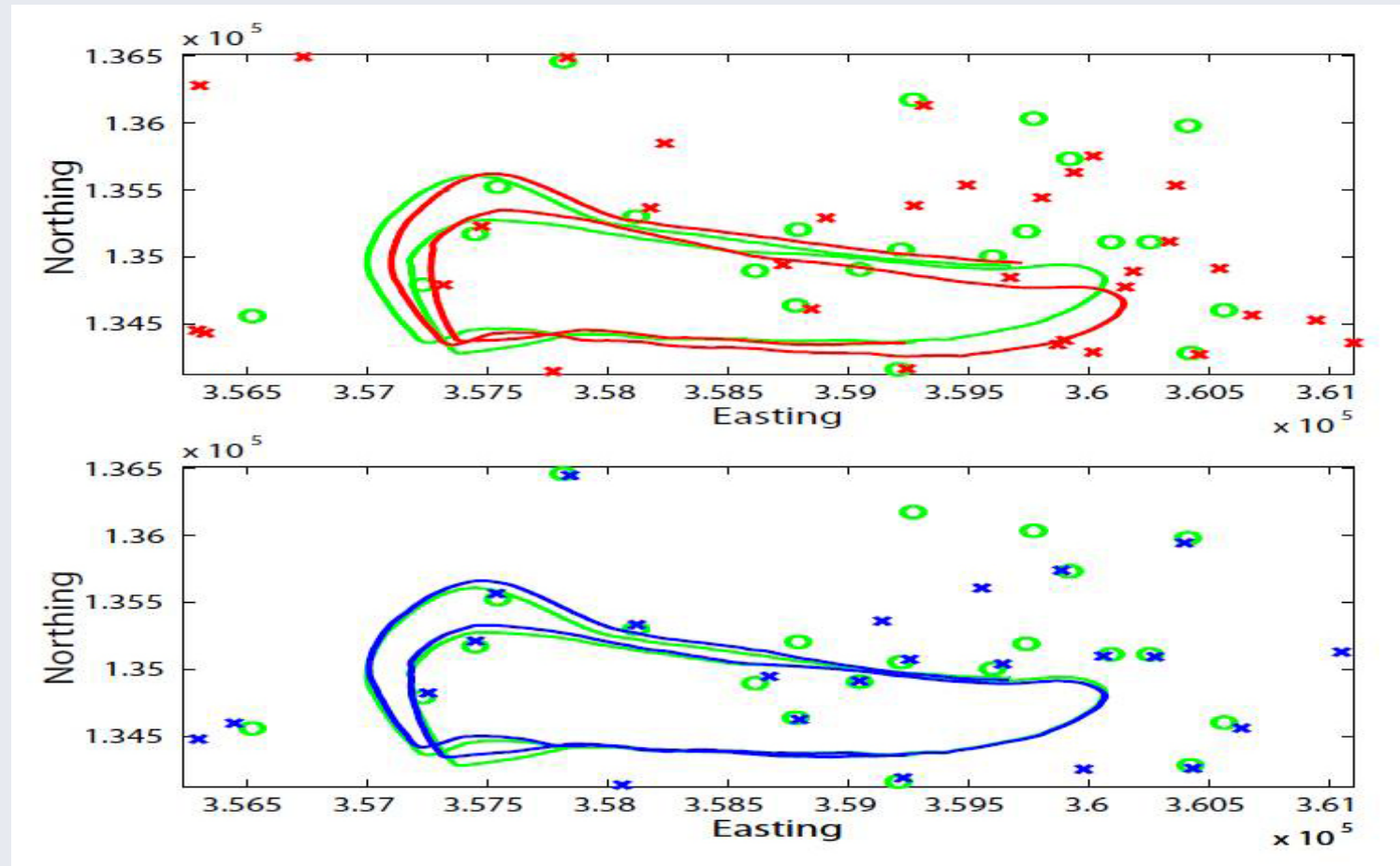


# RFS Versus Vector Based SLAM



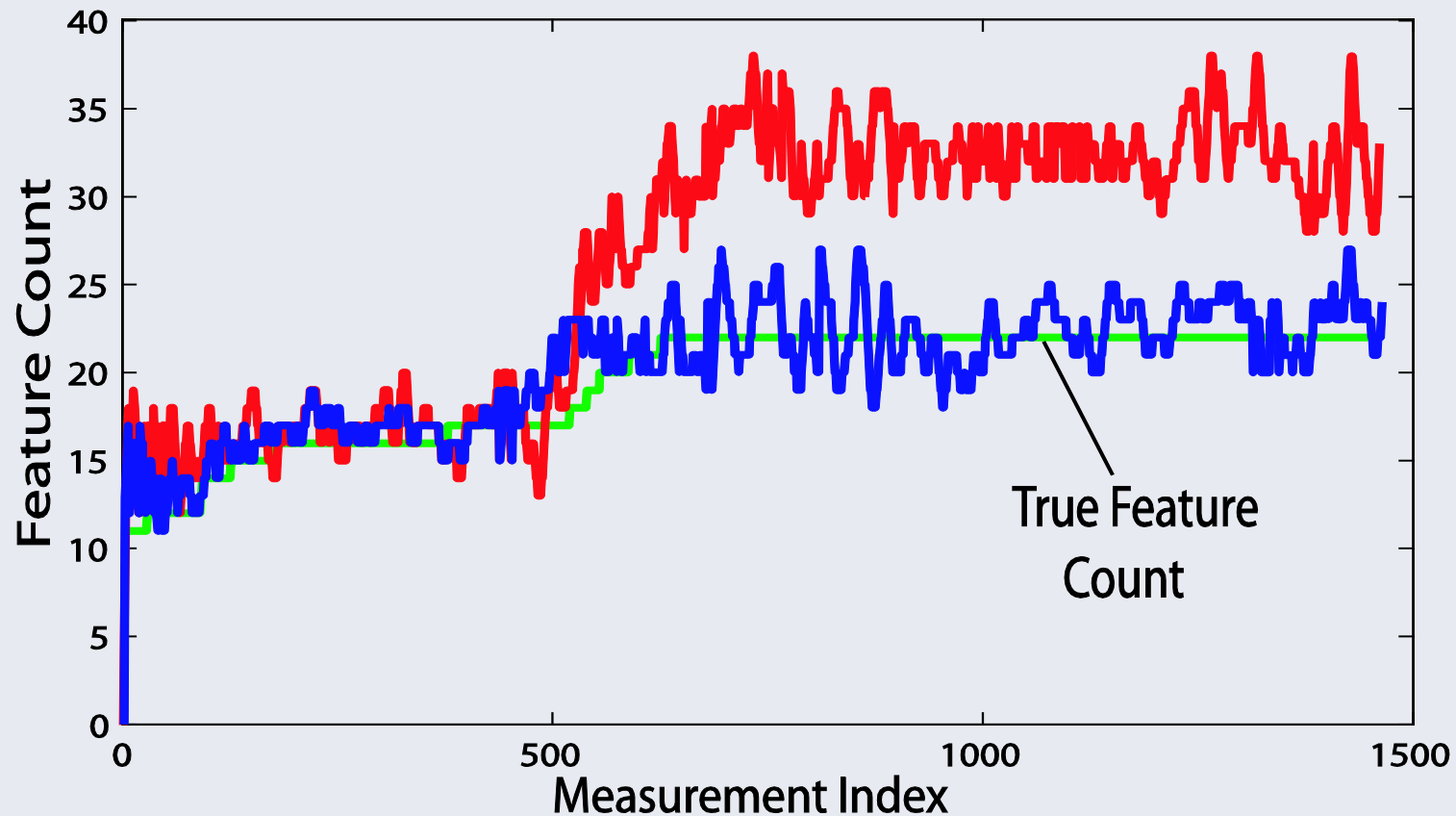
GPS Trajectory (Green Line), GPS point feature coordinates (Green Points), Point feature measurement history (Black dots).

# RFS Versus Vector Based SLAM



Top: Posterior MHT SLAM estimate (red).  
Bottom: Posterior RB-PHD SLAM estimate (blue).  
Ground truth (Green).

# RFS Versus Vector Based SLAM



(Red) MHT SLAM Feature Number estimate.

(Blue) PRB-PHD SLAM Feature Number estimate.

(Green) Actual Number to enter FoV at each time index.

# Conclusions & Future Work

1. Feature based maps more appropriately modelled as RFS than a random vector.
2. RFS Frameworks take into account detection as well as spatial uncertainty information.
3. PHD Filter approximation demonstrated – circumvents fragile data association necessary in vector based methods.
4. Superior results in cluttered environments.





**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



## Session I

### Localization & Mapping

- **Title: Large-Scale Dense 3D Reconstruction from Stereo Imagery**  
**Authors:** Pablo F. Alcantarilla, Chris Beall, Frank Dellaert
  
- **Title: Generation of Accurate Lane-Level Maps from Coarse Prior Maps and Lidar**  
**Authors:** Avdhut Joshi, Michael R. James

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# Large-Scale Dense 3D Reconstruction from Stereo Imagery

Pablo F. Alcantarilla, Chris Beall and Frank Dellaert

**Abstract**—In this paper we propose a novel method for large-scale dense 3D reconstruction from stereo imagery. Assuming that stereo camera calibration and camera motion are known, our method is able to reconstruct accurately dense 3D models of urban environments in the form of point clouds. We take advantage of recent stereo matching techniques that are able to build dense and accurate disparity maps from two rectified images. Then, we fuse the information from multiple disparity maps into a global model by using an efficient data association technique that takes into account stereo uncertainty and performs geometric and photometric consistency validation in a multi-view setup. Finally, we use efficient voxel grid filtering techniques to deal with storage requirements in large-scale environments. In addition, our method automatically discards possible moving obstacles in the scene. We show experimental results on real video large-scale sequences and compare our approach with respect to other state-of-the-art methods such as *PMVS* and *StereoScan*.

## I. INTRODUCTION

Structure from Motion (SfM) and visual Simultaneous Localization and Mapping (vSLAM) algorithms [1, 11] aim to recover a sparse 3D reconstruction and the estimated camera poses in large-scale environments. These methods track features between different frames and optimize 3D structure and camera poses in a nonlinear optimization which incorporates the geometric multi-view constraints between 3D structure, camera poses and image measurements. This nonlinear optimization problem is normally solved by using bundle adjustment variants [10].

Sparse 3D models do not provide enough detail to fully appreciate the underlying structure of the environment. To this end, there have been various efforts towards automated dense 3D reconstruction in the last few years [8, 14, 3, 4, 13, 6]. Automated dense 3D modeling facilitates scene understanding and has countless applications in different areas such as augmented reality, cultural heritage preservation, autonomous vehicles and robotics in general.

One of the key ingredients in dense 3D reconstruction methods is *Multi-View Stereo* (MVS) [16]. MVS algorithms can be roughly classified into four different categories: *deformable polygonal meshes* [2], requiring a visual hull model as an initialization; *voxel-based* [13], requiring a bounding box that contains the scene and the accuracy is limited by the voxel grid size; *patch-based* [3], requires reconstruction of a collection of multiple small surface patches, and *multiple depth maps* [8, 14, 6], that demands fusing multiple maps into a single global model. As mentioned in [3], MVS algorithms can also be thought of in terms of the datasets they

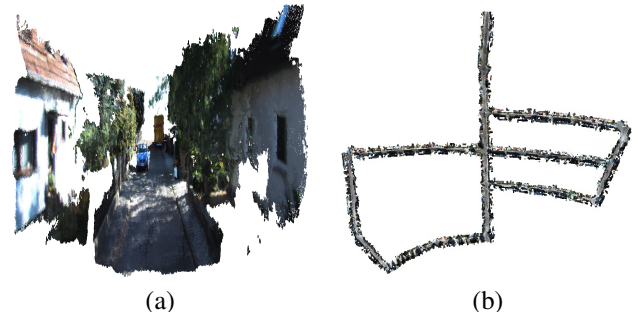


Fig. 1. Details (a) and aerial view (b) of dense 3D reconstruction results for a sequence of 2.2 Km and 2760 frames. The number of reconstructed 3D points is 5,770,704.

can handle: a single object, large-scale scenarios, crowded environments, etc. The choice of a particular MVS algorithm highly depends on the type of dataset and application of interest.

In this paper, we are interested in dense 3D reconstruction of large-scale environments using stereo imagery from a moving platform. We focus on the scenario of a stereo camera mounted on a vehicle or a robot exploring a large scene such as the one depicted in Figure 1. Large-scale environments pose new challenges to the dense 3D reconstruction problem such as large storage requirements and computational complexity.

We propose a novel MVS approach that efficiently combines the best of previous MVS approaches for our target application. Instead of fusing raw disparity maps from each stereo frame (which invariably yields large storage requirements), we use the dense disparity maps as an initialization for a patch-based surface reconstruction considering multiple views. In this way, taking advantage of the flexibility of patch-based methods, we can check for geometric and photometric consistency of each individual patch, which facilitates discarding moving objects from the final reconstruction. Then, we use efficient voxel grid filtering to down-sample the dense point cloud for dealing with large storage requirements.

Our algorithm makes the assumption that the stereo rig calibration and camera motion are already known. Stereo calibration can be obtained offline, while camera motion can be obtained either online by incremental egomotion estimation methods such as visual odometry [6] or with an offline bundle adjustment optimization including loop closure constraints. Our algorithm has the following advantages:

- Exploits dense disparity maps using efficient stereo matching.

<sup>3</sup> The authors are with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. {pfa3, cbeall13, frank}@cc.gatech.edu



- Performs efficient data association, checking for geometric and photometric consistency in a multi-view setup taking into account the uncertainty of stereo measurements.
- Handles large storage requirements due to the use of voxel grid filtering techniques.
- Is able to reject outliers and moving objects or obstacles in the scene.
- It is faster than state-of-the-art techniques.

## II. RELATED WORK

One of the most popular MVS techniques is the patch-based approach also known as PMVS [3]. This method builds a dense 3D reconstruction of a scene based on collections of multiple small surface patches. PMVS basically consists of three different steps: feature matching, expansion and filtering. In the matching step, a sparse 3D reconstruction of the scene is obtained from a set of 2D features. Then, in the expansion step, this sparse 3D point cloud is densified by an iterative procedure that estimates patch geometry by minimizing a photometric cost function. Finally, outliers are removed in the filtering step. PMVS is able to handle moving objects thanks to the photometric consistency check between different images. The main limitation of PMVS is that it is computationally very expensive due mainly to the patch expansion step. For large-scale scenarios, such as the ones we are interested in, PMVS would require several days to obtain dense 3D reconstructions even when using efficient clustering techniques for the set of input images [4].

Pollefeys *et al.* [14] presented an efficient approach for real-time 3D reconstruction from video of urban scenes. Their approach considers a system equipped with 8 cameras plus GPS/INS data mounted on a moving car, exploiting parallelization and GPU processing. They use plane-sweeping stereo as a stereo matcher for obtaining dense disparity maps from different views. Then, multiple depth maps are fused into a single global model by exploiting visibility information.

Recently, Newcombe *et al.* presented an impressive voxel-based dense 3D reconstruction approach from monocular imagery [13]. This approach works well for small scale environments and requires prior knowledge for a bounding box that contains the scene, limiting the accuracy of the 3D reconstruction to the voxel grid resolution.

The approach most similar to ours is the *StereoScan* system described in [6]. In this approach, the authors propose a dense 3D reconstruction pipeline fusing information from dense disparity maps obtained from stereo imagery. In order to deal with the large amount of data from the fusion of multiple disparity maps, the authors propose a greedy approach for solving the data association problem between two consecutive stereo frames. This greedy approach simply reprojects reconstructed 3D points of the previous frame into the image plane of the current frame. When a point projects to a valid disparity, the 3D points from the current and previous frames are fused by computing their 3D mean. Similar to our approach, the authors assume that the camera motion

is obtained from an independent visual odometry pipeline working in parallel. The main limitation of *StereoScan* is its greedy data association approach that considers only two consecutive frames without checking for geometric and photometric consistency between the reconstructed points. Limiting the data association to just two frames and without checking for geometric and photometric consistency introduces many noisy points into the final model, without being able to deal with possible artifacts caused by dynamic objects that will corrupt the 3D model. In addition, without filtering, the storage requirements quickly become prohibitive for large-scale scenarios.

## III. STEREO VISION

Stereo vision makes it possible to estimate 3D scene geometry given only two images from the same scene. We consider a conventional stereo rig in which two cameras are separated by a horizontal baseline. Rectification [9] considerably simplifies the stereo correspondence problem and allows for straight-forward computation of dense disparity maps, which form the base for the dense 3D reconstruction. Each value in the disparity map can be reprojected to a 3D point  $h_i = (x, y, z)^t \in \mathbb{R}^3$  with respect to the camera coordinate frame based on the projective camera equations:

$$\begin{aligned} z &= f \cdot \frac{B}{u_R - u_L} = f \cdot \frac{B}{d_u} \\ x &= \frac{z \cdot (u_L - u_0)}{f} \\ y &= \frac{z \cdot (v - v_0)}{f} \end{aligned} \quad (1)$$

where  $f$  is the camera focal length,  $(u_0, v_0)$  is the principal point,  $B$  is the stereo baseline and  $(u_L, v_L)$  and  $(u_R, v_R)$  are the stereo measurements in the left and right images, respectively. Note that for rectified stereo images  $v_L = v_R = v$ . The horizontal disparity  $d_u$  is the difference in pixels between the horizontal image projections of the same 3D point in the right and left images.

Similarly to [12], our sensor error model is composed of two parts: *pointing error*  $\sigma_p$  and *matching error*  $\sigma_m$ . Pointing error is the error in image measurements due to camera calibration inaccuracy, whereas matching error is due to the inaccuracy of the stereo matching algorithm. Given these values, we can compute the covariance matrix of the stereo measurements  $(u_L, v, d_u)^t$  in the disparity space as:

$$S_i = \begin{pmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_p^2 & 0 \\ 0 & 0 & \sigma_m^2 \end{pmatrix} \quad (2)$$

To obtain the covariance matrix  $P_i$  of the reconstructed 3D point  $h_i$  associated with stereo measurements  $(u_L, v, d_u)^t$ , the error is propagated from the 2D measurement space to 3D by means of linear uncertainty propagation as:

$$P_i = J_i \cdot S_i \cdot J_i^t \quad (3)$$

$$J_i = \begin{pmatrix} \frac{\partial x}{\partial u_L} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial d_u} \\ \frac{\partial y}{\partial u_L} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial d_u} \\ \frac{\partial z}{\partial u_L} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial d_u} \end{pmatrix} = \begin{pmatrix} \frac{B}{d_u} & 0 & -\frac{u_L B}{d_u^2} \\ 0 & \frac{B}{d_u} & -\frac{v B}{d_u^2} \\ 0 & 0 & -\frac{f B}{d_u^2} \end{pmatrix} \quad (4)$$

where  $J_i$  is the Jacobian of the 3D point  $h_i$  with respect to the stereo measurements  $(u_L, v, d_u)^t$ . The covariance matrix  $P_i$  estimates the uncertainty we can expect from a reconstructed 3D point. The uncertainty error grows quadratically with respect to the depth. We denote by  $w_i = \text{trace}(P_i)$  the trace of the covariance matrix  $P_i$ , and this is used as a measure of the uncertainty and as a weighting function for the reconstructed 3D points and color information in our MVS approach.

#### IV. DENSE 3D RECONSTRUCTION

Our approach assumes that stereo camera calibration and motion are known. In addition, we assume that images are given in a time-ordered sequence. Our approach is applicable in batch as well as incremental modes. Camera motion can be obtained online by egomotion estimation methods such as visual odometry or after an offline bundle adjustment optimization including possible loop closure constraints.

Our dense reconstruction approach has these main steps:

- 1) Dense stereo matching.
- 2) Patch-based reconstruction with multi-view geometric and photometric consistency analysis.
- 3) Outlier removal and voxel grid filtering.

We first select a subset of stereo keyframes from the input images to enforce a minimum distance in camera motion between frames which will be processed. This is to avoid adding redundant images which would not contribute any new information to the dense 3D model, but only increase computational complexity. Each stereo keyframe  $F_k$  with  $k = 1 \dots N$  comprises:

- Camera rotation,  $R^k \in SO(3)$ .
- Camera translation,  $\mathbf{t}^k \in \mathbb{R}^3$ .
- Left rectified RGB image,  $I_L^k: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .
- Normalized zero mean and unit variance left rectified RGB image,  $I_{Lnorm}^k: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .
- Right rectified RGB image,  $I_R^k: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .
- Disparity map,  $I_D^k: \mathbb{R}^2 \rightarrow \mathbb{R}$ .

The camera rotation and translation are defined such that a 3D point  $Y_i = (x, y, z)^t \in \mathbb{R}^3$  in the world coordinate frame can be transformed into the camera coordinate frame with:

$$h_i = R^k (Y_i - \mathbf{t}^k) \quad (5)$$

and assuming a pin-hole camera model, the projection of the 3D point  $h_i$  into the image plane is:

$$U_i = K (R^k (Y_i - \mathbf{t}^k)) \quad (6)$$

where  $K$  is the matrix representing the camera intrinsics and  $U_i = (u, v, 1)^t$  is the vector of pixel measurements in

homogeneous coordinates. In addition, each 3D point  $h_i$  has an associated RGB color vector  $c_i = (r, g, b)^t \in \mathbb{R}^3$ . Now, we will describe in detail each of the main steps in our MVS algorithm.

##### A. Dense Stereo Matching

Reliable stereo matching is critical in order to obtain accurate dense 3D point clouds. For this purpose, we use the Efficient Large-Scale Stereo Matching (ELAS) method which is freely available [5]. ELAS provides dense high quality disparity maps without global optimization, while remaining faster than many other stereo methods. For each stereo keyframe  $F_k$  we obtain a dense disparity map  $I_{Disp}^k$  image from the left and right rectified images.

##### B. Multi-View Geometric and Photometric Consistency

Considering that each stereo frame gives rise to thousands of 3D points, transforming all of these into a global 3D model would yield a very noisy reconstruction with lots of redundant points, and consequently storage requirements of prohibitive proportions for large scenarios. Therefore, to avoid the introduction of many redundant points we solve the data association problem between multiple stereo frames and verify geometric and photometric consistency for all points. This is in principle similar to the photometric consistency employed in MVS approaches [8, 3] with the key difference that for each pixel we rely on the depth provided by the stereo matching algorithm instead of minimizing a photometric cost function to find the globally optimal depth of each patch. Figure 2 depicts a graphical example of our multi-view stereo approach considering three views.

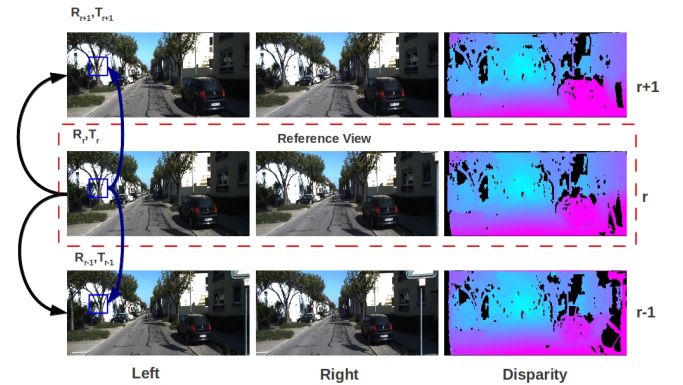


Fig. 2. Multi-view stereo approach checking geometric and photometric consistency. For a pixel  $p$  in the left reference image that has a valid disparity, we check first for geometric consistency between the different views. If the geometric consistency is successful, we perform the photometric consistency analysis.

We choose a central reference stereo keyframe  $F_r$  in a local neighborhood of  $m$  stereo views (in our experiments we consider  $m = 3, 5$ ). The index of the reference stereo keyframe  $r$  in the local neighborhood of  $m$  stereo frames is taken as the central view,  $r = (m - 1) / 2 + 1$ .

For each pixel  $p = (u_L, v_L)$  from the left reference keyframe image  $I_L^r$  which has a valid disparity  $d_u$ , we first

perform a geometric consistency check with respect to the other views in the neighborhood. We compute the 3D point  $h_i$  and the associated covariance  $P_i$  as described in Eq. 1 and Eq. 3 respectively. If the trace of the covariance matrix  $w_i$  is below some threshold  $T_{cov}$ , we then project the point  $h_i$  into the left images for the other  $m - 1$  views in the neighborhood. We then check that the projection of each 3D point  $h_i$  from the reference view into neighboring frames has a valid disparity and low uncertainty. Finally, we also check that the 3D difference between all reconstructed 3D points expressed in the world coordinate frame is within a threshold  $T_{dist}$ .

For all 3D points from the reference image which passed the geometric consistency check, our algorithm then proceeds to a photometric consistency check with respect to the other views in the neighborhood. For each pixel  $p = (u_L, v_L)$  from the left reference image, we compute the normalized cross correlation  $NCC(F_r, F_k, p)$  between a  $\mu \times \mu$  window centered on  $p$  and the corresponding windows centered on the projections in each of the views  $F_k$  with subpixel accuracy. For the  $NCC$  we use the textures from the normalized zero mean unit variance left images  $I_{Lnorm}^k$ . Similar to [8] we use a version of  $NCC$  for  $l$ -dimensional RGB color vectors with normalization per color channel.

$$NCC(c_0, c_1) = \frac{\sum_{j=0}^{l-1} (c_0(j) - \bar{c}_0) \cdot (c_1(j) - \bar{c}_1)}{\sqrt{\sum_{j=0}^{l-1} (c_0(j) - \bar{c}_0)^2 \cdot \sum_{j=0}^{l-1} (c_1(j) - \bar{c}_1)^2}} \quad (7)$$

The  $NCC$  returns a scalar value between  $[-1, 1]$ , where 1 indicates perfect correlation. We compute an average photometric score  $g(p)$  that comprises the sum of photometric scores for the pixel  $p$  between the reference image and the rest of views  $F_k \in V$  where the point is visible:

$$g(p) = \frac{1}{|V|} \sum_{k=r-n}^{k=r+n} NCC(F_r, F_k, p) \quad (8)$$

where  $n = (m - 1)/2$  for the sake of brevity, and  $|V|$  denotes the number of views where the point  $p$  is predicted to be visible, i.e. the number of views for which the point passed the geometric consistency check. If the mean photometric score  $g_p$  exceeds a threshold value  $T_{photo}$  and  $|V|$  is 3 or greater, we proceed to fuse the 3D point with respect to the world coordinate frame and color information into the dense reconstruction as the following weighted average:

$$Y_i = \frac{\sum_{k=r-n}^{k=r+n} w_{i,k} \cdot Y_{i,k}}{\sum_{k=r-n}^{k=r+n} w_{i,k}}, \quad c_i = \frac{\sum_{k=r-n}^{k=r+n} w_{i,k} \cdot c_{i,k}}{\sum_{k=r-n}^{k=r+n} w_{i,k}} \quad (9)$$

where  $w_{i,k}$  is the uncertainty weight of the reconstruction of point  $h_i$  from the view  $k$ . Similarly,  $Y_{i,k}$  and  $c_{i,k}$  denote the 3D point with respect to the world coordinate frame and color information for point  $i$  from view  $k$ .

In order to reduce computational complexity and to avoid adding redundant 3D points as the neighborhood window

slides through the sequence, we keep track of image projections of already reconstructed 3D points in their respective images using a mask. In this way, for each new reference view, we check the visibility masks to reconstruct only those 3D points which were not reconstructed previously.

### C. Outliers Removal and Voxel Grid Filtering

Once we have computed a dense 3D point cloud from a reference stereo keyframe  $F_r$ , we filter possible outliers by means of a *radius removal* filter. This filter removes those 3D points that do not have at least some number of neighbors within a certain range. Then, in order to reduce the computational burden and storage requirements, we downsample the 3D point cloud using a voxel grid filter that fits to the dimensions of the input point cloud. In each voxel, the 3D points are approximated with their centroid, representing more accurately the underlying surface. Once we have processed one stereo keyframe, we repeat the same procedure for the next keyframe until the sequence finishes. After processing all stereo keyframes, we apply the voxel grid filter over the whole dense 3D point cloud to fuse the 3D points into a global voxel grid structure.

## V. RESULTS

We use the KITTI visual odometry RGB dataset [7] for the evaluation of our dense 3D reconstruction approach. This dataset consists of stereo imagery with accurate stereo calibration. The images have a resolution of  $1241 \times 376$  pixels. For the greedy projection surface reconstruction and the radius removal and voxel grid filters, we use the efficient implementations from the Point Cloud Library (PCL) [15].

Typical values for the parameters in our method are:  $\sigma_p = 0.5$  pixels,  $\sigma_m = 1.0$  pixel,  $T_{cov} = 0.5$ ,  $T_{dist} = 0.5$  m,  $T_{photo} = 0.7$  and patch size  $7 \times 7$  pixels. All timing results were obtained with an Intel Core i7-3770 CPU.

### A. Comparison to PMVS and StereoScan

We compare our dense 3D reconstruction approach to PMVS and StereoScan. For PMVS we use the PMVS2 implementation<sup>1</sup>. We configure PMVS options so that it processes images in sequence, enforcing the algorithm to use only images with nearby indices to reconstruct 3D points. For the StereoScan case we use our own implementation and fuse the information between two corresponding 3D points if both disparities are valid and the distance between reconstructed 3D points is below the threshold  $T_{dist}$ . In our method we consider  $m = 3$  views, a voxel grid resolution of 5 cm and a photometric consistency threshold  $T_{photo} = 0.7$ . This value is also used in PMVS.

Figure 3(a) depicts a comparison of our method to PMVS and StereoScan showing the computation time versus the number of input images for the first sequence in the KITTI dataset. We observe that our method is the fastest one. The reason why it is faster than StereoScan is due to the use of a visibility mask, keeping track of image projections of the reconstructed 3D points in their visible images, reducing

<sup>1</sup>Available from: <http://www.di.ens.fr/pmvs/>



computational complexity. PMVS is highly time consuming even for a small set of images. This is because it tries to optimize the 3D position and normal of each patch in each reference image by minimizing a cost function based on the photometric error in a multi-view setup. In contrast, our method and StereoScan use the available 3D geometry from the disparity map and perform data association between different views, which is faster than running an iterative non-linear optimization per patch.

Figure 3(b) shows a comparison of our method to PMVS and StereoScan showing the number of reconstructed 3D points versus the number of input images. The number of reconstructed 3D points in the StereoScan case was scaled down by a factor of ten for clarity reasons. StereoScan produces large amount of 3D points, some of which are noisy and redundant. In large-scale environments the storage requirements of StereoScan can become prohibitive. In contrast, our method returns a more reasonable number of 3D points. In addition, one can control the output number of 3D points with the photometric threshold and the voxel grid resolution. PMVS returns the lowest number of reconstructed 3D points. PMVS is more targeted to *Photosynth-type* systems [1], where there is a large number of images from the same object in a small area. In this case, redundant viewpoints improve the estimation of the patch geometry.

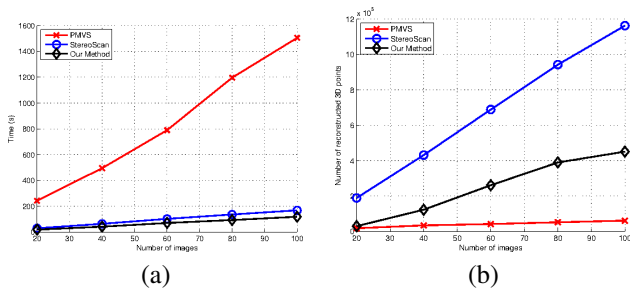


Fig. 3. Comparison to PMVS and StereoScan: (a) Computational time vs number of images (b) Number of reconstructed 3D points vs number of images. Note that the number of reconstructed 3D points that is reported for StereoScan is scaled by a factor of ten for clarity reasons.

Table I shows information about the number of reconstructed 3D points at each level of our MVS approach considering two different photometric thresholds  $T_{photo} = 0.2$  and  $T_{photo} = 0.8$ . In addition, we also show the percentage between the number of accepted points at each step and the number of points that have a valid disparity for each stereo frame. We can observe that in both cases the number of 3D points obtained after the voxel grid filtering is a small fraction of the original number of points facilitating storage requirements in large-scale scenarios.

### B. Detection of Moving Objects

One of the nice properties of PMVS and similar patch-based methods such as ours, is that they can discard specular highlights or moving objects in the scene (pedestrians, cars, etc.). Assuming that the surface of an object is Lambertian, the photometric score function  $g(p)$  will give low scores for

Step	Our method	Our method
	$T_{photo} = 0.2$	$T_{photo} = 0.8$
# Points Disparity	323,420	323,420
# Points Geometric	133,334	133,334
% Accepted	41.23	41.23
# Points Photometric	57,675	9,310
% Accepted	17.83	2.88
# Points Fusion	8,851	1,885
% Accepted	2.74	0.58

TABLE I  
AVERAGE NUMBER OF RECONSTRUCTED 3D POINTS PER STEP AND PERCENTAGE OF ACCEPTED POINTS WITH RESPECT TO POINTS WITH VALID DISPARITY PER STEREO FRAME.

areas which have specular highlights or moving objects in the image, and therefore these points will not be added to the final 3D model. Figure 4 depicts an example of one sequence where there are several moving objects (cars). StereoScan fails to reject these points and adds them to the final model, creating artifacts in the final model. This occurs because StereoScan only considers two consecutive stereo frames for data association based on the disparity information. In such a limited multi-view setup moving objects are not detected properly. In contrast, our method and PMVS are able to discard those 3D points from the final model.

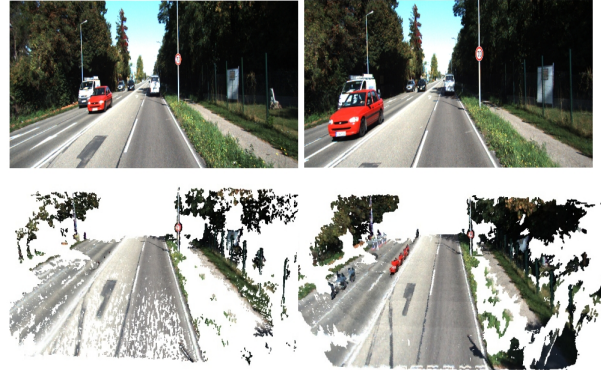


Fig. 4. Detection of moving objects. Top: Two frames from a sequence where there are moving objects in the scene. Bottom left: view of the dense 3D reconstruction with our method. Bottom right: view of the dense 3D reconstruction with StereoScan. Notice how artifacts due to the moving objects are introduced in the final model.

### C. 3D Reconstruction Results

Figure 5 depicts some dense 3D large-scale reconstruction results from different viewpoints. It can be noticed that the dense 3D point clouds contain high level of detail, enough for visualization purposes.

### D. Timing Evaluation

Table II shows average timing results for the most important operations in our MVS approach. We can observe that on average obtaining one incremental update to the dense



Fig. 5. Details of large-scale dense 3D reconstruction results. The cars in the point clouds correspond to static objects in the environment.

3D point cloud takes slightly less than 2 seconds for one stereo view. This time could be further reduced by using GPU implementations since the operations in the multi-view 3D reconstruction approach are independent per pixel.

Step	Time (ms)
Stereo Matching	157.74
RGB Normalization	2.51
Multi-view 3D (m=3)	1303.28
Outlier Removal	351.32
Voxel Grid Filter	2.76
<b>Total</b>	<b>1811.61</b>

TABLE II  
COMPUTATION TIMES IN MS FOR THE MAIN STEPS OF OUR MVS  
APPROACH.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel MVS approach for dense 3D reconstruction in large-scale environments using stereo imagery. We have shown that efficiently fusing disparity maps, while checking geometric and photometric consistency of patches in a multi-view setup, yields detailed 3D models with low storage requirements. In the future we are interested in possible applications of the dense 3D models for planning and scene understanding.

## REFERENCES

- [1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Intl. Conf. on Computer Vision (ICCV)*, 2009.
- [2] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. *Intl. J. of Computer Vision*, 81(1):53–67, 2009.
- [3] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(8):1362–1376, 2010.
- [4] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1434–1441, 2010.
- [5] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Asian Conf. on Computer Vision (ACCV)*, pages 25–38, 2010.
- [6] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.
- [7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.
- [8] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2402–2409, 2006.
- [9] R. Hartley. Theory and practice of projective rectification. *Intl. J. of Computer Vision*, 35:115–127, 1999.
- [10] Y. Jeong, D. Nistér, D. Steedly, R. Szeliski, and I. S. Kweon. Pushing the envelope of modern methods for bundle adjustment. *IEEE Trans. Pattern Anal. Machine Intell.*, 34(8):1605–1617, 2012.
- [11] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. REAL: A system for large-scale mapping in constant-time using stereo. *Intl. J. of Computer Vision*, 94(2):198–214, 2011.
- [12] D. R. Murray and J. J. Little. Environment modeling with stereo vision. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3116–3122, 2004.
- [13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2320–2327, 2011.
- [14] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénus, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *Intl. J. of Computer Vision*, 78(2-3):143–167, 2008.
- [15] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [16] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. S. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528, 2006.

# Generation of Accurate Lane-Level Maps from Coarse Prior Maps and Lidar

Avdhut Joshi<sup>1</sup> and Michael R. James<sup>1</sup>

**Abstract**—While many research projects on autonomous driving and advanced driver support systems make heavy use of highly accurate lane-level maps covering large areas, there is relatively little work on methods for automatically generating such maps. Here, we present a method that combines coarse, inaccurate prior maps from OpenStreetMap (OSM) with local sensor information from 3D Lidar and a positioning system. The algorithm leverages the coarse structural information present in OSM, and integrates it with the highly accurate local sensor measurements. The resulting maps have extremely good alignment with manually constructed baseline maps generated for autonomous driving experiments.

## I. INTRODUCTION

Many existing approaches [1], [2] for autonomous driving systems make heavy use of maps that encode lane-level information at high levels of precision. The lane-level information is used in a variety of situations, from generating smooth trajectories for path planning [1], to predicting the behavior of other vehicles [3], [4], and for planning and reasoning about proper behavior in intersections [5]. In many cases, such maps are generated either through a tedious manual annotation process [2], or by driving the exact lane layout with a test vehicle [6] or by analyzing a collection of GPS tracks [7]. These methods require significant amounts of manual work, either through annotation or in the amount of data collection required. In this paper, we present a method for overcoming these limitations, which opens the door for creating more robust systems with the ability to create their own high-fidelity maps with less manual work.

While there has been extensive work in lane detection [8], [9], [10], [11] and lane tracking [12], [13] using a variety of sensors, we are unaware of any previous work that combines coarse prior information with sensor data to result in consistent, high-quality, large-scale maps such as we generate.

This paper presents a method for estimating the structure and layout of lanes within real-world road scenes, by combining

- 1) structurally informative, easily obtained, coarse maps used as prior information from the Open Street Map (OSM) [14] project, with
- 2) sensor data obtained from a test vehicle with 3D Lidar as well as a high-precision positioning system.

We are typically able to infer lane structure for an entire road by driving the road once, and not once for each lane.

<sup>1</sup>Toyota Research Institute, North America, Toyota Technical Center, Ann Arbor, MI 48105, USA avdhutj@gmail.com, michael.r.james@gmail.com



Fig. 1: TRI-NA test vehicle, showing the sensors used for various advanced safety and autonomous driving research projects. Details of the sensors used for lane mapping are in the text.

This method is robust to differing styles of driving of the test vehicle, as the algorithm is based only on road-paint detection using intensity returns from the Lidar, and does not use the path of the test vehicle in lane estimation. Specifically, we contribute the following:

- Modeling of the inference problem that combines coarse structural prior map data with precise Lidar measurements in a (number of) tractable inference algorithms
- Algorithm for MAP inference of lane positions and identity management
- Evaluation on a real-world dataset

## II. APPROACH

The lane estimation algorithm is based on a dataset that has first been refined using Slam in order to ensure consistent position estimates on loop closures. In Section III, we provide an overview of this algorithm; a variant of GraphSlam [15] run on datasets gathered from multiple runs on only partially overlapping roads, see Figure 2. This serves to both align the laser scans, as well as to ground the data in a consistent and physically meaningful reference frame. However, our main contribution, the lane estimation algorithm, can be run on any dataset that has been refined using this or similar methods.

The lane estimation algorithm Section IV is comprised of two phases, the first of which is the generation of mid-level lane features, which then serve as observations for the measurement function within the second phase which uses particle filtering to estimate the lanes. We experiment with





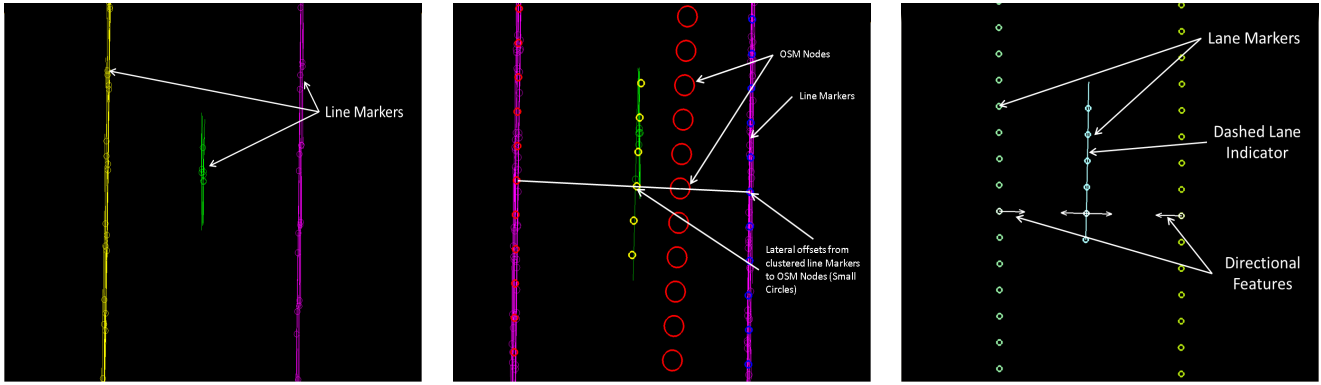


Fig. 3: Lane Features. At left are the lane marker responses from Laplacian filter on the Velodyne intensity data, shown as line segments. In the middle are results of clustering the responses perpendicularly and tying the results to OSM nodes, defining the observations (small circles) for the PF. On the right are the derived binary features. See text for details.

purpose of this step is to generate additional binary features for each lane marker. Some groups of lane markers, such as those corresponding to solid, well-painted lines, will extend for a long longitudinal distance (tens or hundreds of meters) on rural or semi-rural roads, while in other cases such as dashed lines, or areas with many intersections, the groups will be short, on the order of meters.

Given these groupings, three additional features are computed which prove to be useful for lane estimation. First we calculate two binary features which encode on which side(s) of the lane marker a lane can exist (e.g. for a right most lane marking, a lane on the right cannot exist). We compute these binary features namely, *has-l* and *has-r* by looking at the entire lane segment data. For the entire lane segment, we count the number of lane marking observations ( $z_i$ ) that lie on the either side ( $c_k^l$  and  $c_k^r$ ). Then,

$$has-j = (c_k^j \geq \delta_c), j \in \{l, r\}$$

where  $\delta_c$  is a threshold parameter. The third binary variable encodes whether a lane marker is dashed. We first filter out all the lanes which are bigger than a standard dashed lane found in US. Then we connect lane marker groups which are at a set distance apart and have similar orientation. These are marked as a dashed.

The above binary features illustrated in Figure 3 (Right), give important cues to interpreting the lane data, as will be shown in the development of the measurement function for the particle filters described in Section IV-C

### B. Particle Filtering

We have experimented with multiple approaches for particle filtering for this domain. In the following sections, we outline these approaches starting here with some basic definitions that are common to all. As noted above, the particle filter evolution is based on the structure of the OSM nodes, with successive steps in the filter transitioning from one OSM node to the next. The state of each particle is based on its relation to the OSM node (which then ties it to a physical location). With this in mind, we now derive

our filtering equations starting from a basic definition of the state of the map that we want to estimate:

$$X_n : \{x_n^1, x_n^2 \dots x_n^m\};$$

where  $m$  is number of lanes estimated at  $n^{th}$  node in OSM way and  $x_n^i$  is state of the lane estimate. The state of each lane is its offset from the OSM node and its width  $\{o_n^i, w_n^i\}$ . Using the observations  $z_n \rightarrow \{\text{Lane markers observed at } n^{th} \text{ OSM node}\}$  from Section IV-A, our belief state is

$$Bel(x_n) = p(x_n | z_n, z_{n-1} \dots z_0) \quad (1)$$

Using recursive Bayes filtering as defined in [17] for equation (1) we have

$$Bel(x_n) \propto p(z_n | x_n) \int p(x_n | x_{n-1}) Bel(x_{n-1}) dx_{n-1} \quad (2)$$

To implement a particle filter, we need to estimate the quantities  $p(z_n | x_n)$  and  $p(x_n | x_{n-1}) Bel(x_{n-1})$ . For all algorithms, we represent  $Bel(x_n)$  as a set of  $m$  weighted particles

$$Bel(x) \approx \{x^{(i)}, \phi^{(i)}\}_{i=1, \dots, m}$$

where  $x^{(i)}$  is a sample of state (lane estimate) and  $\phi^{(i)}$  is a non-negative parameter called the importance factor or weight. The other necessary quantities are described in depth in each of the following sections.

### C. Conventional Particle Filter

Our implementation of the conventional particle filter follows the following three steps:

- 1) Sampling: Sample  $x_{n-1}^{(i)} \sim Bel(x_{n-1})$  from the weighted sample set representing  $Bel(x_{n-1})$ .
- 2) Proposal Distribution: We sample  $x_n^{(i)} \sim p(x_n | x_{n-1}^{(i)})$ . Since the particle state only evolves in relation to OSM nodes, and OSM maps are highly inaccurate in both position and direction, we sample  $x_n$  by adding Gaussian noise to  $x_{n-1}^{(i)}$ .

$$x_n^{(i)} : \{o_{n-1} + \mathcal{N}(0, \sigma_o), w_{n-1}^{(i)} + \mathcal{N}(0, \sigma_w)\}$$

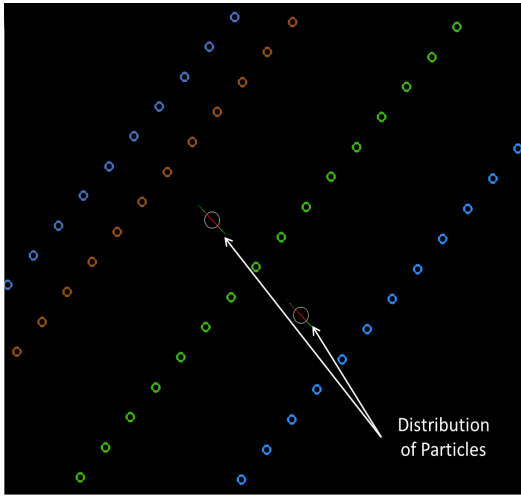


Fig. 4: Conventional Particle Filter. Small circles are observations, with observations displayed for all OSM nodes. Circles crossed with lines are lane-center estimates for the current OSM node, and close inspection reveals that the crossed lines are actually particles, distributed laterally and colored by importance. To make this approach tractable, only a limited number of new proposed particles may be added each step, and so the filter fails to capture narrow bike lane on the top.

Now pair  $(x_n^{(i)}, x_{n-1}^{(i)})$  is distributed according to

$$p(x_n|x_{n-1})Bel(x_{n-1})$$

- 3) Update Function: We update the weight of each sample according to following distribution.

$$\phi_n^{(i)} = p(z_n|x_n^{(i)})$$

$z_n : \{l_1, l_2, \dots, l_k\}$ , where  $l_j$  are lane markers observed at  $n^{th}$  node.

- For each  $x_n^{(i)}$ , perform data association with lane observations. i.e determine associated lane markings for  $x_n^{(i)}$ .
- Compute new observed lane offset and lane width from the observations  $\{\tilde{o}_n^{(i)}, \tilde{w}_n^{(i)}\}$
- Compute  $\phi_n^{(i)}$  using following equation

$$\phi_n^{(i)} = \frac{1}{2\sigma_o} e^{-\frac{(\tilde{o}_n^{(i)} - o_n^{(i)})^2}{2\sigma_o^2}} \frac{1}{2\sigma_w} e^{-\frac{(\tilde{w}_n^{(i)} - w_n^{(i)})^2}{2\sigma_w^2}}$$

where  $\sigma_o$  and  $\sigma_w$  are parameters selected to fit typical standard deviations on width and location based on our data.

During the data association, we check for the appropriate binary variable *has-l* and *has-r* and remove ambiguous data associations. (e.g. if the state of a particle is to the left of the left most lane, then it is not associated with the any lane markings). If the above data association fails, we penalize the  $\phi_n^{(i)}$  by a penalty factor  $\gamma$ . We relax this penalty factor if dashed lane markings are present as we expect them to be missing periodically.

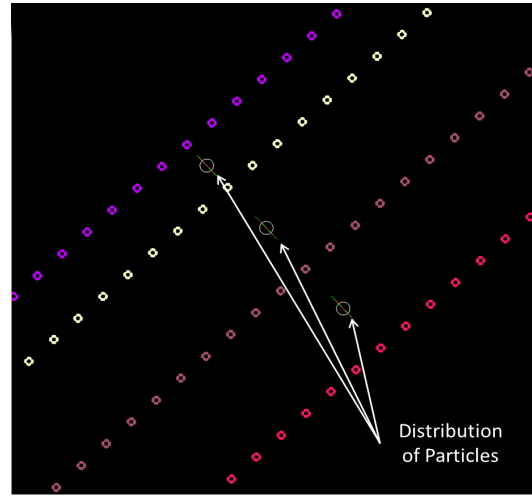


Fig. 5: Dual Particle Filter, showing observations for many OSM nodes, the lane center estimates, and particle distributions for the current node. This approach proposes particles based on observations, and so is able to estimate bike lane on the top.

In order to recover from lane additions or exits in natural road environment, we extend the sampling scheme stated above. We introduce a new parameter  $t$  which is percentage of new particles introduced in the system on every update. Hence we sample according to Algorithm 1, where  $\mu_w$  is the expected lane width. Note that selecting large standard deviations means that a large number of new proposed particles (and corresponding computational cost) are required to sufficiently cover the state space. Further, having large standard deviations increases the chance of proposing an erroneous particle that matches noise in the data.

Figure 4 illustrates the output of Regular Particle Filter estimating lanes at one of the OSM nodes.

**Input:**  $m \rightarrow$  number of particles

**Input:**  $t \rightarrow$  percentage of new particles

**for**  $i = 1:m * (1 - t)$  **do**

    Sample  $x_{n-1}^{(i)} \sim Bel(x_{n-1})$ ;

    Add  $x_{n-1}^{(i)} \rightarrow \tilde{Bel}(x_{n-1})$ ;

**end**

**for**  $i = 1:m * t$  **do**

    Generate new state  $x_{n-1}^{(i)} : \{\mathcal{N}(0, \sigma_o)\mathcal{N}(\mu_w, \sigma_w)\}$ ;

    Set  $\phi_{n-1}^{(i)} : \epsilon$ ;

    Add  $x_{n-1}^{(i)} \rightarrow \tilde{Bel}(x_{n-1})$ ;

**end**

Replace  $Bel(x_{n-1})$  with  $\tilde{Bel}(x_{n-1})$ ;

**Algorithm 1:** Modified Re-sampling algorithm

#### D. Dual Particle Filter

One major limitation observed when applying the conventional particle filter is its failure to capture lanes with abnormal specifications (like biking lanes or extra wide

ramps) as shown in Figure 4. While this could be addressed by increasing the standard deviation of new particles, this solution is suboptimal for reasons discussed above. We will now describe a dual method in order to tackle this problem formally. In the dual configuration, we reverse the role of proposal distribution and measurement function as stated above. At every iteration we sample new particles based on their agreement with the observations

$$x_n^{(i)} \sim p(z_n | x_n)$$

and importance factors are set using

$$\phi_n^{(i)} = \int p(x_n^{(i)} | x_{n-1}^{(i)}) Bel(x_{n-1}) dx_{n-1}$$

The algorithm is then:

- 1) Proposal Distribution: We propose new particles based on the observations. Let  $z_n : \{l_1, \dots, l_k\}$  be  $k$  lane markers observed at  $n^{th}$  OSM node, sorted by location. We uniformly select  $j \in \{1, (k-1)\}$  and propose

$$x_n^{(i)} : \left\{ \frac{l_j + l_{j+1}}{2} + \mathcal{N}(0, \sigma_o); (l_{j+1} - l_j) + \mathcal{N}(0, \sigma_w) \right\}$$

- 2) Update Function: Importance factors for each particle are then corrected using prior belief  $Bel(x_{n-1})$ . To approximate this distribution over the continuous state space, we take a kernel density approach. We first generate  $m$  samples as done for the proposal distribution in a conventional particle filter.

$$\tilde{x}_n^{(i)} \sim p(x_n | x_{n-1}) Bel(x_{n-1})$$

Writing  $h(\{\tilde{x}_n\}; x)$  to denote the parameterized kernel density function approximating this distribution, the importance factor for each particle is given by

$$\phi_n^{(i)} = h(\{\tilde{x}_n\}; x_n^{(i)})$$

As shown in Figure 5, the Dual Particle Filter is able to estimate non-standard bike lane which the Conventional Particle Filter failed to capture.

### E. Mixture Particle Filter

While the pure Dual Particle Filter is able to capture abnormal lane specifications, it will fail in the situation where new lanes are added. Proposed particles for new lanes cannot be matched to any in the previous distribution, thus getting essentially zero weight. The approach described in [17] fixes this problem using a combination of both Conventional and Dual Particle Filter. In the Mixture approach, we use a variable mixture ratio  $\theta$  ( $0 \leq \theta \leq 1$ ) and sample from the Conventional method with probability  $1 - \theta$  and with probability  $\theta$  using the Dual.

Additionally, the Mixture Particle Filter allows for more flexible modeling based on situational information. For instance we can vary the mixture ratio  $\theta$  based on structural information from the OSM map. Specifically, we reduce the ratio closer to intersections where performance of Dual is significantly bad due to the lack of lane markings. Variations on this theme, and whether such dependencies can be learned, are an interesting source of future work.

### F. Clustering and Lane Indexing

Our generated map will have only a finite number of lanes, each with a single lane position and width estimate for each OSM node. Further, these lanes should be linked over iterations using IDs. This requires one further processing step. The particle filters above result in a discrete approximation of  $Bel(x_n | z_n)$  represented by a set of particles. This distribution can be observed in Figure 4. This distribution is multi-modal and number of modes are unknown a priori. We use a EM-based weighted clustering algorithm on the distribution to find the maximum-a-posteriori modes. These cluster centers are final lane estimates. This clustering is done in the space of  $x$  (i.e. on both offset and width).

To generate temporal links between iterations, we assign an index to each cluster using Algorithm 2.

```

Input: p → set of particles
Input: C → Clusters
for c = 1 : C do
  i → Most common index in the set of particles
  belonging to cluster c;
  if i then
    assign index i → to cluster c and all the
    particles belonging to cluster c;
  else
    assign new index → to cluster c and all the
    particles belonging to cluster c;
  end
end

```

**Algorithm 2:** Cluster Indexing algorithm

## V. RESULTS

To evaluate the accuracy of our approach, we compare our lane estimates with hand labeled road network data for 28 km of road. Our hand-labeled road network data consists of rural roads, urban roads and freeways which is shown in Figure 2, but unfortunately, this does not include bicycle lanes. For all our experiments, we set the number of particles to 5000 and for mixture case, we had mixture ratio set to 0.3. As discussed above, the mixture ratio is set to zero near intersections, i.e. we relied only on regular particle filter in that case. Figure 6 illustrates qualitative results of lane estimates for each type of scene. We did not evaluate the dual approach on its own.

We evaluate our results using two metrics, mean positional error and number of nodes for which we incorrectly estimated a lane centers. Quantitative results are shown in Table I. These results show that both the regular and mixture approaches generate highly accurate lane-level maps, with the mixture approach being slightly better in some cases.

For the entire dataset, we were not able to associate our estimates with hand labeled data for 2.3% of the estimated lane center nodes for the regular particle filter, and for 6.2% for the mixture case. The number of missed estimates is higher for the mixture case as we do not have hand labeled

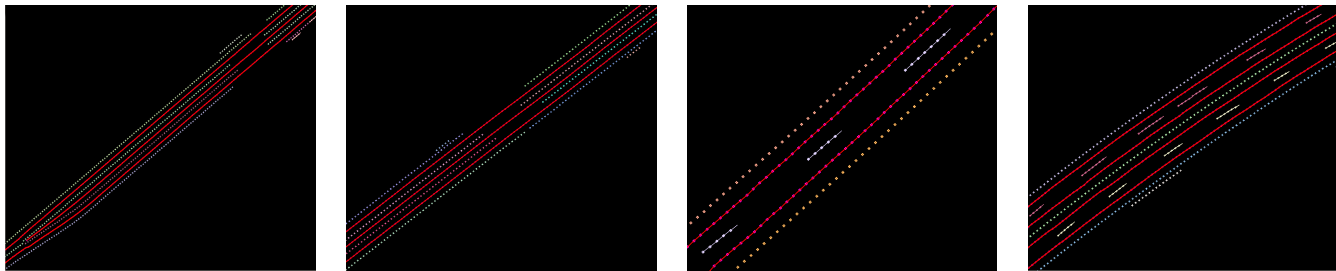


Fig. 6: Qualitative Results. left two figures show lane estimation on country roads. Even though there is missing data at intersections, we are able to track lanes successfully. Third figure shows lane estimation on highway with dashed lanes. Last figure illustrates our results on multi-lane urban roads. Note that for all those roads, we did not have any prior information about number of lanes

Method	Mean Error(m)	Max Error(m)
Regular Particle Filtering - urban	0.06	0.38
Regular Particle Filtering - highway	0.05	0.13
Regular Particle Filtering - all data	0.06	0.38
Mixture Particle Filtering - urban	0.06	0.22
Mixture Particle Filtering - highway	0.04	0.08
Mixture Particle Filtering - all data	0.05	0.22

TABLE I: Quantitative Results

data for bike lanes for which mixture particle filter is able to guess lane estimates correctly, in other words, the more flexible modeling capability of the mixture approach actually hurts it in this metric. Analysis of the locations where errors occur, indicates that errors mainly stem from noisy data at intersections where lanes markings are missing.

## VI. CONCLUSIONS

In this work, we have shown how structural priors can be leveraged in a real-world outdoor mapping task requiring, and compared against, accurate lane-level maps. Our results are encouraging, this approach is able to generate maps that agree with hand-made maps to a high level of accuracy. Application of this work will allow our system to generate accurate large-scale lane-level maps suitable for advanced driver support and autonomous driving applications.

Additionally, we find the general approach of combining structural priors with accurate local information extremely interesting and are working on a number of ideas for extending it including:

- Improve the use of prior information in intersection handling
- Learning a model of how and when to modify inference algorithm based on situations
- Develop a method for detecting and correcting maps when world changes
- Use this approach in an on-line algorithm

## REFERENCES

- [1] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, “Autonomous driving

- in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.
- [3] A. Doshi and M. M. Trivedi, “Tactical driver behavior prediction and intent inference: A review,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1892–1897.
- [4] N. Oliver and A. P. Pentland, “Driver behavior recognition and prediction in a smartcar,” in *AeroSense 2000*. International Society for Optics and Photonics, 2000, pp. 280–290.
- [5] N. Fairfield and C. Urmson, “Traffic light mapping and detection,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5421–5426.
- [6] “How Google’s Self-Driving Car Works (IROS 2011 Keynote),” <http://www.youtube.com/watch?v=z7ub5Doyapk>, Jun. 2013.
- [7] J. Biagioni and J. Eriksson, “Inferring road maps from global positioning system traces,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2291, no. 1, pp. 61–71, 2012.
- [8] H. Wang, Z. Cai, H. Luo, C. Wang, P. Li, W. Yang, S. Ren, and J. Li, “Automatic road extraction from mobile laser scanning data,” in *Computer Vision in Remote Sensing (CVRS), 2012 International Conference on*. IEEE, 2012, pp. 136–139.
- [9] S. T. McMichael, “Lane detection for dexter, an autonomous robot, in the urban challenge,” Ph.D. dissertation, Case Western Reserve University, 2008.
- [10] M. Aly, “Real time detection of lane markers in urban streets,” in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 7–12.
- [11] A. S. Huang and S. Teller, “Probabilistic lane estimation using basis curves,” *Robotics: Science and Systems (RSS)*, 2010.
- [12] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, “Efficient lane detection and tracking in urban environments,” in *Proc. European Conf. Mobile Robots*, 2007, pp. 126–131.
- [13] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Nunn, and A. Kummert, “A novel approach to lane detection and tracking,” in *Intelligent Transportation Systems, 2009. ITSC’09. 12th International IEEE Conference on*. IEEE, 2009, pp. 1–6.
- [14] “OpenStreetMap Website,” <http://www.openstreetmap.org/>, Jun. 2013.
- [15] J. Levinson, M. Montemerlo, and S. Thrun, “Map-based precision vehicle localization in urban environments,” in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- [16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *International Journal of Robotics Research (IJRR)*, to appear.
- [17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust monte carlo localization for mobile robots,” *Artificial Intelligence Journal*, 2001.





## Session II

### Perception

- **Keynote speaker: Davide Scaramuzza (ETHZ, Zurich, Switzerland)**  
**Title: Vision-Controlled Micro Aerial Vehicles: from "calm" navigation to "aggressive" maneuvers**
- **Title: Enabling Efficient Registration using Adaptive Iterative Closest Keypoint**  
**Authors: Johan Ekekrantz, Andrzej Pronobis, John Folkesson, Patric Jensfelt**
- **Title: Information fusion and evidential grammars for object class segmentation**  
**Authors: Jean-Baptiste Bordes, Philippe Xu, Franck Davoine, Huijing Zhao, Thierry Denoeux**

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

IROS'13

PPNIV'13

5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2013 IEEE/RSJ International Conference on Intelligent Robots and Systems

## Session II

Keynote speaker: **Davide Scaramuzza**  
(ETHZ, Zurich, Switzerland)

### **Vision-Controlled Micro Aerial Vehicles: from "calm" navigation to "aggressive" maneuvers**

**Abstract :** In the last two years, we have heard a lot of news about drones, small autonomous flying vehicles. Flying robots have numerous advantages over ground vehicles: they can get access to environments where humans cannot get access to and, furthermore, they have much more agility than any other ground vehicle. Unfortunately, their dynamics makes them extremely difficult to control and this is particularly true in GPS-denied environments. In this talk, I will present challenges and results for both ground vehicles and flying robots, from localization in GPS-denied environments to motion estimation. I will show several experiments and real-world applications where these systems perform successfully and those where their applications is still limited by the current technology.

**Biography:** Born in Terni, Italy, in 1980, Davide received his Master's degree (2004) in Electronics and Information Engineering (Summa cum Laude and Dignity of Printing) at the University of Perugia, Italy. His Master's thesis won the Aica-Federcomin Award (Federcomin is a sector of Confindustria, the confederation of the Italian industries), the most prestigious Italian prize for Master's theses in the field of Information and Communication Technology. His advisors were Prof. Paolo Valigi and Dr. Stefano Pagnottelli, from the University of Perugia. After completing his Master's thesis, between October 2004 to February 2005, he did an internship at the EPFL of Lausanne, Switzerland. Afterwards, he started his PhD studies at the Autonomous Systems Lab of EPFL with Prof. Roland Siegwart. In July 2006, the Autonomous Systems Lab was moved to ETH Zurich. In February 2008, he received his PhD in Computer Vision and Robotics at ETH Zurich with his thesis: Omnidirectional vision: from calibration to robot motion estimation. His PhD advisor was Prof. Roland Siegwart and his committee members were Prof. Patrick Rives (INRIA Sophia Antipolis) and Prof. Luc Van Gool (ETH Zurich). His PhD thesis won the Robotdalen Scientific Award, which is the most prestigious award for PhD theses in the field of Robotics and Automation. The prize was supported by the European Union, the IEEE Instrumentation and Measurement Society, ABB, Volvo, and several other industries.

As of February 2008, he became a postdoctoral researcher at the Autonomous Systems Lab of Professor Siegwart at ETH Zurich. In January 2009, he became leader and project manager of the European project sFly. He will lead this project until its end in December 2011. From 2008 to 2010, he was lecturer of the Autonomous Mobile Robots Masters course. In 2009, he led a group of Bachelor and PhD students to participate in the European Micro Aerial Competition. They won the 2nd place with the first purely vision based autonomous helicopter.

In January 2011, Davide moved to the GRASP Lab of the University of Pennsylvania for a second postdoc under the direction of Prof. Vijay Kumar and Prof. Kostas Daniilidis. In February 2012, he became Assistant Professor in Human Oriented Robotics at the University of Zurich where he heads the Robotics and Perception Lab.

Between 2009 and 2010, he worked as a consultant of the ETH spin-off Dacuda, inventor of the world first scanner mouse, currently distributed by LG (watch promo video).

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# Vision-Controlled Micro Aerial Vehicles: from «calm» navigation to «agile» maneuvers

Davide Scaramuzza

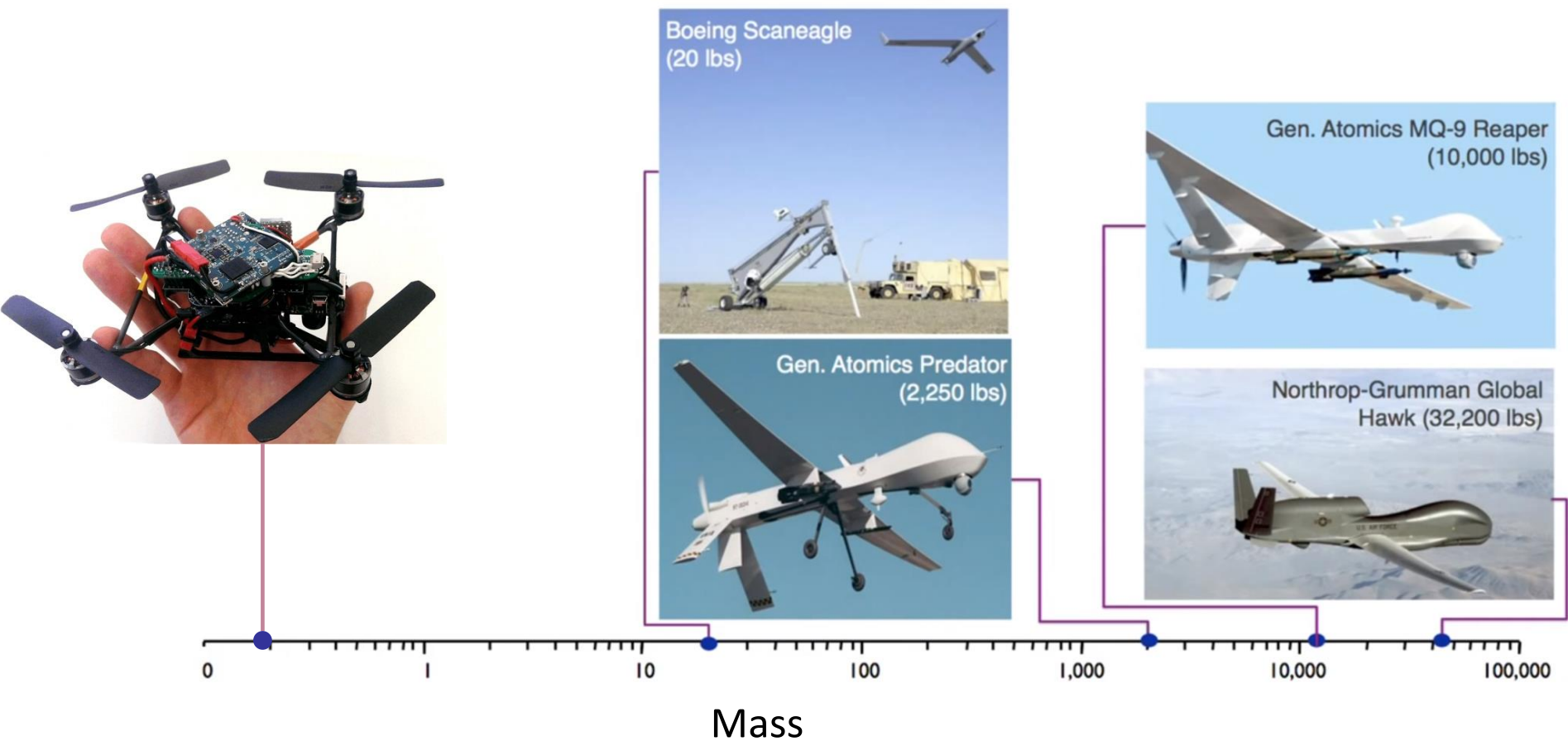
Robotics and Perception Group

*rpg.ifi.uzh.ch*

University of Zurich



# Unmanned Aerial Vehicles





# Motivation

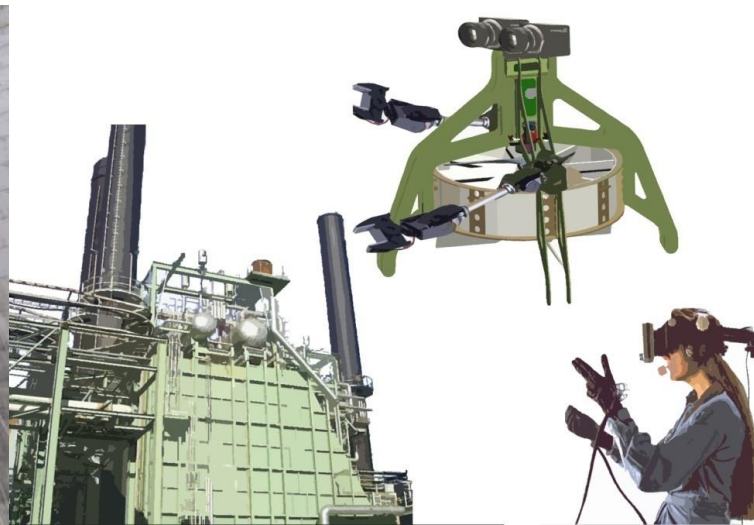
5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan



Search and  
Rescue



Environment  
Monitoring



Remote  
Inspection<sup>107</sup>



### NanoQuad from KMeilRobotics

Visual SLAM running fully onboard (55 fps)

Embedded Computer: Odroid (ARM Cortex A-9)



### Custom made quad

(same embedded computer as NanoQuad)

Davide Scaramuzza – [rpg.ifi.uzh.ch](http://rpg.ifi.uzh.ch)

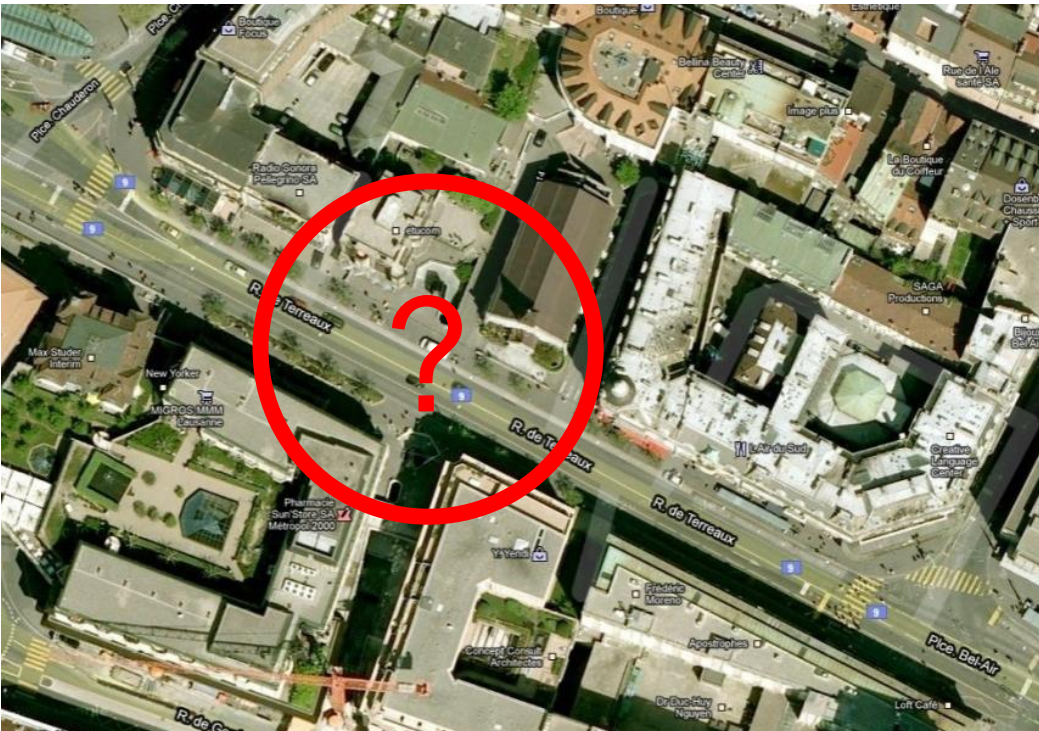
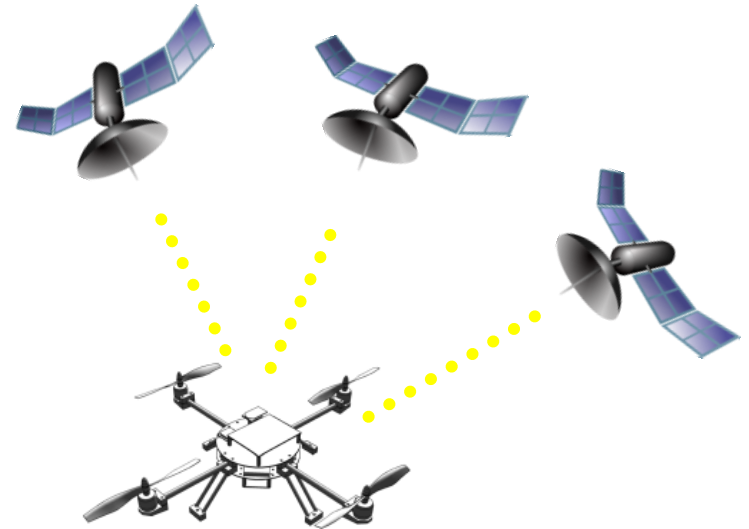




- Visual Navigation in GPS-denied Environments
- Open Problems and Challenges
  - with Vision
  - with Quadrotors
- Air-ground collaboration
- Event-based Vision

# Why not GPS ?

- It does not work indoors
- Even outdoors it is not a reliable service



# Why is Perception Important ?



# Why is Perception Important ?

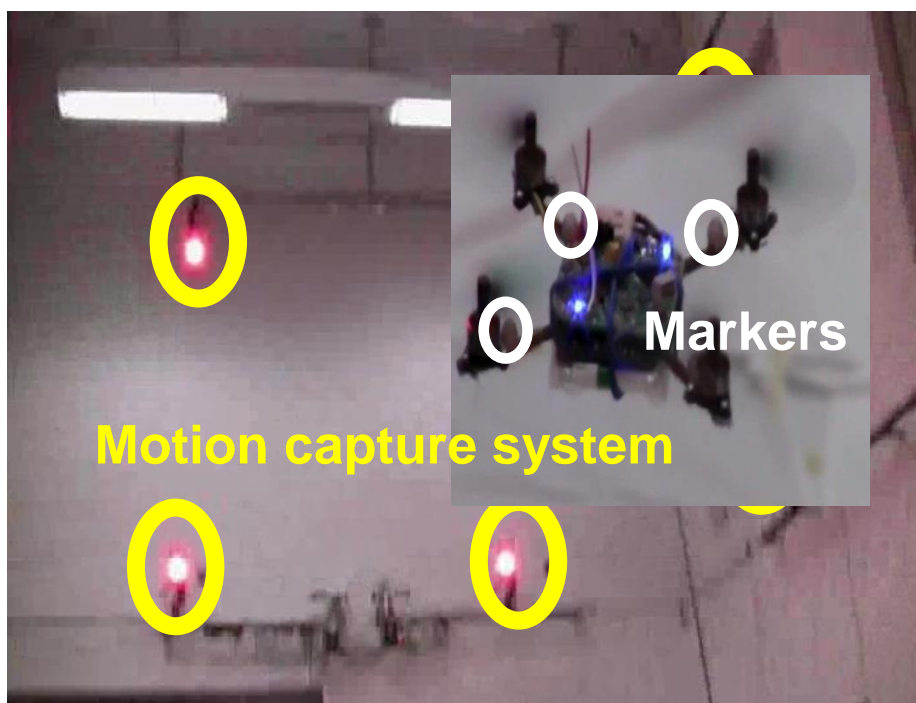
It allows a robot to be truly autonomous

This robot is «*blind*»

Davide Scaramuzza – [rpg.ifi.uzh.ch](http://rpg.ifi.uzh.ch)

# Why is Perception Important ?

It allows a robot to be truly autonomous

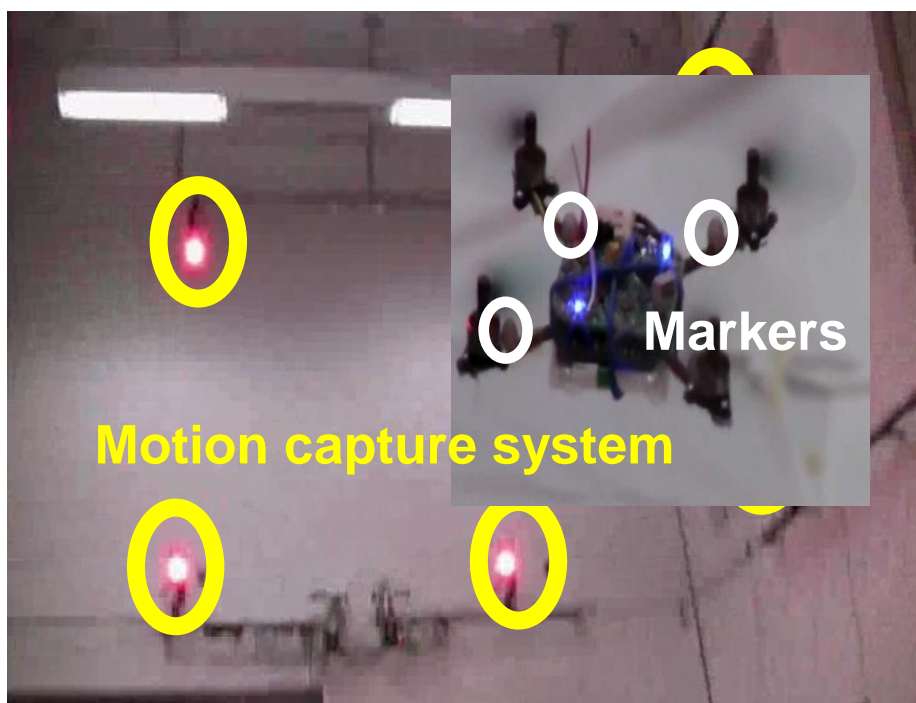


This robot is «*blind*»

Davide Scaramuzza – [rpg.ifi.uzh.ch](http://rpg.ifi.uzh.ch)

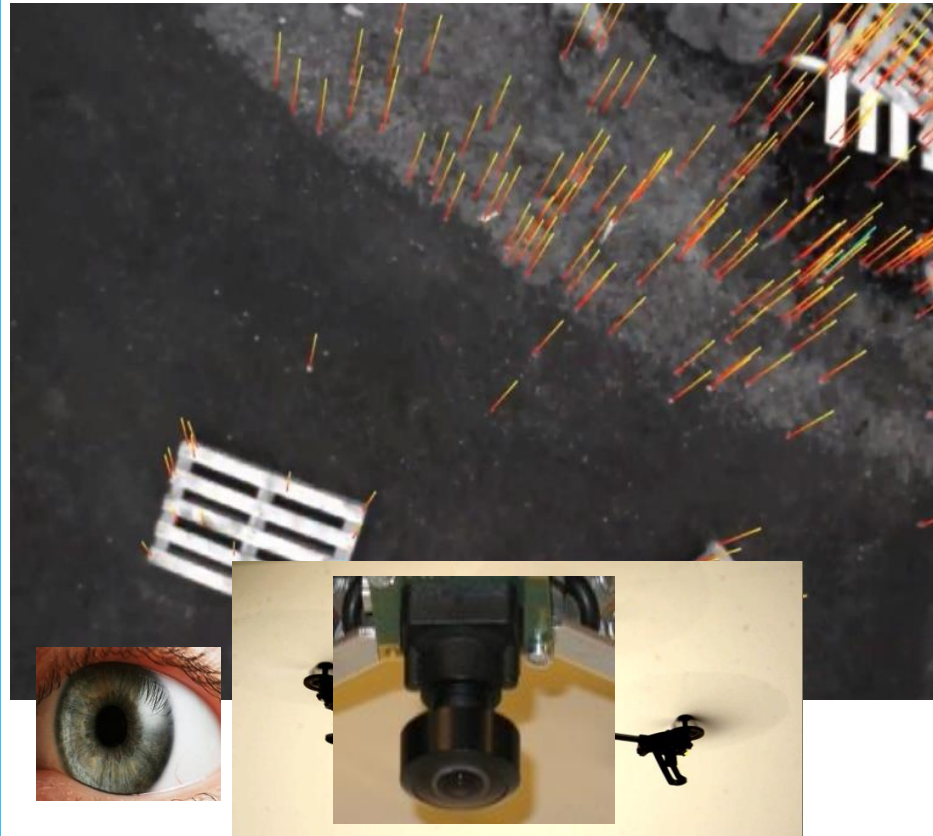
# Why is Perception Important ?

It allows a robot to be truly autonomous



This robot is «*blind*»

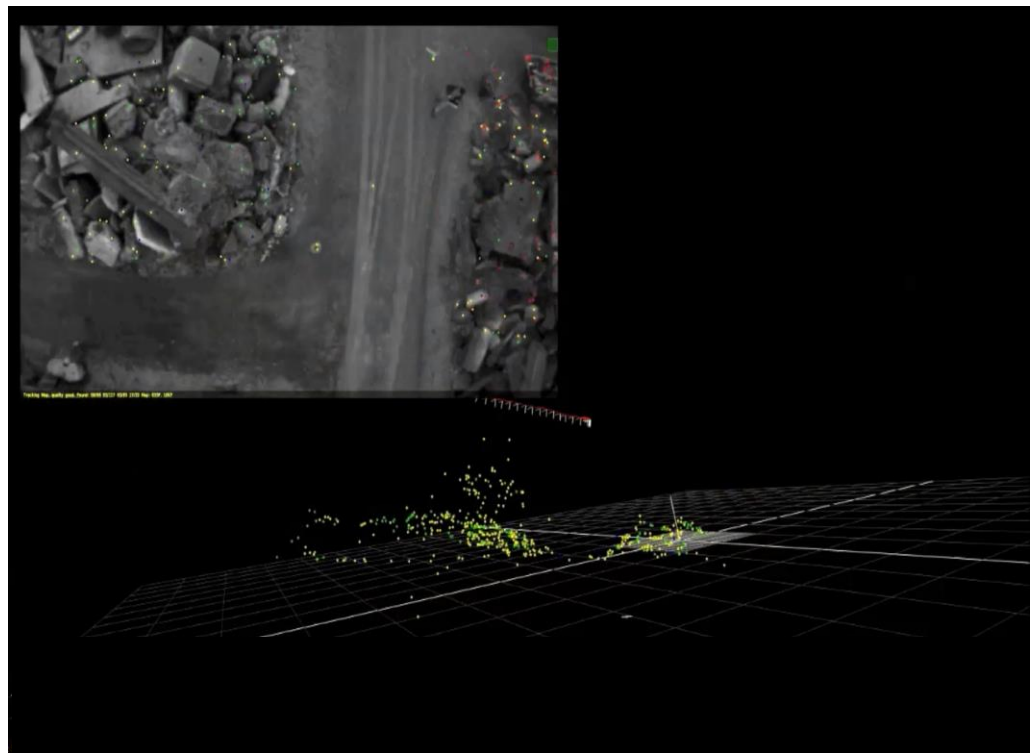
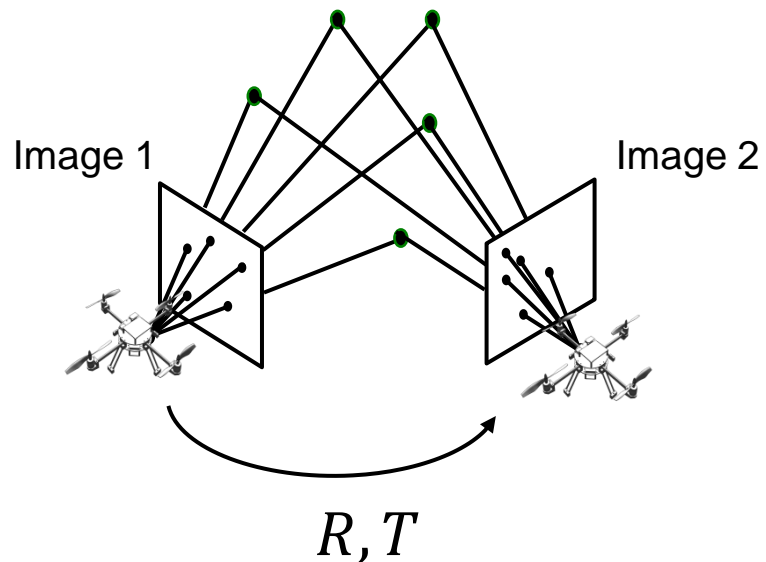
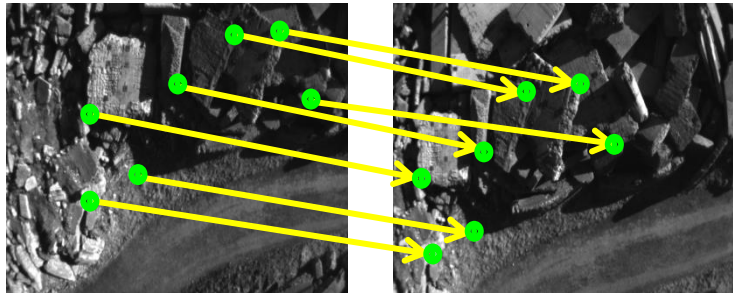
Davide Scaramuzza – [rpg.ifi.uzh.ch](http://rpg.ifi.uzh.ch)



This robot can «see»<sup>14</sup>



# How Does it Work ?





# How do we Globally Localize?

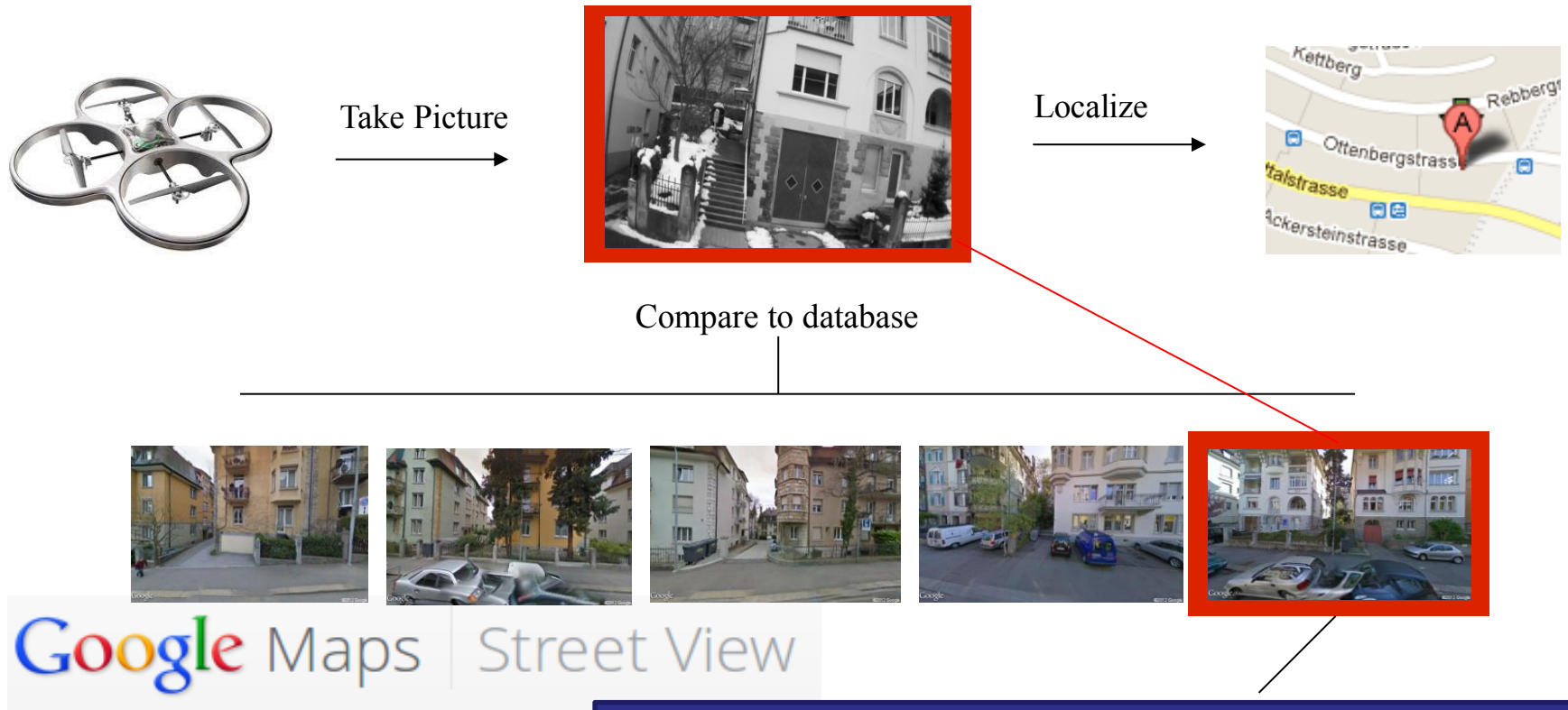
- Visual SLAM helps the robot to stabilize and localize w.r.t. its own map
- But what if we wanted to localize in an urban environment, without GPS?

# MAV Urban Localization from Google Street View Data



# Goal: Image to GPS

- Detect the global position of the MAV by recognizing visually-similar places (appearance based localization for MAVs)
- Large viewpoint changes -> **air-ground matching**



Lat: 47.384345, Long: 8.545037, Heading: 161.01

# Air-ground matching is challenging



Google Maps | Street View



***IROS Presentation TOMORROW – Session: «Unmanned Aerial Vehicles IV»***



# Air-ground matching is challenging



Google Maps | Street View

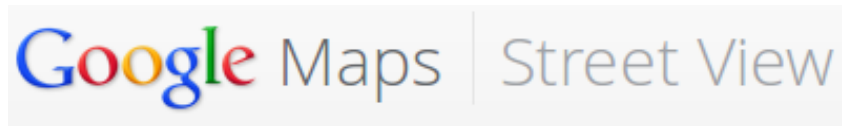


***IROS Presentation TOMORROW – Session: «Unmanned Aerial Vehicles IV»***

*MAV Urban Localization from Google Street View Data, Majdik, Albers-Schoenberg, Scaramuzza*

# Air-ground matching is challenging

➤ Repetitive and self-similar structures





# Localization Results

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan



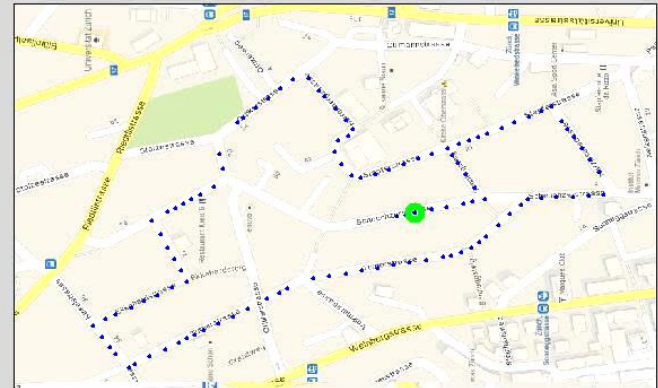
Drone image: left-000500.jpg



Most similar street image: street-003.jpg



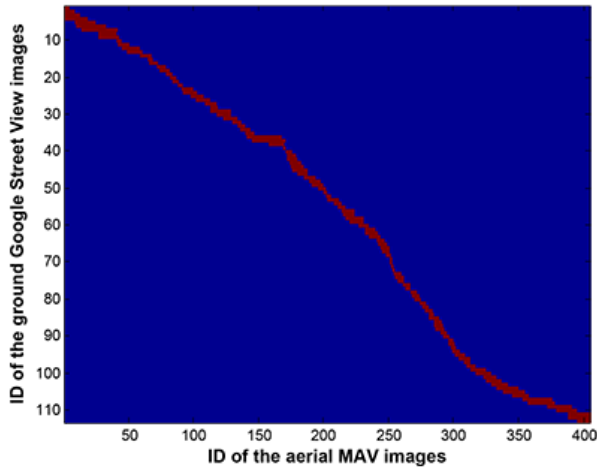
Features matched



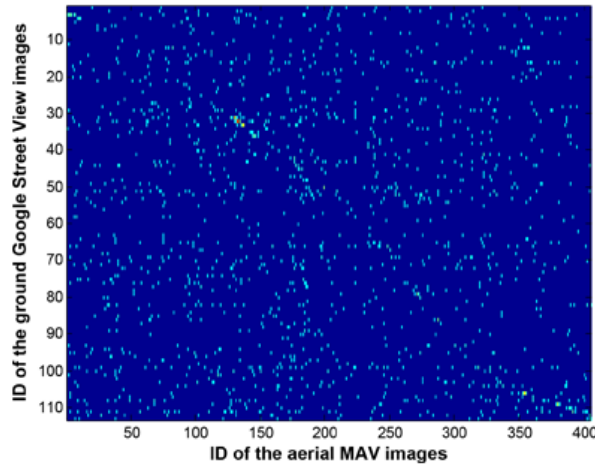
Aerial view



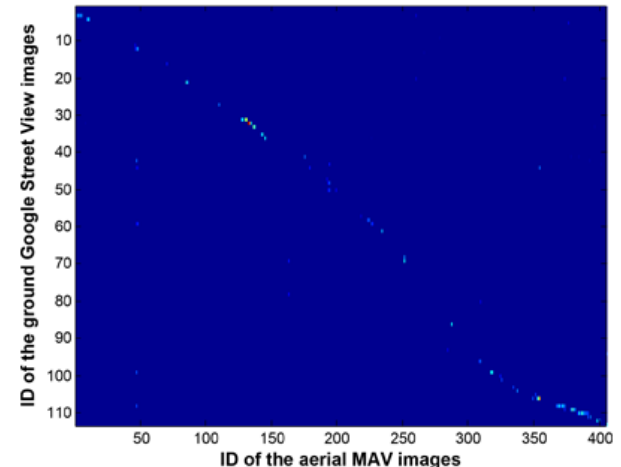
# Comparison among state-of-the-art



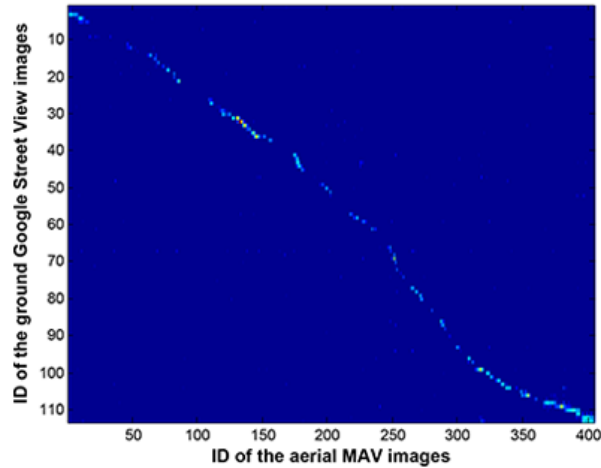
(a) Ground truth



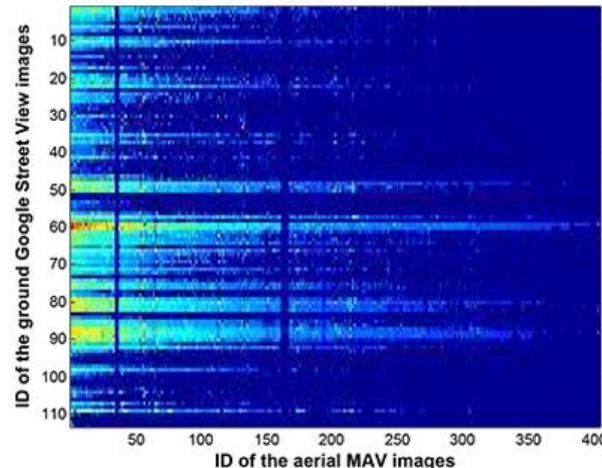
(b) Brute-force feature matching



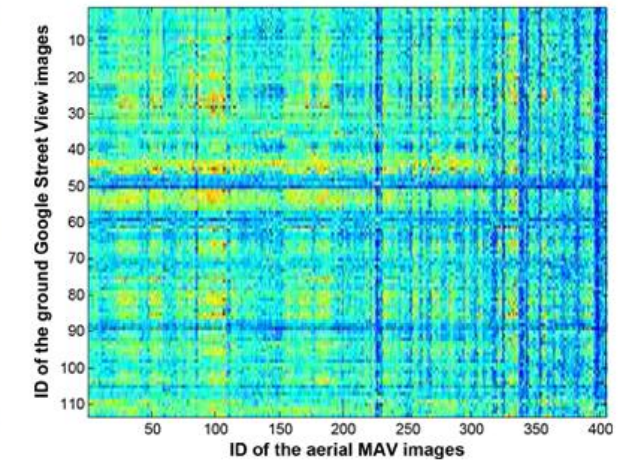
(c) Affine SIFT and ORSA



(d) Proposed approach

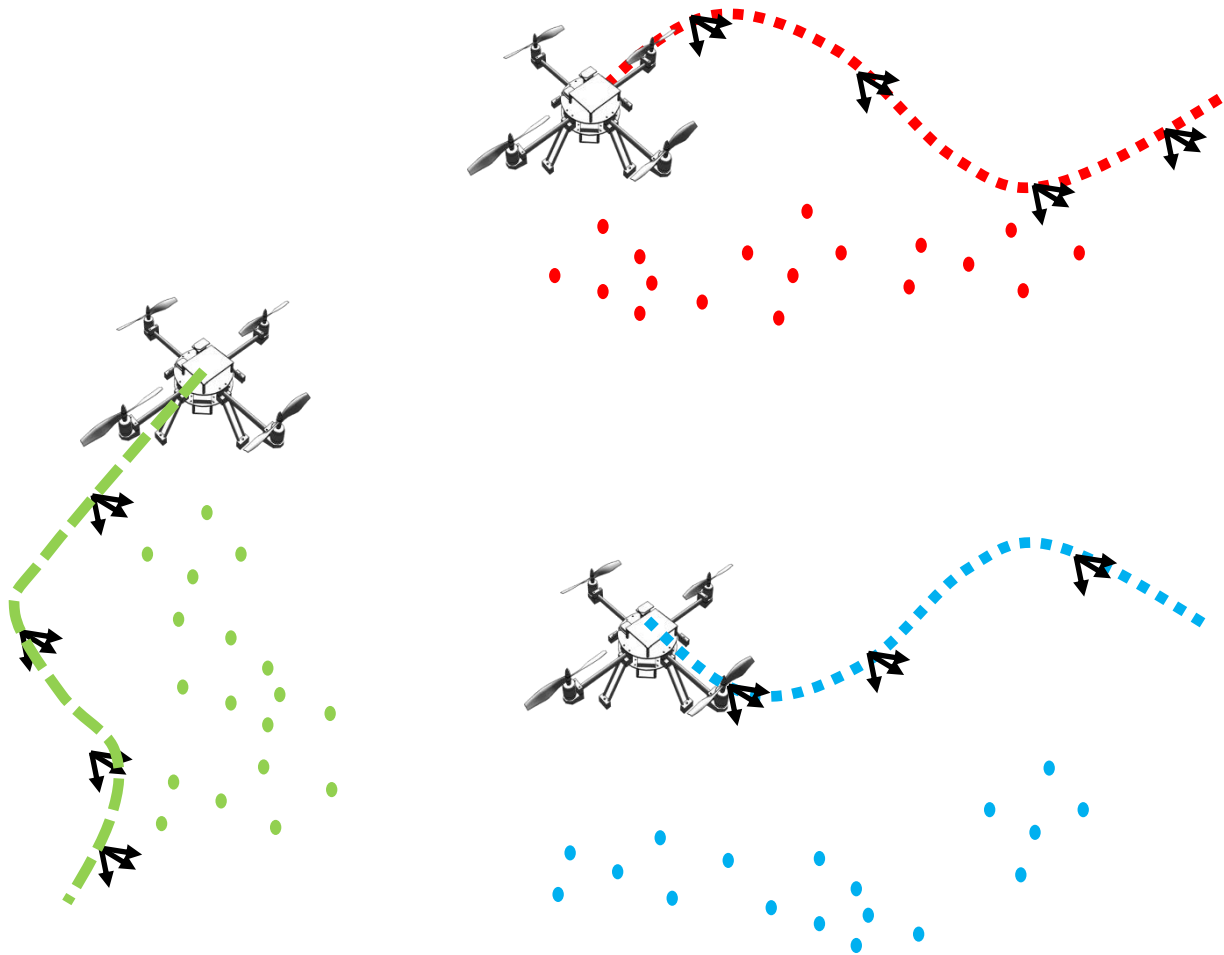


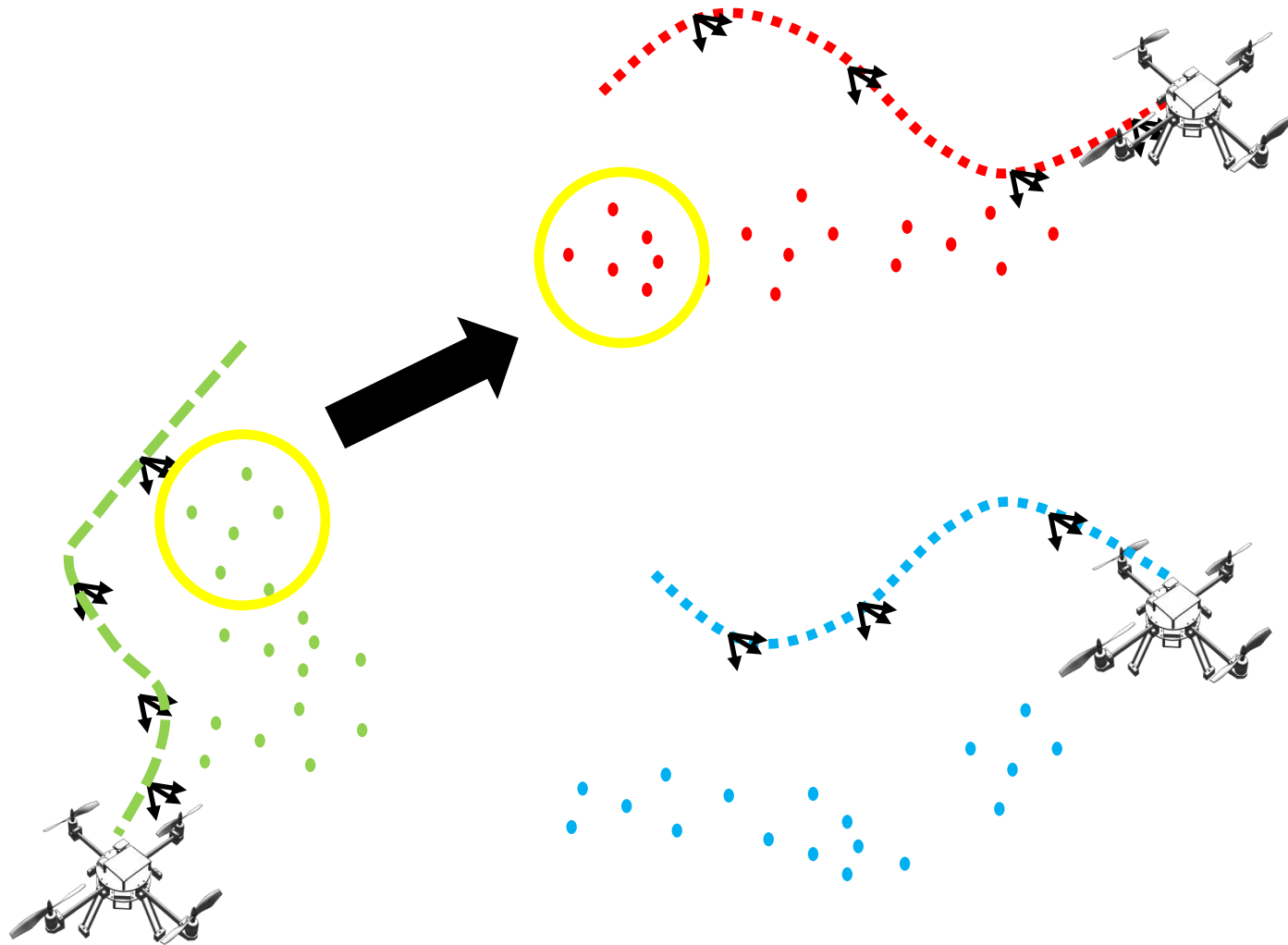
(f) FABMAP

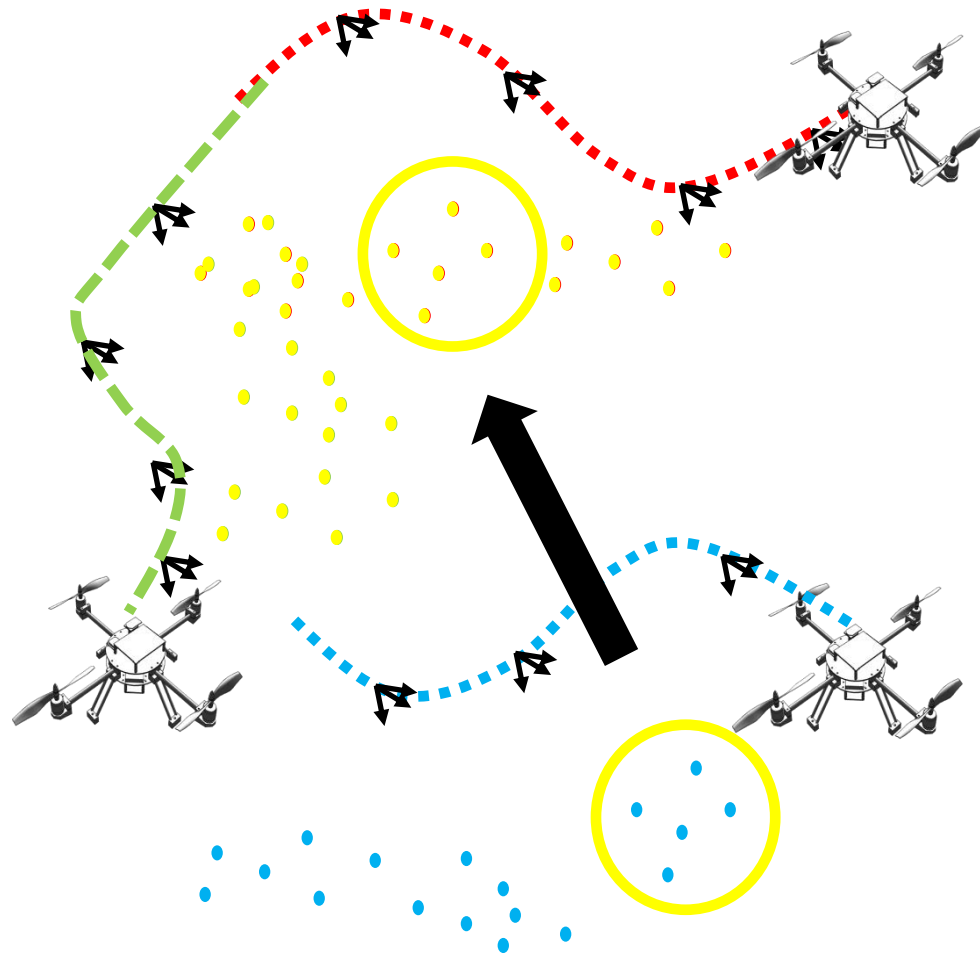


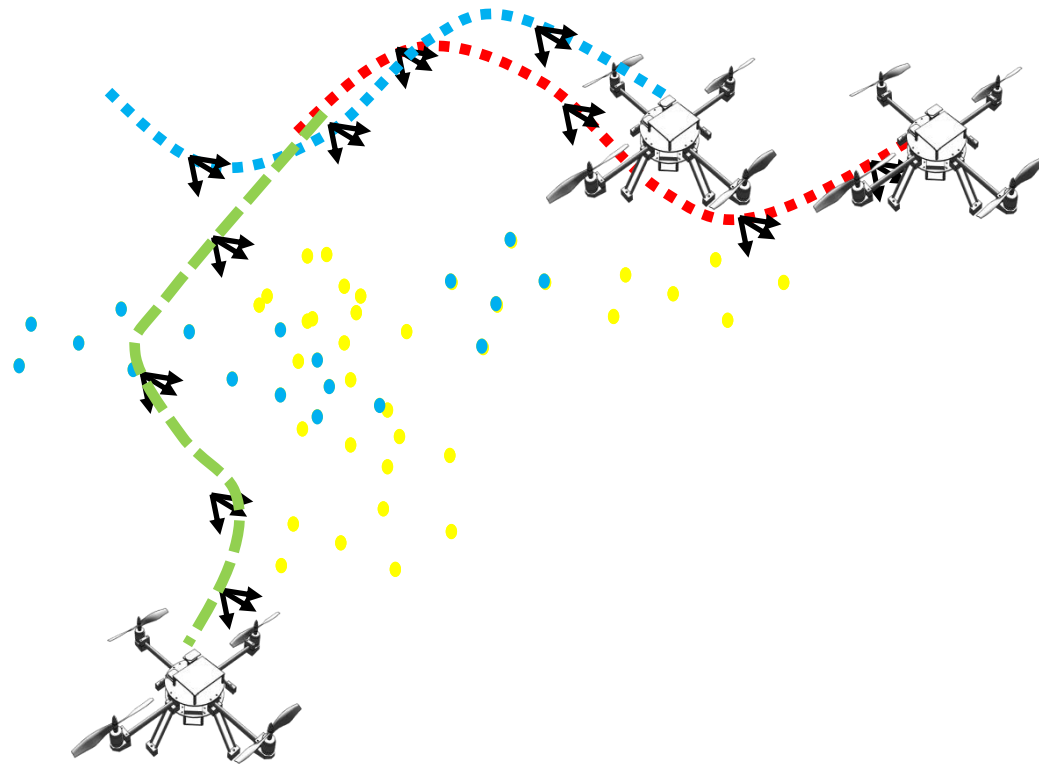
(e) Bag of Words

***IROS Presentation TOMORROW – Session: «Unmanned Aerial Vehicles IV»***

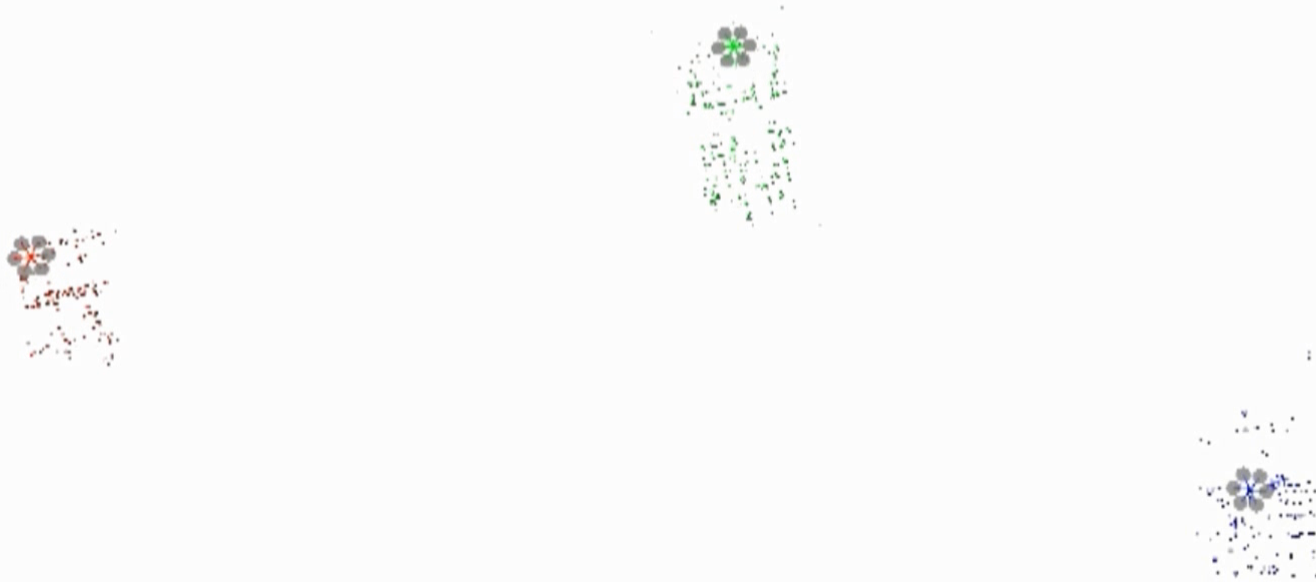






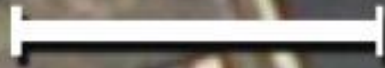


**Watch video at <http://rpg.ifi.uzh.ch>**





**10 m**

A white horizontal scale bar with vertical end caps, indicating a length of 10 meters.

**— MAV 1**  
**— MAV 2**  
**···· GPS**

- Vision-controlled Quadrotors in GPS-denied Environments
- Open Problems and Challenges
  - with Vision
  - with quadrotors
- Air-ground collaboration
- Event-based Vision

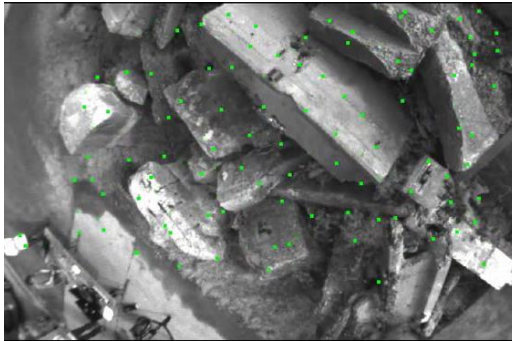
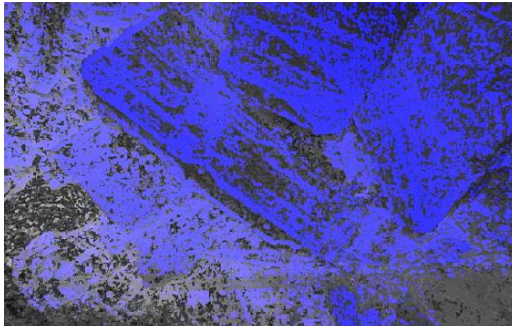
# Open Problems and Challenges in Vision

- Lack of texture
- Illumination changes
- Dynamic environments



# Monocular, Real-Time 3D Dense Reconstruction for MAVs

- Tracks every pixel (like DTAM [Newcombe, CVPR'10])
- Running live from video streamed (8 ms in CUDA, on i7 laptop)
- Allows tracking with low texture surfaces





- Vision-controlled Quadrotors in GPS-denied Environments
- Open Problems and Challenges
  - With Vision
  - With Quadrotors
- Air-ground collaboration
- Event-based Vision

# Motivation

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

- Search and rescue missions can benefit from robotic technologies (Fukushima, Gotthard rock slide, Italy earthquake)
- Current robots are teleoperated by trained professionals
- Ground robots would benefit from an external “flying eye”





# 2011 - Fukushima Nuclear Power Plant



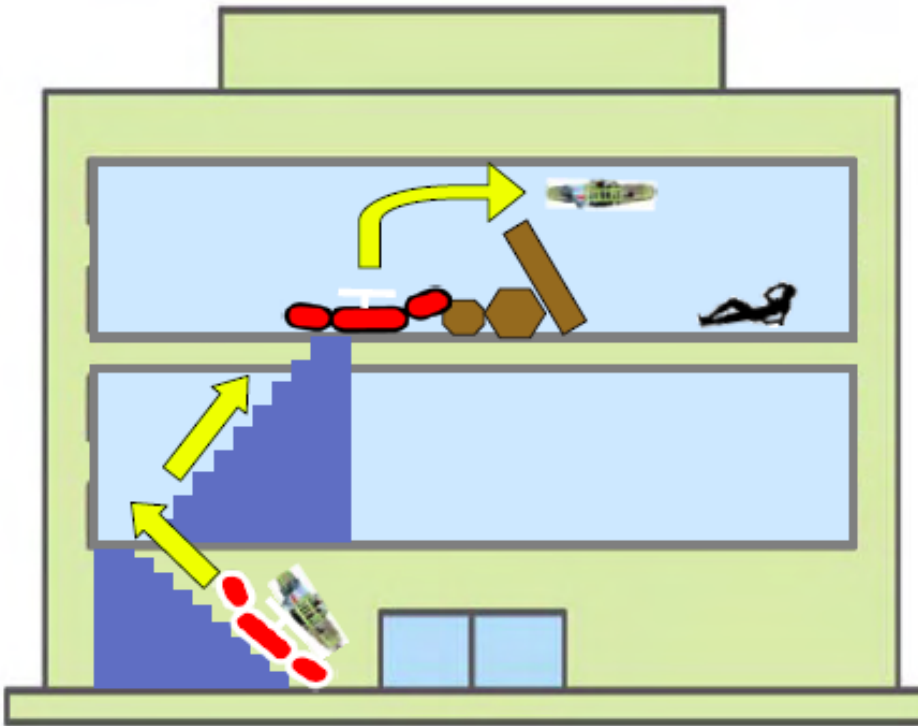
# 2011 - Fukushima Nuclear Power Plant



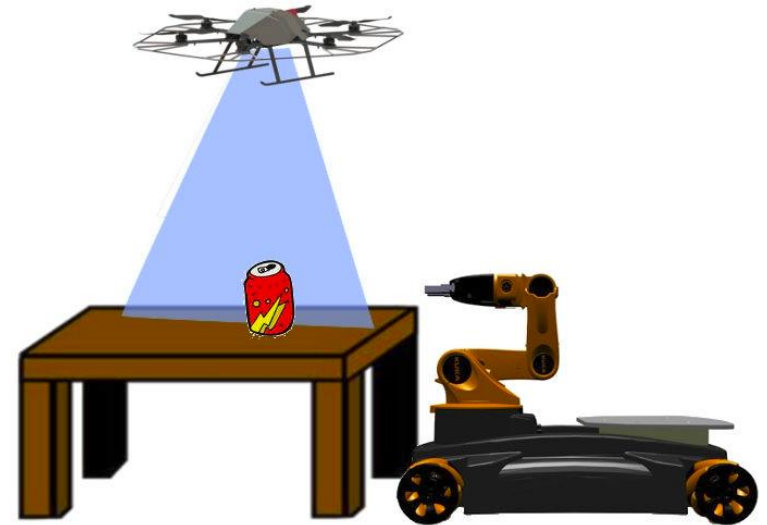


# Synergistic Collaboration between Ground and Aerial Vehicles

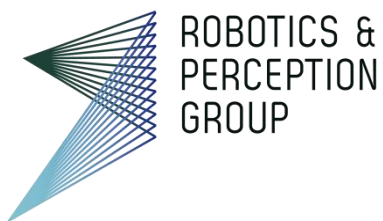
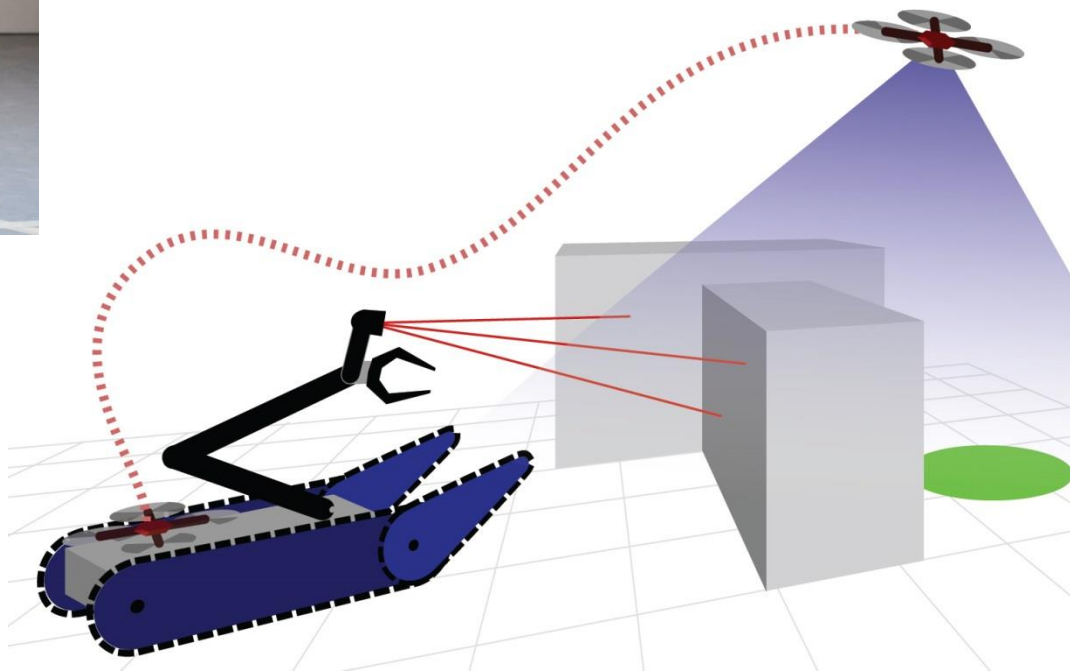
## Air-ground exploration



## Air-ground collaborative grasping



# Collaborative Localization and Mapping with Complementary Sensing Modalities (kinect on the ground robot and single camera on the flying robot)



***IROS Presentation TOMORROW – Session: «Unmanned Aerial Vehicles IV»***

*Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction, Forster, Pizzoli, Scaramuzza*

# Collaborative Localization and Mapping with Complementary Sensing Modalities (kinect on the ground robot and single camera on the flying robot)

Watch video at <http://rpg.ifi.uzh.ch>



***IROS Presentation TOMORROW – Session: «Unmanned Aerial Vehicles IV»***

*Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction, Forster, Pizzoli, Scaramuzza*

# Collaborative Grasping

Watch video at <http://rpg.ifi.uzh.ch>





- Vision-controlled Quadrotors in GPS-denied Environments
- Open Problems and Challenges
  - With Vision
    - With Quadrotors
- Air-ground collaboration
- Event-based Vision

# Open Problems and Challenges with Micro Helicopters

- Current flight maneuvers achieved with onboard cameras are still too slow compared to those attainable with Motion Capture Systems



 D. Mellinger, V. Kumar



 S. Lupashin, R. D'Andrea

# Towards Aggressive Maneuvers with Onboard Vision



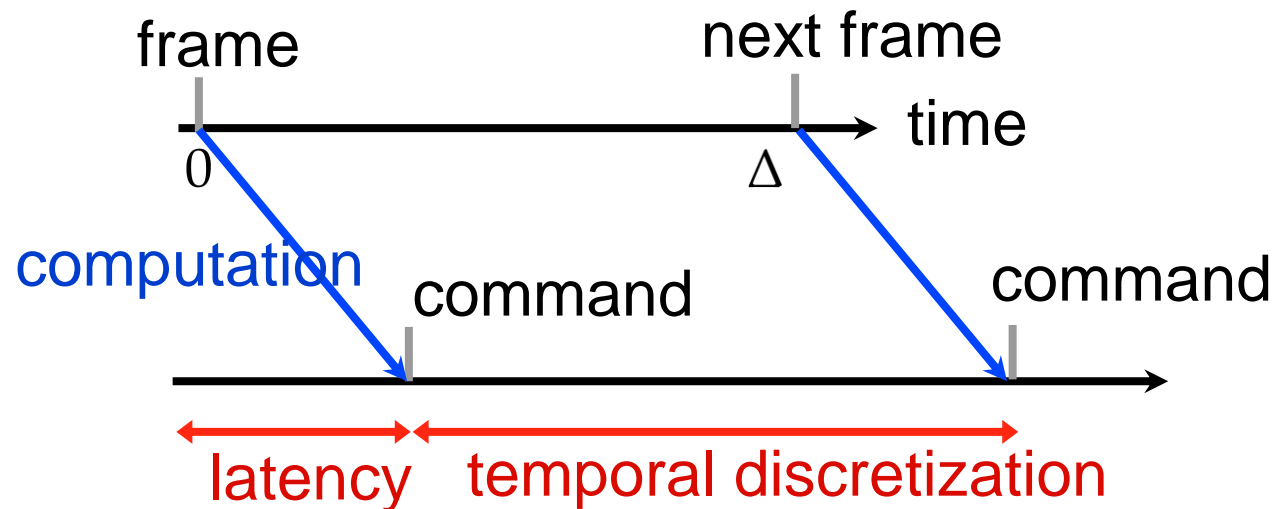
*How fast can we go with a camera?*

*Let's assume that we have perfect perception (i.e., localization)*

*Can we achieve the same flight performances  
attainable with motion capture systems?*

# Towards Aggressive Maneuvers with Onboard Vision

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline



# Towards Aggressive Maneuvers with Onboard Vision

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline
- Can we create a low-latency, low-discretization control architecture?

*Yes...*

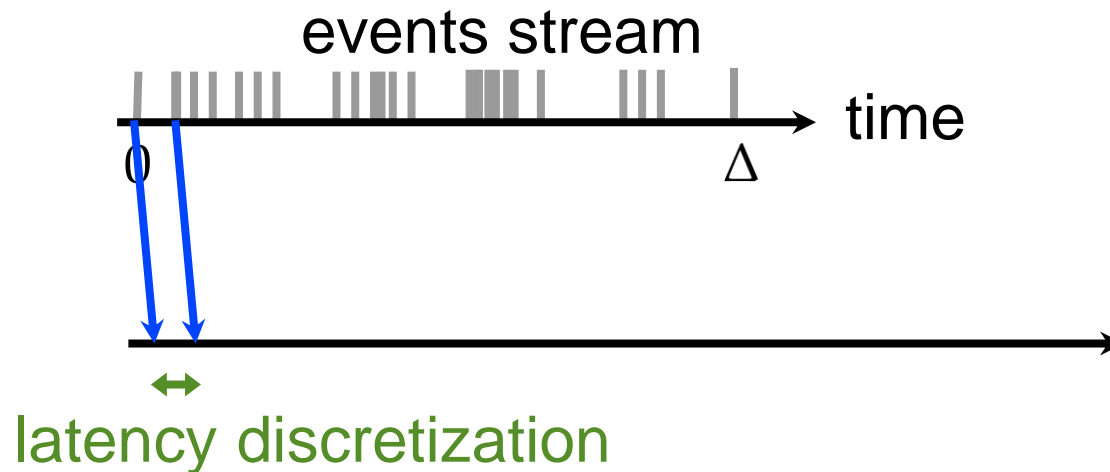
*...if we use a camera where pixels do not spike all at the same time*

*... in a way like we humans do...*



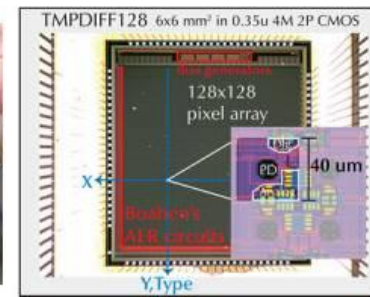
# Towards Aggressive Maneuvers with Onboard Vision

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline
- Can we create a low-latency, low-discretization control architecture?



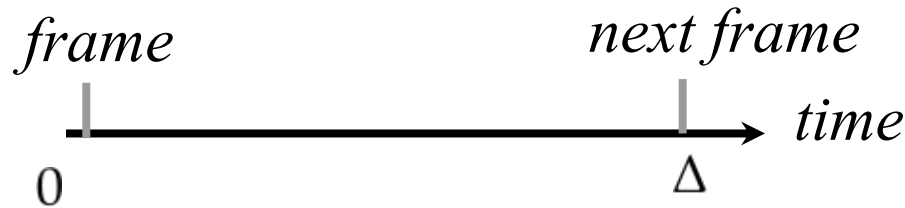
# Dynamic Vision Sensor (DVS)

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

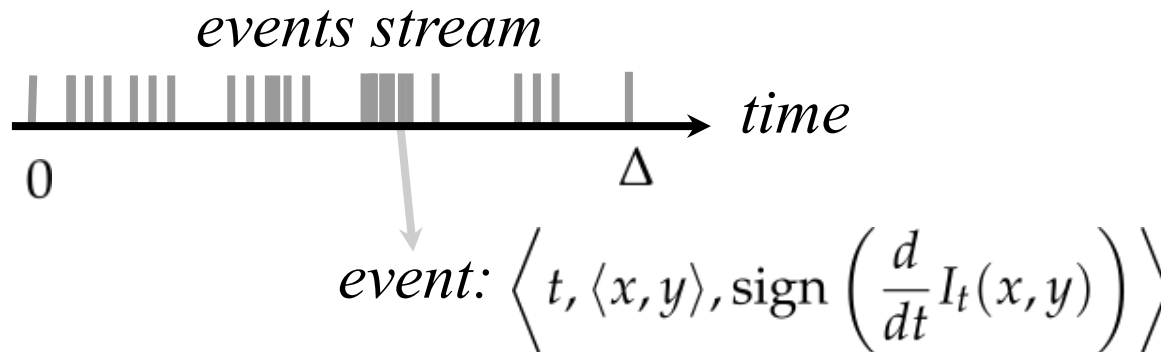


[S. Liu and T. Delbruck, Neuromorphic sensory systems'03 ]

- In a traditional camera, frames arrive at fixed intervals:



- In a DVS, an event is generated each time a single pixel changes value:



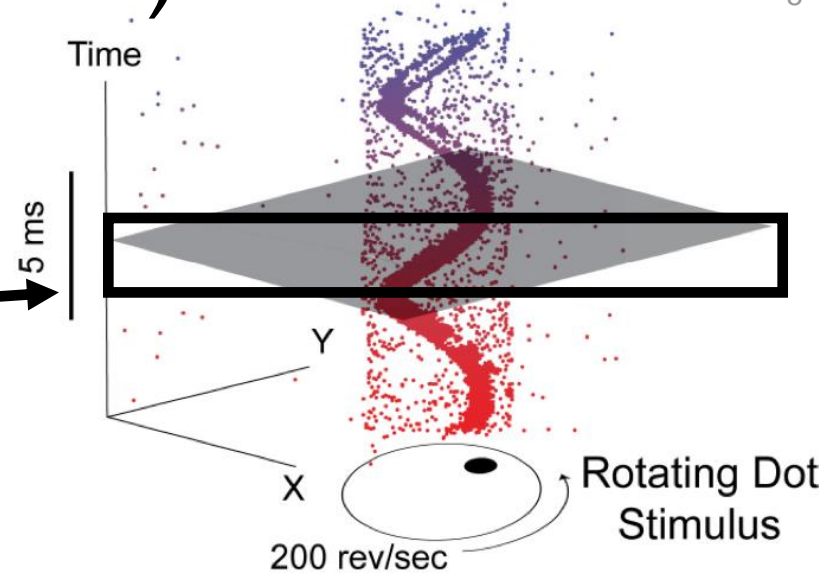
- This is an idealization of a very complicated circuit

# Dynamic Vision Sensor (DVS) Space-Time Spike Events

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

4  
5

- We can render the DVS data as a normal frame-based animation
- Each frame is a histogram of the events received in a given **time slice**
- For visualization only!

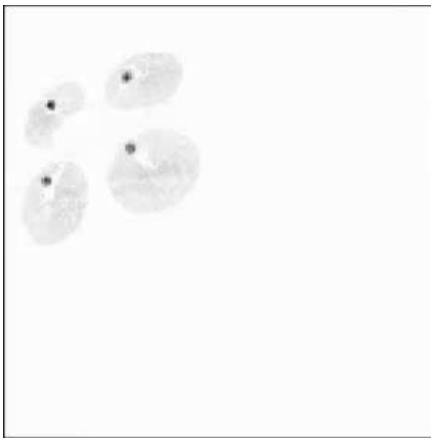
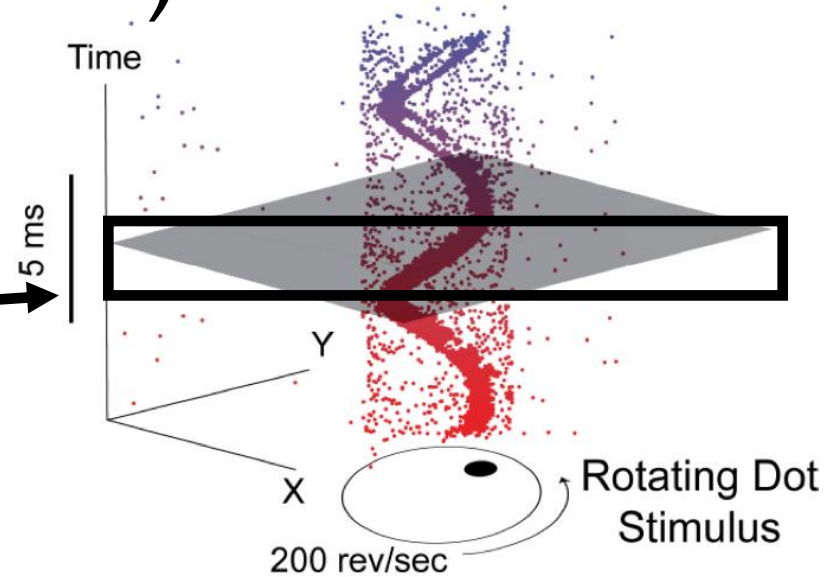


# Dynamic Vision Sensor (DVS) Space-Time Spike Events

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

- We can render the DVS data as a normal frame-based animation
- Each frame is a histogram of the events received in a given **time slice**
- For visualization only!

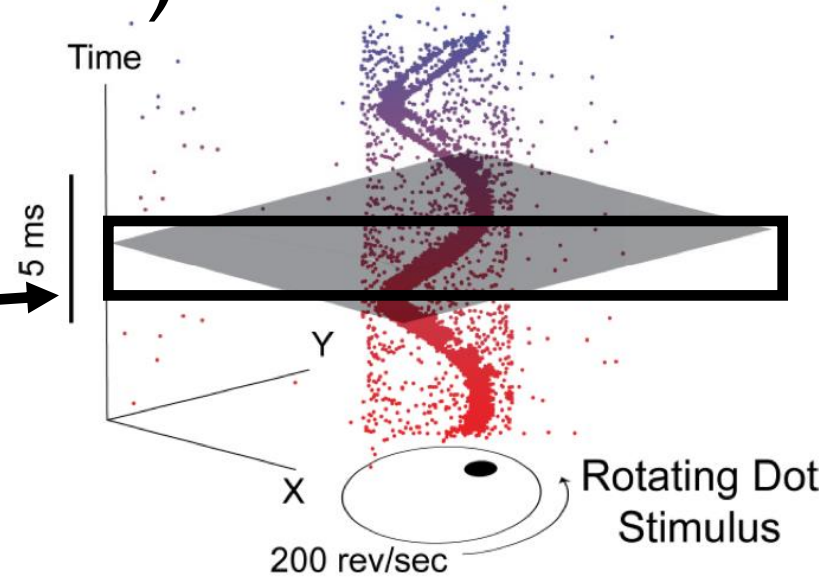
1 video frame = 33 ms (real time)



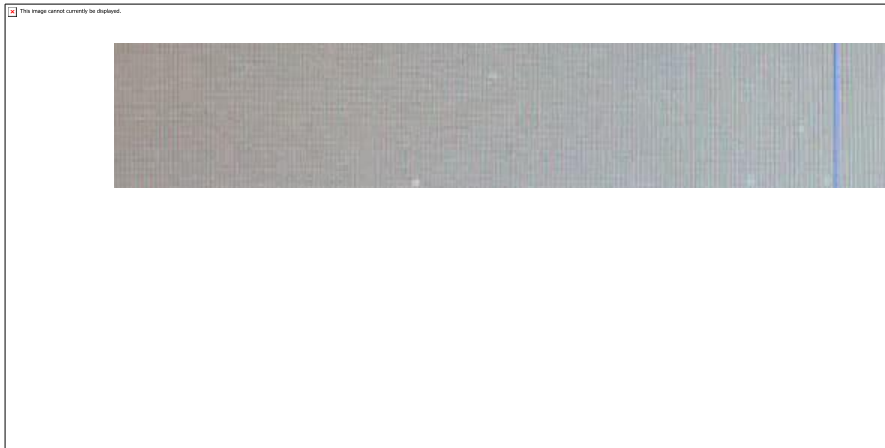
# Dynamic Vision Sensor (DVS) Space-Time Spike Events

- We can render the DVS data as a normal frame-based animation
- Each frame is a histogram of the events received in a given **time slice**
- For visualization only!

1 video frame = 1 ms



# Flip: video frame = 33 ms (i.e., real time)



***IROS Presentation TOMORROW – Session: «Localization II»***

*LocLow-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor , Censi,<sup>52</sup>Brandli, Delbruck, Scaramuzza*

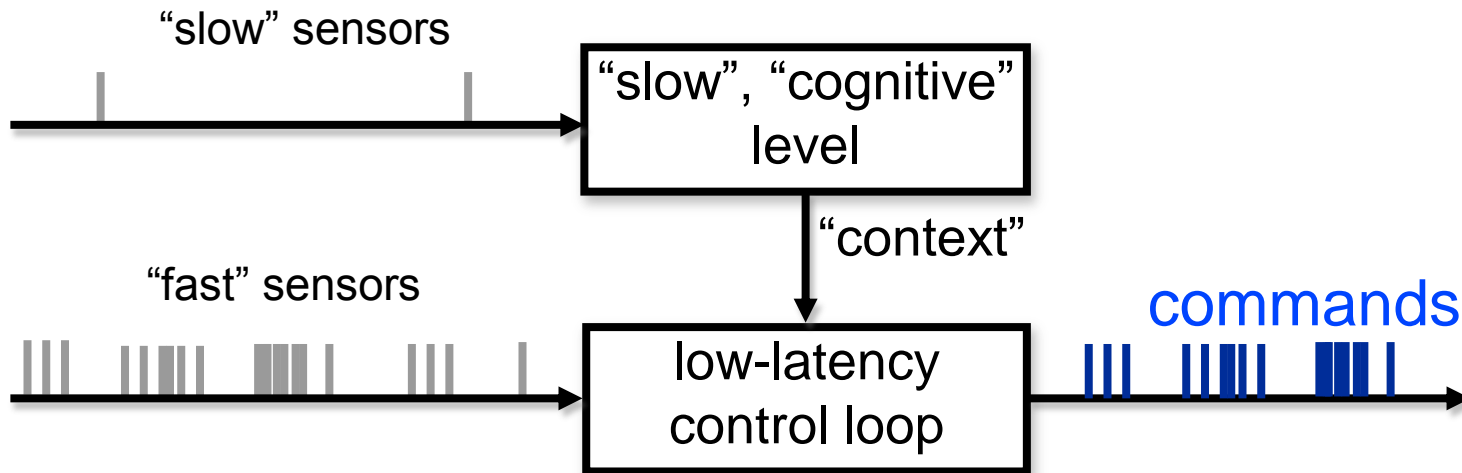


# Flip: video frame = 1ms

***IROS Presentation TOMORROW – Session: «Localization II»***

*LocLow-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, Censi, Brandli, Delbruck, Scaramuzza*

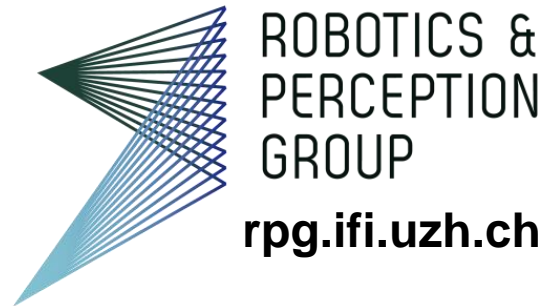
We might imagine a **two-level architecture** in which agile behavior is obtained by **low-latency control action** which uses the data from a sensor like DVS, while, at **slower time-scales**, other tasks such as SLAM are done based on slower traditional sensors.



***IROS Presentation TOMORROW – Session: «Localization II»***

*LocLow-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, Censi,<sup>14</sup>Brandli, Delbruck, Scaramuzza*

# Thanks!



# Our next IROS talks: Monday and Tuesday, Nov. 4-5

## Session Unmanned Aerial Vehicles IV

- ***Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles***  
by Christian Forster
- ***Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction***  
by Christian Forster
- ***MAV Urban Localization from Google Street View Data***  
by Andras Majdik

## Session: Localization II

- **Low-Latency Localization by Active LED Markers Tracking Using a Dynamic Vision Sensor**  
by Andrea Censi



## **Session II**

### **Perception**

- **Title: Enabling Efficient Registration using Adaptive Iterative Closest Keypoint**  
**Authors:** Johan Ekekrantz, Andrzej Pronobis, John Folkesson, Patric Jensfelt
  
- **Title: Information fusion and evidential grammars for object class segmentation**  
**Authors:** Jean-Baptiste Bordes, Philippe Xu, Franck Davoine, Huijing Zhao, Thierry Denoeux

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



# Enabling Efficient Registration using Adaptive Iterative Closest Keypoint

Johan Ekekrantz<sup>1</sup>, Andrzej Pronobis<sup>2</sup>, John Folkesson<sup>1</sup> and Patric Jensfelt<sup>1</sup>

**Abstract**—Registering frames of 3D sensor data is a key functionality in many robot applications, from multi-view 3D object recognition to SLAM. With the advent of cheap and widely available, so called, RGB-D sensors acquiring such data has become possible also from small robots or other mobile devices. Such robots and devices typically have limited resources and being able to perform registration in a computationally efficient manner is therefore very important. In our recent work [1] we proposed a fast and simple method for registering RGB-D data, building on the principle of the Iterative Closest Point (ICP) algorithm. This paper outlines this new method and shows how it can facilitate a significant reduction in computational cost while maintaining or even improving performance in terms of accuracy and convergence properties. As a contribution we present a method to efficiently measure the quality of a found registration.

## I. INTRODUCTION

Data registration is the natural next step after acquisition of sensory data. The goal is to align two frames of sensor data of the same scene taken from different locations. Registration is often used as a way to replace or enhance odometry obtained from wheel encoders. Registration is important because a robot’s behavior is based on its world model and that world model requires accumulation of data in a consistent reference frame. Therefore, a more accurate data registration allows the robot to make better inferences and decisions.

The recent advancements in RGB-D cameras have led to increasing use of range image data in robotics. The availability of both depth and visual information can largely simplify the registration itself. In this work, we focus on the problem of registration of RGB-D views and actively exploit the visual content to improve both accuracy and efficiency.

In [1], we present Adaptive Iterative Closest Keypoint (AICK), a registration algorithm for RGB-D views which builds on the idea of Iterative Closest Point (ICP) [2]. Algorithms based on the principle of ICP are able to provide very accurate estimations, given an initial transformation that is close to the final result. Unfortunately, the performance of standard ICP often deteriorates steeply with the decrease of the quality of the initial guess, as often happens in case of registration of views captured during fast sensor rotations. Additionally, noise can drastically affect the convergence of the iterative optimization method, with local minima being a common problem.

AICK preserves the accuracy of ICP for small transformations, while providing a drastic improvement of robustness to larger view rotations and translations without the need for an initial guess given sufficient overlap between the frames. Our algorithm exploits both depth and visual information and relies on keypoints detected in images associated with 3D positions in the local reference frame and a visual descriptor. The key property of the algorithm is the ability to weigh the importance of the visual descriptor and the 3D position while iteratively optimizing the transformation. This allows us to exploit the distinctiveness of appearance features for improved initial robustness and accuracy of point locations for the final precision.

In this paper we compare the proposed method to generalized ICP (GICP) [3], the 3D normal distribution transform (3D-NDT) [4] and a method based on RANSAC [5] and keypoints which we will call 3-point RANSAC and show how our method provides a significant reduction in computational cost without sacrificing performance and improving it significantly in most use cases. The evaluation of the four algorithms is performed on a publicly available dataset [6] and we base our quantitative analysis on an established benchmarking procedure and performance measure [6]. In this paper we also present an efficient way to assess the quality of the registration between two frames.

In the remaining parts of the paper, we first provide an overview of registration methods. Section III provides details of the proposed algorithm. Section IV covers the setup for the experimental evaluation. Finally, we present the results of the experimental evaluation in Section V.

## II. RELATED WORK

Most of the point cloud registration methods are based on the Iterative Closest Point (ICP) algorithm introduced in [2]. The most computationally expensive part of ICP is typically finding the closest points. The standard way of performing the matching is to use nearest neighbour matching in Euclidean space. This has a complexity of  $\mathcal{O}(N^2)$  in a naive implementation. A common way to speed this up is to use a kd-tree (or a set of trees) which reduces the complexity to  $\mathcal{O}(N \log(N))$ .

Each point cloud is a sample of the real world and even small perturbations in the sensor pose can lead to sampling different structures or parts thereof. A common way to address this is to make a parametric model and then, for example, fit the points in one frame against planes in the other as in [7] or more recently in the GICP algorithm [3] where both point clouds are models with planar surfaces.

<sup>1</sup>The authors are with the Centre for Autonomous System at KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {ekz, johnf, patric}@csc.kth.se

<sup>2</sup>A. Pronobis is with the Robotics and State Estimation Lab at the University of Washington. pronobis@cs.washington.edu

The 3D normal distribution transform (3D-NDT) [4], [8] fit Gaussian ellipsoids to the data which both address the issue of noise and reduces the dimensionality of the data, thus speeding up the processing. Similar work has been presented in [9]. The Multi-scaled EM-ICP [10] share some properties with AICK. It does not assume one data association but rather consider a weighted combination of matches with the scale setting the weight.

Using key points (such as SIFT [11], SURF[12], BRIEF [13], BRISK[14] and FREAK [15]) typically extracted from the RGB information, reduces the need to treat all pixels and using feature descriptors allows for reliable associations. An example of using key points and ICP to register RGB images is given by [16]. In this work we use SURF and ORB [17] which extends BRIEF with invariance to rotation. Key points are often detected by FAST [18] or Harris corners [19].

The Kinect Fusion algorithm [20] uses a dense, non-parametric, representation for the reference frame from which an artificial point cloud is sampled and registered against.

A common data association problem is that of looking for a match between one frame and all frames previously seen. Finding these, so called, loop closures are key to a successful implementation of SLAM. Here the question is first if the two frames match at all and if so what the transformation is. Matching feature by feature in each frame is prohibitively slow. A common approach taken is to make use of visual vocabularies [21]. The basic idea is to form clusters in descriptor space and assign a label to each cluster or word. The discretisation of descriptors into words means that feature matching can be done by comparison two integer indices (the label of the word). This has laid the foundation for FAB-MAP [22] and its follow-ups.

A major part of registration is the problem of outlier rejection i.e. the fact that there may be regions with no overlap. Using a suitable model, RANSAC [5] can be used to separate inliers from outliers and calculate model parameters.

### III. THE AICK ALGORITHM

The AICK algorithm is an efficient and accurate way to register two frames of RGB-D data. It exploits keypoints that have both a 3D position in space as used by ICP and a descriptor which characterizes the surrounding context of the point. In contrast to ICP, it is able to find a good registration even when no initial guess is given. AICK is an iterative algorithm that adaptively changes from emphasizing the descriptor match to emphasizing the geometric fit between the points in the two frames. At the later stages it becomes essentially ICP but having avoided the local minima that result from incorrect initial matches. The results are thus as for ICP with less failures.

As said, in ICP one must start with an initial guess of the transformation between the two frames. One then finds all the matching pairs of points. The matching criteria is the smallest Euclidean distance,  $d_e$  between the 3D points. After finding all matches where  $d_e$  is below a threshold, the

transformation is recomputed to minimize the sum of these distances.

The main strength of the ICP method is that it gives very accurate transformations when the matches are correct. It is most suitable for dense point clouds where sampling artifacts are not significant.

The main weakness of ICP is that if the initial guess leads to too many incorrect matches the solution can get 'stuck' trying to make those fit. It needs most of the initial matches to either be correct or at least on the correct smooth surfaces. The need to have a good guess to start with is rather problematic as it is just this transformation that we are after. It would be better if the method did not require any initial guess, especially when looking for loop closures. In AICK the initial match is independent of the transformation as it is based solely on the descriptor information.

AICK does not match dense point clouds but rather keypoints. Two similar features or the same feature seen from different angles will have descriptors that are close in this descriptor space. This way we reduce the number of points to consider for matching to only those points that have a key point associated with it. This then addresses the problem of which points to select as well.

AICK does the same two phases, match and optimize, as ICP but it uses a different matching metric which adapts over the course of the iterations. Instead of  $d_e$  we use  $d_i$ ,

$$d_i = (1 - \alpha^i)d_e + \alpha^i d_d, \quad (1)$$

where  $i \in \{0, 1, 2, 3, \dots\}$  is the iteration number,  $d_d$  is the distance, L2 norm, in descriptor space and the constant parameter  $\alpha \in [0, 1]$  is the decay factor to move from pure descriptor distance, ( $i = 0$ ) to nearly only Euclidean distance, ( $\alpha^i \ll 1$ )

In addition to assessing what points are closest, the distance metric is also used to reject points that are too far away. The distance  $d_e$  and  $d_d$  have different units and finding a threshold for the combined distances requires some thought. We define this threshold according to

$$\lambda_i = (1 - \alpha^i)\lambda_e + \alpha^i \lambda_d, \quad (2)$$

where  $\lambda_e$  is the outlier rejection corresponding to the euclidean distance and  $\lambda_d$  corresponding to the feature distance.

#### A. Non exhaustive search strategy

AICK reduces the computational requirement compared to standard ICP in several ways. Firstly, because it only uses points with an associated key point. Experiments also show that we do not have to perform an exhaustive search for the best matches. That is, even if we limit the search for the keypoints in one frame to only a small subset and miss some matches performance is maintained high given that we start with enough key points. This opens up ways to make the algorithm more efficient by trading off the expensive step of finding all the matches that fall below our threshold.

A common way to reduce the cost of matching, which we also make use of, is to use a so called 'vocabulary' of words do this we use the method of learning a 'vocabulary' of words

as in the bag of words method.<sup>1</sup> We learn the words using different data from what we test on. Learning corresponds to clustering the descriptors from all the training images into a predetermined number of clusters. The words are then the mean descriptors for each cluster.

With every keypoint,  $p_k$ , we associate a list of its closest words in that frame, which we denote as  $\Psi(p_k)$ .  $\Psi(p_k)$  contains the words to which the descriptor distance of the keypoint is less than a threshold,  $R_w$ . This can be done swiftly if the vocabulary contains few clusters or if the words are arranged in a tree structure that speeds up this search. Note that this is only done once per frame, i.e., if we match the frames to many other frames we need not recompute  $\Psi$ . This is key for applications such as SLAM where detection of loop closures look at the same frame for matches several times. To look for matches for a key point in frame A to key points in frame B we start with the closest words in frame A and match the key point only to the key points associated with the same words in the other frame. This can be made very fast by creating an index per frame from words to keypoints. Instead of having to match all points to all points we only match each point to a (small) subset of the points in the other frame. This can speed up the expensive association step by an order of magnitude in most cases. We consider this a generalization of the original algorithm as using  $R_w = \infty$  is equivalent to the original algorithm.

### B. Quality of registration

To assess the quality of the registration we subsample every RGB-D frame using a grid in the image plane. We store one validation point for each intersection point in this grid. When two frames are matched the validation points in the two frames are backprojected into the depth of the other frame.

These points are scored based on the difference  $d_i$  between the backprojected depth and the measured depth. If the absolute value of  $d_i$  is smaller than a threshold  $\Gamma_{good}$  this point is considered valid. In order to make use of knowledge of open space between the sensor and the depth reading, validation points that end up much closer (quantified in the form of a threshold  $\Gamma_{bad}$ ) to the sensor than the measured depth are penalized by assigning it a value  $\delta < 0$ . The definition of  $score(d_i)$  is summarized in the following equation.

$$score(d_i) = \begin{cases} 1, & |d| < \Gamma_{good} \\ \delta < 0, & d > \Gamma_{bad} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The overall quality measure,  $W$ , is given

$$W = \frac{1}{M} \sum_{i=1}^N score(d_i) \quad (4)$$

where  $N$  is the number of overlapping validation points and  $M = \max(M_{min}, N)$ , with  $M_{min}$  ensuring that  $W$  becomes small when  $N$  is small, i.e. when there is a small overlap.

<sup>1</sup>We do not use the 'bags' in this work only the words. The bags might be useful to chose which two frames to try to register to one another which is a question not addressed here.

## IV. EXPERIMENTAL SETUP

For evaluation we use [6] which is a publicly available dataset designed for the purpose of benchmarking RGB-D SLAM algorithms in realistic indoor environments. The dataset is complete with ground truth and contains sequences of RGB-D data captured using a Kinect. To be specific, we use the sequence *fr1/room* which at the time of writing this paper was the longest of all the sequences in the natural office environment subset. This data set is well suited to its designed purpose of testing state of the art registration algorithms in that the motion has all 6 degrees of freedom and the movement is both rapid and uneven.

### A. Performance Measure

We employed a performance measure provided together with the dataset [6]. The measure is based on the relative pose error, which is found by first transforming the origin pose using the estimated transformation and then transforming it back using the inverse of the ground truth transformation. In a perfect case without error, this results in a pose matching the origin pose.

$$E_i = G_i^{-1}Q_i - I, \quad (5)$$

where  $G_i$  is the ground truth transformation for transformation  $i$ ,  $Q_i$  is the estimated transformation and  $I$  is the identity matrix. We analyze the translation component of  $E_i$  by measuring the relative distance between the pose obtained after the two transformations described above and the origin pose as suggested in [6]. This error will be given in meters, see (6) for mathematical formulation. As a means of summarizing the results for a set of translation errors we define *successratio* as the ratio of translation errors smaller than some threshold  $\lambda_t$  in the set. That is the registration is considered a 'success' if it satisfies (6).

$$E_i^{Translation} = \left( \sum_{j=0}^2 ||E_{i,j,3}||^2 \right)^{1/2} < \lambda_t. \quad (6)$$

It is worth noting that when *successratio* = 0.5,  $\lambda_t$  is the median error. Similarly to using the median error the *successratio* considers all outliers as equal, meaning that gross outliers does not bias the analysis. This formulation allows us to analyze the distribution of errors by varying the threshold  $\lambda_t$ .

### B. Algorithms tested

Three different registration algorithms in addition to AICK<sup>2</sup> were ran and compared on the test set. The parameters for the algorithms were optimized by hand by testing a large set of values to yield good performance within a maximum of roughly five minutes of execution time per pairwise registration.

1) *GICP*: We use the GICP implementation provided by the Point Cloud Library (PCL [23])<sup>3</sup>.

<sup>2</sup>AICK using  $\lambda_e = 0.01m$  and  $\lambda_f = 0.2$ .

<sup>3</sup>GICP was allowed to run for 25 iterations. Rejection threshold = 0.004m.

2) *3D-NDT*: We use the 3D-NDT implementation provided by the Point Cloud Library (PCL [23])<sup>4</sup>.

3) *3-POINT RANSAC*: We used the RANSAC algorithm on this problem by first forming a list of potential matching keypoint pairs based on the similarity of the descriptors only. We then randomly select three of these pairs to define a transformation between the frames, which we will call the 'model'. We then count the number of 'inliers' according to the model. The model with the most inliers is chosen and updated by using all of the found inliers. In forming the list of potential matched pairs only associations between keypoints with descriptor distance  $d_d \leq \lambda_f$  are used. Inliers are calculated by transforming the keypoints in one frame by the model and associating the transformed keypoints to the closest keypoint in the other frame. If the euclidean distance  $d_e \leq \lambda_e$  between these keypoints the association is counted as an inlier<sup>5</sup>. For the 3-point RANSAC algorithm we use SURF keypoints.

We will use two different types of keypoints, SURF [12] and ORB [17]. The Surf keypoints will be extracted using OpenSURF Library[24]. Using our test set we found an average of 906 surf keypoints with valid depthdata in an average of 0.12 seconds. To extract the ORB keypoints we use OpenCV [25]. Using our test set we found an average of 857 ORB keypoints with valid depthdata in an average of 0.011 seconds.

### C. Experimental Procedure

The registration experiments were performed by estimating transformations between consecutive frames of the data sequence. In order to test robustness to larger transformations, we performed the experiments for pairs of frames separated by different lengths of time. Performance is measured quantitatively using the measure described in (6). The point clouds were created with calibrated camera parameters. In section V-C we visualize the effects of accumulating a sequence of consecutive frame transformations and transforming the appropriate pointclouds into a common coordinate frame.

## V. EXPERIMENTAL RESULTS

We plot the *successratio* versus a varying  $\lambda_t$  for (6) up to 0.05 meters using consecutive frames (around 30ms apart) for the different algorithms in fig. (1)<sup>6</sup>. This allows us to see both the size and variation of the translation error of the different methods when the transformation between frames is relatively small. A steep curve can be interpreted as good performance as that would mean that the method often yields a transformation with a small translation error. One sees that, for consecutive frames, all of the methods reach nearly 100% *successratio* at a relatively small  $\lambda_t$ . The conclusion is

<sup>4</sup>To keep the runtime reasonably low the pointclouds were subsampled through the use of a voxelgrid with a voxel size of 0.02m. 3D-NDT was allowed to run for 25 iterations, with *resolution* = 0.1 and *stepsize* = 0.09.

<sup>5</sup>We iterated the RANSAC over 400 random models in searching for the best model using  $\lambda_e = 0.02m$  and  $\lambda_f = 0.2$ .

<sup>6</sup>AICK was run for 25 iterations with  $\alpha = 0.8$  and  $R_w = \infty$

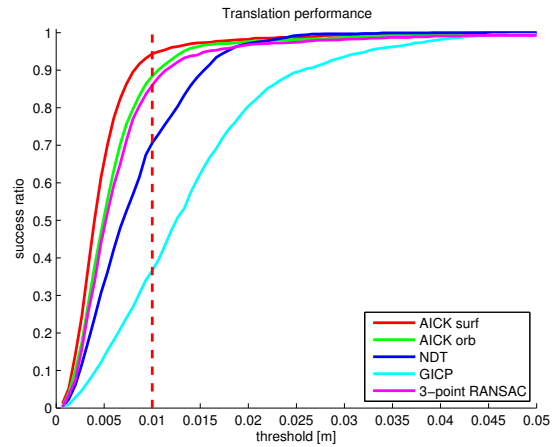


Fig. 1. The *successratio* as a function of the threshold on the translation error in m. Here we use all the found keypoints. The red dashed line shows the threshold used in fig. (2). Meaning that the intersections with the red dashed line are equivalent to values for the *successratio* in fig. (2) when the time difference between frames equals 30 ms.

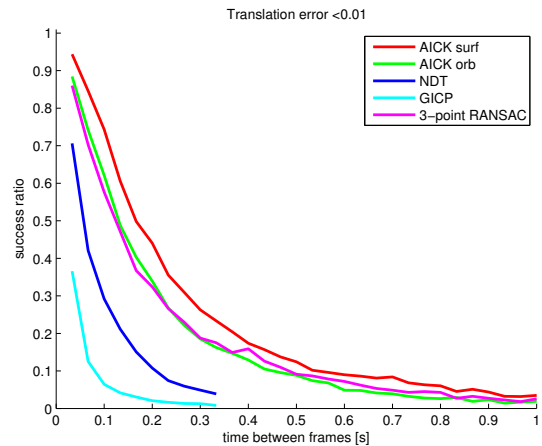


Fig. 2. *successratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m. Here we use all the found keypoints.

that while AICK using surf keypoints outperforms the other methods in this test all of the methods are fairly accurate given small displacements of the camera. It is also interesting to note that the difference between the use of surf and orb keypoints is relatively small for AICK.

It is also informative to see the result on the *successratio* by using a fixed threshold and varying the time difference between the frames being matched. This is shown in fig. (2)<sup>6</sup> for a threshold of 0.01 meters. It is clear that the AICK and 3-point RANSAC degrades much slower than GICP and NDT when the camera displacement increases.

### A. Quality of registration

By rejecting bad transformations we can ensure a higher performance for the non-rejected transformations. Fig. (3)<sup>7</sup> shows the effects on the *successratio* for two different thresholds on  $W$  (see eq. (4)). Notice the large difference

<sup>7</sup> Using a 10-by-10 subsampling grid with a minimum overlap of 500 samples,  $\Gamma_{good} = 0.01m$ ,  $\Gamma_{bad} = 0.075m$  and  $\delta = -2$ .



Algorithm		$R_w$	Iterations	Avg runtime [s]	successratio for threshold $\lambda_t$		
Keypoints	$\lambda_t = 0.0033$				$\lambda_t = 0.01$	$\lambda_t = 0.05$	
AICK	on average 906 surf keypoints	$\infty$	25	0.180	0.374	0.944	0.993
AICK	on average 857 orb keypoints	$\infty$	25	0.135	0.276	0.885	0.999
AICK	max 200 surf keypoints	$\infty$	5	0.00385	0.281	0.902	0.993
AICK	max 350 orb keypoints	$\infty$	10	0.0113	0.209	0.833	0.998
AICK	max 200 surf keypoints	0.26	5	0.000445	0.258	0.888	0.992
AICK	max 200 surf keypoints $W > 0.7$	0.26	5	0.000480	0.313	0.953	0.999
AICK	max 200 surf keypoints $W \leq 0.7$	0.26	5	0.000480	0.156	0.740	0.977
AICK	max 200 surf keypoints $W > 0.5$	0.26	5	0.000480	0.285	0.931	0.999
AICK	max 200 surf keypoints $W \leq 0.5$	0.26	5	0.000480	0.075	0.500	0.932
AICK	max 350 orb keypoints	0.165	10	0.000717	0.209	0.828	0.995
GICP			25	224	0.070	0.366	0.996
NDT			25	237	0.177	0.706	1
3-point ransac			400	4.09	0.255	0.860	0.993

TABLE I

RUNTIME COSTS AND PERFORMANCES FOR THE TESTED ALGORITHMS.  $R_w$  IS THE RADIUS AROUND THE KEYPOINT TO FIND MATCHING WORDS.

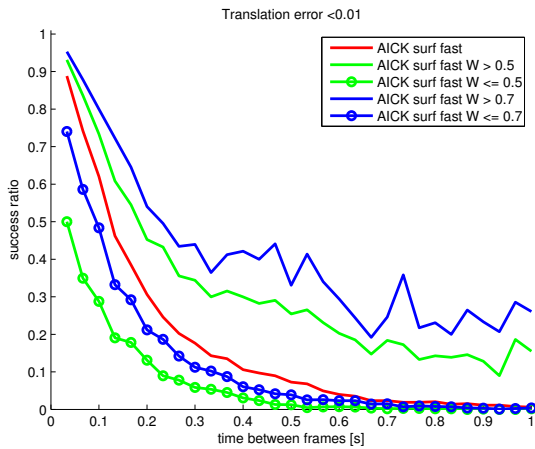


Fig. 3. *successratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

between the cases when the  $W$  is over a specified threshold as compared to being under this threshold.

### B. Runtime

We can control the runtime to performance trade-off of the algorithm using three main parameters: the number of keypoints used, the number of iterations the algorithm is allowed to run and the threshold  $R_w$ . The effects on the *successratio* from limiting these parameters can be seen in Table I for registration of two consecutive views. The cost for extracting keypoints used by AICK or 3-point RANSAC is not included in the table. The reason being that in many applications keypoint extraction is only done once per frame whereas frame to frame registration may be run multiple times per frame. For the frames in the test set we found an average of 906 surf keypoints with valid depthdata in an average of 0.12 seconds and an average of 857 ORB keypoints with valid depthdata in an average of 0.011 seconds. It can be seen that the keypoint based methods are much faster than the non-keypoint based methods. Obviously runtime is dependent on implementation but since the keypoint methods deal with a lot less data there are less calculations to be done. By controlling the parameters for the AICK algorithm results

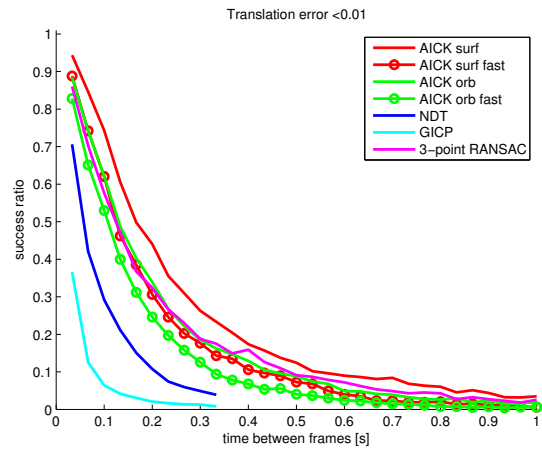


Fig. 4. *successratio* as a function of the time difference between frames with a fixed threshold on the translation error of 0.01 m.

similar to that of the 3-point RANSAC can be achieved in a fraction of the time. It can also be seen in table (I) that extracting an estimate of the quality of the registration can be done at a small computational load.

The performance of AICK using different parameter settings is shown in table (I) and in fig. (4)<sup>8,9</sup>.  $R_w = \infty$  indicates not using words at all. Table (I) shows that tuning the parameters of the algorithm can greatly speed up the registration while fig. (4)<sup>8,9</sup> shows that the drop in performance was relatively small.

### C. Visual inspection

The AICK algorithm clearly outperforms the other methods in both robustness and precision as the above results show. In fig. (5) we visualize the results of accumulating transformations estimated by AICK over a sequence of 1000 frames. This is a common and effective way to allow for a qualitative evaluation by visual inspection. Because

<sup>8</sup>AICK orb fast was run 10 iterations with  $\alpha = 0.6$ , a maximum of 350 orb keypoints and  $R_w = 0.165$ .

<sup>9</sup>AICK surf fast was run 5 iterations with  $\alpha = 0.3$ , a maximum of 200 surf keypoints and  $R_w = 0.26$ .



Fig. 5. Rendering of the the points given by frame-to-frame transformation estimates when walking past a series of bookshelves in the KTH library. The data is displayed from three different view points. The bookshelves are lined up in the library and the upper part of the image shows that our method produces results very close to this even using pure dead-reckoning.

transformations are added frame by frame, i.e. pure dead-reckoning, errors, especially in orientation, will result in clearly visible distortions. To remove the background and avoid displaying noisy data, only data captured close to the sensor is displayed. The absence of distortions lends credibility to the practical use of the AICK method on real world systems.

## VI. SUMMARY AND CONCLUSIONS

In this paper we have clearly shown that AICK is natural choice for small robots, mobile devices and other embedded systems with limited resources but where high performance is needed. This is made possible by transitioning between coarse, appearance-based registration such that no initial estimate is needed and fine registration using position-based ICP on distinctive keypoints. In order to verify the performance of our method, we employed a standard benchmark consisting of a dataset and performance measure [6]. We compared the method to three different high performance registration techniques. In the experiments our method showed a significant improvement of both robustness to larger transformations and precision of the final result which can be attributed to the adaptive distance metric. Furthermore, sub-sampling of the point cloud into a selection of keypoints resulted in an algorithm orders of magnitudes faster than algorithms used for comparison.

## ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 600623 and by SSF through its Centre for Autonomous Systems.

## REFERENCES

- [1] J. Ekekrantz, A. Pronobis, J. Folkesson, and P. Jensfelt, "Adaptive iterative closest keypoint," in *Proceedings of the 6th European Conference on Mobile Robots (ECMR '13)*, 2013.
- [2] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intel.*, no. 2, pp. 239–256, 1992.
- [3] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proceedings of Robotics: Science and Systems*, (Seattle, USA), June 2009.

- [4] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, pp. 803–827, 2007.
- [5] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [7] Y. Chen. and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation*, pp. 2724–2729, 1992.
- [8] T. Stoyanov, M. Magnusson, and A. J. Lilienthal, "Point Set Registration through Minimization of the L2 Distance between 3D-NDT Models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 14–19 2012.
- [9] L. Montesano, J. Mingeuz, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2005.
- [10] S. Granger and X. Pennec, "Multi-scale em-icp: A fast and robust approach for surface registration," in *European Conference on Computer Vision*, 2002, pp. 418–432, 2002.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision—ECCV 2006*, pp. 404–417, Springer, 2006.
- [13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Computer Vision—ECCV 2010*, pp. 778–792, Springer, 2010.
- [14] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.
- [15] A. Alahi, R. Ortiz, and P. Vanderghenst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517, IEEE, 2012.
- [16] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai, "Registration of challenging image pairs: Initialization, estimation, and decision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1973–1989, 2007.
- [17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.
- [18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision—ECCV 2006*, pp. 430–443, Springer, 2006.
- [19] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, pp. 147–151, 1988.
- [20] R. Newcombe, A. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 127–136, IEEE, 2011.
- [21] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1470–1477, IEEE, 2003.
- [22] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [23] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *International Conference on Robotics and Automation*, (Shanghai, China), 2011 2011.
- [24] C. Evans, "Notes on the opensurf library," Tech. Rep. CSTR-09-001, University of Bristol, January 2009.
- [25] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.



# Information fusion and evidential grammars for object class segmentation

Jean-Baptiste Bordes<sup>1</sup> Philippe Xu<sup>1,2</sup> Franck Davoine<sup>2</sup> Huijing Zhao<sup>2</sup> Thierry Dencœur<sup>1</sup>

**Abstract**—In this paper, an original method for traffic scene images understanding based on the theory of belief functions is presented. Our approach takes place in a multi-sensors context and decomposes a scene into objects through the following steps: at first, an over-segmentation of the image is performed and a set of detection modules provides for each segment a belief function defined on the set of the classes. Then, these belief functions are combined and the segments are clustered into objects using an evidential grammar framework. The tasks of image segmentation and object identification are then formulated as the research of the best parse graph of the image, which is its hierarchical decomposition from the scene, to objects and segments while taking into account the spatial layout. A consistency criterion is defined for any parse tree, and the search of the optimal interpretation of an image formulated as an optimization problem. We show that our framework is flexible enough to include new sensors as well as new classes of object. The work is validated on real and publicly available urban driving scene data.

## I. INTRODUCTION

Automatic understanding of the scene in front of a car is an essential task for advanced driver assistance or safety systems. Automatic understanding denotes generally a segmentation of the image scene into its constituting objects, augmented eventually with spatial or functional relationships. However, there are many classes of objects which can be found in traffic scenes, and for most of them, their level of variability is very high. Indeed, detecting even a single kind of object can be very challenging since the highly cluttered environment as well as the dynamically changing backgrounds, among others, contribute to the difficulty of such a task. Many approaches have been proposed recently to tackle individual problems such as road detection or pedestrian detection, and they can use different kinds of sensors.

### A. Related Work

In the last decade, the accuracy of object detection methods has increased substantially thanks to the appearance of efficient visual descriptors in images such as SIFT as well as the success of computer vision challenges such as PASCAL. In the field of intelligent vehicles, they are mainly applied to pedestrian detection which is the most studied case [7], even if more classes have also been considered [8]. However, to reach better performances, more sensors are generally used:

<sup>1</sup>Jean-Baptiste Bordes, Philippe Xu and Thierry Dencœur are with UMR CNRS 7253, Université de Technologie de Compiègne, BP 20529, 60205 Compiègne Cedex, France. [bordes@nlpr.ia.ac.cn](mailto:bordes@nlpr.ia.ac.cn)

<sup>2</sup>Philippe Xu, Franck Davoine and Huijing Zhao are with LIAMA, CNRS, Key Lab of Machine Perception (MOE), Peking University, Beijing, P.R. China. [philippe.xu@hds.utc.fr](mailto:philippe.xu@hds.utc.fr)

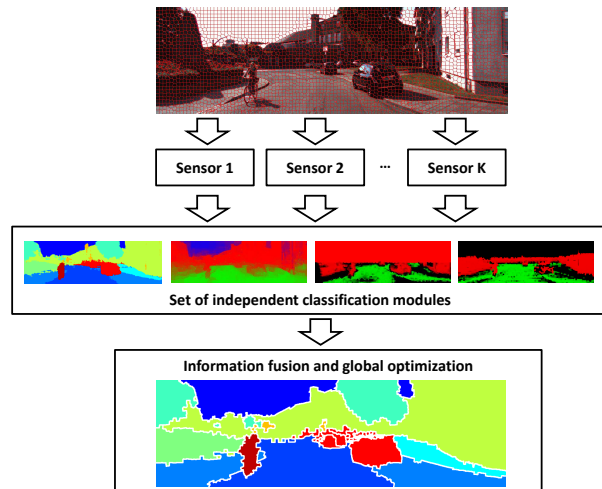


Fig. 1. Overview of the system. The scene is perceived by several sensors among which a camera provides an over-segmented image. A set of independent classification modules then gives some partial information which are finally combined through a global optimization scheme.

LIDAR sensors are widely used to detect static structures but also moving objects [14]. Depth information from stereo camera systems has also been used by Ess et al. [8] as well as Gavrila et al. [10] for pedestrian detection, it has also proven to be efficient to detect obstacles and navigable space [2]. Most of these methods are based on local visual clues, but some other approaches add to this local step a post-processing to take advantage of some consistency clues. Wojek et al. [15] perform joint object detection to take into account the spatial relationships between objects. Brehar [4] uses openCyc ontology to exploit inter-class relationships between classes in traffic scenes. Impressive results have also been obtained in [17] on a great variety of databases including traffic scenes using visual grammars, which is an adaptation of formal grammars for visual data. The objects and their components are first defined in the model and then, given a new image, a parse graph is computed, which is the decomposition of the scene into objects and parts of objects, down to the image primitives. Visual grammars have shown generalization capabilities and provide efficient way to face problems such as occlusion and scale.

### B. Contribution

In this work, instead of presenting new efficient descriptors which are already numerous in the literature, we present a method to make the most of the existing works. For this purpose, the Dempster Shafer theory on belief functions is

used to properly fuse a set of relevant sources of information, that we call in this article "modules", even when each one of them is reasoning independently in its own decision space. This framework has several strong advantages. First of all, it provides a high level of flexibility to the system: new sensors and modules can be added easily, and their output will be fused in a common space. Reversely, the independence of the modules before fusion makes our system robust to sensor failure. Moreover, we will show that new classes can be added easily as well, since belief functions make it possible to work on sets of classes and not only on individual classes. Some expert information on the relative position of the objects in a scene is also taken into account as an other source of information by the use of an innovative framework called "evidential grammar".

### C. Overview

The architecture of the system we consider, illustrated on Fig. 1, consists of a set of sensors including a camera. The image provided by the camera is over-segmented as a first step of image processing. We also consider a set of independent modules (road detection module, pedestrian detection module, etc.) receiving data from the sensors, the output of which is transformed into belief function before being fused at the segment level. Finally, the evidential grammars provide some kind of "global fusion" to this segment level information and strengthen weak detections as well as prune misdetections. We will show how this framework can be applied in practice by considering a monocular camera, stereo camera and a LIDAR. Our system is validated on the KITTI Vision Benchmark Suite [9].

## II. MULTI-MODAL AND MULTI-CLASS FUSION

When working in a multi-modal context, several challenges arise. First of all, the sources of information may be of very different nature, they may come from several types of sensors or even from prior knowledge. Each source having its own specificity, complementary information can be fetched from them. For example, 3D information from a stereo camera or a LIDAR can be used to detect obstacles while texture and color, from a monocular camera, can be used to detect vegetation or the sky. The second challenge is now to properly combine information about different classes of objects.

We follow the framework proposed in [16] which can deal with those two issues. The information from all the sources are projected onto the image space and formulated as an image labeling problem. Meaning that each pixel of the image has to be classified. A first over-segmentation is however done so that the classification do not have to be done at the pixel level which is often too local. The combination over different sets of classes is handled using the theory of belief functions.

### A. Dempster Shafer's theory of belief functions

1) *Reasoning on sets with belief functions:* The belief functions theory is an extension of classical probability

which is especially well adapted for reasoning on sets. Given a set of classes  $\Omega = \{\omega_1, \dots, \omega_K\}$ , a *mass function*, or *basic belief assignment* (BBA), is a function  $m : 2^\Omega \rightarrow [0, 1]$  verifying:

$$m(\emptyset) = 0, \quad \sum_{A \subseteq \Omega} m(A) = 1. \quad (1)$$

Contrary to a probability distribution which assigns a probability to every class, a mass function can assign a mass on any set of classes. Let us notice that a mass function whose non-zero values are only on singletons is equivalent to a Bayesian probability.

The plausibility is another measure often used to manipulate mass functions, it is defined as:

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Omega. \quad (2)$$

When a decision has to be made, the singleton with maximum plausibility is usually a good choice.

Given two mass functions  $m_1$  and  $m_2$ , they can be combined by using the Dempster's rule of combination to give a new mass  $m_{1,2} = m_1 \oplus m_2$  defined as:

$$\begin{aligned} m_{1,2}(\emptyset) &= 0, \\ m_{1,2}(A) &= \frac{1}{1 - \kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \end{aligned} \quad (3)$$

where  $\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$  measures the amount of conflict between the two mass functions.

2) *Reasoning in the product space:* In the method described in this paper, it will be necessary to introduce a set of evidential variables, and thus mass functions have to be manipulated on product spaces. Some well known operations that are used for Bayesian functions have to be introduced for mass functions. In all this section, two evidential variables  $X$  and  $Y$  are defined respectively on  $\Omega_X$  and  $\Omega_Y$ .

a) *Marginalization:* In this problem, the joint mass function  $m_{XY}$  is assumed to be known, the operation of marginalization can be used to get  $m_X$ :

$$m_{XY \downarrow X}(B) = \sum_{A \subseteq \Omega_{XY} | A \downarrow \Omega_X = B} m_{XY}(A), \quad \forall B \subseteq \Omega_X. \quad (4)$$

b) *Vacuous extension:* In this problem, the belief mass  $m_X$  is assumed to be known and we wish to extend it to the product space. The belief function theory suggests to choose the least informative mass function which provides  $m_X$  after it marginalization:

$$m_{X \uparrow XY}(A) = \begin{cases} m_X(B) & \text{if } A = B \times \Omega_Y, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

c) *Conditioning:* In this problem, the joint mass function  $m_{XY}$  is assumed to be known and  $X$  is supposed to belong to  $B$ , we denote:  $m_X^B(B) = 1$ . The conditioning operation is defined by:

$$m_{Y|X}(\cdot | B) = (m_{X \uparrow XY}^B \oplus m_{XY})_{XY \downarrow Y}. \quad (6)$$

The mass function  $m_{Y|X}(\cdot | B)$  is called conditional mass function knowing that  $B \subseteq \Omega_X$ .

d) *Deconditioning*: In this problem, the conditional mass function  $m_{Y|X}(\cdot|B)$  and we wish to evaluate  $m_{XY}$ . The belief function theory suggests to choose the least informative mass function which provides  $m_{Y|X}(\cdot|B)$  after conditioning:

$$m_{XY}(C) = \begin{cases} m_{Y|X}(A|B) & \text{if } C = (B \times A) \cup (B \times \Omega_Y), \\ 0 & \text{if different for all } C \subseteq \Omega_{XY}. \end{cases} \quad (7)$$

### B. Constructing belief functions

There are different ways to construct a belief function from data. Several classifiers such as the evidential  $k$ -nearest neighbors and neural network from Denoeux [5], [6] directly give a mass function as output.

For binary classification problem ( $\Omega = \{C, \bar{C}\}$ ), the general formulation proposed by Xu et al. [16] is used to transform the classifier output into a mass function. In this paper, our method is also enriched by taking into account the outputs of classical multiclass classifiers such as SVM or boosting which provide a set of score measures for each class which is denoted here  $(s_1, s_2, \dots, s_K)$ . To extract from this output a mass function, the following steps are processed, similarly to [1]:

- The scores are transformed into a probability distribution using a softmax function:

$$p(\omega_k) = \frac{\exp(s_k)}{\sum_{j=1}^K \exp(s_j)}. \quad (8)$$

- The probability are then transformed into a possibility:

$$\text{poss}(\{\omega_k\}) = \sum_{\omega_j \in \{\omega_1, \dots, \omega_K\}} \min(p(\omega_k), p(\omega_j)). \quad (9)$$

- The possibilities  $\pi_k = \text{poss}(\{\omega_k\})$  are sorted so that:

$$\pi_1 \geq \pi_2 \geq \dots \geq \pi_K. \quad (10)$$

- The possibility is finally transformed into a consonant mass function:

$$m(A) = \begin{cases} \pi_k - \pi_{k+1} & \text{if } A = \{\omega_1, \dots, \omega_k\}, \\ \pi_K & \text{if } A = \Omega, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

## III. GLOBAL FUSION PROCESS USING EVIDENTIAL GRAMMARS

The previous step is local since for every segment, the belief function describing its class is computed only with the information lying inside the segment. In this section, a global fusion process on the top of this local fusion step will be presented using evidential grammars. Thus, the mass functions of the segments will be combined, and prior information provided by experts about the possible relative positions of objects in a traffic scene will be added as well. The goals which are expected from this stage are: segmentation of the scene into objects by grouping the segments corresponding to a single instance of a class and disambiguation of the belief functions at the local level as well as reduction of false positives.

### A. Evidential Grammars

A grammar is defined as a 4-tuple  $\{V_N, V_T, S, \Gamma\}$  where  $V_N$  is a finite set of non-terminal nodes,  $V_T$  a finite set of terminal nodes,  $S$  a start symbol at the root, and  $\Gamma$  is a set of production (or derivation) rules. A production rule  $\gamma \in \Gamma$  changes a string of symbols (containing at least one non-terminal symbol) into another string of symbols. The production process starts with the  $S$  symbol and stops when the string is composed only of terminal symbols. The set of all the possible strings which can be produced by a grammar is called a *language*. The strength of grammars lies in the fact the language generated by a grammar can be large even when the *vocabulary*, that is to say  $V_T$  and  $V_N$  contain few elements.

To deal with image grammars, the natural left-to-right ordering is replaced with spatial relationships such as “hinge”, “border”, or “occlude”, which are used to combine segments into complex and structured objects. Moreover, to rank alternative interpretations and take into account uncertainty (on the class of the objects, on their relationships and on the derivation process), the grammar is augmented to a 5-tuple  $\{V_N, V_T, S, \Gamma, \mu\}$  by adding a fifth component  $\mu$  containing a set of conditional mass functions expressing our knowledge about the decomposition of the scene and the objects. This 5-tuple is called “evidential grammar”, the global framework of which has been detailed in [3], we thus expose here briefly the main aspects of this method.

### B. Model of an image interpretation

The image interpretation is represented by a parse hypergraph. A parse tree is a decomposition of a scene into its components. For this purpose, several partitions of the image into regions are considered, each one corresponding to a level of description: objects, parts-of-objects, segments etc. An evidential variable  $X_i$  is set for every region  $R_i$  to describe its class, and every region is assumed to contain one single instance of an object: let us emphasize that uncertainty on the value of  $X_i$  doesn't mean that several classes might be mixed in  $R_i$ . To group them into a single entity, the pair  $(R_i, X_i)$  is called a “node” denoted  $N_i$ . Except in the case when  $X_i$  is associated to a region at the segment level,  $R_i$  is partitioned into regions of the lower level of description the corresponding nodes of which will be called “children nodes” of  $N_i$ . To get a parse hypergraph, the parse tree is augmented with spatial and contextual relationships between the children of a given node. These relationships depend of the level of interpretation where the nodes are lying, relationships such as “aligned” or “borders” can be used at a part-of-object level, “occludes” or “supports” at an object level. These relationships are taken into account by adding an evidential variable  $\Xi_i$  in the graph taking its value in the discernment frame composed of the set of relationships for the corresponding level of description. This makes it possible for instance to model a pedestrian as a head “over a” body.

In [3], it is shown that a parse hypergraph can be set in relationship with an evidential network by assuming that the joint belief function of a node and its children can be

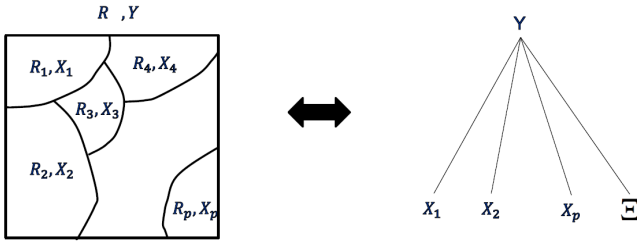


Fig. 2. Correspondence between a partition of an object into  $p$  components and the graphical dependency of the related variables. The variable  $\Xi$  describes the spatiale relationships between the regions  $R_1, R_2 \dots R_p$ .

expressed independently of the other nodes of the graph. The evidential variables describing the content of the segments are provided as the output of the local fusion step. Given an evidential network, the belief is propagated from the leave nodes to the other nodes of the network up to the scene level through a bottom-up inference stage. This is performed through a succession of classical operations of belief functions: deconditioning, vacuous extension, Dempster's combination in the production space, and marginalization on the variable of the father's node. More precisely, if  $Y$  is a node the children nodes of which are denoted  $X_1, X_2, \dots, X_p$  and the spatial relationship between those latter nodes is denoted as an evidential variable  $\Xi$  as illustrated on Fig. 2. In a first step, the vacuous extension is applied to the functions  $m_{X_1}, m_{X_2}, \dots, m_{X_p}$  and  $m_{\Xi}$ . The resulting functions are denoted  $m_{X_1 \uparrow X_1, X_2, \dots, X_p, \Xi, Y}, m_{X_2 \uparrow X_1, X_2, \dots, X_p, \Xi, Y}, \dots, m_{X_p \uparrow X_1, X_2, \dots, X_p, \Xi, Y}$ , and  $m_{\Xi \uparrow X_1, X_2, \dots, X_p, \Xi, Y}$ . These belief functions characterize the contents of disjoint regions and are thus supposed to be independent pieces of evidence. These belief functions are then combined using Dempster's rule:

$$m_{X_1, X_2, \dots, X_p, \Xi, Y}^1 = \left( \bigoplus_{i=1}^p m_{X_i \uparrow X_1, X_2, \dots, X_p, \Xi, Y} \right) \dots \bigoplus m_{\Xi \uparrow X_1, X_2, \dots, X_p, \Xi, Y}. \quad (12)$$

In a second step, all the  $N$  conditional belief functions corresponding to grammar rules involving the rewriting of a symbol into  $p$  symbols are deconditioned into a set of  $N$  functions denoted here  $m^k$  defined on the product space  $\{X_1, \dots, X_p, \Xi, Y\}$ . These belief functions correspond to distinct production rules which themselves encode different semantic information about the decomposition of the objects and the scene. They are thus supposed to be independent pieces of information and Dempster's rule of combination is consequently applied. We have:

$$m_{X_1, X_2, \dots, X_p, \Xi, Y}^2 = \bigoplus_{k=1}^N m_{X_1, X_2, \dots, X_p, \Xi, Y}^k. \quad (13)$$

where  $\Xi$  is the observable variable defining the spatial relation between the regions.  $m^2$  is then combined with  $m^1$ , and a belief function taking into account all the available information is thus obtained:

$$m_{X_1, X_2, \dots, X_p, \Xi, Y} = m_{X_1, X_2, \dots, X_p, \Xi, Y}^1 \oplus m_{X_1, X_2, \dots, X_p, \Xi, Y}^2. \quad (14)$$

The joint mass  $m_{X_1, X_2, \dots, X_p, \Xi, Y}$  is finally marginalized to extract  $m_Y$ :

$$m_Y = m_{X_1, X_2, \dots, X_p, \Xi, Y \downarrow Y}. \quad (15)$$

### C. Search for the optimal interpretation

By using the scheme detailed in the previous section, the belief is propagated from the segments up to the root to get an interpretation of an image. However, a large number of possible parse trees can be considered and consequently as many possible interpretations of a same image. We choose here to define the optimal parse tree as the one minimizing the conflict on the root node. Since, the non-normalized Dempster combination is applied, the root node aggregates all the conflict contained in the evidential network and thus gives a measure of the quality of the hierarchy.

A greedy algorithm is finally used to search for the optimal parse hypergraph in reasonable computation time. The main idea of this algorithm is to initiate a complex configuration which is simplified step by step as long as the consistency measure of the parse tree decreases:

- A parse tree is first initialized by linking all the nodes corresponding to the segments of the image directly to the root node. This is equivalent to considering that every segment is interpreted as one object.
- As long as the consistency measure of the parse tree decreases:
  - The consistency measure is computed for a set of alternative hypergraphs, each one being obtained by applying one single elementary modification to the current parse hypergraph. The elementary modifications that we consider are the merging of every pair of nodes of the same level of the hierarchy of the parse graph. If the nodes are terminal nodes, a new node is created which is linked with this pair of nodes. If the nodes are not terminal nodes, a new node is created which is composed of all the children of this pair of nodes.
  - The parse hypergraph minimizing the consistency measure is kept for the next iteration.
- The last parse hypergraph is kept as the output of the method.

## IV. EXPERIMENTS

The KITTI Benchmark Suite [9] was used to validate our approach. A set of 140 images has been annotated manually with a total of 14 classes as listed in Tab. I. Several modules were trained on 100 images and tested on the 40 others.

### A. Sensors and modules

We used a monocular camera, a stereo camera and a LIDAR as sensors. The principal monocular classification module is the Automatic Labeling Environment (ALE) proposed by Ladický et al. [12], which can be directly learned over all the previously defined classes. The second monocular module, from the works of Hoiem et al. [11], estimates the scene geometry from one single image. The classification output

Ground	Sky	Static structures	Building	Moving obstacles
	Road		Pole	
	Sidewalk		Fence	
	Lane marking		Other	
	Grass		Car/trucks	
	Tree	Person/cyclist		
	Other	Other		
	Vegetation			

TABLE I

CLASSES OF OBJECTS CONSIDERED IN OUR EXPERIMENTS. THERE ARE 14 CLASSES IN TOTAL, SOME OF THEM CAN BE GROUPED INTO SETS.

is limited to the three classes: ground, obstacles and sky. In our case, the ground class is the union of road, sidewalk, lane marking and grass, while the obstacles class includes everything else except the sky. The class membership scores from those two modules are transformed into a mass function following the steps (8-11).

Then the 3D information from the stereo camera and the LIDAR are used to detect the ground as in [16], by assuming that the ground is planar. Again, the classes ground and non-ground are actually sets of other classes. We clearly see the interest of working with sets of classes.

### B. Grammar model

A three levels grammar model was considered: scene, objects, and segments. No parts of objects (such as “wheel” or “head”) were considered in these experiments, and the objects are supposed to be derived directly into the set of their constituting segments which can be considered as elementary pieces of the objects. Consequently, the size of the discernment space at the segment level is the same as the one at the object level. The pairwise links “occlude”, “is occluded by”, “bordering” and “disjoint” were used to describe spatial relationships between the objects. The root node can produce any arbitrary combination of instances of the 14 types of objects we consider under a set of 30 spatial constraints. These constraints correspond to the derivation rules which are formulated as a set of prohibited configurations between pairs of objects, for example: “The sky cannot occlude an other object”, “A car cannot border a building”, “The road cannot occlude an other object”, etc. Each object can be decomposed in an arbitrary combination of patches of the corresponding object under a spatial constraint of neighborhood. It should be noticed that this model takes little advantage of the potential of the grammars to decompose complex objects in structured reusable components. Indeed, no database annotated with parts of objects were available for that purpose.

### C. Results

The inputs and outputs of our system are illustrated on Fig. 3. The average precision of the multi-class classification is showed on Tab. II. We can see that using more information often improve the results, and it never significantly degrades

them. At the local level, no notion of object is handled. After using the global approach, different cars can be separated and segmented using geometric constraints.

## V. CONCLUSIONS

We have showed an information fusion based system which is flexible to include many sensors and modules defined over different sets of classes. It is based on the framework proposed in [16] and has been augmented by a global grammar-based reasoning [3]. Fusion at the segments level improves local accuracy and global fusion enables to have object level reasoning.

Future works will enhance object classification by introducing sliding windows based approaches as well as part-based object detections so as to use the full potential of visual grammars. New sources of information such maps will also be considered.

## ACKNOWLEDGMENT

This work is supported and financed by the Cai Yuanpei program from the Chinese Ministry of Education, the French Ministry of Foreign and European Affairs, the French Ministry of Higher Education and Research, and by the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). It is also supported by the Blanc International ANR-NSFC Sino-French PRETIV project and the ICT-ASIA PREDiMap project.

## REFERENCES

- [1] A. Aregui and T. Denœux. Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. *International Journal of Approximate Reasoning*, 49(3):575-594, 2008.
- [2] H. Badino, U. Franke and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *ICCV Workshop on Dynamical Vision*, Brazil, 2007.
- [3] J.-B. Bordes, F. Davoine, P. Xu, and T. Denœux. Evidential grammars for image interpretation. application to multimodal traffic scene understanding. In *Third International Symposium on Integrated Uncertainty in Knowledge Modeling and Decision Making*, pp. 65-78, China, 2013.
- [4] R. Brehar, C. Fortuna, S. Bota, D. Mladenic and S. Nedevschi. Spatio-temporal reasoning for traffic scene understanding. In *Proc. of ICCP*, pp. 377-384, USA, 2011.
- [5] T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):804-813, 1995.
- [6] T. Denœux. A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 30(2):131-150, 2000.
- [7] P. Dollár, C. Wojek, B. Schiele and P. Perona. Pedestrian detection: an evaluation of the state of the art. *PAMI*, 34(2):743-761, 2011.
- [8] A. Ess, T. Müeller, H. Grabner and L. J. Van Gool. Segmentation-Based Urban Traffic Scene Understanding. In *Proc. of BMVC*, pp. 84-91, UK, 2009.
- [9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, pp. 3354-3361, USA, 2012.
- [10] D. M. Gavrila and S. Munder. Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. *IJCV*, 73(1):41-59, 2007.
- [11] D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):1-23, 2007.
- [12] L. Ladick y, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H.S. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *Proc. of BMVC*, pp. 1-11, UK, 2010.



	Ground							Static structures				Moving obstacles			Overall
	Sky	Road	Sidewalk	Lane marking	Vegetation			Building	Pole	Fence	Other	Car/truck	Person/cyclist	Other	
					Grass	Tree	Other								
ALE	<b>96.7</b>	82.5	<b>88.5</b>	93.5	89.9	86.7	78.5	85.1	90.5	88.4	94.7	88.8	86.0	91.1	86.7
ALE + Geo	94.8	82.6	86.4	93.4	89.9	<b>88.1</b>	87.5	<b>85.5</b>	<b>93.4</b>	<b>94.9</b>	98.7	<b>92.9</b>	94.1	<b>100</b>	87.8
ALE + Geo + 3D	<b>96.7</b>	<b>83.0</b>	87.9	<b>94.5</b>	<b>95.4</b>	87.0	<b>88.1</b>	85.1	92.8	94.8	<b>98.8</b>	92.7	<b>94.7</b>	<b>100</b>	<b>88.0</b>

TABLE II

AVERAGE PRECISION (IN PERCENTAGE) OF THE MULTI-CLASS CLASSIFICATION. ALE IS THE MONOCULAR MULTI-CLASS MODULE FROM [12]. GEO REFERS TO THE MONOCULAR GEOMETRIC CONTEXT FROM [11]. 3D IS TO THE GROUND DETECTION MODULE USING STEREO AND LIDAR AS IN [16].

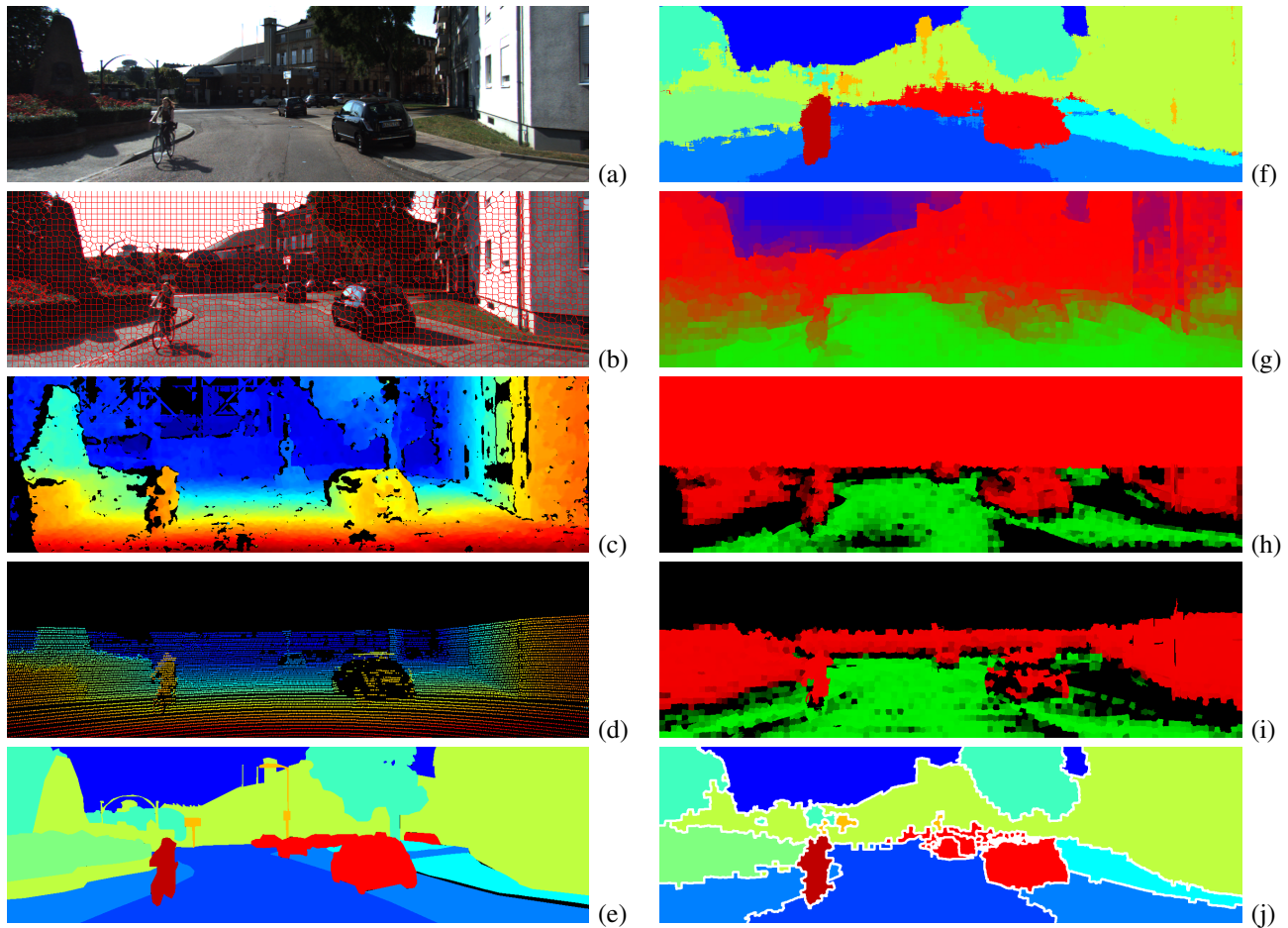


Fig. 3. Input data and results from the different modules. (a) Raw image from the left camera. (b) Over-segmented image. (c) Disparity computed from the stereo camera. (d) Lidar impact points. (e) Ground truth with 14 classes. (f) Output from ALE, the color of each pixel is the class with highest score. (g) Classification probability from the geometric context, the red, green and blue intensities represent the probability of having an obstacle, the ground and the sky respectively. (h-i) Ground/Non-ground classification using 3D information, the green color represents the mass put on the class “ground” and the red the one on “non-ground”, the black color represents the ignorance. The results are from the data (c-d) respectively. (j) Final combined information and segmentation from the evidential grammar.

- [13] D. Mercier, B. Quost, and T. Dencux. Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246-258, 2008.
- [14] S. Thrun, W. Burgard and D. Fox. Probabilistic robotics. *The MIT Press*, Cambridge, Massachusetts, 2005.
- [15] C. Wojek and B. Schiele. A Dynamic Conditional Random Field Model for Joint Labeling of Object and Scene Classes. In *Proc. of ECCV*, pp. 733-747, France, 2008.
- [16] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Dencux. Information fusion on oversegmented images: An application for urban scene understanding. In *Proc. of Intl. Conference on MVA*, pp. 189-193, 2013.
- [17] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259-362, 2006.





## Session III

### Interactive session

- **Title: Hierarchical Traffic Control for Partially Decentralized Coordination of Multi AGV Systems in Industrial Environments**  
Authors: Valerio Digani, Lorenzo Sabattini, Cristian Secchi, Cesare Fantuzzi
- **Title: TLD based Real-Time Weak Traffic Participants Tracking for Intelligent Vehicles**  
Authors: Linji Xue, Ming Yang, Yongkun Dong, Chunxiang Wang, Bing Wang
- **Title: On Keyframe Positioning for Pose Graphs Applied to Visual SLAM**  
Authors: Andru Putra Twinanda, Maxime Meilland, Désiré Sidibé, Andrew I. Comport
- **Title: Online Spatiotemporal-Coherent Semantic Maps for Advanced Robot Navigation**  
Authors: Ioannis Kostavelis, Konstantinos Charalampous, Antonios Gasteratos
- **Title: Use of a Monocular Camera to Analyze a Ground Vehicle's Lateral Movements for Reliable Autonomous City Driving**  
Authors: Young-Woo Seo, Ragnathan Rajkumar
- **Title: Object-Level View Image Retrieval via Bag-of-Bounding-Boxes**  
Authors: Ando Masatoshi, Yuuto Chokushi, Yousuke Inagaki, Shogo Hanada, Kanji Tanaka
- **Title: Cart-O-matic project : autonomous and collaborative multi-robot localization, exploration and mapping**  
Authors: Antoine Bautin, Philippe Lucidarme, Remy Guyonneau, Olivier Simonin, Sebastien Lagrange, Nicolas Delanoue, Francois Charpillet
- **Title: Driving Intention Assistance for Front-wheel-drive Personal Electric Vehicle**  
Authors: Satoshi Fujimoto, Zhencheng Hu, Nobutomo Matsunaga, Claude Aynaud, Roland Chapuis, Han Wang
- **Title: Ad-hoc heterogeneous (MAV-UGV) formations stabilized under a top-view relative localization**  
Authors: Martin Saska, Vojtech Vonasek
- **Title: Toward Smooth and Stable Reactive Mobile Robot Navigation using On-line Control Set-points**  
Authors: Lounis Adouane

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# Hierarchical Traffic Control for Partially Decentralized Coordination of Multi AGV Systems in Industrial Environments

Valerio Digani, Lorenzo Sabattini, Cristian Secchi and Cesare Fantuzzi

**Abstract**—This paper deals with decentralized coordination of Automated Guided Vehicles (AGVs) used for logistics operations in industrial environments. We propose a hierarchical traffic control algorithm, that implements path planning on a two layer architecture. The high-level layer describes the topological relationships among different areas of the environment. In the low-level layer, each area includes a set of fixed routes, along which the AGVs have to move. In the proposed control architecture, each AGV autonomously computes its path, on both layers. The coordination among the AGVs is obtained exploiting shared resources (i.e. centralized information) and local negotiation (i.e. decentralized coordination). The proposed strategy is validated by means of simulations. This work is developed within the PAN-Robots European project.

## I. INTRODUCTION

This paper deals with the path planning and coordination of multiple Automated Guided Vehicles (AGVs) in an automated warehouse.

The standard approach to coordinate a fleet of AGVs lies in a centralized supervisor (the control center) which manages all the information coming from the Warehouse Management System (WMS) and from the environment. The control center handles the coordination of the fleet, solving a multi-robot path planning problem. Several works can be found in the literature that face this kind of problem. Generally speaking, multi-robot path planning can be solved exploiting centralized or decentralized strategies.

With *centralized* strategies, a single decision maker determines the entire path plan for all the robots. These approaches can theoretically find optimal solutions for multi-robot path planning problems [1], but they are restrictive in the number of robots for which they can plan, as the complexity of planning grows exponentially with the number of robots. Thus, while they provide the highest-quality solutions overall, they are generally intractable for large teams. Several centralized strategies can be found in the literature. For instance [1], [2] solve the problem of coordinating a multi-robot system using a coordination space representation of the robot motions. The basic idea is to reduce the size of the problem (exponential with the number of robots involved) exploiting a path decomposition method, which decomposes it into its elementary pieces consisting of either straight line segments or arcs of a circle.

Authors are with the Department of Science and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {valerio.digani, lorenzo.sabattini, cristian.secchi, cesare.fantuzzi}@unimore.it

This paper is written within PAN-Robots project. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 314193.

Another method to reduce the search space is to weakly constrain the allowable paths that robots can follow by limiting the motion of the robots to lie on route maps in the environment. Intuitively, route maps are akin to automotive highways, where robots move from their starting position to a route map, move along the route map to the proximity of the goal, and then move off the route map to the specific goal location. Several strategies can be found in the literature for the coordination of multi-robot systems on a route map, based on different optimality principles [3]–[7]. In order to solve local conflicts, traffic rules may be defined [8]. If robots are allowed to locally exchange information, several strategies can be found in the literature that consider the segments of the route map as resources to be allocated [9], and solve the allocation problem by means of negotiation [10].

The dimension of the multi-robot space may be reduced using of a multi-layer structure to represent the world. As explained in [11], [12], the approach is to construct a hierarchical route map which can abstract the traversable areas using the adequate number of nodes and edges of a graph. The path is searched using the graphs of the several layers.

Conversely, completely *decentralized* approaches are very attractive. In these approaches, each robot autonomously determines its routes, dissolving the conflicts and collecting information from other robots. Decentralized techniques are generally faster than centralized ones, but they present several drawbacks, such as failing in finding valid paths for all robots due to deadlocks [13], [14].

In this paper we present a partially decentralized control strategy for the coordination of multi AGV systems. Specifically, our idea is based on a hierarchical control architecture [13]. In detail, two layers are used in order to reduce the total complexity and to simplify the control. The first layer is a topological graph of the plant. The global map of the plant is divided into several macro-areas, called sectors. Each sector corresponds to a node of the graph. Its main purpose is to permit a dynamic re-planning of the paths in case of dynamic events. The second layer is the real route map on which the AGVs move. The coordination on the route map is limited only to a single sector of the first layer. In other words, in each sector, the traffic is managed in a decentralized manner on a local route map. Preliminary results on these topics were introduced in [15].

## II. PROBLEM STATEMENT

In this paper we will present a strategy for path planning of multiple AGVs over a route map.

We first introduce the definition of a *path* on a route map.

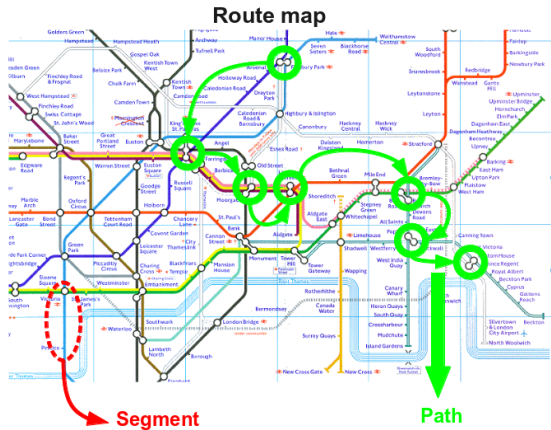


Fig. 1: Elements of a route map

**Definition 1 Paths** Given a route map, a path is a set of consecutive segments.

A path can be assigned to an AGV, that is then allowed to move along the segments in the path.

We introduce the following definition of *admissible set of paths*.

**Definition 2 Admissible set of paths** Given a fleet of  $n$  AGVs moving on a route map, an admissible set of paths is a set of  $n$  paths defined in such a way that, assigning a path to each AGV, it is possible to define a velocity profile for each AGV such that collisions are avoided.

The problem can be formally stated as follows:

**Problem 1 Multi AGV path planning** Consider:

- a fleet of  $n$  AGVs
- a route map
- the initial and final positions for all the AGVs

Define an admissible set of paths such that each AGV is able to move from its initial position to its assigned final position.

Therefore, the problem consists in planning a path for each AGV so that conflicts and deadlocks are avoided. Each AGV starts its path in a initial position, and has to reach its own pick/drop position.

The following **Assumptions** are made on the system

- A1 The environment is represented with a 2D static layout, in which free areas and occupied ones are depicted.
- A2 Each AGV has a prior knowledge of the geometry of the environment, and of the route map.

A3 Each AGV can communicate with the others in its neighborhood.

A4 Each AGV has access to shared data stored in a centralized layer.

A5 The maximum velocity and acceleration are the same for all the AGVs. In other words, the fleet is composed of homogeneous AGVs.

A6 Each AGV is modeled as a kinematic agents, whose linear and angular velocities can be controlled.

A7 No unforeseen events, such as the presence dynamic obstacles (manual forklift, people, etc.) are considered.

It is worth noting that removing Assumption 7 would lead to the solution of the dynamic version of the *multi AGV path planning* problem, in which the set of path initially defined has to be modified in case of unforeseen events. In other words, a path re-planning or a dynamic planning is needed.

The task assignment (that is, the goal given to each AGV) is out of the scope of this paper, and will therefore not be considered.

## III. TWO LAYER CONTROL ARCHITECTURE

In this section, the main idea of the paper is explained. The problem of coordinating a elevated number of AGVs is faced splitting the control through a multi-layer architecture. In our idea two layers are used. The top-layer, or *Topological Layer*, is a topological map representing the global map, with different macro-cells called sectors. The layer below, or second layer, is the geometric map of each sector of the first layer, and will be hereafter referred to as *Route Map Layer*.

Therefore the path planning is done on two levels. Topology path planning searches for the best path to the final goal (actually to the final sector where the real goal is) from the current sector. Route map planning computes the path on the route map and makes the coordination inside the sector.

### A. Topological layer

The first layer is the most abstract layer, it is generated by a subdivision of the geometric layout in several sectors.

1) *Sector division*: A **sector** is an area, or a region, which can be distinguished from the other ones based on topological aspects, material flow, logistical aspects and geometrical ones. The layer gives a topological representation of the real map.

A *sector* is an abstract entity, that owns the following properties:

- Geometric space
- Topological information
- Constraints

The constraints are defined based on the characteristics of the operational environment. For instance, constraints can be defined in terms of maximum number of AGVs contained in a single sector, or in terms of maximum number of operations of loading/unloading. This kind of information is owned by the sectors and is stored in a *centralized manner*. In this way, the information is visible to all the AGVs and is shared among them from the centralized storage.

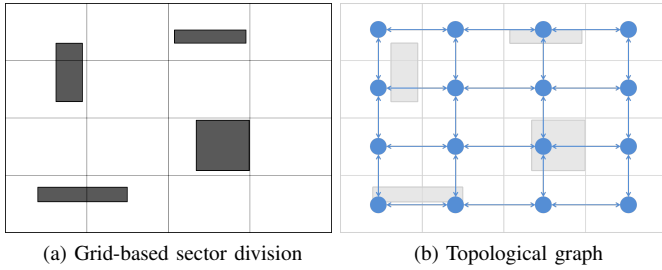


Fig. 2: The topological sector division: in 2a the geometric map is divided into regular sectors; in 2b the graph representation generated by the sector division is shown

2) *Path planning on the topological layer*: The information owned by the sectors are used to plan the sub-optimal route for an AGV. Each vehicle has to reach its destination minimizing several cost functions, such as the crossing time, average velocity, travel distance, etc.

The map of sectors (i.e. the first layer) can be represented by means of a graph: each sector is a node of a directed graph, and the links among neighboring sectors are the edges of the graph.

The path from the start sector to the goal one is searched by means of the D\* algorithm [16]. D\* algorithm is an incremental search algorithm which solves the path planning problems where a robot has to navigate to given goal coordinates in unknown terrain. It makes assumptions about the unknown part of the terrain and finds a shortest path from its current coordinates to the goal coordinates under these assumptions. During the path following, the new information (such as previously unknown obstacles) is added to the map, and, if necessary, the algorithm re-plans a new shortest path from its current coordinates to the given goal coordinates. The choice of this search algorithm is due to the need of re-planning the path in a dynamic way on the topological layer.

Furthermore a MPC (model predictive control [17], [18]) mechanism has been added. That is, at each step, the AGV checks if the previously assigned path is still admissible. Actually the MPC approach lies in the fact that only a portion of the path is checked and the horizon can be extended or reduced. This approach provides an optimal local solution but a sub-optimal global one, because only the part of the path inside the horizon is interested by the optimization.

Therefore each AGV computes autonomously its own path through the grid of sectors without paying attention to the other AGVs' planned routes. The relationship among them is provided only by the shared data about the state of the sectors.

The procedure described so far is summarized in Algorithm 1.

### B. Route Map layer

The route map layer contains the geometric information of the environment and the route map itself. The route map is a set of routes as a highway, and it is composed

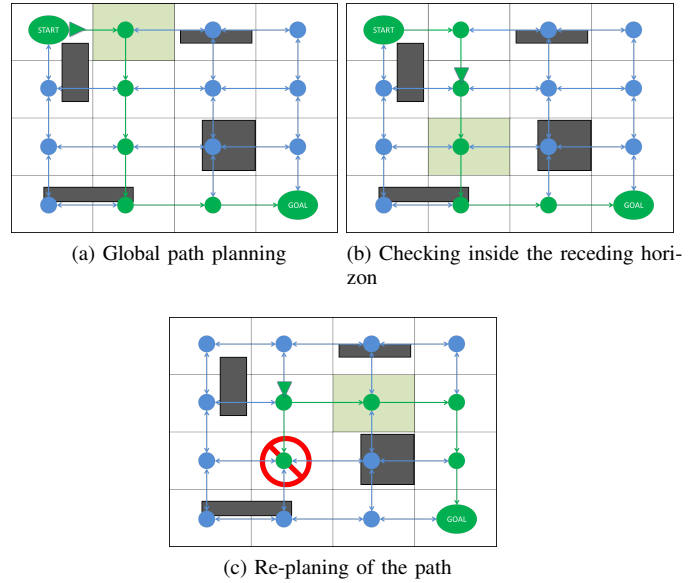


Fig. 3: The path planning on the topological layer: in 3a the path is searched by means the D\* algorithm; in 3b the AGV moves along its path and checks the next sectors; in 3c a re-planning of the path is needed due to the new condition of the next sector

---

#### Algorithm 1: Path planning on the topological layer

---

```

1 Each AGV computes its path from the current sector to
  the sector of destination;
2 for each step do
3   if next sector still available then
4     | go to next sector;
5   else
6     | Re-compute the path, avoiding that sector;
7   end
8 end

```

---

by distinguished elements called segments. The AGVs are constrained to follow the route map and its segments.

Inside each sector the coordination among AGVs is needed. The second layer manages the real path following of the route map and the avoidance of deadlocks and conflicts among AGVs or among AGVs and obstacles. The coordination is managed locally (in each sector) in a decentralized manner. With this hierarchical architecture it is possible to simplify the whole control in order to focus the coordination of the AGVs only inside each sector in a local way.

Two main contributions appear in this layer. The first deals with the fact that the route map is built according to specific constraints, and the second is the real coordination algorithm.

1) *Route map properties*: We will hereafter assume that the route map is designed according to the following **Properties**:

P1 The route map is a directed graph: each edge is unidirectional in order to avoid the situation in which two



or more AGVs are on the same road but with opposite directions

- P2 There are at least two exits, two entries and one intersection in a sector
- P3 AGVs on different segments can't collide. The minimum distance between two segments has to be sufficient to ensure the passage of the AGVs without collision. If this condition is not possible, then the two segments will be intersected with an intersection
- P4 An *intersection* (node shared by several segments) is a resource to be allocated to a single AGV and it can be free or occupied
- P5 The intersection is defined with a *cross point* and some *attention points* (see Fig. 4a)
- Cross points: it is the real intersection due to the collision of two or more segments
  - Attention points: points linked with the cross point. They are the extremes of the colliding segments

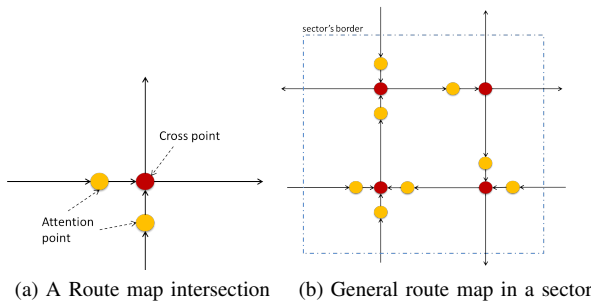


Fig. 4: Route map properties

2) *Coordination on the route map layer*: The objective for each AGV is to compute a path to reach the next assigned sector. The path planning inside a sector consists in assigning a set of segments to each AGV. The algorithm used to find the path is the simple A\*. The choice is due to the fact that the route map is fixed, and local dynamic changes are not considered.

When a path has been assigned to every AGV, a coordination to avoid conflicts in the intersection among the paths is needed. The coordination is fulfilled by means of a hybrid approach, exploiting a resource allocation and negotiation mechanisms in order to obtain the advantages of both. It is managed *locally*, because it takes place exclusively inside the sector: the AGVs share information among each other, without the participation of a centralized supervisor. The data exchange among AGVs concerns:

- *AGV priority*: each AGV can have different tasks with different levels of priority.
- *Intersection request*: an AGV which is approaching an intersection has to communicate this intention to the others-
- *Intersection allocation*: an AGV allowed to go through an intersection has to communicate this to the others.

The coordination is defined as a combination of *negotiation* and *resource allocation*: the resource (intersection) is

allocated only to a single AGV in order to avoid conflicts, and the negotiation permits to avoid deadlocks.

The coordination procedure is described in details in Algorithm 2.

---

**Algorithm 2:** Coordination on the route map layer

---

```

1 if AGV approaching intersection then
2   request intersection;
3   if other AGVs requested intersection then
4     share priority;
5     perform negotiation;
6     establish the winner;
7   end
8 end
9 ;
10 move to the attention point;
11 if AGV is the winner and intersection is free then
12   go through the intersection;
13   leave the intersection;
14   withdraw the previous request;
15 else
16   stop;
17   go to line 2;
18 end

```

---

To sum up, the global flowchart diagram is shown in figure 5.

#### IV. SIMULATIONS

In order to evaluate the idea, the algorithm is implemented using Matlab. A portion of a real plant is used to simulate the environment.

Fig. 6 shows the scenario: the black rectangles are the real block storages and the free space is shown in white. Based on that plant, a simple geometric route map is built. In order to simulate the current behavior of the AGVs, a segment division of the route map is fulfilled. In this way, no more than one AGV is allowed to stay on a segment, at a given time.

The tests are conducted under the same conditions, in particular, the scenario consists in:

- map of a real plant
- 25 sectors
- maximum limit of 4 AGVs allowed in each sector
- one AGV allowed per segment
- start positions of the AGVs are different
- a queue of tasks is generated randomly
- the simulation stops when all the AGVs reach their goals and the queue of tasks is empty
- the priority is generated randomly for each AGV
- 20 AGVs

In order to simulate a fleet of decentralized AGVs, the algorithm is executed in a parallel manner by implementing one single dedicated process per AGV. Fig. 7 shows the sequence of events and actions of several AGVs. In particular two AGVs are approaching an intersection and, based on



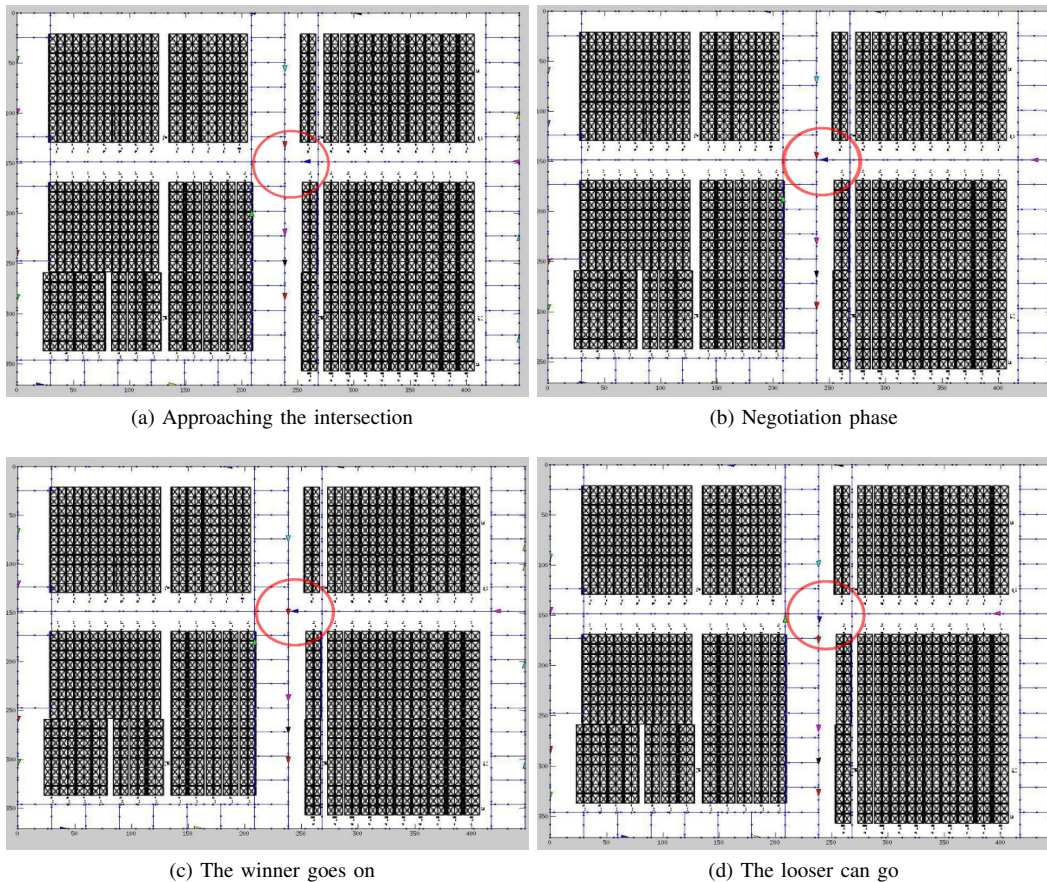


Fig. 7: The screen-shots show the coordination at intersection. In these pictures, the red lines are the segment's borders and the dots are the nodes of the route map

the priority, one AGV goes on and the other one has to stop temporarily.

Although the development of the idea is in an early phase, a statistical analysis is made in order to figure out the computational needs of the algorithm. Several tests are executed changing the number of AGVs, in particular the tests concern 5, 10, 15 and 20 AGVs. In all of them the elapsed time is monitored. The results (see Fig. 8) show a linear behavior of the elapsed time in function of the number of AGVs. The higher is the number of AGVs, the higher is the time for the computation. It is worth noting that with the increase of the number of AGVs, also the variance of results increases. This is due to the high number of negotiations which, depending on the random priority of the AGVs, can provide different results on tests performed in similar conditions.

## V. CONCLUSION

The paper describes a coordination strategy for a fleet of AGVs, through an architecture based on a two-layer approach. The presented idea tries to treat the planning and the path optimization as a common entity. The coordination and the traffic management are treated as global functions. In order to achieve this, a hybrid path planning and coordination

is achieved. The path planning is split on the two layers in order to simplify the problem. The path planning executed by each AGV is totally decentralized, but the information about the occupation of the sectors is managed in a centralized way. The local coordination is also totally decentralized. In this case, the AGVs share the information among them in order to negotiate the access to the shared resources (i.e. the intersections).

The simulations have shown that it is easily possible to manage a high number of AGVs with this approach avoiding conflicts among them. The studied scenario is actually a strong simplification of a real plant. Therefore in the next steps, it will be necessary to validate the algorithm on a realistic scenario using the route map of a real plant both through virtual simulation and implementation in a real industrial environment. An experimental comparison with existing methodologies will also be performed. Current efforts also aims at mathematically analyzing the complexity of the proposed algorithm.

Moreover, future work will aim at implementing an automatic procedure for the definition of the route map itself, possibly considering bi-directional segments and more complex intersection structures.

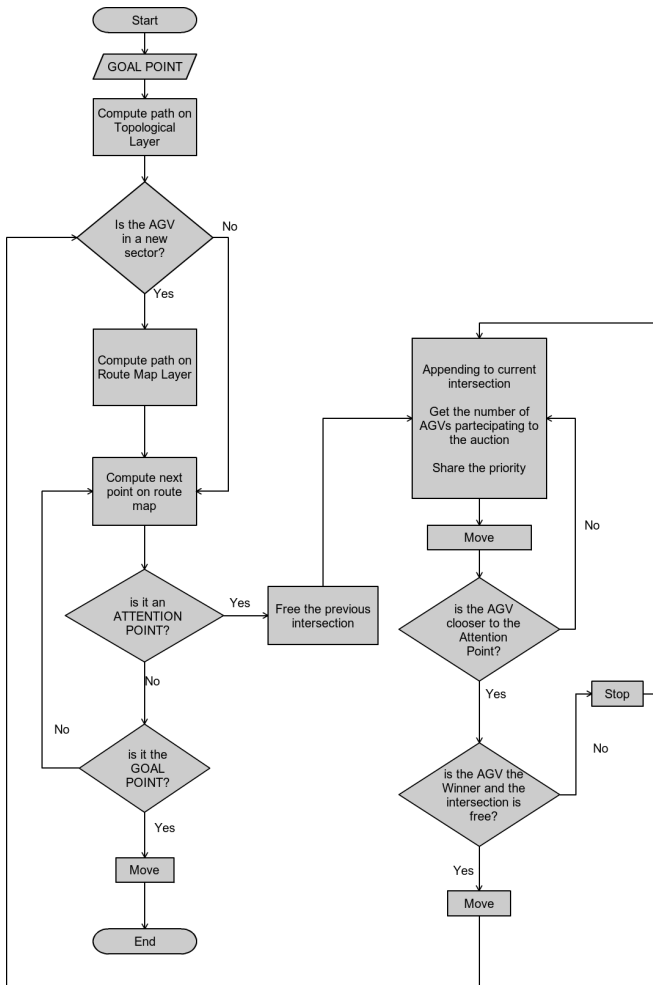


Fig. 5: Flowchart diagram of the path planning and coordination procedures.

## REFERENCES

- [1] S. LaValle and S. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.
- [2] T. Simeon, S. Leroy, and J.-P. Laumond, "Path coordination for multiple mobile robots: a resolution-complete algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.
- [3] I. F. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, May 2006.
- [4] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2557–2562, Oct. 2012.
- [5] N. Hui, "Coordinated motion planning of multiple mobile robots using potential field method," in *2010 International Conference on Industrial Electronics, Control Robotics (IECR)*, 2010, pp. 6–11.
- [6] R. Olmi, C. Secchi, and C. Fantuzzi, "Coordination of industrial AGVs," *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 1, pp. 5–25, 2011.
- [7] —, "Coordination of multiple agvs in an industrial application," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.*, 2008, pp. 1916–1921.
- [8] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, Dec. 2007.
- [9] S. Reveliotis and E. Roszkowska, "Conflict resolution in free-ranging

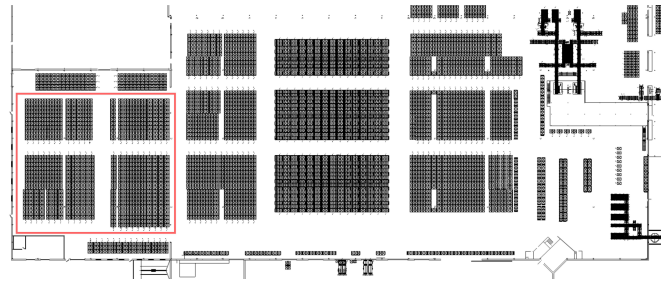


Fig. 6: Geometric layout of a real warehouse and the portion used in the simulation.

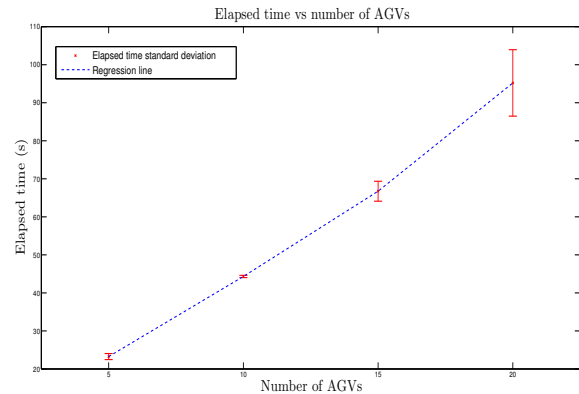


Fig. 8: Elapsed time versus number of AGVs

- multivehicle systems: A resource allocation paradigm," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 283–296, 2011.
- [10] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
  - [11] B. Park, J. Choi, and W. K. Chung, "An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment," *2012 IEEE International Conference on Robotics and Automation*, pp. 180–186, May 2012.
  - [12] T. W. Min, L. Zhe, H. K. Yin, G. C. Hiang, and L. K. Yong, "A rules and communication based multiple robots transportation system," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1999. CIRA '99.*, 1999, pp. 180–186.
  - [13] W. Zhang, M. Kamgarpour, D. Sun, and C. Tomlin, "A hierarchical flight planning framework for air traffic management," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 179–194, 2012.
  - [14] M. Jager and B. Nebel, "Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.*, vol. 3, 2001, pp. 1213–1219 vol.3.
  - [15] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Towards decentralized coordination of multi robot systems in industrial environments: a hierarchical traffic control strategy," in *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2013.
  - [16] A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, no. May, pp. 3310–3317, 1994.
  - [17] J. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.
  - [18] C. E. Garca, D. M. Prett, and M. Morari, "Model predictive control: Theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335 – 348, 1989.

# TLD based Real-Time Weak Traffic Participants Tracking for Intelligent Vehicles

Linji Xue<sup>1</sup>, Ming Yang<sup>1</sup>, Yongkun Dong<sup>2</sup>, Chunxiang Wang<sup>2</sup>, Bing Wang<sup>1</sup>

1. Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, [mingyang@sjtu.edu.cn](mailto:mingyang@sjtu.edu.cn)
2. Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai 200240

**Abstract**—Pedestrians and bicycles are the most vulnerable participants in urban traffic. Numerous Pedestrian detection methods have been proposed in the last ten years. However, most of them cannot meet the real-time requirement of intelligent vehicles. At the same time, there is little paper on bicycle detection or tracking. This paper proposes a real-time pedestrian and bicycle tracking method based on TLD (Tracking-Learning-Detection), which is an award-winning, real-time algorithm for tracking of unknown objects. In order to solve the background movement arising from the moving observation platform, the location of feature points of the tracking part of TLD is adjusted according to the characteristic of moving pedestrian and bicycle. Then gradient feature is used instead of gray feature in original TLD algorithm, in order to solve the pedestrians and bicycles' deformation problem. Experimental results demonstrated that the effectiveness and real-time performance of the proposed method.

**Key words:** pedestrian tracking, bicycle tracking, gradient feature, TLD, intelligent vehicles

## I. INTRODUCTION

Over the past decade, a lot of pedestrian detection algorithms have been proposed. For example, the typical pedestrian detection algorithm HOG (Histogram Of Gradient, HOG) + SVM (Support Vector Machine)<sup>[1]</sup> was proposed by the French Dalal, as well as the Discriminatively trained deformable part models method<sup>[2]</sup> which was proposed by P. Felzenszwalb. This algorithm's most significant characteristic is detecting the whole and the part of the target separately. These pedestrian detection algorithms are able to get great effectiveness on detecting pedestrians on a single frame. However, if these algorithms are used in tracking pedestrians in a video stream, they won't meet the real-time requirement.

For tracking pedestrian algorithm, R. Benenson proposed a stereo vision based pedestrian tracking algorithm, called Stixel computation<sup>[3]</sup>. This algorithm's processing speed in pedestrian tracking can reach 100 frames per second. In other words, it's able to meet the real-time requirement in tracking pedestrian fully. However, this algorithm has a limitation, the stereo camera is necessary in this tracking algorithm as a hardware support. So it increases the cost of the pedestrian tracking.

Meanwhile, the existing pedestrian tracking algorithm is basically used in the fixed camera, such as the surveillance cameras which is installed on crossroads or parking. In other words, only pedestrians are moving within the field of vision. The background is stationary. For pedestrian tracking in intelligent vehicles, the camera is located in the intelligent vehicles. The background movement arising from the moving observation platform will affect the pedestrian tracking.

The tracking algorithm which is suitable for static background can not simply be transplanted to the dynamic background environment. With the continuous development of the pedestrian detection and tracking algorithm, there is a little detection and tracking algorithm for bicycles. This paper proposes a tracking algorithm for weak traffic participants based TLD algorithm, which is used in the dynamic background occasions pedestrians and bicycles tracking.

TLD<sup>[4]</sup> is a tracking algorithm for long-term tracking of unknown object, which is proposed by Zdenek Kalal who is a Czech doctoral student at University of Surrey Guildford, UK. It explicitly decomposes the long-term tracking task into tracking, learning and detection. The tracker follows the object from frame to frame. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates detector's errors and updates it to avoid these errors in the future.<sup>[4]</sup> It's an award-winning, real-time tracking algorithm.

After experiment, the success rate in pedestrian tracking is low and the bounding box has serious deviation in the tracking process. And the reason is the complex deformation of moving pedestrian and the background movement arising from the moving observation platform. Based on the above two points, this paper makes a series of improvements to TLD algorithm.

## II. OBJECT DETECTION

### A. Pedestrian Detection

Every time users want to track pedestrian using TLD algorithm, they should draw a bounding box to localize the location of the tracking target before. Of course this is a waste of time step. And if users waste too much time in this step, the tracking effectiveness will be influenced. So this paper uses the default HOG + SVM<sup>[1]</sup> pedestrian detector of OpenCV to initialize the pedestrian tracking of TLD algorithm. It can



bring convenience to users and ensure the effectiveness of tracking.

### B. The Choice of Pedestrian Detecting Features

The detector scans the input image by a scanning-window and for each patch decides about presence or absence of the object.<sup>[4]</sup>

TLD algorithm's detector uses a cascade classifier which is composed of three parts, and they are the Variance classifier, Ensemble classifier and Nearest Neighbor classifier.

The detector of TLD algorithm uses gray feature to distinguish foreground and background. But when people are walking, its gray feature will change a lot because the complex deformation of pedestrian. And the useless background information in the bounding box will have a large difference in gray feature according to the environment. As shown in Figure 1 and Figure 2.



Fig. 1. The Same Pedestrian in Different Environment and Pose



Fig. 2. The Difference of Pedestrian's Gray Feature

Therefore, gray feature is not suitable for tracking pedestrian. In order to solve this problem, this paper substitutes the original gray feature of TLD for the gradient feature. Most of all pedestrian detection algorithms use gradient feature. When people are moving, the changes of their gradient feature are slighter than gray feature. And the gradient feature is mainly from the boundary of foreground and background. In other word, the useless background information in the bounding box will have a smaller impact of gradient feature. As shown in Figure 3.

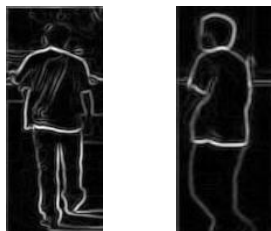


Fig. 3. The Difference of Pedestrian's Gradient Feature

Because of these, this paper create a new classifier based on HOG feature. Then this paper uses this HOG classifier to replace the original NNClassifier (Nearest Neighbor Classifier) of TLD algorithm's detection cascade.

This paper removes the NNClassifier from the detection cascade of TLD algorithm, but the Variation classifier and the Ensemble classifier are retained. The gradient features of different pedestrians are different in some degree, but they are largely the same. That is to say the gradient feature is hard to distinguish between different people. So this paper retains two of the classifiers based on gray feature, to distinguish between different people and reduce the burden of HOG classifier.

### C. Screen Scan Windows According to the Limitation of Target Motion

The movement of the target which the algorithm tracks must be limited between frame and frame. Pedestrian, bicycle and most of all objects' movements are limited in limited time. According to this, this paper adds a new part in the detector of TLD algorithm, to screen the scan windows before they enter into the detection cascade.

If the current bounding box's location is confirmed, and it can represent the position and scale of the target, the position of the target in next frame must be very close to the current bounding box. And the scale of the bounding box won't change too much. So if the confidence coefficient of current bounding box is high enough, only the scan windows which are very close to it and the scale of them is similar to it will enter into the detection cascade.

After experiment, it proves that the screen scan windows part can surely enhance the real-time performance of TLD algorithm. But it will bring some false detection problem.

### D. The Method of Eliminating Positive Examples and Negative Examples

The number of positive examples and negative examples will also affect the processing speed of TLD algorithm.

This paper has finished some experiments to show the influence of the number of examples.

The experimental result is shown in Figure 4.

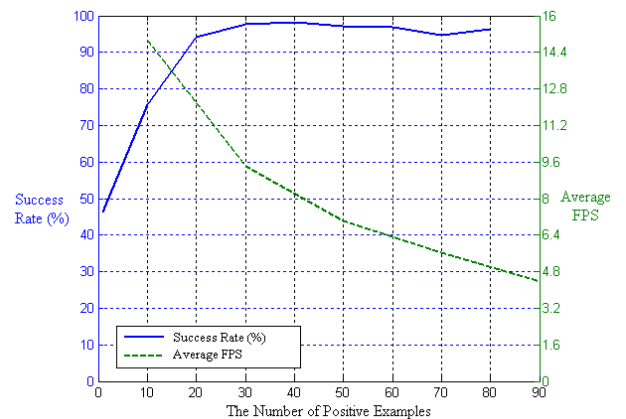


Fig. 4. Tracking Success Rate and Average FPS on a Function of the Number of Positive Examples

The experimental results show that the processing speed of TLD algorithm will decrease with the growing of the number of positive examples. In addition, the effectiveness of tracking won't increase when the number of positive examples reaches a constant value. Because of this, it's necessary to limit the

number of positive examples and the negative examples. So this paper proposes a method to eliminate the positive examples and negative examples to control the number of them.

For positive examples, because every positive example may be useful in the future tracking, this paper uses the easiest way to eliminate positive examples. When the number of positive examples reaches a constant value, they will be eliminated randomly. In addition, the first positive example is the most believable positive example, so it won't be eliminated until the tracking is over.

For negative examples, because the negative examples are collected from the background, and the background is moving, the early negative examples won't be used in future tracking. This paper uses a queue to store the negative examples. When the number of negative examples reaches a constant value, the negative example on the top of the queue will be eliminated. By this method, the number of negative examples can be controlled, and the useless negative examples can be eliminated.

The method of eliminate positive examples and negative examples will enhance the real-time performance of TLD algorithm, and save the memory space.

### III. OBJECT TRACKING

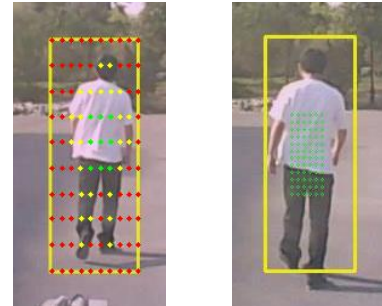
#### A. Adjustment of the Pedestrian Tracking Feature Points' Location

The tracking method in TLD algorithm is called the median flow tracking method. Median flow tracking method is mainly used Lucas-Kanade optical flow tracking<sup>[5][6]</sup>, and the Forward-Backward Error<sup>[7]</sup> plus NCC (Normalized Cross-Correlation) approach which is proposed by the original author of TLD. For the Lucas-Kanade optical flow tracking results, the Forward-Backward Error plus NCC approach as feedback, to screen the significant feature points of tracking. Finally, on this basis, the original author of TLD proposes a tracking failure detection algorithm.<sup>[4]</sup>

Before the tracking is start, the tracker of original TLD algorithm will put the tracking feature points evenly in the bounding box. But there is not only foreground information in the bounding box. If the tracking feature points are put evenly, they will inevitably be put in the background. And because of the background movement arising from the movement of observation platform, the tracking feature points in the background will have a bad influence in tracking. In addition, the tracking feature points will be put in the arms and legs of people. These are the most deformable parts of pedestrian. The feature points on them are also bad to tracking. As shown in Figure 5.

In Figure 5 (a), the points which are put on the background are red. The points which are put on the most deformable parts of pedestrian are yellow. But the green points mean that they are put on the location where the deformation is small. Only 9 points in 100 points are green. The rest of 91 red and yellow points will have a serious influence in tracking.

So, this paper adjusts the location of feature points of the tracker of TLD algorithm according to the characteristic of moving pedestrian. In other word, this paper put the feature points in the back and hips of pedestrian, which the deformation is slight. As shown in Figure 5 (b).



(a) Before Adjustment (b) After Adjustment  
Fig. 5. Adjustment of the Pedestrian Tracking Feature Points' Location

After the modification, the tracker of TLD algorithm can eliminate the influence cause by the movement of background and the complex deformation of walking people.

#### B. Adjustment of the Bicycle Tracking Feature Points' Location

After pedestrian tracking, this paper wants to use this new TLD algorithm to track bicycles. This paper adjusts the location of feature points of the tracker of TLD algorithm according to the characteristic of bicycle. The feature points of tracking bicycle are similar to the pedestrian. They are all on the rider's back and hip. As shown in Figure 6.



Fig.6. the Location of the Feature Points in Bicycle Tracking

Then, this paper adjusts some parameters in TLD algorithm according to the characteristic of bicycles.

### IV. EXPERIMENT RESULTS AND ANALYSIS

The experimental platform of all this paper's experiments is PC and the system is Microsoft Windows XP, the processor is Pentium (R) Dual-Core CPU T4200 2.00GHz. At last, the memory is 3.00GB.

For comparison purposes, all this paper's experiments will use the TLD algorithm which is modified to pure C++ version by Alan Torres.

In order to describe the tracking results better, this paper has the following definitions:

- Valid Frames: The number of frames which contain the target in a video sequences.

- Tracking Success Frames: The number of frames which the bounding boxes in them contain the target.
- False Detection Frames: The number of frames which the bounding boxes in them don't contain the target.
- Success Rate: Tracking Success Frames accounted for the percentage of Valid Frames.

#### A. Pedestrian Tracking Experiment by Original TLD Algorithm

Use the TLD algorithm which is not adding any modifications to complete the pedestrian tracking experiment.

The test videos are test data 1 and test data 2. They are shot by vehicle-mounted camera. The pedestrian in test data 1 wears white shirt and black pants. And the pedestrian in test data 2 wears black jacket and black pants.

The experimental result is shown in Table I.

TABLE I. THE PEDESTRIAN TRACKING RESULTS OF ORIGINAL TLD ALGORITHM

Experimental Data	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)	Average FPS
test data 1	832	585	151	70.31	6.821
test data 2	744	642	9	86.29	6.826

Experimental parameters of test data 1: the start frame: 46, initial bounding box:  $x = 347$   $y = 181$  width = 84 height = 217.

Experimental parameters of test data 2: the start frame: 40, initial bounding box:  $x = 324$   $y = 197$  width = 79 height = 198.

As can be seen from the table I, using the original TLD algorithm to track pedestrians in two video sequences, the tracking success rate is not high. Not only that, the bounding box has serious deviation while the process of tracking. The deviation can be shown in Figure 7 and Figure 8.

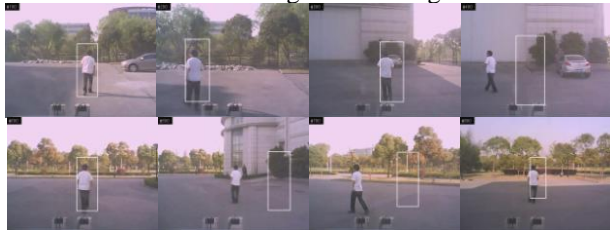


Fig. 7. The Original TLD Algorithm Experimental Results of Test Data 1.

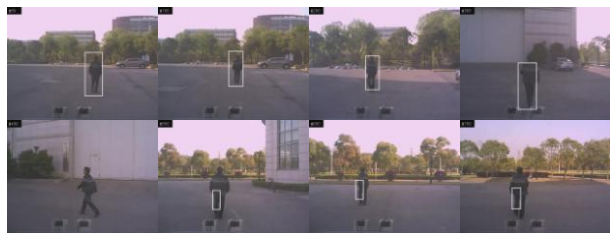


Fig. 8. The Original TLD Algorithm Experimental Results of Test Data 2.

From the experimental results, the reason why the tracking success rate is low and the bounding box has serious deviation in the tracking process is the complex deformation of moving pedestrian and the background movement arising from the moving observation platform. Based on the above two points, this paper makes a series of improvements to TLD algorithm.

#### B. Screen Scan Windows Experiment

This paper has finished some experiments to show the screen scan windows part's influence in algorithm's real-time performance and false detection. The two kinds of algorithms in the experiment are completely the same except the screen scan windows part.

The experimental result is shown in Table II, Table III and Figure 9.

TABLE II. THE SCREEN SCAN WINDOWS PART'S INFLUENCE IN FALSE DETECTION

	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)
Original Algorithm	2233	2182	84	97.72
Using Screen Scan Windows	2233	2130	117	95.39

Experimental parameters: test video: test data 5, starting frame: 31, initial bounding box:  $x = 306$   $y = 267$  width = 70 height = 63.

TABLE III. THE SCREEN SCAN WINDOWS PART'S INFLUENCE IN REAL-TIME PERFORMANCE

	The Average Number of Scan Windows Enter into the Detector	Average FPS	Average Processing Time (ms)
Original Algorithm	294975	4.9911	203.177
Using Screen Scan Windows	166114.7	10.41	103.687

Experimental parameters: test video: test data 5, starting frame: 141, initial bounding box:  $x = 294$   $y = 272$  width = 81 height = 60, the total number of frames involved in the calculation: 1000, the total number of scan windows: 294975.

And the data in Table II and III can be shown in Figure 9.

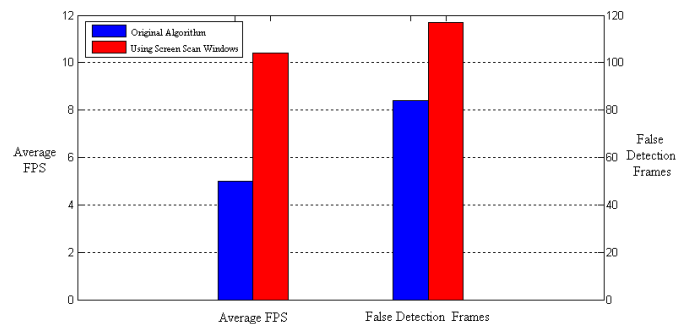


Fig. 9. The Screen Scan Windows Part's Influence in Real-Time Performance and False Detection Frames

The experimental results show that the screen scan windows part can surely enhance the real-time performance of TLD algorithm. And the experimental results also show the screen scan windows part will bring some false detection problem. Because if the false detection happens, the screen scan windows part will hinder the error correction. But because of the movement of background, the false detection won't last too long. So the false detection problem is within the acceptable range.



### C. Pedestrian Tracking Experiment by Modified TLD Algorithm

After the modification above, this paper finishes a lot of experiments to prove that this paper's TLD algorithm has better effectiveness of tracking and has higher processing speed than original TLD algorithm in pedestrian tracking.

The test videos are not only test data 1 and test data 2, but also the test data 3 from the ETH pedestrian datasets' LINTHESCHER video sequence

The experimental result is shown in Table IV, V, VI, and Figure 7, 8, 10, 11, 12, 13, 14 and 15.

TABLE IV. THE EXPERIMENTAL RESULTS OF TEST DATA 1

	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)	Average FPS
Original TLD Algorithm	832	585	151	70.31	6.821
This Paper's TLD Algorithm	832	806	0	96.88	10.69

Experimental parameters of test data 1: the start frame: 46, initial bounding box:  $x = 347$   $y = 181$   $width = 84$   $height = 217$ .



Fig. 10. The Modified TLD Algorithm Experimental Results of Test Data 1.

TABLE V. THE EXPERIMENTAL RESULTS OF TEST DATA 2

	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)	Average FPS
Original TLD Algorithm	744	642	9	86.29	6.827
This Paper's TLD Algorithm	744	684	0	91.94	10.30

Experimental parameters of test data 2: the start frame: 40, initial bounding box:  $x = 324$   $y = 197$   $width = 79$   $height = 198$ .



Fig. 11. The Modified TLD Algorithm Experimental Results of Test Data 2.

TABLE VI. THE EXPERIMENTAL RESULTS OF TEST DATA 3

	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)	Average FPS
Original TLD Algorithm	578	103	8	17.82	2.922
This Paper's TLD Algorithm	578	571	0	98.79	11.67

Experimental parameters of test data 3: the start frame: 601, initial bounding box:  $x = 406$   $y = 104$   $width = 108$   $height = 285$ .



Fig. 12. The Original TLD Algorithm Experimental Results of Test Data 3.



Fig. 13. The Modified TLD Algorithm Experimental Results of Test Data 3.

Experimental results above show that this paper's modified TLD algorithm greatly improves the tracking success rate in pedestrian tracking. In addition, it enhances the algorithm's real-time performance, and eliminates false detection problems, but also solves the bounding box deviation problem which is shown in Figure 7 and Figure 8.

### D. Bicycle Tracking Experiment by Modified TLD Algorithm

After the adjustment of tracking feature points' location and some parameters, this paper finishes some experiments to prove that this paper's TLD algorithm has better performance in bicycle tracking.

The test video is test data 4. It's shot by vehicle-mounted camera. The experimental result is shown in Table VII, Figure 14 and 15.

TABLE VII. THE EXPERIMENTAL RESULTS OF TEST DATA 4

	Valid Frames	Tracking Success Frames	False Detection Frames	Success Rate (%)	Average FPS
Original TLD Algorithm	1557	1490	54	95.70	6.722
This Paper's TLD Algorithm	1557	1548	0	99.42	7.469

Experimental parameters of test data 4: the start frame: 48, initial bounding box:  $x = 352$   $y = 290$   $width = 62$   $height = 167$ .



Fig. 14. The Original TLD Algorithm Experimental Results of Test Data 4.



Fig. 15. The Modified TLD Algorithm Experimental Results of Test Data 4.

Experimental results above show that this paper's modified TLD algorithm greatly improves the tracking success rate in bicycle tracking, and enhances the algorithm's real-time performance, and eliminates false detection problems in bicycle tracking.

## V. CONCLUSIONS

This paper proposes a lot of methods to modify the original TLD algorithm, in order to make the original TLD algorithm more suitable for pedestrian and bicycle tracking. These methods not only enhance the effectiveness of tracking pedestrian and bicycle, but also greatly improve the real-time performance of the TLD algorithm. At last, this paper finished a lot of experiments to show the new TLD algorithm's advantages in effectiveness of tracking and real-time performance.

In a single pedestrian tracking, this paper has achieved good effectiveness. The future work should be devoted to track more than one pedestrian in a video sequence.

## ACKNOWLEDGMENT

This paper is sponsored by the Major Research Plan of National Natural Science Foundation (91120018/91120002), the General Program of National Natural Science Foundation of China (61174178/51178268).

## REFERENCES

- [1] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection" CVPR 2005.
- [2] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in IEEE Conference on Computer Vision and Pattern Recognition, 2008.
- [3] R. Benenson, M. Mathias, R. Timofte, L. Van Gool, "Fast stixel computation for fast pedestrian detection," In: ECCV, CVVT workshop, 2012.
- [4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," Pattern Analysis and Machine Intelligence, 2011.
- [5] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artificial intelligence, vol. 17, no. 1-3, pp. 185–203, 1981.
- [6] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," International Joint Conference on Artificial Intelligence, vol. 81, pp. 674–679, 1981.
- [7] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," International Conference on Pattern Recognition, pp. 23–26, 2010.
- [8] Z. Kalal, J. Matas, and K. Mikolajczyk, "Weighted Sampling for Large-Scale Boosting," British Machine Vision Conference, 2008.
- [9] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints," Conference on Computer Vision and Pattern Recognition, 2010.
- [10] C. Bibby and I. Reid, "Real-time Tracking of Multiple Occluding Objects using Level Sets," Computer Vision and Pattern Recognition, 2010.
- [11] A. Adam, E. Rivlin, and I. Shimshoni, "Robust Fragments-based Tracking using the Integral Histogram," Conference on Computer Vision and Pattern Recognition, pp. 798–805, 2006.
- [12] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1296–1311, 2003.

# On Keyframe Positioning for Pose Graphs Applied to Visual SLAM

Andru Putra Twinanda<sup>1,2</sup>, Maxime Meilland<sup>2</sup>, Désiré Sidibé<sup>1</sup>, and Andrew I. Comport<sup>2</sup>

<sup>1</sup> Laboratoire Le2i UMR 5158 CNRS, Université de Bourgogne, Le Creusot, France.

<sup>2</sup> CNRS-I3S, Université de Nice Sophia Antipolis, Sophia Antipolis, France.

**Abstract**—In this work, a new method is introduced for localization and keyframe identification to solve a Simultaneous Localization and Mapping (SLAM) problem. The proposed approach is based on a dense spherical acquisition system that synthesizes spherical intensity and depth images at arbitrary locations. The images are related by a graph of 6 degrees-of-freedom (DOF) poses which are estimated through spherical registration. A direct image-based method is provided to estimate pose by using both depth and color information simultaneously. A new keyframe identification method is proposed to build the map of the environment by using the covariance matrix between relative 6 DOF poses, which is basically the uncertainty of the estimated pose. This new approach is shown to be more robust than an error-based keyframe identification method. Navigation using the maps built from our method also gives less trajectory error than using maps from other methods.

**Index Terms**—SLAM, spherical system, keyframe identification, covariance matrix.

## I. INTRODUCTION

**S**IMULTANEOUS Localization and Mapping (SLAM) has been one of the most discussed research topics in the domain of autonomous robotics. In the general visual SLAM problem, the camera pose and environment structure are estimated simultaneously and incrementally using a combination of sensors. A visual SLAM approach is interesting in a wide range of robotics applications where a precise map of the environment does not exist.

In the last decade, many methods have been explored to perform robust full translation and rotation (6DOF) localization and mapping. In particular, some of the visual SLAM approaches [1], [2] have used feature-based techniques combined with depth and pose estimation. Unfortunately, these methods are still based on error-prone feature extraction techniques. Furthermore, it is necessary to match the features between images over time which is also another source of error since feature mapping sometimes is not necessarily one-to-one.

One can also refer to appearance and optical flow based techniques to avoid the feature-based problems, by directly minimizing the errors between image measurements. Methods that have similar approach like this fall into the category of image-based or direct methods. One of the earlier works [3] uses a planar homography model, so that perspective effects or non-planar objects are not considered. Recent work [4], [5] uses a stereo rig and a quadrifocal warping function which closes a non-linear iterative estimation loop directly with images. Visual odometry methods are however incremental

and prone to small drifts, which when integrated over time become increasingly significant over a large distance.

A solution to reduce drift in visual odometry is to use image-based keyframe techniques such as in [6], [7], where each pose is estimated with respect to a reference image (keyframe) that has been acquired from learning phase. This is one of the solutions for mapping problem in SLAM, where the environment is represented by a set of connected image keyframes. This approach is also referred to as graph-based SLAM. Most of the work in this domain focused on the back-end which optimizes the obtained graph, such as the method in [8] that performs pose graph optimization by exploiting the sparseness of the Jacobian of the system. However, such methods do not investigate the importance of a keyframe, subsequently do not reduce the number of keyframes. Traditionally, the choice of keyframes is solely based on the travelled distance by the robot or the passing time in between keyframes. This is, however, not the best way to select, from an image sequence, the best images to build the structure of the environment. In the earlier work [9], a statistical approach to identify keyframes using a direct-method was proposed, which is based on the median absolute deviation (MAD) of the residuals. The drawback of this method is that it depends on a threshold value that does not apply for all types of sequences, so that different values are given for different kind of environment, making the map learning process totally empirical.

In the last few years, dense techniques have started to become popular. In particular, an early work [10] performing dense 6DOF SLAM over large distances was based on warping and minimizing the intensity difference using omnidirectional spherical sensors. Alternatively, other approaches have focused only on the geometry of the structure [11]. However, these techniques limit themselves either to photometric optimization only or to geometric information only. Dropping one or the other information means that there are important characteristics from the complete information that are being overlooked which might degrade in terms of robustness, efficiency and precision.

More recently, some techniques have considered to include both photometric and geometric information in the pose estimation process. In [12], a direct ICP technique was proposed which minimizes the error of both information simultaneously. Unfortunately, the approach is not well constrained in the technique because the minimization of the geometric error is only performed on the  $Z$ -component of the scene, not the whole 3D component. In this paper, it is argued that the

error minimization should incorporate all information provided from an omnidirectional spherical camera system, i.e. the photometric and depth (thus, 3D geometric) information, as also proposed in [10], [9]. By using all data, it is ensured that nothing will be overlooked while performing localization. The main contribution of this paper is to investigate a new keyframe identification method for graph-based SLAM that can be applied to general visual SLAM problem. However, in this case, a model of the environment is built by incrementally selecting a subset of the images from the learning sequence to be our reference spheres.

## II. SPHERICAL TRACKING AND MAPPING

An environment will be represented as a graph containing nodes that correspond to robot poses and to all information obtained from those poses, as laid out in [10]. Every edge between two nodes corresponds to the spatial constraints between them. The 3D model of the environment is defined by a graph  $\mathcal{G} = \{\mathcal{S}_1, \dots, \mathcal{S}_k; \mathbf{x}_1, \dots, \mathbf{x}_m\}$  where  $\mathcal{S}_i$  are augmented spheres that are connected by a minimal parameterisation  $\mathbf{x}_i$  which is the 6 degree of freedom (DOF) velocity twist between two spheres, expressed in exponential map. For every sphere  $\mathcal{S}$ , it is defined by a set of  $\{\mathcal{I}, \mathcal{Q}, \mathcal{Z}\}$  where

- $\mathcal{I} = \{i_1, \dots, i_n\}$  is the spherical photometric image.
- $\mathcal{Z} = \{z_1, \dots, z_n\}$  is the depth image.
- $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$  is a set of equally spaced and uniformly sampled points on unit sphere where  $\mathbf{q} \in S^2$  is expressed in spherical coordinate system  $(\theta, \phi, \rho)$  and belongs to a unit sphere ( $\rho = 1$ )

### A. Localization

Robot motion can be represented by a transformation  $\mathbf{T}(\mathbf{x})$  that takes the parameter  $\mathbf{x}$  that consists of two vectors representing: translation velocity  $\mathbf{v} = [v_x \ v_y \ v_z]^T$  and rotation velocity  $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ . The parameter  $\mathbf{x} \in \mathbb{R}^6$  is defined by the Lie algebra as  $\mathbf{x} = \int_0^1 (\boldsymbol{\omega}, \mathbf{v}) dt \in \mathbb{SE}(3)$  which is the integral of a constant velocity twist which produces a transformation  $\mathbf{T}$ . The transformation and twist are related via the exponential map as  $\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]_\wedge}$ , where the operator  $[\cdot]_\wedge$  is defined as follows:

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{v} \\ 0 & 0 \end{bmatrix} \quad (1)$$

where  $[\cdot]_\times$  represents the skew symmetric matrix operator.

For localization of a sphere  $\mathcal{S}$ , an initial guess  $\hat{\mathbf{T}} = (\hat{\mathbf{R}}, \hat{\mathbf{t}}) \in \mathbb{SE}(3)$  of the current vehicle position with respect to a reference sphere  $\mathcal{S}^* = \{\mathcal{I}^*, \mathcal{Q}^*, \mathcal{Z}^*\}$  is available, where  $\hat{\mathbf{R}} \in \mathbb{SO}(3)$  is a rotation matrix and  $\hat{\mathbf{t}} \in \mathbb{R}^3$  is a translational vector. Since it is assumed that the initial guess  $\hat{\mathbf{T}}$  is available, the tracking problem boils down to the estimation of an incremental pose  $\mathbf{T}(\mathbf{x})$  such that  $\bar{\mathbf{T}} = \mathbf{T}(\mathbf{x})\hat{\mathbf{T}}$ , where  $\bar{\mathbf{T}}$  is the estimated pose of the current sphere.

The pose and the trajectory of the camera can be estimated by minimizing a non-linear least squares cost function[13]:

$$\mathcal{C}(\mathbf{x}) = \mathbf{e}_{\mathcal{I}}^T \mathbf{e}_{\mathcal{I}} + \lambda_P^2 \mathbf{e}_P^T \mathbf{e}_P \quad (2)$$

where, for every pair of spherical point and depth  $\{\mathbf{q}_i^*, z_i^*\} \in \mathcal{S}^*$ :

$$\mathbf{e}_{\mathcal{I}} = \begin{bmatrix} \mathcal{I}(w(\bar{\mathbf{T}}; \mathbf{q}_1^*, z_1^*)) - \mathcal{I}^*(\mathbf{q}_1^*) \\ \vdots \\ \mathcal{I}(w(\bar{\mathbf{T}}; \mathbf{q}_n^*, z_n^*)) - \mathcal{I}^*(\mathbf{q}_n^*) \end{bmatrix} \quad (3)$$

$$\mathbf{e}_P = \begin{bmatrix} (\bar{\mathbf{R}}\mathbf{n}_1^*)^T (\bar{\mathbf{P}}(w(\bar{\mathbf{T}}; \mathbf{q}_1^*, z_1^*)) - \bar{\mathbf{T}}\bar{\mathbf{P}}_1^*) \\ \vdots \\ (\bar{\mathbf{R}}\mathbf{n}_n^*)^T (\bar{\mathbf{P}}(w(\bar{\mathbf{T}}; \mathbf{q}_n^*, z_n^*)) - \bar{\mathbf{T}}\bar{\mathbf{P}}_n^*) \end{bmatrix} \quad (4)$$

where  $\mathbf{e}_{\mathcal{I}}$  is a vector containing the intensity errors,  $\mathbf{e}_P$  is a vector containing the structural errors,  $\mathbf{P}_i$  is the  $i$ -th 3D point on the current sphere,  $\bar{\mathbf{P}}_i^*$  is the homogeneous coordinate of  $\mathbf{P}_i^*$  on the reference sphere,  $\mathbf{n}_i^*$  is the surface normal at point  $\mathbf{P}_i^*$ ,  $\bar{\mathbf{R}}\mathbf{n}_i^*$  is the normal at point  $\bar{\mathbf{T}}\bar{\mathbf{P}}_i^*$ , and  $w(\cdot)$  represents the warping of a 3D point from a sphere to another, as shown in Figure 1.

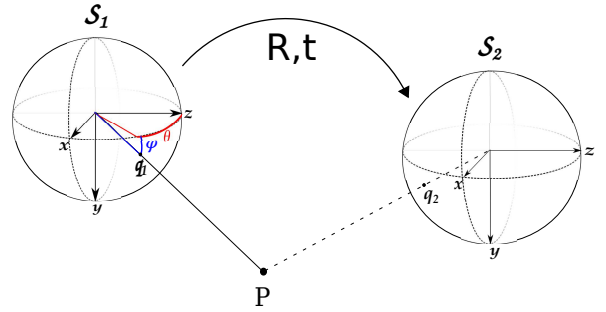


Figure 1. Illustration of spherical warping where the warping goes from  $\mathcal{S}_1$  to  $\mathcal{S}_2$  and  $P$  is a 3D point in the coordinate frame of  $\mathcal{S}_1$ .

The localization can now be considered as a minimization problem. The aim is to minimize simultaneously the cost function in an accurate, robust and efficient manner. Using an iterative approach, the estimate is updated at each step by a homogeneous transformation  $\hat{\mathbf{T}} \leftarrow \mathbf{T}(\mathbf{x})\hat{\mathbf{T}}$ . Using Gauss-Newton algorithm, the pose update  $\mathbf{x}$  can be obtained iteratively from:

$$\mathbf{x} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \begin{bmatrix} \mathbf{e}_{\mathcal{I}} \\ \lambda_P \mathbf{e}_P \end{bmatrix} \quad (5)$$

where  $\mathbf{J}$  is the Jacobian of the cost function which is its derivative with respect to the 6DOF twist, and  $(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$  is the pseudo-inverse of the Jacobian. The Jacobian can be expressed as  $\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \mathbf{J}_{\mathcal{I}} \\ \lambda_P \mathbf{J}_{\mathcal{E}P} \end{bmatrix}$ . By using chain rule, one can rewrite the Jacobian into more modular parts:

$$\mathbf{J}_{\mathcal{E}P} = \mathbf{J}_{\mathcal{I}} \mathbf{J}_w \mathbf{J}_{\mathbf{T}P^*} \quad (6)$$

$$\mathbf{J}_{\mathcal{E}P} = \Delta \mathbf{P}^T \mathbf{J}_{\mathbf{R}n^*} + (\bar{\mathbf{R}}\mathbf{n}^*)^T (\mathbf{J}_P \mathbf{J}_w \mathbf{J}_{\mathbf{T}P^*} - \mathbf{J}_{\mathbf{T}P^*}) \quad (7)$$

where:

- $\mathbf{J}_{\mathcal{I}}$  is the intensity gradient with respect to its spherical coordinate position  $(\theta, \phi)$ . It is of dimension  $n \times 2n$ .



In [10], an efficient way to compute  $\mathbf{J}_{\mathcal{I}}$  using efficient second-order minimization is presented. The same technique is applied in this paper.

- $\mathbf{J}_w$  is the derivative of Cartesian to spherical conversion function. It is of dimension  $2n \times 3n$ .
- $\mathbf{J}_{\mathbf{T}_{P^*}}$  is the derivative (velocity) of point transformation with a dimension  $3n \times 6$ .
- $\mathbf{J}_P$  is the 3D point gradient with respect to its spherical coordinate position  $(\theta, \phi)$ . It is of dimension  $3n \times 2n$ .
- $\Delta P$  is the difference between the transformed points and the warped points  $(\bar{P}^w - \bar{\mathbf{T}}P^*)$
- $\mathbf{J}_{\mathbf{R}_{n^*}}$  is the derivative with respect to the normal rotation. It is of dimension  $3n \times 6$ .

### B. Robust Estimation

During the navigation, the environment can vary between the keyframe and the current images due to moving objects, illumination changes and occlusions. To deal with them, a robust M-estimator is used. The idea of M-estimator is to reduce the effect of outliers by replacing the residuals with another function of the residual. After applying the M-estimator to the residual, the pose update  $\mathbf{x}$  can be obtained from:

$$\mathbf{x} = -(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \begin{bmatrix} \mathbf{e}_I \\ \lambda_P \mathbf{e}_P \end{bmatrix} \quad (8)$$

where  $\mathbf{W}$  is the weighting matrix where the diagonal corresponds to the weight computed by the weight function [14].

## III. KEYFRAME IDENTIFICATION

In graph-based SLAM, selecting keyframes (i.e. reference spheres in this case) to be put as nodes in the final map is an important step. Taking too many references will cause the system to suffer from a high accumulated error because every time a new reference is taken, the residual error of the new reference will always be integrated in the following pose estimates, resulting in an accumulated drift. The error can be due to interpolations, occlusions, illumination change, and the dynamic of the environment (e.g moving cars). Yet needless to say, taking a new reference is also necessary to perform localization because the already mapped reference image goes out of view over large distances. Several strategies for keyframe selection will now be presented.

### A. Median Absolute Deviation (MAD) [9]

One technique to achieve this goal locally is to observe the statistical dispersion of the residual error  $\mathbf{e}$  obtained from the pose estimation process. The most common way to measure this is by computing the standard deviation (STD). However, the standard deviation is not a robust method because of its sensitivity to outliers. The MAD, on the other hand, is one of the simplest robust methods. It has a breakdown point of 50%, which means that the measurement still holds up close to 50% contamination of outliers, while STD has 0% breakdown point since a single large outlier can throw it off.

A new reference sphere is then placed according to the MAD of the weighted error:

$$\gamma < \text{med}(|\mathbf{W} \mathbf{e} - \text{med}(\mathbf{W} \mathbf{e})|) \quad (9)$$

where  $\text{med}(\cdot)$  is a function to extract the median of data and  $\gamma$  is the threshold for keyframe placement decision.

This approach is computationally cheap and optimized in many frameworks, resulting in a possibility to be applied for real-time applications. However, the criterion signifies that a new reference should be taken when the robust variance is too high, while 'too high' is an open statement. A drawback of this criterion is that we need to define a value to be the threshold. This process is totally empirical based on experiments and highly dependent on the characteristics of the sequence. Note that MAD can be applied to univariate data, hence the MAD is applied only on the intensity error since there isn't a good way to merge the two errors into the same scale and unit.

### B. Incremental Ellipsoid

In the pose estimation process, one can compute the uncertainty of the estimation by using the covariance matrix. We propose a method that further observes the error ellipsoid. The orientation of the ellipse can be obtained by computing the eigenvector of the sub-covariance matrices. The orientation of the ellipsoid is, however, not used in the proposed criterion since the orientation of the error is invariant because it is based on the magnitude of the uncertainty. Instead, the semi axes  $\mathbf{s} = [s_x \ s_y \ s_z]^T$  of the error ellipsoid are more interesting to monitor since they are directly connected to the magnitude of the uncertainty. A new keyframe will be added to the map whenever:

$$\|\mathbf{s}_{t|t^*}\| > \|\mathbf{s}_{t|t-1}\| + \|\mathbf{s}_{t-1|t^*}\| \quad (10)$$

where  $\mathbf{s}_{t|t^*}$  are the semi-axes resulting from warping the current sphere to the reference sphere,  $\mathbf{s}_{t|t-1}$  are the semi-axes for warping the current sphere to the previous current sphere,  $\mathbf{s}_{t-1|t^*}$  are the semi-axes for warping the previous current sphere to the reference sphere. The diagram of the comparison is shown in Figure 2.

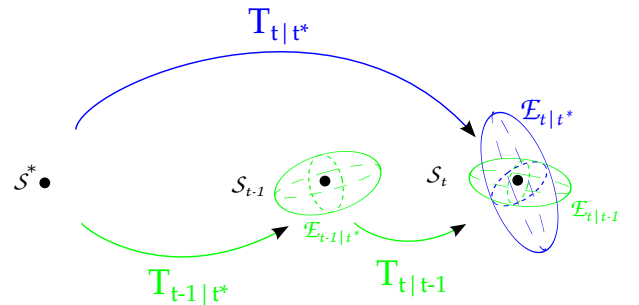


Figure 2. Illustration of incremental ellipsoid criterion

### C. Symmetric Ellipsoid

The incremental error ellipsoid is, however, biased to the direction of computing the sequence. It is almost certain that if the direction of the exploration is inverted (i.e moving from

the end to the beginning of the sequence), the selected nodes will not be the same. This shows the method for selecting reference spheres is not based on the underlying information in the measurement, but depends on the computation order. If it is assumed that the complete sequence and its connectivity is already acquired (before the map learning is performed), a less biased method can be implemented. Instead of selecting the references incrementally, all the images in the sequence will initially be considered as references in the graph. In order to get the best nodes symmetrically, a symmetric comparison is added in the three-node groups. In this case, both forward and backward uncertainty is considered, as shown in Figure 3. The inequality in Equation 10 is now:

$$\|s_{t|t^*}\| + \|s_{t^*|t}\| > \|s_{t|t-1}\| + \|s_{t-1|t^*}\| + \|s_{t^*|t-1}\| + \|s_{t-1|t}\| \quad (11)$$

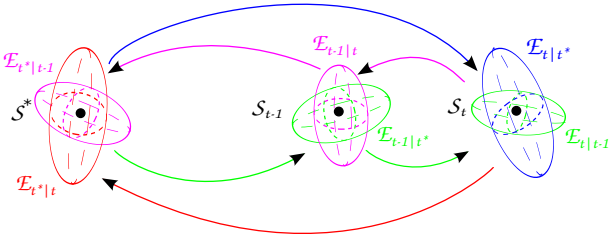


Figure 3. Illustration of symmetric ellipsoid criterion

#### IV. EXPERIMENTS

##### A. Experimental Setup

To test the method, four synthetic sequences have been made. These sequences simulate indoor environment, however the system is designed to work in both outdoor and indoor environments. The detail of the sequences can be seen in Table I and some images are shown in Figure 4. The first two sequences are used to build the map and the last two are used during the map testing phase. We will compare the performance of our keyframe identification methods with the MAD criterion. In this experiment, two MAD thresholds  $\gamma$  are used: 8 and 12.



Figure 4. Image with (a) spherical and (b) diffuse illumination

Our quantitative evaluation involves the number of references during the map building as well as the trajectory error with respect to the ground truth that can be computed from:

$$\Delta \mathbf{T} = \tilde{\mathbf{T}}^{-1} \hat{\mathbf{T}} \quad (12)$$

where  $\tilde{\mathbf{T}}$  is the ground truth and  $\hat{\mathbf{T}}$  is the estimated pose. The 6 DOF error between the estimated and the ground truth

Table I  
SEQUENCE DATA

Seq	#Images	Size	Illumination	Distance Traveled
1	1549	1024×512	Spherical	142 m
2	1549	1024×512	Diffuse	142 m
3	1400	512×256	Spherical	169 m
4	1400	512×256	Diffuse	169 m

can be obtained by computing the logarithmic map of  $\Delta \mathbf{T}$ , such that  $\Delta \mathbf{x} = \log(\Delta \mathbf{T})$ . The trajectory error  $\Delta \mathbf{x}$  will be a 6-element vector that contains the difference of translation velocity  $\Delta \mathbf{v}$  and rotation velocity  $\Delta \boldsymbol{\omega}$ .

##### B. Map Building Result

From Table II, it can be seen that there is a huge increase of number of references in the maps using MAD criteria on the sequence with spherical illumination (Sequence 1) compared to the sequence with diffuse illumination (Sequence 2). This is inevitable due to the higher intensity error introduced in the Sequence 1, meaning that the MAD threshold is easily reached after only a few images. The number of references using the incremental ellipsoid criterion, however, does not vary much with respect to the change in illumination: 32 and 30 for Sequence 1 and 2 respectively. In contrast, the number of keyframes in the maps using the MAD criteria varies with changes in lighting condition: 30 to 150 for MAD-8, and 19 to 77 for MAD-12.

From this result, it can be seen that the ellipsoid criteria are better in terms of automatically choosing a consistent number of references for both types of sequences because it does not include a scalar threshold that has to be tuned before map learning process. In other words, the value 8 or 12 is not the best threshold value for the MAD criterion to select keyframes from Sequence 1. This verifies our argument that the MAD has a disadvantage due to its threshold that needs to be adjusted depending on the condition on the sequence, unlike the ellipsoid-based criteria that do not need any adjustments.

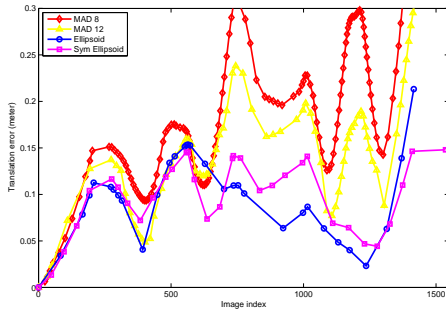
To observe the pose error, we can refer to Figure 5 that shows graphs of the chosen keyframes index against their pose error. If we look closely on the graphs, keyframes in the maps built by using the MAD are rather uniformly picked along the sequence. On the other hand, the ellipsoid criteria do not behave the same way and pick more keyframes at certain points along the sequence. These are the points where the robot is taking a turn and going to another corridor. By doing such turns, there will be a lot of new information introduced in the sequence and naturally it is favorable to take new keyframes when a lot of new information is introduced. The implication of this new information in the sequence is that higher error and higher uncertainty will be computed, resulting in more keyframes during the turns. However at some other points, the criteria pick less keyframes. This is the counter part of taking a turn which is going through a straight trajectory. Since we are working with a dense spherical system, going through such straight trajectory (in a corridor) does not introduce a lot of new information in the images. So, the criteria will only decide



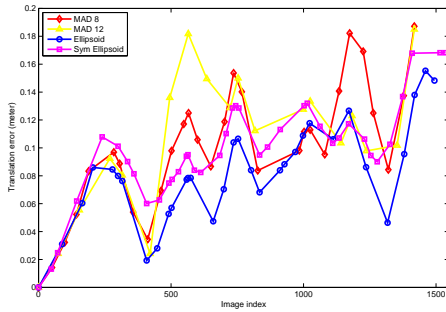
to take a new keyframe when the interpolation error starts to decrease the accuracy of the tracking system.

Table II  
MAP BUILDING RESULT

Seq.	Criterion	#Ref	Avg. transl. err. (m)	Avg. rot. err.
1	MAD-8	150	0.1972	0.0183
	MAD-12	77	0.1484	0.0121
	Inc. ell.	32	0.0979	0.0065
	Sym. ell.	33	0.0982	0.0055
2	MAD-8	30	0.0999	0.0091
	MAD-12	19	0.1045	0.0086
	Inc. ell.	30	0.0814	0.0066
	Sym. ell.	34	0.0983	0.0061



(a)



(b)

Figure 5. Keyframe's translational error for: (a) Sequence 1 and (b) Sequence 2. The rotational error is similar.

So far, we can conclude that in Sequence 1 the two ellipsoid criteria are superior compared to the MAD, in terms of number of references and pose error. Almost at every point in the maps obtained by ellipsoid criteria, the keyframes' pose error is less than the ones by the MAD. Although it is also the case for Sequence 2, we can not conclude yet whether the ellipsoid criteria are better than MAD criteria since the keyframes' pose error is not very different in the maps. However, we can see in Figure 6 that the reconstructed structures using ellipsoid criteria are slightly better, as the reconstructed structures of the second floor from MAD criteria are slightly slanted compared to the ground truth because the MAD criteria give more rotational error in the maps compared to the ellipsoid criteria. This can be the effect of reference placement choice by the criteria which has been mentioned previously, in which ellipsoid criteria select more keyframes on the turns than on straight trajectories. The reconstructed

structures from Sequence 1 also have similar results, in which the structures reconstructed using MAD criterion are slanted compared to the ellipsoid criteria.

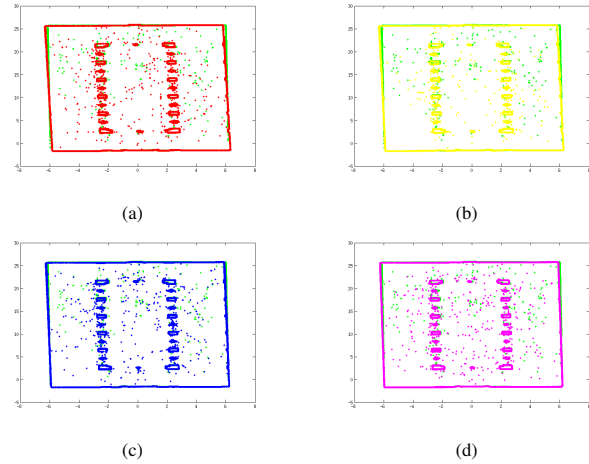


Figure 6. Map quality on the second floor in the order of MAD-8, MAD-12, incremental, and symmetric ellipsoid ((a),(b),(c),(d)) on Sequence 2. The structure in green is the ground truth.

### C. Map Testing Result

The first noticeable result from the trajectory error in Figure 7 is that there are a lot of spikes in the translational error graph. These spikes are caused by the changing of reference during navigation because the minimization process is still biased to the previous reference. This can be avoided by taking multiple keyframes simultaneously as references during navigation, as mentioned in [13], such that when a new reference is considered, change is not so radical since other references are kept during reference switching.

Referring to the trajectory error for environment with spherical illumination (Sequence 3) in Figure 7-a, it can be seen that tracking with the maps obtained by using MAD gives higher error. This drift is naturally caused by the reference pose error during the map learning. In addition to higher pose errors, other problems might appear in maps with high number of keyframes. Such maps make creating edges in the graph challenging, making it necessary to consider more sophisticated methods to build the connections between keyframes. With a high number of keyframes, they can be easily connected by false connections (false loop closures). These wrong connections can lead to wrong changes during navigation process, which will result in failure in tracking and higher trajectory errors.

The incremental and symmetric ellipsoid methods seem to perform equally well, with slightly better performance from incremental ellipsoid, except at the end of the sequence. This might be the result of bias in direction. The incremental ellipsoid only considers one direction of the trajectory during learning which is the same direction as the testing sequence. So, the minimization scheme favors the incremental ellipsoid more than the symmetric ellipsoid.

If the case with diffuse illumination is considered, as shown in Figure 7-b, the performance of all four criteria pretty much

the same. Even so, at some points the ellipsoid criteria perform better than the MAD criteria and vice versa. This is highly related to the reference pose estimation error during the map building phase.

Table III  
MAP TESTING RESULTS

Seq.	Criterion	Avg. transl. err. (m)	Avg. rot. err.
3	MAD-8	0.2026	0.0161
	MAD-12	0.1706	0.0116
	Inc. ell.	0.1193	0.0077
	Sym. ell.	0.1207	0.0065
4	MAD-8	0.1247	0.0102
	MAD-12	0.1292	0.0088
	Inc. ell.	0.102	0.0069
	Sym. ell.	0.1241	0.0065

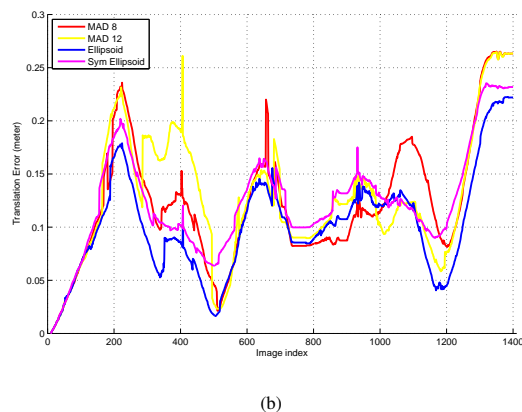
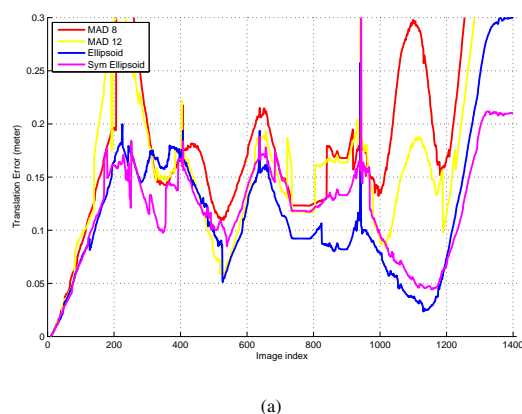


Figure 7. Translational error during navigation test on: (a) Sequence 3 and (b) Sequence 4. The rotational error is similar.

## V. CONCLUSIONS

A new spherical localization method was proposed that uses all photometric and geometric information for dense visual SLAM. A novel keyframe identification method (incremental ellipsoid) was proposed that incorporates the covariance matrix and compares the uncertainty ellipsoid between spheres. We have also extended it to work on symmetric navigation paths within the pose graph (symmetric ellipsoid) to ensure best selection of the keyframes. Although the MAD has the advantage of computational efficiency, it has been shown that

the MAD has a drawback due to its scalar threshold value that needs to be adjusted accordingly to the characteristics of the sequence. On the other hand, the proposed methods don't need this adjustment and have better statistical properties, in terms of number of references as well as the quality of the maps. It has been shown that the method is more robust to variations in the lighting condition of the map.

There are still several aspects that remain to be explored within the proposed model. All the criteria presented in this paper are still biased to the first image in the sequence since it has to be included in the final map. By combining the symmetric ellipsoid criterion and loop-closure detection during keyframe identification, this bias can be eliminated since the first keyframe can be pruned during the map building phase. It has been mentioned beforehand that the work presented here is just improving the front-end of the graph-based SLAM. Some testing should also be done after applying it to the back-end. By doing this, the graph optimization method that will adjust the position of the nodes in the graph accordingly to its constraints. However, no pruning is needed since the selected nodes are already optimized in the mapping process.

## REFERENCES

- [1] A. Davison and D. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 865–880, 2002.
- [2] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IROS*, 2008, pp. 3946–3952.
- [3] S. Benhimane and E. M., "Real-time image-based tracking of planes using efficient second-order minimization," 2004.
- [4] A. Comport, E. Malis, and P. Rives, "Accurate Quadri-focal Tracking for Robust 3D Visual Odometry," in *IEEE International Conference on Robotics and Automation, ICRA'07*, Rome, Italy, April 2007.
- [5] —, "Real-time quadrifocal visual odometry," *International Journal of Robotics Research, Special issue on Robot Vision*, vol. 29, no. 2-3, pp. 245–266, 2010.
- [6] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," 2004.
- [7] A. Remazeilles, F. Chaumette, and P. Gros, "Robot motion control from a visual memory," in *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, vol. 4, New Orleans, Louisiana, April 2004, pp. 4695–4700.
- [8] M. Kaess, A. Ranganathan, and F. Dellaert, "isam : Fast incremental smoothing and mapping with efficient data association," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 1670–1677.
- [9] M. Meilland, A. I. Comport, and P. Rives, "Dense visual mapping of large scale environments for real-time localisation," in *IEEE International Conference on Intelligent Robots and Systems*, sept. 2011, pp. 4242–4248.
- [10] —, "A spherical robot-centered representation for urban navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10. Taipei, Taiwan*, Paris, France, 8-9 Novembre 2010.
- [11] D. Cole and P. Newman, "Using laser range data for 3d slam in outdoor environments," in *ICRA*, 2006, pp. 1556–1563.
- [12] T. Tykkälä, C. Audras, and A. Comport, "Direct Iterative Closest Point for Real-time Visual Odometry," in *The Second international Workshop on Computer Vision in Vehicle Technology: From Earth to Mars in conjunction with the International Conference on Computer Vision*, Barcelona, Spain, November 6-13 2011.
- [13] M. Meilland and A. Comport, "Super-resolution 3D Tracking and Mapping," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany., May 6-10 2013.
- [14] P. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

# Online Spatiotemporal-Coherent Semantic Maps for Advanced Robot Navigation

Ioannis Kostavelis, Konstantinos Charalampous and Antonios Gasteratos

**Abstract**—In this paper we introduce a novel online semantic mapping framework apt to establish the seamless cooperation between the low level geometrical information and the high level environment’s perception. Its main contribution involves the online formation of a semantic map, relying on the memorization of abstract place representations and capitalizing both on space quantization and time proximity. A time evolving Augmented Navigation Graph is formed describing the semantic topology of the explored environment and the connectivity among the places visited, which is expressed as the inter-places transition probability. A side contribution of this paper involves the utilization of the learned semantic maps for efficient navigation in the explored environment. Moreover, a specific human-robot interaction paradigm is proposed by illustrating a competent methodology to address the *go-to* tasks. The performance of the proposed framework was evaluated on long range robot datasets in an unstructured office environment and it exhibits remarkable performance by inferring semantic maps in previously unexplored environments.

## I. INTRODUCTION

In modern human societies it is of great importance to build up machines that can be operated by non specialists or even by technologically illiterate people, such as youngsters or elderly. An obvious solution to this challenge is to build up cognitive robot companions ample to competently perceive and interpret their surroundings. In particular, considering navigation the robots should retain cognitive interpretation capacities to understand human concepts for places and objects and, in order to proficiently deploy in domestic environments, they should be able to construct geometrical maps and simultaneously draw semantic inferences about their ambient.

Multiple definitions about *semantic mapping* have been proposed in the literature [1], the majority of which converges to the aspect that a semantic map is an augmented representation of the explored environment, which -complementary to the geometrical information- entails high level qualitative features. Those features might be the abstraction of the spatial knowledge, the place labeling and even the connectivity information regarding the perceived places. Significant research efforts have been devoted to geometrical map construction and their results can be distinguished into three main categories, viz. metric [2], topological [3] and hybrid [4]. Although these methods proved to be capable of driving robots into specific target positions, they

lack of high level cognition attributes, which would allow them to bring the human-robot interaction one step beyond.

Aiming to remove this barrier, the proposed work is oriented towards the direction of incorporating the following goals:

- online partitioning of the places visited into distinguished annotated rooms to form a semantic layer on top of the geometrical one
- spatiotemporal connectivity among the detected places expressed in terms of transition probability
- functionalities that enable the user to pass high level navigation orders directly to a mobile robot, such as “*go to the living room*”

More precisely, the sole prior knowledge this paper utilizes is a learned visual vocabulary encompassing abstract representations of the possible place categories the robot may visit during its journey. As the robot wanders, it partitions its surroundings into places according to their spatiotemporal relation employing the space quantization functionality of the Hierarchical Temporal Memory (HTM) networks [5]. The used nodes allow the extraction of specific semantic and localization attributes able to outline the space and, thus, to form a topological map, which evolves with time. Considering the temporal proximity, a Markov Model is evolved progressively and grouped online in consequent frames, proportionally to the already learned places, which derives the formation of an interconnected Augmented Navigation Graph (ANG), describing both the learned places and the transition probability among them. The formulated scheme constitutes the proposed semantic map which brings together navigation and cognitive information by seamlessly integrating the low level topological graphs into the high level ANG, thus incorporating spatiotemporal and semantic attributes at the same time. Given the semantic map in hand, the user is able to pass high level commands to redirect the robot from one learned place to another. The execution of these actions is treated hierarchically by employing simple graph traversing algorithms that firstly detect the sequence of the places the robot should progressively follow and then retrieving the corresponding topological graphs that connect those places in terms of robot’s localization. A video illustrating the proposed framework is also available <sup>1</sup>

The rest of the paper is organized as follows: in Sec. II a review of the related literature is outlined. In Sec. III the framework for the formation of the semantic map is presented, while in Sec. IV its application is illustrated. Section V describes the experimental evaluation proving the

I. Kostavelis, K. Charalampous and A. Gasteratos are with the Department of Production and Management Engineering, Democritus University of Thrace, Robotics and Automation Laboratory {*gkostave, kchara, agaster*}@pme.duth.gr

significance of the proposed method and, last, in Sec. VI conclusions are drawn.

## II. RELATED WORK

Albeit the plethora of laborious research conducted in the specific field [6], each of the respective work tackles the problem of semantic mapping only partially. Typically this problem can be further decomposed to its primitives such as the localization, the mapping, the navigation and the place categorization [7], [8], [9], [10], where significant findings have already been accomplished. The most common characteristic of all these methods is that they encode the space attributes using abstract spatial representations of the sensory input, howbeit, they utilize this knowledge to improve the geometrical aspect of the navigation framework only.

Contrary to the aforementioned methods, there are various semantic mapping approaches that embody both geometrical and cognitive characteristics for navigation. In a preliminary method described in [11], an intelligent object recognition system operating on a mobile robot is illustrated. This work relies solely on the ability of the system to correctly recognize and locate different objects in an environment. The authors in [12] introduced an augmented method for semantic mapping utilizing semantic symbols, yet assigned by a human. In [13] a similar algorithm based on the place geometry and the object information is presented. In particular, a laser scanner mounted on a mobile robot enables the acquisition of dense point clouds, which are then partitioned and annotated with semantic labels using object recognition techniques. In a more sophisticated manner, the work in [14] encloses significant semantic characteristics to form concept oriented representations of space, as well as to infer about the explored environment relied on object recognition. The main advantage of such methods is the employment of conceptual strategies to determine the boundaries of the detected places. However, in cases that similar objects appear in different locations this method would fail to produce a consistent semantic map. Moreover, the method described in [15], utilizes Image Sequence Partitioning (ISP) techniques to group visually similar images as topological graph nodes. An interesting aspect of the semantic mapping described in [16] utilizes a clustering algorithm according to the recognized objects in a scene retaining both spatial and appearance based information. In this algorithm the semantic map is formed by superimposing an object map over the geometrical one. Additionally, the authors in [17] designed a method that forms augmented semantic maps integrating multiple cues, such as place geometry and object information by employing both laser and vision data. This method also integrates the geometrical and the semantic maps in terms of a navigation graph under the supervision of the conceptual abstraction of the detected places. In a more recent work [1], the authors presented a probabilistic framework combining heterogeneous cues such as object observations, geometrical

characteristics, conceptual common-sense and even human assentation to form a semantic map. Although this scheme exhibits encouraging performance, it highly depends on the human input during the formation of the semantic map. Moreover, it needs to detect doors in order to annotate a topological graph with a specific place label, which might be tricky in arbitrary uneven environments with irregular passages.

## III. PART A: FORMING THE SEMANTIC MAP

### A. Learning a Bag-of-Words

Learning a visual vocabulary in the arrangement of Bag-of-Words (BoW) [18] comprises the sole off-line procedure in the proposed framework, as depicted in Fig. 1. We considered a labeled sequence of images that corresponds to a robot's trajectory containing various instances from all the place categories to be memorized. This sequence of images was independent of the target one, on which the proposed framework will be later evaluated. Specifically, the scale-invariant feature transform (SIFT) [19] is applied on each single image of the sequence and the detected feature points are concatenated framewise. The resulting feature space is denoted by a data matrix  $\mathbf{S}$ , which represents a BoW problem and comprises a substantial description of the entire space to be memorized. Following the work

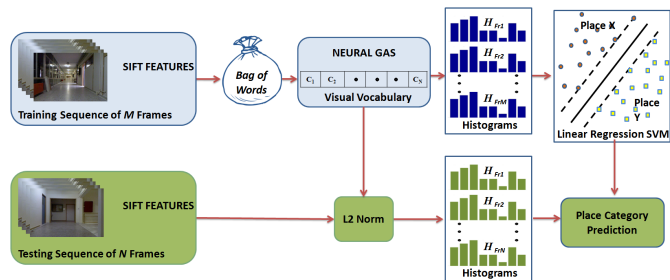


Fig. 1. The upper line depicts the procedure concerning the creation of the vocabulary, the appearance based histograms and the SVM training, whilst the lower line describes the formation of the appearance based histograms using the pre-computed vocabulary and the L2 norm for the SVM classification.

presented in [20]  $\mathbf{S}$  is clustered by a vector quantization algorithm, namely the Neural Gas one, which has been preferred instead of the k-means to avoid local minimum solutions. Given  $\mathbf{S}$ , a subset of visual words should be defined that characterize the entire input space in an abstract form. Thereupon, the set of  $Q$  centers of the resulting space quantization  $\mathbf{C}^{128 \times Q} = [c_1, c_2, \dots, c_Q]$  comprises the visual vocabulary and provides a satisfactory representation of the initial space. The visual words are then utilized to create an appearance based histogram for each respective image of the sequence. Given the detected features we form a representative consistency histogram  $h_{S_k} \in \mathbb{R}^Q$  for each image  $k = 1, 2, \dots, M$  spread over the  $Q$  visual words. The L2 norm between the detected features and the visual words is calculated and the representative histogram is formed; the binning is performed according to the smaller distance.

<sup>1</sup>You may find the full version of the video at: [http://utopia.duth.gr/~gkostave/downloads/semantic\\_video.rar](http://utopia.duth.gr/~gkostave/downloads/semantic_video.rar).



Consequently, each image in the sequence has been replaced by a respective appearance based histogram, which is utilized to execute any further comparison. Hence, the computational burden is simplified due to the fact that there is no need to elaborate full images.

The learning of the different place categories is accomplished by means of Support Vector Machines (SVM) [21]. The reason of this choice is based on the excessive performance of the SVMs in several visual recognition tasks [22]. Given the fact that the robot should learn various place categories, the one-against-all strategy has been preferred, i.e. for each different class a respective SVM is trained to separate this single class from all the others. The linear SVM yielded remarkable recognition accuracy, while it kept low the number of parameters that have to be tuned.

### B. Spatiotemporal Place Memorization

The place memorization is undertaken by a *spatial* and *temporal pooler* which evolves simultaneously with time in an online fashion. This architecture is inspired by the single node functionality in the HTM networks, as analytically described in our previous work [23].

The **spatial pooler** is exposed to the target sequence and memorizes in an online fashion the quantization space of the appearance based histograms. The subset of such histograms -eventually added to the spatial pooler- are the quantization centers. The pooling of new centers is governed by two specific conditions that should be both satisfied as follows:

- The extracted histogram  $h_{S_i} \in \mathcal{R}^Q$ , corresponding to frame  $i$ , is checked against the existing quantization centers according to a threshold value and it is considered different in case that the respective L2 norm is found greater. In this case  $h_{S_i}$  is marked as a candidate to retain in the spatial pooler as a new quantization center, otherwise, it is ignored.
- A voting procedure of the SVM models decides about the place label of the candidate quantization center. To infer about the place label of a new candidate, the  $w$  neighbors of the  $h_{S_i}$  participate in a majority vote process with aim to infer about the place label of the new candidate. This constrain exploits the time proximity of the successive frames during the robot's exploration and boosts confidence in the place categorization.

Each quantization center is accompanied by the robot's current location estimation and, therefore, it is considered as a node in the topological graph of the explored environment. A unique attribute in our method is that a new node is added in the topological graph each time the current appearance based information, i.e. the set of nodes on the topological graph, is not sufficient to describe the explored environment. Moreover, the nodes classified in the same class describe a specific area providing semantic attributes in the formed topological graph. The latter is decomposed into multiple interconnected subgraphs, each of which describes a detected place in the explored environment and exhibits excessive spatial coherence. Figure 2(a) depicts the resulted topological graph during a robot's exploration, which contains semantic

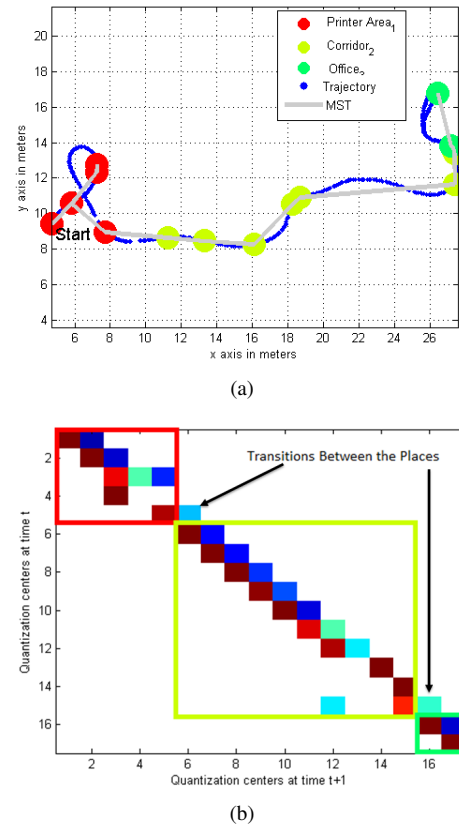


Fig. 2. a) The topological graph superimposed over the robots trajectory. Three different places have been detected: the "Printer Area", the "Corridor" and the "Office". The topological graph is expressed as the MST among the existed nodes; b) The normalized TAM, where the partitioning of the quantization centers according to their class label and the transitions among the different groups are illustrated. Note that among the "Printer Area" and the "Office", there are no transitions in the TAM, which indicates that passing directly from the one place to another is not possible.

information about the places visited. This comprises an important novelty comparing to similar works [4], where a node is added in the topological graph whenever the robot has traveled a certain distance, thus the system's agility in managing intelligence weakens.

The **temporal pooler** operates simultaneously with the spatial one forming a temporal adjacency matrix (TAM). Assuming that the spatial pooler consists of  $N$  quantization centers -nodes in the topological graph- an  $N \times N$  square matrix  $T$  is created. The rows and the columns of the matrix correspond to the nodes triggered at time  $t$  and  $t + 1$ , respectively. Matrix  $T$  follows the first order Markov Model as its elements contain the respective number of transitions among the existing quantization centers.  $T$  constantly updates itself by examining the consecutive occurrences within two successive histograms. In the specific cases that the spatial pooler does not add a new quantization center, the transition among the consecutive input histograms updates the TAM as follows: the respective input histograms at times  $t$  and  $t + 1$  are examined with respect to the minimum L2 norm among all the existed quantization centers in the spatial pooler. Assuming that the  $i$ th,  $j$ th quantization centers

have been triggered during this procedure at times  $t$  and  $t + 1$ , then the  $T(i, j)$  element of the TAM is increased by one. In a different case, when a new quantization center is added in the spatial pooler, the TAM expands, maintaining its square form and initializing the new row and column elements. The physical meaning of this procedure is that the nodes in the topological graph that share great spatial and temporal proximity are grouped together around the diagonal of the TAM. The mutual transition probabilities among the quantization centers, which represent the nodes in the topological graph, are obtained by normalizing the TAM (Fig. 2(b)).

### C. Augmented Navigation Graph and Place Connectivity

The ANG derives from the online segmentation of the normalized TAM. The aim of this procedure is to group the labeled places, while it computes the transition probability from one group to the other according to the physical arrangement of the nodes on the topological graph as well as the intra-node transitions. Thereupon, the high level semantic information is amalgamated online with the low level geometrical one along the robot's perambulation.

More precisely, the last formed TAM is partitioned into groups: the goal is to divide the set of quantization centers -nodes in the topological graph- into spatiotemporal coherent subgroups corresponding to the different places. On the one hand, the temporal coherence is obtained by utilizing the sequential temporal transitions of TAM. On the other hand, the spatial consistency of the places is ensured due to the fact that the TAM is partitioned into specific groups by taking into consideration the place label of the quantization centers. Each of the groups formed contains quantization centers with identical labels and, consequently, they belong to the same place. The quantization centers that belong to the same group are most likely to exhibit both spatial and temporal adjacency, expressed by nodes' locations on the topological graph and their transitions on the TAM, respectively. The transition probability among different places (groups) is estimated by computing the intersection of the transitions between the quantization centers of different groups in the TAM. The detected places and their mutual transition probabilities comprise the *Augmented Navigation Graph*. Considering the TAM in Fig. 2(b) the partitioning procedure results into three detected places, namely the "Printer Area", the "Corridor" and the "Office", hence three different respective groups are formed, each of which constitutes a node in the ANG as depicted in Fig. 3. It is worth noting that each node in the graph consists solely of quantization centers similarly classified by the SVM models and the weights in the graph denote the transition probability between different groups.

A unique attribute in our method is that it simultaneously handles the existence of more than one similar types of places appearing in the explored environment, e.g. in a school building, where multiple classrooms exist. The formation of the ANG is further constrained by the currently estimated location of the robot. The detection of multiple places e.g. "Office1" and "Office2", that bear the same label is accom-

plished by applying the Minimal Spanning Tree algorithm (MST) on the set of the nodes in a topological graph. In particular, the MST is applied on the localization output of the nodes being grouped together during the partitioning of the TAM. The resulting edges connect all the quantization centers according to their minimum Euclidean distance. In case that an edge possesses a value greater than an adaptive threshold, then the corresponding group is partitioned and this procedure repeats until no in-between edge is greater than the threshold. The distribution of the intra-node distances is computed and the outliers determine the value of the threshold. The utilization of the MST ensures the detection of similar places according to the low level geometrical adjacency, while it simultaneously retains the formation of the ANG in case that the robot visits the same place more than once.

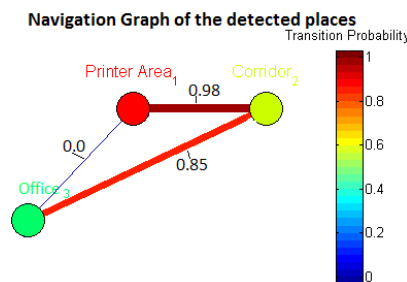


Fig. 3. The ANG that corresponds to the topological graph and the TAM depicted in Fig. 2. The weights on the graph indicate the transition probability among the places. Note that the transition probability from the "Printer Area" to the "Office" is zero, indicating that the robot should firstly pass through the "Corridor" in order to traverse from the one place to another.

## IV. PART B: TRAVERSING THE SEMANTIC MAP

The annotated topological graph accompanied by the ANG comprises the semantic map of the proposed work. This high level map can be utilized by a user to pass directly high level orders to the robot in order to move from one place to another. Since the proposed method is an online one, the user may intervene at any moment and pass a *go-to* command to the robot relative to the places already seen in the explored environment.

Given that the topological graph possesses low level spatial attributes and remains transparent to the user, the latter may interact only with the ANG. Towards this direction, the traversing of the semantic map can be treated as a hierarchical procedure that comprises two distinct phases. The user is constantly aware about the current location of the robot, i.e. the robot infers about its current location and, therefore, it can be redirected to any other known place. In the first phase the robot examines the ANG to find the conductivities among its current place and the target one. Given the fact that the ANG retains information about the place connectivity, the robot can plan a high level route. Considering the example presented in Fig. 3 the robot is not able to move from the "Printer Area" to the "Office",



unless it passes through the "Corridor". This high level graph traversing is accomplished using the Dijkstra algorithm, where the corresponding weights in the ANG are expressed as the inverse transition probabilities. The first phase imposes the sequence of places the robot has to pass through in order to reach the target. Given the detected nodes that have been triggered on the ANG during the first phase, only the respective quantization centers of the topological graph participate in the second one. More precisely, for each group corresponding to a specific place, the MST is computed for the respective nodes in the topological graph. Next, the derived paths are concatenated forming the minimum cost path that -in geometrical terms- connects the corresponding places. The formation of such a constrained path is physically explained taking into consideration that the corresponding sequence of places is indeed a feasible route connecting the current and the selected place. In a similar fashion to the one described in [12] the sequence of the nodes that form the selected route can be embodied to the robot's path planner in order to autonomously navigate towards the target place.

## V. EXPERIMENTS

The proposed semantic mapping framework has been evaluated by means of the COLD dataset [24]. This is a large database suitable for vision based recognition systems, inasmuch as it consists of three sub-datasets, acquired in different Universities viz. Freiburg, Ljubljana and Saarbrücken. Given the fact that the introduction of a localization and mapping algorithm is not within the goals of this paper, the location data provided by the COLD dataset have been used in our experiments. During the off-line learning phase, the SVM classifiers were trained on the Freiburg and Saarbrücken sub-datasets utilizing the classes that intersect all sub-datasets. Moreover, we considered that the proposed framework should operate in previously unseen places, hence it was evaluated on the Ljubljana sub-dataset.

As far as the formation of the vocabulary in the BoW is concerned, the parameter  $Q$  in the Neural Gas was set equal to 200, to offer a decent compromise between the classification accuracy and the computational cost. The system has been trained off-line using linear SVMs in a ten-fold cross validation fashion resulting in the relaxation parameter  $C$  equal to 100. The classification accuracy is summarized in the confusion matrix as shown in Table I, exhibiting more than 93% accuracy in all cases. The trained SVM models

TABLE I

CONFUSION MATRIX CONCERNING THE CLASSIFICATION ACCURACY OF THE SVM MODELS AFTER THE TEN-FOLD CROSS VALIDATION.

	Pr. Area	Kitchen	Corr.	Office	Lab.	Bath.
Pr. Area	97%	0.0	3%	0.0	0.0	0.0
Kitchen	0.0	100%	0.0	0.0	0.0	0.0
Corr.	0.0	0.0	100%	0.0	0.0	0.0
Office	1%	0.0	0.0	94%	5%	0.0
Lab.	0.0	0.0	0.0	4%	96%	0.0
Bath.	0.0	1%	0.0	0.0	0.0	99%

where used for the evaluation of the semantic mapping

framework during the online experiments. In particular, the parameter  $w$  in the voting procedure was set equal to 4, i.e. the four neighbors around the candidate quantization center are examined. A trajectory for the evaluation of the semantic mapping framework on the Ljubljana sub-dataset (Fig. 4(a)) that consists of six places ("Printer Area", "Corridor", "Office", "Bathroom" "Kitchen" and "Laboratory"), has been utilized. It is clear that the semantic annotated topological graph has been correctly formed by detecting precisely all the existed places that observed during the robot's exploration. One attribute that is spotted is that the places labeled as "Corridor" have been sufficiently memorized and the topological graph does not pool additional nodes until the appearance based histograms significantly differentiate. Another attribute also highlighted here is that the "Corridor" is split into two different places according to the MST partitioning of the topological graph. The spatial connectivity in all cases is correctly formed indicating the precise formation of the groups in topological graphs, given their geometrical attributes. Regarding the respective ANG of the places visited, it has also been correctly evolved during the robot's exploration. The edges with the omitted transition probability values are zero, thus indicating the absence of direct connections among non-adjacent places. As an example, in order to traverse from the "Printer Area" to the "Bathroom", the direct transition is unfeasible and the robot has to pass through the "Corridor<sub>2</sub>" and "Corridor<sub>5</sub>", thus optimizing its route by including only the essential known places, in order to reach its target, as depicted in Fig. 4(b). Additional experiments were conducted proving the correct execution of the *go-to* orders for arbitrary positions operating precisely on the high level ANG, while the low level nodes in the topological graph are correctly triggered, Fig. 4(c).

## VI. CONCLUSIONS

In this paper a method to seamlessly conjugate semantic and geometrical maps has been presented with ultimate aim to bring closer the robot navigation to the human one. An online semantic mapping framework has been introduced based solely on visual input and, given an off-line learned vocabulary, the proposed framework draws accurate semantic inferences about its surroundings. The proposed method reveals spatiotemporal coherency leading to rational topological graphs. The main innovation in our method is that the high level ANG is formed online during the robot's exploration and it exhibits semantic attributes of the detected places, while it express the spatiotemporal connectivity among them in terms of transition probability. Wherefore, the proposed ANG reveals qualitative navigation characteristics similar to the ones apprehended by a user, whilst it preserves also tokens of quantitative ones, in terms of traversability quantification among places. To this end, the ANG can be further exploited by the user to pass direct high level commands to the robot. The execution of such orders is performed hierarchically, firstly by planning an abstract route and then by triggering the appropriate sequence of nodes in the topological graph. Moreover, the proposed method has

been thoroughly examined on long range indoor datasets yielding remarkable performance.

## REFERENCES

- [1] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *ICRA*. IEEE, 2012, pp. 3515–3522.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [3] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Visual topological slam and global localization," in *ICRA*. IEEE, 2009, pp. 4300–4305.
- [4] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *ICRA*. IEEE, 2011, pp. 3041–3047.
- [5] D. George and J. Hawkins, "Towards a mathematical theory of cortical micro-circuits," *PLoS Computational Biology*, vol. 5, no. 10, 2009.
- [6] S. Thrun *et al.*, "Robotic mapping: A survey," *Exploring artificial intelligence in the new millennium*, pp. 1–35, 2002.
- [7] J. Courbon, Y. Mezouar, and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 392–402, 2009.
- [8] A. Pronobis, O. Martínez Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 298–320, 2010.
- [9] F. Dayoub, G. Cielniak, and T. Duckett, "A sparse hybrid map for vision-guided mobile robots."
- [10] B. Steder, G. Grisetti, and W. Burgard, "Robust place recognition for 3d range data based on point features," in *ICRA*. IEEE, 2010, pp. 1400–1405.
- [11] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.
- [12] C. Nieto-Granda, J. G. Rogers, A. J. Trevor, and H. I. Christensen, "Semantic map partitioning in indoor environments using regional analysis," in *IROS*. IEEE, 2010, pp. 1451–1456.
- [13] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.
- [14] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, 2008.
- [15] H. Korrapati, J. Courbon, Y. Mezouar, and P. Martinet, "Image sequence partitioning for outdoor mapping," in *ICRA 2012*. IEEE, 2012, pp. 1650–1655.
- [16] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. K. Mackworth, "Automated spatial-semantic modeling with applications to place labeling and informed search," in *Canadian Conference on Computer and Robot Vision*, 2009., 2009, pp. 284–291.
- [17] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.
- [18] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*. IEEE, 2003, pp. 1470–1477.
- [19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
- [20] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV, Workshop on Statistical Learning in Computer Vision*, vol. 1, 2004, p. 22.
- [21] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [22] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A realistic benchmark for visual indoor place recognition," *Robotics and autonomous systems*, vol. 58, no. 1, pp. 81–96, 2010.
- [23] I. Kostavelis and A. Gasteratos, "On the optimization of hierarchical temporal memory," *Pattern Recognition Letters*, 2011.
- [24] M. Ullah, A. Pronobis, B. Caputo, J. Luo, R. Jensfelt, and H. Christensen, "Towards robust place recognition for robot localization," in *ICRA*. IEEE, 2008, pp. 530–537.

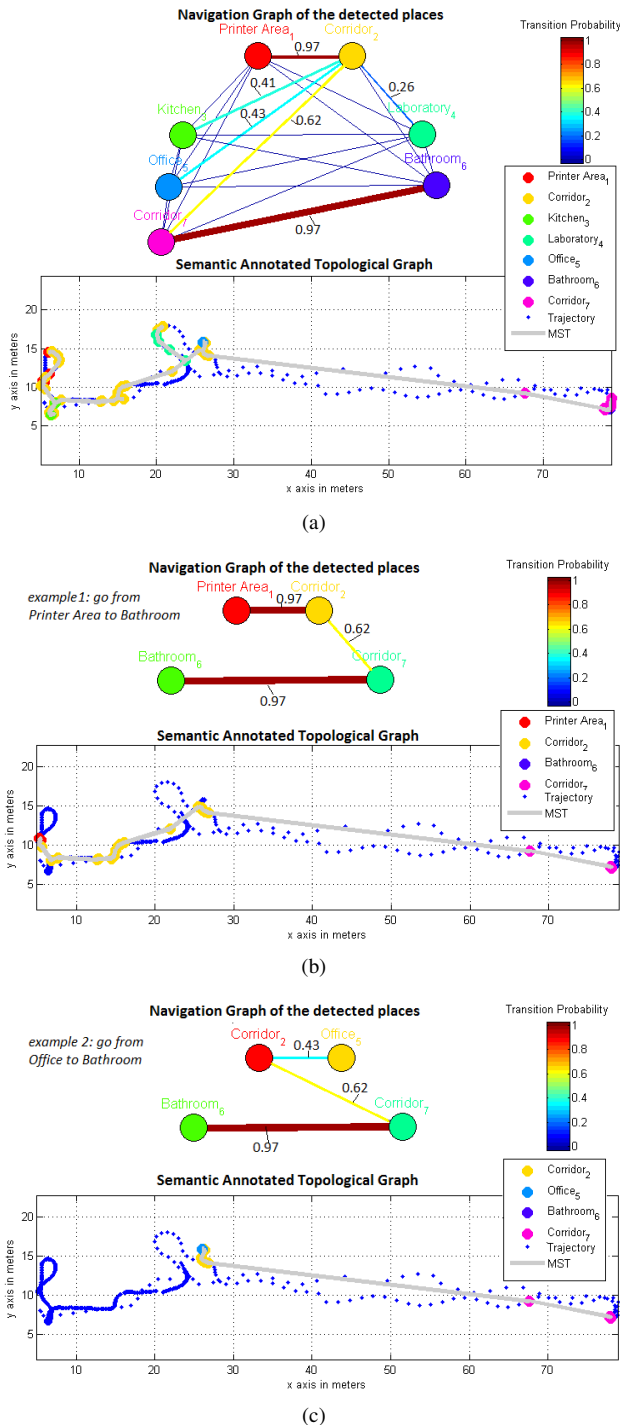


Fig. 4. a) The ANG and the respective semantic annotated topological map during the robots exploration in the small sequence Sunny<sub>1</sub> of the COLD dataset. The existed places are a "Printer Area", a "Kitchen", a "Laboratory", a "Corridor", an "Office" and a "Bathroom". Examples of *go-to* actions b) from the "Printer Area to the Bathroom" and c) from the "Office to the Bathroom", are presented.

# Use of a Monocular Camera to Analyze a Ground Vehicle's Lateral Movements for Reliable Autonomous City Driving

Young-Woo Seo and Rangunathan (Raj) Rajkumar

**Abstract**—For safe urban driving, one prerequisite is to keep a car within a road-lane boundary. This requires human and robotic drivers to recognize the boundary of a road-lane and the location of the vehicle with respect to the boundary of a road-lane that the vehicle happens to be driving in. We present a new computer vision system that analyzes a stream of perspective images to produce information about a vehicle's lateral movements, such as distances from a vehicle to a road-lane's boundary and detection of lane-changing maneuvers. We improve existing work in this field and develop new algorithms to tackle more challenging cases, such as driving on inter-city highways. Tests on real inter-city highways showed that our system provides stable and reliable performance in terms of computing lateral distances, while yielding reasonable performance in detecting lane-changing maneuvers.

## I. INTRODUCTION

In city-driving scenarios, an essential component of safe driving is keeping the vehicle in a road-lane boundary. In fact, such a capability is a prerequisite for various advanced driver assistance systems (ADAS) [3], [5], [12] as well as for executing reliable autonomous driving [15], [20]. One way to achieve this capability, for human drivers, is to design lane-departure warning systems. By analyzing steering commands from in-vehicle data and lane-markings through a forward-looking camera, such a warning system can alert drivers when they unintentionally deviate from their paths. A self-driving car, to be deployed on urban streets, should be capable of keeping itself in a road lane before executing any other urban autonomous driving maneuvers, such as changing lanes and circumventing stalled or slow-moving vehicles.

The task of staying within a road-lane begins with perceiving longitudinal lane-markings. A successful detection of such lane-markings leads to the extraction of other important information – the vehicle's location with respect to the boundary of the road-lane. Such information about lateral distances of the vehicle to the left and right boundaries of a road-lane help a human driver and a robot driver keep the vehicle in the road-lane boundary. The capability of driving within designated lanes is critical for autonomous driving on urban streets, where GPS signals are either degraded or can be readily disrupted.

Some earlier work, using 3D LIDARs, demonstrated impressive results in understanding road geometry. In particular, four of the autonomous driving applications installed multiple off-the-shelf laser range finders toward the ground and measured the reflectivity values of road surfaces. In such manner they analyzed the geometry of the current roadway [6], [11], [15], [20]. Two of ADAS applications proposed

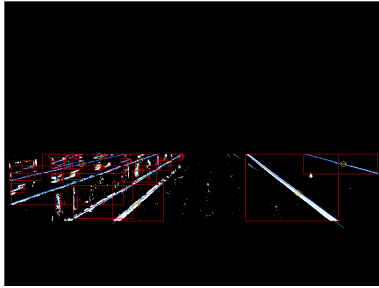
lane-departure warning systems using automotive-grade laser range scanners. Instead of multiple LIDARs, they used a single LIDAR with multiple horizontal planes: six for Ogawa and Takagi [17] and four for Kibbel et al. [10]. Both methods recognized lane-markings in a similar way: 1) handpicking some of the scan points, 2) finding a list of parameters, (e.g., curvature, lane-width, lateral offset, and yaw-angle), and 3) representing the lane with a polynomial (e.g., quadratic or cubic).

However, such a high-end, expensive LIDAR may not always be available. Instead of relying on such active range sensors, many researchers as an alternative, with an eye on lower costs and installation flexibility, have studied the use of vision sensors. Researchers have actively studied road geometry understanding through lane-marking detection; some research results have been successfully commercialized as well [3]. Some utilize inverse perspective mapping to remove perspective distortions [1], [16], others use in-vehicle data, such as steering angle, velocity, whether a wiper is turned on [3], [12]. Some have implemented Bayes filters, to make their lane-detection methods robust [9], [10], [16], [17]. However, most of this research using a vision sensor focuses on developing driver-assistance systems for manual driving, where the outputs are not always expected to be produced and human drivers can, if necessary, override the incorrect outputs [1], [3], [5], [12], [16]. For a self-driving car, in contrast, the information about a vehicle's location with respect to a road-lane boundary should be available throughout navigation and in a bounded performance. Otherwise, when driving on regions with unreliable GPS signal reception (e.g., urban canyons), an autonomous vehicle might easily veer from the centerline of a road-lane, resulting in unacceptable consequences.

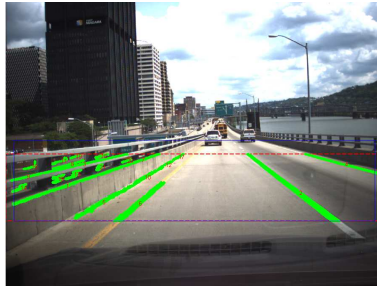
To produce a vehicle's relative motions within a road-lane, we develop a vision algorithm that analyzes perspective images acquired from a monocular camera to extract information about a vehicle's lateral movements, metric offset measurements from road-lane boundaries, and detection of lane-changing maneuvers. To this end, our algorithm first extracts longitudinal lane-markings from input perspective images and, on inverse perspective images, analyzes their geometric relation. This step yields the local geometry of a current roadway. The algorithms then solve a homography between a camera plane and a roadway plane to assign the identified geometry with metric information.

The contributions of this paper include 1) a method of analyzing the geometry of a current roadway, 2) a method of computing metric information of points on the ground plane, and 3) a new vision system for computing a vehicle's lateral movements.

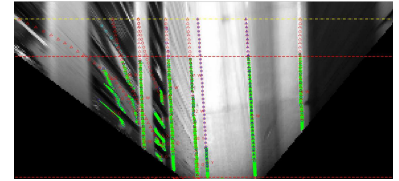
Young-Woo Seo is with the Robotics Institute and Rangunathan Rajkumar is with Dept of Electrical Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, young-woo.seo@ri.cmu.edu, raj@ece.cmu.edu



(a) An example of lane-marking detection results. Our lane-marking detector produces a binary image about lane-markings and the detected lane-markings are represented as a list of pixel groups (or blobs). Each of the red boxes shows a bounding-box of a detected lane-marking blob.



(b) The initial lane-marking detection results are overlaid onto, after false positive removal, the input image. The blue rectangle defines the image region that is transformed into an inverse perspective image.



(c) This subfigure shows only a part of an inverse perspective image to enlarge the image sub-region where lane-marking blobs appear.

Fig. 1: Results of a lane-marking detection.

## II. UNDERSTANDING LATERAL MOTIONS OF A GROUND VEHICLE FROM A SINGLE IMAGE

Our goal in this work is to provide a ground vehicle with information about its lateral movements. We call the road-lane, which our vehicle happens to be driving on, the host road-lane. The information provided includes the vehicle's lateral location in meters relative to the host road-lane's boundary and occurrences of lane-changing maneuvers. To acquire such information, our vision algorithms first detect longitudinal lane-markings on the images acquired from a forward-looking camera, and classify their colors (e.g., yellow or white); then transform a perspective image into an inverse perspective image to obtain the information about the geometric structure of the host roadway, such as the number of road-lanes in the current roadway and the index of the host road-lane from the leftmost road-lane; and, finally, we compute metric measurements of the identified regions to obtain information about the vehicle's lateral motion.

In what follows, we detail how we recognize lane-markings from perspective images and compute the geometry of a local roadway from inverse perspective images. We then explain how we compute 3-dimensional world coordinates of 2-dimensional image coordinates of the identified roadway geometry so as to produce the information about the vehicle's lateral motion in meters.

### A. Recognizing Lane-Markings for Understanding Local Geometry of Roadway

Road-markings define how drivable regions are used to guide vehicles' navigation. They are obviously important cues to understanding the geometric structure of a roadway. Among these, the ones we want to detect are those that longitudinally depict boundaries of individual road-lanes. In a forward-looking image of urban-streets, we can readily, with the naked eye, distinguish lane-markings. They have distinguishing colors (white and yellow), relatively higher intensity than their neighboring pixels, and occupy approximately known locations. However, these salient features are not always available for image processing; after all the actual values of lane-marking pixels vary based on image acquisition conditions.

Instead of dealing directly with these challenging variations in lane-marking pixels' appearances, we identify lane-marking image regions by implementing a simple filter, which emphasizes the intensity contrast between lane-marking pixels and their neighboring pixels. Our lane-marking detection algorithm was inspired by the one developed by Nieto and his colleagues [16].

Normal longitudinal pavement lane markings on highways (i.e., inter-city and inter-state highways in the U.S.) are 4~12 inches wide (10~30.48 centimeters) [13]. Given this fact, we can readily compute the number of pixels used to depict lane-markings on each row of the input image. For example, for a given pre-computed lane-marking pixel width,  $w_i$ , our filter transforms the original image intensity value,  $I(u, v)$ , into  $I(u, v)'$  by

$$I(u, v)' = 2 \times I(u, v) - \{I(u - w_i, v) + I(u + w_i, v)\} - |I(u - w_i, v) - I(u + w_i, v)|$$

If  $I(u, v)'$  is greater than a predefined maximum value, we set it to that maximum value (e.g., 255). If  $I(u, v)'$  is lesser than zero, we set it to 0. To produce a binary image of lane-markings from this filter response, we do a thresholding that keeps only pixels of which values are greater than a given threshold. Figure 1a shows an example of lane-marking detection results. Even with many (readily discernible) false positive outputs, our lane-marking detection outputs are sufficient because their false negatives are quite small, meaning that our detector picked up almost all true longitudinal lane-markings appearing in the image. We then represent the lane-marking detection result as a list of pixel groups (or blobs) and analyze their geometric properties, such as heading and length, to filter out some non-lane-marking blobs. To further filter out false positives, we also compute the ratio of the sum of a blob's width to that of a true lane-marking to estimate the likelihood that a lane-marking blob is a true lane-marking.

$$\gamma(b_i) = \frac{\sum_{v_j} |u_{j,1} - u_{j,|u_j|}|}{\sum_{v_j^*} |u_{j,1}^* - u_{j,|u_j|}^*|}$$

where  $b_i$  is the  $i$ th lane-marking blob,  $u_{j,1}$  ( $u_{j,|u_j|}$ ) is the  $j$ th row's first (last) column of the  $i$ th blob, and  $v_j^*$  is the corresponding information of the true lane-marking blob.



The color of a lane-marking plays an important role of determining its semantics. For example, in the U.S., a yellow (or white) longitudinal lane-marking separates traffic flows in the opposite (same) direction [13]. To obtain such semantic information about a lane-marking, we classify, using a Gaussian mixture color model, the color of a detected lane-marking blob into one of three categories: yellow, white, and other. In particular, the color class of a detected lane-marking blob is determined by computing,  $\arg \min_{c \in C} (\mu_b - \mu_c)^T (\Sigma_b + \Sigma_c)^{-1} (\mu_b - \mu_c)$ , where  $\mu_b$  and  $\Sigma_b$  are the mean and covariance of HSV (Hue-Saturation-Value) color of a lane-marking blob and  $\mu_c$  and  $\Sigma_c$  are a color model's mean and covariance. We reserve an "other" class for handling all other colors of lane-marking blobs other than the two major color classes: yellow and white.

To obtain the information about the geometric structure of the current roadway, we compute an inverse perspective image from a given perspective image. The inverse perspective mapping is an image warping technique that is frequently used to remove the perspective effect from the field of lane-marking detection [1], [5], [12], [16]. This mapping essentially defines two transformations of a point,  $\mathbf{X}$ , from a perspective image to an inverse perspective image,  $\mathbf{X}^{inv} = \mathbf{T}_{per}^{inv} \mathbf{X}^{per}$ , and vice versa,  $\mathbf{X}^{per} = \mathbf{T}_{inv}^{per} \mathbf{X}^{inv}$ . Figure 1c shows a part of the inverse perspective image of the perspective image shown in Figure 1b.

Before we analyze the geometry of the current roadway, we further filter out false-positive lane-marking blobs from inverse perspective images where two parallel lane-markings are (nearly) parallel to each other. We removed lane-marking blobs from further consideration if their orientations were not aligned with the primary orientation. The primary orientation of lane-marking blobs is that of the longest lane-marking blob. This selection is based on the assumption that the longest lane-marking blob is always aligned with the roadway's driving direction, regardless of whether it is truly a lane-marking. For the remaining lane-marking blobs, we select any lane-marking blob pairs if their distance is probabilistically significant. In other words, we assume that the widths of legitimate road-lanes follow a Gaussian distribution,  $P(w_i) \sim N(\mu, \sigma)$ . We pick a lane-marking blob and a neighboring lane-marking blob. And then we compute the average distance between  $k$  selected points from the lane-marking blob pair and use that as the width between the pair. We keep the pair for further consideration if the probability of the approximated width is significant (e.g., within  $1\sigma$ ). This process results in a list of lane-marking blobs, some of which are in fact true longitudinal boundary lane-markings. Our approach of selecting a road-lane hypothesis is similar to that of [9] in terms of probabilistic hypothesis generation, but different in that Kim [9] used a combination of RANSAC and a particle filter to generate road-lane hypotheses.

To finalize the search of road-lane boundary lane-markings, we use the lane-marking color classification results to handpick some of the selected lane-marking blob pairs. In addition, we use two pieces of prior information: the most frequent number of road-lanes and the semantic meaning of lane-markings' colors. In particular, from government-published highway statistics [14], the majority of highways are four-lane, with two lanes each for traffic in each driving direction. In the U.S., where the vehicles drive on the right

side of a road, when a driver observes a yellow lane-marking on the left side, that lane-marking almost certainly indicates the left boundary of the road-lane. This also holds true when one observes a (solid) white lane-marking on his left side to the immediate left.<sup>1</sup> Once we find one of lane-marking blobs on the left, either white or yellow, we choose its right-side counterpart based on the pre-defined maximum number of road-lanes. The strength of each individual hypothesis is also probabilistically evaluated as before. Figure 2 shows the results of our algorithm on recognizing the structure of a highway in Pittsburgh, PA USA. Although there are many false positive lane-marking blobs (depicted in green), the appearances of which are legitimate, our algorithm was able to pick up the right combination of lane-markings for delineating road-lane boundaries. For the internal representation, we interpolate the centerline of two identified boundary lane-markings of the host road-lane and fit a quadratic function to estimate the curvature of the current roadway.

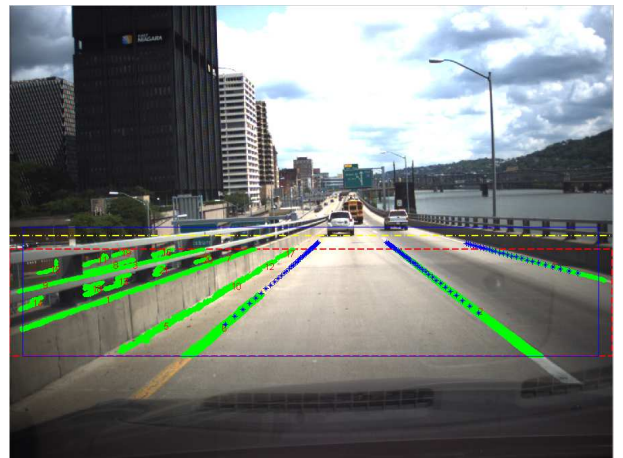


Fig. 2: The road-lane boundaries detected by our algorithm are depicted by a series of blue stars.

### B. Metric Information Computation

Using the method described in the previous section, we recognized some of the detected lane-marking blobs as boundary lane-markings for the current roadway. This information enables us to understand 1) how many road-lanes are in the current roadway and 2) the index of the current road-lane from the leftmost road-lane. In the example shown in Figure 2, we know that our vehicle is driving on the leftmost road-lane of a two-lane (inter-city) highway. We now need to compute the lateral distances of our vehicle from the left and the right boundaries of the host road-lane.

To this end, we define a homography between a roadway plane and an image plane to estimate 3-dimensional coordinates of interesting points on the roadway plane. A 3D world coordinate computation through such a homography works well when the camera plane and the roadway plane are perpendicular to one another. Occasionally, however, such an assumption falls apart because of the vehicle's ego-motion and uneven ground surface. To handle with such cases, we

<sup>1</sup>We know which lane-marking blobs are located at the left of our vehicle because we know the image coordinates of the point our camera is projected on, in a perspective image.

estimate the angle between the camera plane and the ground plane using the vanishing point.

In what follows, we first explain how we compute a vanishing point along the horizon line and then details how we compute world coordinates of interesting points on the ground plane.

1) *Vanishing Point Detection for Estimating Pitch Angle:* Knowledge of a vanishing point's location and the horizon line on a perspective image provides a great deal of useful information about road scene geometry. Among these, we are interested in estimating the angle between the camera plane and the ground plane. A vanishing point is an intersection point of two parallel lines on a perspective image. In urban street scenes, one might obtain plenty of parallel line pairs, pairs such as longitudinal lane-markings and building contour lines. To obtain these contour lines and other lines, we tried three methods: Kahn's method [8], the probabilistic, and the standard Hough transform [18]. We found that the Kahn's method works best in terms of the number of resulting lines and their lengths. The Kahn's method basically uses the principal eigen vector of a pixel group's coordinates, to compute the orientation of a line fitting to that group. Figure 3 shows one result of our line extraction, where each of the extracted lines is depicted in a different color based on its orientation.

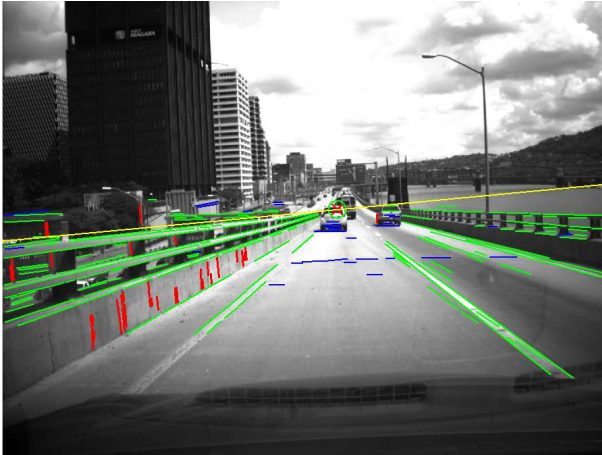


Fig. 3: An example of vanishing point detection result. The red "x" in a green circle represents the computed vanishing point along the horizon line. The yellow line represents the identified horizon line.

Given a set of extracted lines, we use RANSAC to find the best estimation of a vanishing point. In particular, we first set two priors for the horizontal and vertical line groups as  $\mathbf{vp}_h = [0, 0, 1]^T$ ,  $\mathbf{vp}_v = [0, 1, 0]^T$  in the camera coordinate. We then categorize each of the extracted lines into one of these two groups based on the Euclidean distance to horizontal and vertical priors. For each line pair randomly selected from the horizontal and vertical line groups, we first compute the cross-product of two lines,  $\mathbf{vp}_{ij} = l_i \times l_j$ , to find an intersection point. We use this intersection point as a vanishing point candidate. We then claim the vanishing point candidate with the smallest outliers as the vanishing point for that line group. A line pair is regarded as an outlier if the angle between a vanishing point candidate and

the vanishing point obtained from the line pair is greater than a pre-defined threshold (e.g., 5 degrees). We repeat this procedure until a vertical vanishing point is found and more than one horizontal vanishing point is obtained. The horizon line is obtained by linking all of those horizontal vanishing points. Figure 3 shows one result of our vanishing point computation.

2) *A Perspective Transformation between Camera Plane and Road Plane:* This section details how we model the perspective transformation between an image plane,  $\pi$ , and a road plane,  $\mathbf{n}$ . We assume that a world coordinate frame aligned with the camera center and the roadway plane is flat. Figure 4 illustrates the perspective transformation we used in our study. The camera coordinate is oriented such that the  $z_c$ -axis is looking along a road's driving direction, the  $y_c$ -axis is looking down orthogonal to the road plane, and the  $x_c$ -axis is oriented perpendicular to the driving direction of the road. In addition, we model, based on our vehicle coordinates, the coordinate frame of the road plane such that the  $X_R$ -axis of the road plane is aligned with the  $z_c$ -axis of the camera (or world) coordinate and the  $Y_R$ -axis of the road plane is aligned with the  $x_c$  axis of the camera (or world) coordinate.

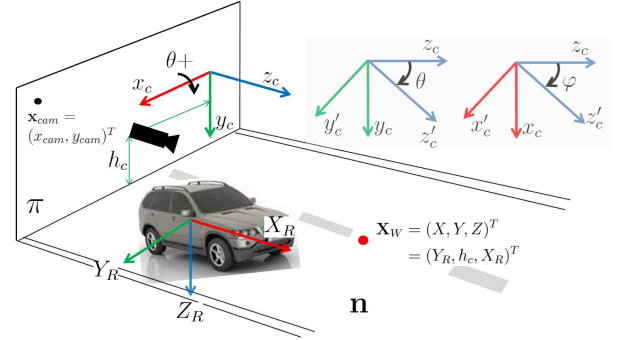


Fig. 4: A perspective transformation between the camera plane and the roadway plane.

In this setting, a point in the real-world,  $\mathbf{X}_W = (X, Y, Z)$ , can be represented as  $\mathbf{X}_W = (Y_R, h_c, X_R)$ , where  $h_c$  is the camera's mounting height from the road plane. We use the basic pinhole model [4] to define the perspective central projection between a point in the world,  $\mathbf{X}_W$  and a point in a camera plane,  $\mathbf{x}_{cam} = (x_{cam}, y_{cam})$ . Note that a point in an image plane is further mapped through  $\mathbf{x}_{im} = K\mathbf{x}_{cam}$ , where  $K$  is a camera calibration matrix defining a camera's intrinsic parameters [4].

$$\mathbf{x}_{cam} = \mathbf{P}\mathbf{X}_W \quad (1)$$

where  $\mathbf{P}$  is the camera projection matrix that defines the geometric relationship between two points,  $\mathbf{x}_{cam}$  and  $\mathbf{X}_W$ . The projection matrix, in particular, consists of a rotation matrix,  $\mathbf{R}_{3 \times 3}(\phi, \theta, \psi)$  and a translation matrix  $\mathbf{t}_{3 \times 1}(h_c)$ ,  $\mathbf{P} = [\mathbf{R}(\phi, \theta, \psi) | \mathbf{t}(h_c)]$ , where  $\phi, \theta, \psi$  define roll, pitch, and yaw angles. Assuming that roll and yaw angles are zero, the central projection equation is detailed as

$$\mathbf{x}_{cam} = [\mathbf{R}_{3 \times 3} | \mathbf{t}_{3 \times 1}] \begin{bmatrix} \mathbf{X}_W \\ 1 \end{bmatrix}_{4 \times 1}$$



$$\begin{aligned} &= \mathbf{R}\mathbf{X}_W + \mathbf{t} \\ \mathbf{X}_W &= \mathbf{R}^T \mathbf{x}_{cam} - \mathbf{R}^T \mathbf{t} = [\mathbf{R}^T] - \mathbf{R}^T \mathbf{t} \mathbf{x}_{cam} \\ \text{where } \mathbf{R} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{bmatrix}, \mathbf{t} = \begin{bmatrix} 0 \\ h_c \\ 0 \end{bmatrix} \end{aligned}$$

where  $c\theta$  is  $\cos\theta$  and  $s\theta$  is  $\sin\theta$ . We solve Equation 1 algebraically to obtain the coordinates of a point in the real world,  $(X_R, Y_R)$ .

$$\begin{bmatrix} X_R \\ Y_R \end{bmatrix}_{2 \times 1} = \begin{bmatrix} x_{cam}p_{33} - p_{13} & x_{cam}p_{31} - p_{11} \\ -y_{cam}p_{33} + p_{33} & -y_{cam}p_{31} + p_{21} \end{bmatrix}_{2 \times 2}^{-1} \times \begin{bmatrix} -x_{cam}(p_{32}h_c + p_{34}) + p_{12}h_c + p_{14} \\ y_{cam}(p_{32}h_c + p_{34}) - p_{22}h_c - p_{24} \end{bmatrix}_{2 \times 1} \quad (2)$$

where  $(X_R, Y_R)$  is a point on the road plane in the world coordinate. Once we obtain these coordinates, it is straightforward to compute metric measurement of a point on the road plane. For example,  $X_R$  is the distance from the camera center.

To precisely compute such a metric measurement, it is necessary to obtain Euler angles, particularly the pitch angle, the angle between the camera plane and the ground plane. We approximate the pitch angle from a vanishing point computation in the following way. Suppose that a vanishing point at the horizon line is defined as [7]:

$$\mathbf{vP}_h^*(\phi, \theta, \psi) = \left[ \frac{c\phi s\psi - s\phi s\theta c\psi}{c\theta c\psi}, \frac{-s\phi s\psi - c\phi s\theta c\psi}{c\theta c\psi} \right]^T$$

Suppose that the yaw and the roll angles are zero, the above equation yields:

$$\mathbf{vP}_h^*(\phi = 0, \theta, \psi = 0) = \left[ \frac{0}{c\theta}, -\frac{s\theta}{c\theta} \right]$$

If a road plane is flat and perpendicular to an image plane, the vanishing point along the horizon line is exactly mapped to the camera center, resulting in the pitch angle being zero. From this fact, we can compute the pitch angle by analyzing the difference between the  $y$  coordinate of a vanishing point and that of the principal point,  $\tan^{-1}(|P_y - vp_y|)$ , where  $P_y$  is the  $y$  coordinate of the principal point.

Figure 5 presents an example result from our local roadway geometry analysis. At the top left, we display information about the geometric structure of the host roadway, such as the number of road-lanes, the index of the host road-lane, and the host road-lane's width in meters. In particular, our vehicle is driving on the first lane of a two-lane road in which the width of the host road-lane is estimated to be 3.52 meters and the true road-width is 3.6 meters. Two (red) bars along the left road-lane boundary indicate the estimated distances from the camera center (in this case, 3.80 and 9.82 meters). Finally, the lateral distances of our vehicle from the left and right boundaries are computed as 1.028 and 0.577 meters.

With this information, we can also detect whether our vehicle ever crosses a boundary of the host road-lane. In particular, we represent the estimated lateral distances of our vehicle from the left with negative numbers and from the right with positive numbers. To detect a lane-detection maneuver, we first observe these numbers up to  $k$  previous time steps (or frames), determine which lateral offset is

smaller (or which side is closer to the vehicle), and claim a lane-changing maneuver when the sign of the closest side is changed. To go back to normal driving status, we observe these sequential values again and claim "normal" driving if we observe  $k-l$  number of the same signs. It is important to observe a series of similar values before triggering the state change. If we only respond to a sign change, our algorithm would fail to distinguish zig-zagging from a lane-changing maneuver. Figure 6 presents a series of images as an example of lane-changing maneuver detection.

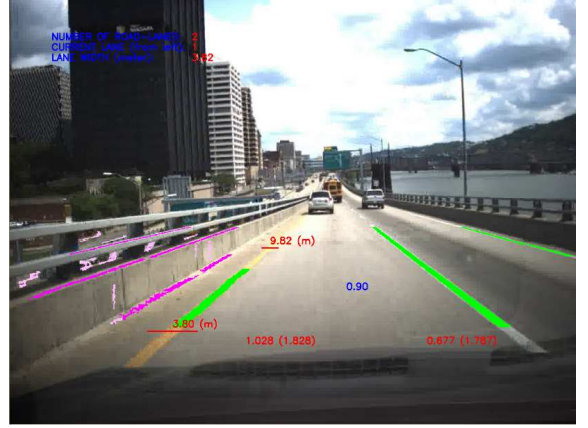


Fig. 5: An example result from our local roadway geometry analysis.

### III. EXPERIMENTS

In this section, we present our detailed experimental settings and results. We drove a robotic vehicle equipped with a pose estimation system. The accuracy of our pose estimator is from approximately 0.1 to 0.3 meter. We drove the vehicle one km along a curvy and hilly segment of road. Our manual measurement recorded a true lane width of 3.6 meters, but some regions of the testing path had different widths due to road geometry (i.e., intersections) or designated U-turn areas.

Figure 7 shows results of metric computation for the estimated local roadway geometry. The  $x$ -axis is time and the  $y$ -axis is computed metric in meters. A (green) dashed horizontal line is depicted at 3.6 to indicate the true lane-width. We intentionally drove the vehicle along the centerline of the testing roads until time step 400 and then, before taking a U-turn between 790 and 910, we drove the vehicle in a zig-zag fashion. While making a U-turn, our system generated no outputs, which were correct. After the U-turn, we zig-zagged at a higher fluctuation. At the upper part of the Figure, the results of lane-width computation are shown, whereas at the lower part, the results of lateral offset computation are shown, where the magenta circles (the cyan triangles) represent the left (right) lateral offsets.

On average, the lane width estimation varied between 3 and 4.5 meters with a variation of 0.342 meter. To clearly differentiate measurement errors, a different shape is depicted at the top of a lane-width estimate: A blue square for when the error is less than 0.2 meter, a cyan circle for when it is between 0.2 and 0.3 meter, and a green circle for all remaining estimates. We could improve the performance if we intentionally removed the lane-width

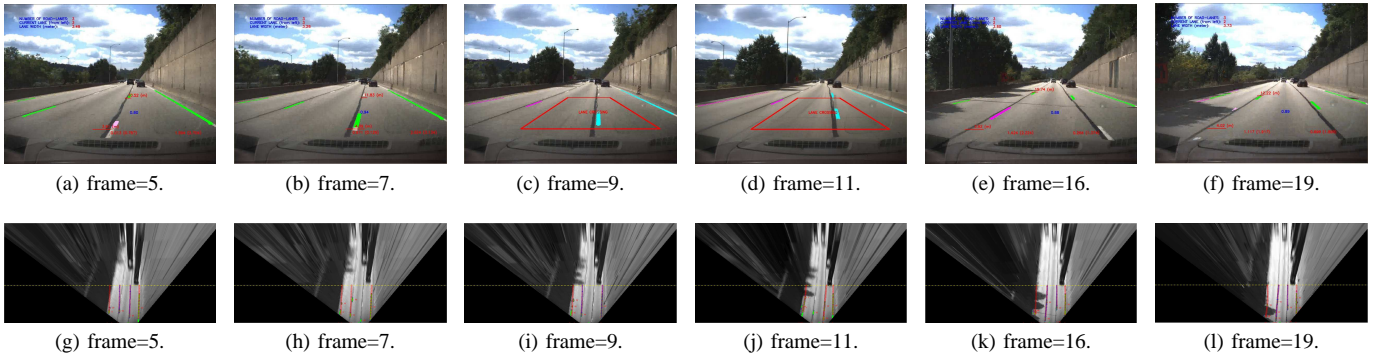


Fig. 6: An example of lane-changing maneuver detection. Images at the upper row show a series of perspective images whereas the ones at the lower row present a corresponding pairs of inverse-perspective images.

estimate, when its value is greater than 3.9 meters. Although such a thresholding is valid, in terms of using a prior information, we did not do this, to measure the accuracy as it is.

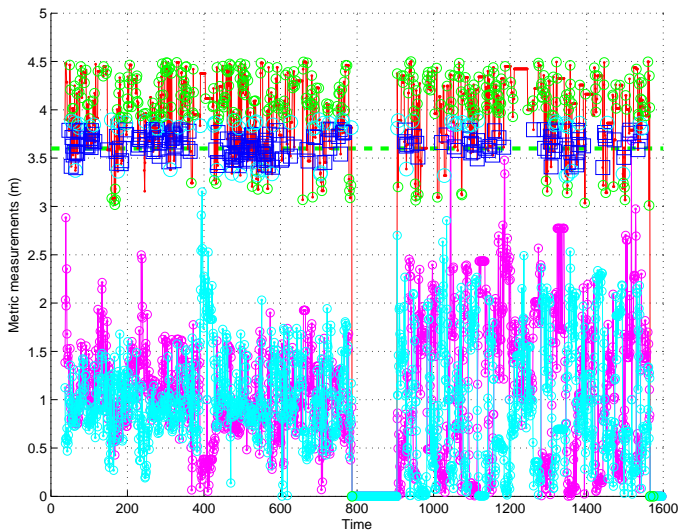


Fig. 7: Results of metric computation for the local roadway geometry. In the upper part, the results of lane-width computations are presented with different shapes based on the computation error whereas, in the lower part, the results of lateral offsets are presented in different colors (magenta (cyan) for left (right) lateral offset).

While conducting this experiment, we had no means, unfortunately, of measuring the true lateral offsets. One way we could possibly measure the performance of lateral offset computation is to look into the accuracy of the lane-width computation. This is because the error of the lateral offset computation is basically a sum of its own error and that of the lane-width computation.

To evaluate the performance of our system's lane-changing maneuver detection, we also recorded several hours of videos on different dates that included highway and city drivings. We manually identified 33 lane-changing maneuvers. We could also measure the performance of our system's metric

computation from this data, but only looked at these manually identified maneuvers. For the  $k$  and  $l$ , which are the parameters for the temporal window of observing the closest lane-marking, we found 20 and 5 worked best. Our system was able to detect 27 out of 33 lane-changing maneuvers, resulting in a recall rate of  $(27/33 =) 0.81$ . Twice the system incorrectly produced outputs, resulting in a precision rate of  $(27/29 =) 0.93$ .

#### IV. CONCLUSIONS AND FUTURE WORK

This paper has presented a computer vision system that analyzes a stream of perspective images from a forward-looking camera to acquire information about a ground vehicle's lateral movements. The outputs include the information about the geometric structure of the host roadway such as the number of road-lanes, the index of the host road-lane, and the width of host road-lane in meters. These pieces of information enabled us to determine the lateral distances of our vehicle from the left and right boundaries of the host road-lane in meters and whether our vehicle crossed any road-lane boundaries. From the actual road-tests, we found our system showed stable and reliable performance in computing lateral distance and reasonable performance in detecting lane-changing maneuvers.

As future work, we would like to investigate whether a Bayes filter would help improve the current implementation, which analyzes image frames individually in order to understand the geometric structure analysis of the host roadway. For the lane-markings' color classification, we learned, under a batch mode, the color model from a set of manually labeled color samples and used the model for the classification. The learned model is biased to the sample data and may result, when the color distribution of testing data is significantly different, in unacceptable performance. To find a remedy to this problem, we also would like to investigate whether an incremental update of the color model would help improve the performance of the color classification.

#### V. ACKNOWLEDGMENTS

The author would like to thank Dr. Myung Hwangbo for his helps and fruitful discussions on 3D geometry.

## REFERENCES

- [1] Mohamed Aly, Real time detection of lane markers in urban streets, In *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 7-12, 2008.
- [2] Nicholas Apostoloff and Alexander Zelinsky, Vision in and out of vehicles: integrated driver and road scene monitoring, *The International Journal of Robotics Research*, 23(4-5): 513-538, 2004.
- [3] Itay Gat, Meny Benady, and Amnon Shashua, A monocular vision advance warning system for the automotive aftermarket, In *Proceedings of SAE 2005 World Congress and Exhibition*, 2005.
- [4] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [5] Aharon Bar Hillel, Dan Levi, and Guy Raz, Recent progress in road and lane detection: a survey, *Machine Vision and Applications*, 2012.
- [6] Albert S. Huang, David Moore, Matthew Antone, Edwin Olson, and Seth Teller, Finding multiple lanes in urban road networks with vision and lidar, *Journal of Autonomous Robots*, 26(2-3): 103-122, 2009.
- [7] Myung Hwangbo, Vision-based navigation for a small fixed-wing airplane in urban environment, *Tech Report CMU-RI-TR-12-11*, PhD Thesis, The Robotics Institute, Carnegie Mellon University, 2012.
- [8] P. Kahn and L. Kitchen and E.M. Riseman, A fast line finder for vision-guided robot navigation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11): 1098-1102, 1990.
- [9] ZuWhan Kim, Robust lane detection and tracking in challenging scenarios, *IEEE Transactions on Intelligent Transportation Systems*, 9(1): 16-26, 2008.
- [10] Jorg Kibbel, Winfried Justus, and Kay Furstenberg, Lane estimation and departure warning using multilayer laserscanner, In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, pp. 777-781, 2005.
- [11] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun, Map-based precision vehicle localization in urban environments, In *Proceedings of Robotics Science and Systems*, 2007.
- [12] Joel C. McCall and Mohan M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, *IEEE Transactions on Intelligent Transportation Systems*, 7(1): 20-37, 2006.
- [13] U.S. Department of Transportation, Federal Highway Administration, *Manual on uniform traffic control devices for streets and highways*, <http://mutcd.fhwa.dot.gov/>, 2009 Edition.
- [14] U.S. Department of Transportation, Federal Highway Administration, *Highway Statistics*, <http://www.fhwa.dot.gov/policy/ohim/hs06/index.htm>, 2006.
- [15] Michael Montemerlo et al., Junior: the Stanford entry in urban challenge, *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge, Part II*, 25(9): 569-597, 2008.
- [16] Marcos Nieto, Jon Arrospe Laborda, and Luis Salgado, Road environment modeling using robust perspective analysis and recursive Bayesian segmentation, *Machine Vision and Applications*, 22:927-945, 2011.
- [17] Takashi Ogawa and Kiyokazu Takagi, Lane recognition using on-vehicle lidar, In *Proceedings of IEEE Intelligent Vehicle Symposiums (IV-06)*, pp. 540-545, 2006.
- [18] John C. Russ, *The Image Processing Handbook*, CRC Press, 2011.
- [19] Sebastian Scherer, Lyle Chamberlain, and Sanjiv Singh, Autonomous landing at unprepared sites by a full-scale helicopter, *Journal of Robotics and Autonomous Systems*, 60(12): 1545-1562, 2012.
- [20] Chris Urmson et al., Autonomous driving in urban environments: Boss and the Urban Challenge, *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge, Part I*, 25(8): 425-466, 2008.

# Object-Level View Image Retrieval via Bag-of-Bounding-Boxes

Ando Masatoshi   Chokushi Yuuto   Inagaki Yousuke   Hanada Shogo   Tanaka Kanji

**Abstract**—We propose a novel bag-of-words (BoW) framework to build and retrieve a compact database of view images, toward robotic localization, mapping and SLAM applications. Our method does not explain an image by many small local features (e.g. bag-of-SIFT-features) as most previous methods do. Instead, the proposed bag-of-bounding-boxes (BoBB) approach attempts to explain an image by fewer larger object patterns, which leads to a semantic and compact image descriptor. To make the view retrieval system more practical and autonomous, the object patterns are discovered in an unsupervised manner, via common pattern discovery (CPD) between the input and a known reference images, which does not require pre-trained object detector. Moreover, our CPD task does not rely on good image segmentation and can handle scale variations, exploiting the recently developed CPD technique, spatial random partition. By exploiting traditional bounding box -based object annotation and knowledge transfer, we compactly describe an image in a form of bag-of-bounding-boxes (BoBB). With a slightly modified inverted file system, we efficiently index/search the BoBB descriptors. Experiments with publicly available “RobotCar” dataset show that the proposed method achieves accurate object-level view image retrieval with significantly compact image descriptors, e.g. 20 words per image.

## I. INTRODUCTION

View image retrieval on *compact* database of view images is a fundamental building block for robotic localization, mapping and SLAM systems [1]–[3]. Applications include large scale maps and information sharing, where the spatial cost for storage [1], [2] and information transfer [3] of view database becomes critical issue. One of best known ways to address this problem is the popular bag-of-visual-features (BoVF) [4]–[7], which was originally inspired by the traditional bag-of-words (BoW) model from text information retrieval, and where the indexing (or retrieval) process proceeds as follows:

- 1) extract local visual features from an input database (or query) view image;
- 2) translate the features into visual words using a feature dictionary;
- 3) index (or exact search) the inverted file system using the visual words.

Our approach proposed in this paper also follows a similar pipeline consisting of three steps 1)-2)-3), but it does not explain an image by *many small local features* (e.g. bag-of-SIFT-features) as most BoVF frameworks do. Instead, we attempt to explain an image by *fewer larger object patterns*, which leads to a semantic and compact image descriptor.

This work was partially supported by MECSST Grant (23700229, 30325899), by KURATA grants and by TATEISI Science And Technology Foundation.

The authors are with Graduate School of Engineering, University of Fukui, Japan. [tnkknj@u-fukui.ac.jp](mailto:tnkknj@u-fukui.ac.jp)

This study is motivated by recent success in object-level correspondence techniques (e.g. co-segmentation) for common pattern discovery, i.e. mining common object patterns across images [8]–[11]. A known limitation of feature-level correspondence techniques (e.g. BoVF) is that they are largely influenced by the extracted features, and cannot exploit further information beyond the detected features, whose size and shape are typically small and must be defined prior to the feature extraction (i.e. 1st) stage. To counter this, different lines of researches on object-level correspondence, including common pattern discovery [8], co-segmentation [9], subimage search [10], and visual phrase [11] have been developed. By simultaneously looking at a pair of images, those techniques attempt to find larger object-level correspondences based on the fact that true correspondences are supported by larger object region than false ones.

We are particularly inspired by the spatial random partition (SRP), a common pattern discovery (CPD) technique originally proposed in [12] and recently developed in [11], where an input image is characterized by a pool of overlapping subimages randomly sampled from it. For CPD, each subimage is queried and matched against the subimage pool, based on the fact that a common pattern is likely to be present in a good number of subimages across different images. From our viewpoint of object-level view retrieval, SRP has several desirable properties: 1) It does not rely on good image segmentation techniques; 2) It does not require a priori knowledge on how many common object patterns exist in the input views; 3) It does not rely on quantization of visual features; and 4) It is able to handle scale variations of the object. Our proposed approach is designed to leverage those desirable properties of SRP.

In this paper, we focus on use of the object-level correspondence techniques within the general BoW framework. Accordingly, our indexing (or retrieval) process is slightly different from that of the BoVF framework, and proceeds as follows (Fig.1):

- 1) extract *object patterns that well explain an input image from a known reference image*;
- 2) translate the *object patterns discovered* to visual words;
- 3) index (or *similarity search*) the inverted file system using the visual words.

Following the BoW literature, the 1st and 2nd stages for database images are done in offline and ready for parallelization and large-scale view retrieval. At the 1st stage, a known reference image is simply used as a view dictionary, in contrast to the pre-learned feature dictionary used by traditional BoVF frameworks.



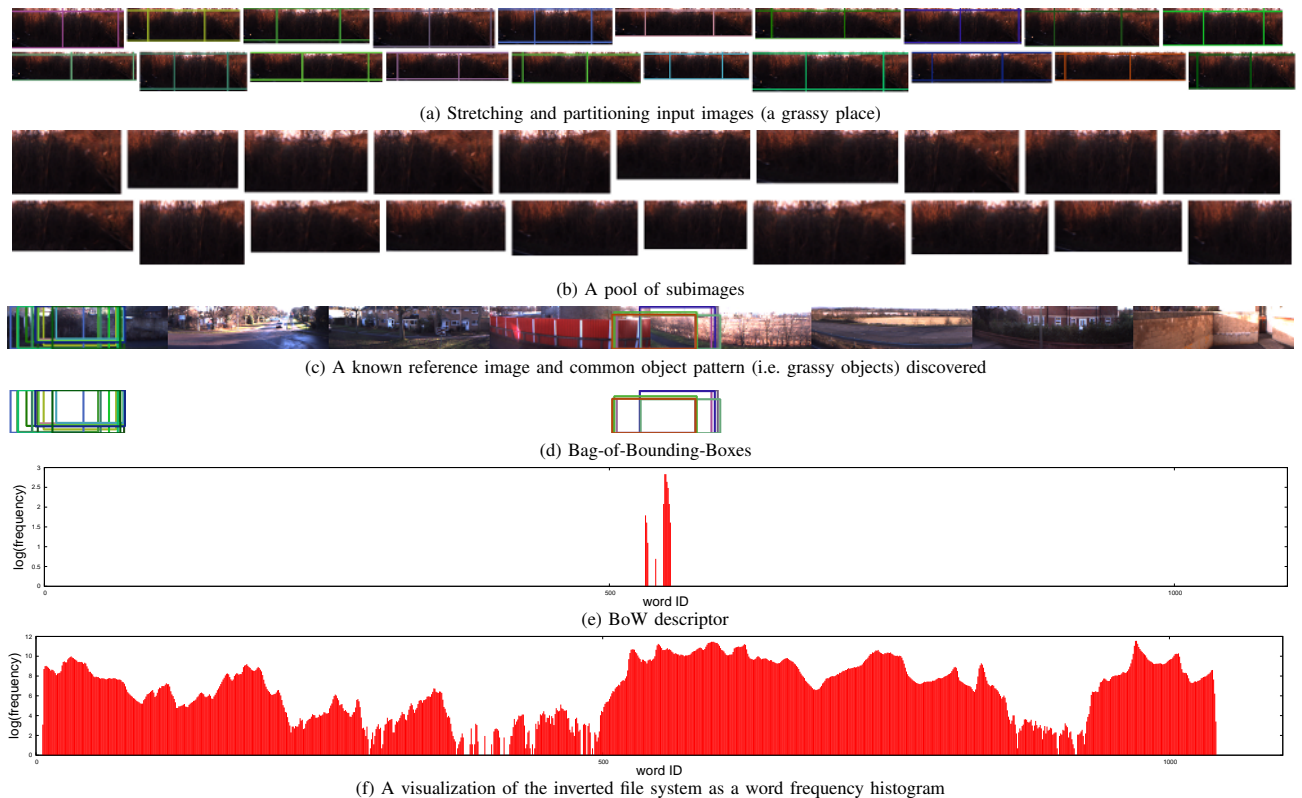


Fig. 1. Our BoW pipeline. An input image is stretched vertically and horizontally as shown in (a), and randomly partitioned into a pool of subimages (b). The subimages are matched between the input image and a known reference image, based on the fact that a common pattern is likely to be present in a good number of subimages across different images (c). The resulted bag-of-bounding-boxes (d) is used as a compact BoW descriptor (e) for indexing and retrieving the inverted file system (f).

There are five key properties about the proposed approach:

- An image is *semantically* characterized by object-level information, in contrast to feature-level image characterization in existing BoW frameworks;
- The object pattern discovery process is *unsupervised*, without requiring manually labeled examples and/or pre-trained object detector;
- An image is *compactly* described in a form of bag-of-bounding-boxes (BoBB), employing traditional BB-based object annotation and knowledge transfer [13];
- The BoBB framework inherits the *efficiency* in indexing and retrieval from the general BoW framework by using a slightly modified inverted file system;
- The BoBB framework leverages the state-of-the-art CPD technique, spatial random partition (SRP) [11], which has desirable properties as aforementioned.

Experiments with publicly available “RobotCar” dataset [1] show that the proposed approach achieves accurate object-level view image retrieval using significantly compact description of view images, e.g. 20 words per image.

## II. RELATION TO OTHER WORK

Image retrieval in a large number of images has recently received increasing attention [1], [7], [14]–[22]. Previous studies have dealt with various aspect of the BoW framework, including the quantization method and its speed [1], [7], [14], the post processing based on a global spatial geometric verification [15], the matching distance of descriptors

[16], and with various types of visual features including local feature (e.g. SIFT, SURF), global feature (e.g. GIST), filter bank (e.g. color, texture, object), and other feature modeling techniques. While most of the above systems work on large image databases, several efforts also focused on *compactness* of the image database [19]–[22]. [21] has improved the memory usage per image introducing a method for projecting the BoW vectors onto a set of pre-defined sparse projection functions. In [22], we also employed the BoW projection technique and used it within a multi-cue BoW framework for scalable scene retrieval applications. However, almost all of those efforts to compact view database focused on feature-level correspondence, and little study has attempted on the object-level correspondence, as we propose to do in this study.

The problem of object retrieval, whose goal is to accurately locate the target object in image collections [11], is clearly different from our view retrieval problem. Object retrieval is a challenging task due to the fact that the target object usually occupy only a small portion of an image with cluttered background, and can differ significantly from the query in scale, orientation, viewpoint and in color. One of most effective ways to address this problem is the use of spatial context [11], where the input images are partitioned into small subimages and then matched against one another based on the fact that a common object pattern is likely to co-exist in a good number of subimages across different images. However, existing works focus on object retrieval

tasks, and often concerned with setting where feature-based inference is possible, e.g. demanding rich features for geometric verification. From our view retrieval standpoint, the object retrieval approaches would waste a large amount of memory resource to index those individual objects, and not suited for our objective, i.e. compact description of view images.

Although object-level scene representation is a central importance in robotic mapping, localization and SLAM [23], existing efforts to compact the view image database focus on feature-level approaches relying on dimension reduction techniques. [24] developed a self-localization system by combining the SIFT feature descriptor with principal component analysis (PCA) dimension reduction techniques, and achieved accurate track of the position of a robot in a real environment. Many efforts have also been made on various types of feature descriptors and advanced dimension reduction techniques [1], [2], [25]–[28]. In our previous papers, we also have developed localization systems exploiting dimension reduction techniques, including locality sensitive hashing (LSH) [26], semantic hashing (SH) [27], and compact projection (CP) [28]. In contrast, our current paper focuses on an object-based scene characterization.

The problem of common pattern discovery (CPD), multiple objects co-segmentation, or co-recognition, which aims at automatic discovery of common object patterns across images is an active and open research issue [8], [29]–[31]. Because no prior knowledge is available on the common object patterns, this task is very challenging, and much more difficult than traditional tasks such as detection and retrieval of object patterns, since the search space (e.g. appearance, size, shape, number of objects) is enormous. The existing solutions include earth mover’s distance (EMD) [8] and other model learning techniques, co-segmentation [29], and correspondence growing [30]. In [31], we also have developed a CPD technique in a form of correspondence growing algorithm by employing a probabilistic MCMC framework. However, improving existing common pattern discovery techniques is not the objective of our current paper. In this study, we focus on *use* of common pattern discovery as a method for the object-level image characterization within the general BoW framework.

### III. BAG-OF-BOUNDING-BOXES (BOBB) FRAMEWORK

The proposed BoBB framework is slightly different from the BoVF framework in three important aspects: 1) definition of visual word; 2) representation of dictionary; and 3) search criteria. We describe the basic idea behind each of them in the following, and then explain the BoBB framework step-by-step in subsections III-A, III-B, III-C, III-D, and III-E.

First, we define a visual word as a common object pattern that well explains an input query/database image, discovered from a known reference image via common pattern discovery (CPD). Accordingly, our visual word extraction process becomes an iteration of the CPD between an input and the reference images, which consists of hypothesization and verification of common object patterns: 1) Each iteration

begins by randomly stretching and shrinking each image to deal with variations of scale, viewpoint and occlusions (Fig.1a); 2) For the hypothesization, inspired by the spatial random partition technique [11], a pool of subimages are randomly sampled from both images and each pair of subimages is used as a hypothesis of common object pattern (Fig.1b); 3) For the verification, correspondence between the subimage pair is verified (Fig.1c) by using any type of correspondence measure (e.g. EMD [8], multiple objects co-segmentation [29], correspondence growing [30]); In this paper’s experiments, the normalized image correlation will be used as the correspondence measure; 4) Common object patterns discovered are compactly described in a form of bag-of-bounding-boxes (Fig.1d), employing traditional bounding box -based object annotation and knowledge transfer [13].

Second, we use a known reference image as a view dictionary. This is in contrast to the feature dictionary used by the BoVF framework for dimension reduction or quantization of visual features. To make the view retrieval more practical and autonomous, we do not assume any special indexing architecture for the dictionary, such as ImageNet. Instead, our view dictionary consists of raw images (e.g. JPEG images) being acquired by the robot-self or shared via distributed robot networks, without supervised categorization. Although a dictionary for the BoW framework in general should be designed to contain visual words that are frequently used [4], it is beyond the scope of this paper to discuss such an optimal design or adaptive learning of the dictionary image. In this paper’s experiments, we will simply use dictionary images consisting of 8-64 raw images, as shown in Fig.3.

Third, our search criteria is based on similarity search, in contrast to the exact search (either single- or multi- probe strategy [21]) commonly used by the BoVF framework. Because our visual word is defined as a bounding box with its pose and shape attributes, the similarity used for search criteria is designed to evaluate similarity of those attributes. In the current paper, the area of overlap between bounding boxes will be simply used as the similarity measure.

#### A. Problem: View Image Retrieval

The goal of view image retrieval is to retrieve images similar to a given query image  $I^Q$  by comparing the query image  $I^Q$  and each image  $I^D$  in the image database  $D = \{I^D\}$ , given a reference image  $R$  and an object-level correspondence measure  $S$ .

#### B. Common Pattern Discovery

Unlike the BoVF framework, the database building process consists of an iteration of the common pattern discovery between an input  $I$  and the reference  $R$  images, which proceeds as follows:

- 1) randomly partition the input and the reference images  $I$  and  $R$  for multiple times, and obtain a pool of overlapping subimages  $\{I_k\}$  and  $\{R_k\}$  (Fig.1a,b);
- 2) evaluate the likelihood of each subimage pair  $(I_k, R_k)$  being a match pair by using the correspondence measure  $S$ ;



- 3) rank all the subimage pairs in descending order of the likelihood score;
- 4) select a set  $\{(I_k, R_k)\}_{k=1}^T$  of  $T$  top ranked subimage pairs as common object patterns (Fig.1c).

Currently, the likelihood at the step 2 is evaluated by comparing geometry and appearance between the subimage pair. More formally, height and width of bounding box is compared between the subimage pair, and if width or height of taller bounding box does not exceed a pre-defined ratio  $(1+r)$  than shorter box, the subimage pair is viewed as a potential match, and then, the likelihood for such a potential match is evaluated by the given correspondence measure:  $S(I_k, R_k)$ .

### C. Visual Word Extraction

We compute a bounding box for each of the regions of the  $T$  common object patterns output by the above process, and represent it by the coordinates  $x_{\min}, x_{\max}, y_{\min}, y_{\max}$  (Fig.1d). The pose  $(x_{\min}, y_{\min})$  and the shape  $(w, h)$  of each bounding box, where  $2w$  and  $2h$  respectively represent the width and height of the box, is computed and the 4D parameter  $(x_{\min}, y_{\min}, w, h)$  is mapped to the visual word.

### D. Indexing

The procedure for indexing the inverted file system given bag-of-bounding-boxes (BoBB) descriptors is straightforward. The BoBB descriptor is represented by a 4D parameter and transformed to a 1D visual word. Since we have to store one entry for each bounding box existing in the pool [21], each input image requires space linear to the number of bounding boxes (i.e. visual words) per image.

### E. Similarity Search

Given a query image  $I^Q$ , the similarity search process aims to retrieve and score images in the database  $D$ , and proceeds in the following steps:

- 1) extract a bag-of-bounding-boxes  $\{I_i^Q\}$  from the query image  $I^Q$  in the same manner as in III-B, III-C (Fig.1d);
- 2) For each query BB  $I_i^Q = (x, y, w, h)$ ,
  - a) retrieve images  $D_i (\subset D)$  whose BB can overlap with  $I_i^Q$  or belongs to the following area

$$Z = [x - (2+r)w, x + (2+r)w] \times [y - (2+r)h, y + (2+r)h] \\ \times [w/(1+r), w(1+r)] \times [h/(1+r), h(1+r)],$$

- in the BB parameter space;
- b) evaluate similarity between every pair of BBs from  $I^Q$  and each  $I_j^{D_i} (\in D_i)$ , according to the area of overlap between BBs;
- 3) compute the aggregate score  $v^j$  for each of the retrieved database images  $D_i$ , while set score  $v^j = 0$  for those database images that are not retrieved;
- 4) Rank all the database images in descending order of the aggregate score  $v^j$ .

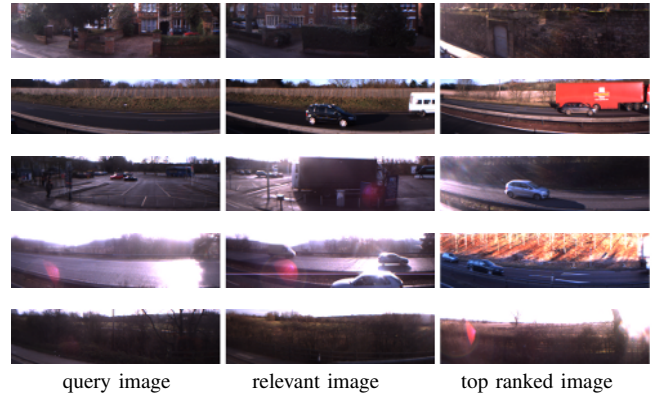


Fig. 2. Input images and retrieval results for 5 different retrieval tasks. In each of them, the retrieval was successful and the relevant images are assigned high ANR rankings [%], 3, 3, 9, 4 and 2, respectively.

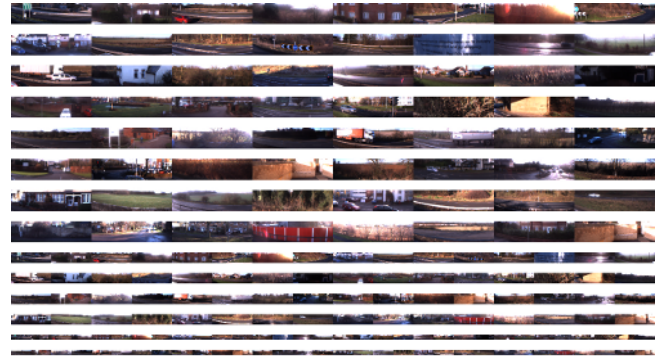


Fig. 3. Reference images. From top to bottom, images named “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “12”, “34”, “56”, “78”, “1234”, “5678”, “12345678” are shown (zoom in for detail).

## IV. EXPERIMENTS

We conducted view retrieval experiments by utilizing the “RobotCar” dataset provided by the authors of [1] (“FAB-MAP 2.0”). The original dataset consists of GPS, stereo, and omni-directional image data acquired by a car robot during its driving 1,000km in outdoor environments. An omni-directional image consists of 5 images from each of the five side-facing cameras #1-#5 of Ladybug cameras mounted on the robot car. For view retrieval experiments, we chose images from the camera#1 which is directed to the right and use image data within a rectangular region,  $y \in [100, 250]$ , and GPS data for ground truth. We expect the input images to be contaminated by variations in viewpoints, illumination and partial occlusions. To counter this, each view is represented by a small set of 10 frames sampled from a short frame sequence, and similarity between a given view pair is defined as the average of the  $10 \times 10$  frame pairs from the view pair. Each retrieval experiment uses independent dataset, which consists of a query view and a size  $N = 100$  view database. Each database consists of one relevant view and a set of random  $(N - 1)$  distracter views which do not overlap with either query or relevant view. We utilize the information of loop closing provided as a part of the “RobotCar” dataset, and use each of the beginning and the ending of a loop respectively as a query and a relevant views. The resulted datasets consist of images

TABLE I

ANR PERFORMANCE OF DIFFERENT BOW FRAMEWORKS

Dataset name	ANR(%)
“RobotCar” 10K words BoVF vocabulary	42.59
1K words BoVF vocabulary (20 words per image)	41.85
BoBB vocabulary (20 words per image)	<b>35.38</b>

TABLE II

INFLUENCE OF SUBIMAGE PROPERTIES.

			crop			
			w: 0.5		w: 0.9	
			h: 0.5	h: 0.9	h: 0.5	h: 0.9
scale	w: 1.0	h: 1.0	38.73	36.19	41.22	44.41
		h: 1.25	39.14	36.66	38.44	44.60
	w: 1.25	h: 1.0	38.72	<b>35.38</b>	42.01	45.35
		h: 1.25	40.34	36.89	40.56	44.13

taken of the same scene from different viewpoints, and the appearance variations are attributed to various factors including viewpoint change, illumination change, occlusion, which makes the view retrieval tasks challenging.

For performance evaluation, we use the averaged normalized rank (ANR) [4] as performance measure. The normalized correlation is used as the correspondence measure  $S$ . The number of common patterns per image is set  $T = 20$  as default. The size of bounding box for a size  $w \times h$  input image is set  $(w \cdot \text{crop.w}) \times (h \cdot \text{crop.h})$ , where  $\text{crop.w} = 0.5$  and  $\text{crop.h} = 0.9$  are used as default. The scaling factor for shrinking/stretching bounding boxes along the horizontal and vertical directions are respectively set  $\text{scale.w} = 1.25$  and  $\text{scale.h} = 1.0$  in default. The default reference image is constructed by appending 8 images sampled from the image set, each of which does not overlap with any query or database image, and shown as “1” in Fig.3.

Although our BoBB (i.e. object-level) framework is complementary to existing BoVF (i.e. feature-level) frameworks, for the sake of evaluation, the BoVF framework was also implemented and compared with the proposed BoBB framework. In this study, two types of BoVF view retrieval systems were developed. One is based on the BoVF data provided as a part of the “RobotCar” dataset. We weighted the original BoVF vectors with standard TF-IDF weighting scheme, then index and retrieve the view database using an inverted file system, and evaluate the performance using the ANR measure. From our standpoint, a major inconvenience of the above publicly available BoVF data is that its vocabulary is learned from images acquired by the whole omni-directional camera, i.e. not the camera#1 we use. We hence constructed another independent BoVF data which is learned from the images acquired by the camera#1. A set of training images that are independent from the query and the database images are randomly sampled from the entire image set, and for each training image, a bag of SIFT feature vectors are extracted at keypoints extracted by a grid sampling technique, and then quantized into a bag of visual words using the approximated k-means (AKM) quantization technique in [15]. In this study, we set the vocabulary size to 1K words. Table I reports the ANR performance comparing the proposed BoBB framework with the other two BoVF frameworks. The BoVF frameworks on our view retrieval problem is not as impressive as we ex-

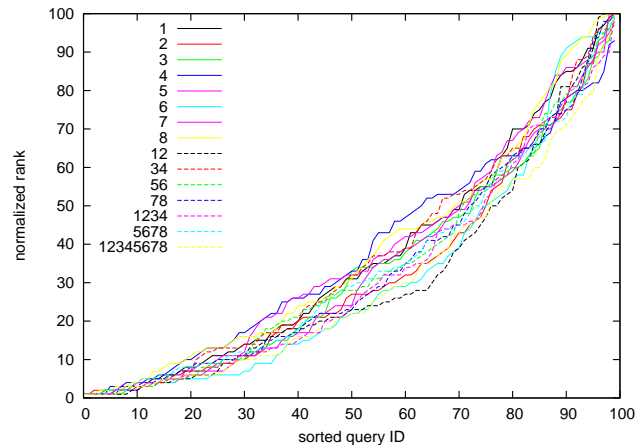


Fig. 4. ANR performance for different reference images.

pected. This is because of that in the current dataset, matched objects often occupy only a small portion of an image, which are very difficult to be identified (Fig.2). Although the object retrieval techniques (e.g. geometric verification) have been used to counter this problem in literature, they require many words per image, and not suited for compact view database as discussed in II. In contrast, our BoBB framework based on compact description of object patterns achieves much better retrieval performance with requiring only 20 words per image.

So far, the experiments have focused on the case where an input image is characterized by a pool of *small* subimages. To evaluate effectiveness of this strategy, we also implemented an alternative strategy where the pool of subimages consists of *big* subimages almost the same size (e.g. 90%) as the input image, and compare it with the proposed strategy. In this study, we evaluate 16 different cases (crop.w, crop.h, scale.w, scale.h) =  $\{0.5, 0.9\} \times \{0.5, 0.9\} \times \{0.5, 0.9\} \times \{0.5, 0.9\}$ . Tab II reports the comparison results. It can be seen that the strategies with  $\text{crop.w}=0.5$  clearly outperform the ones with  $\text{crop.w}=0.9$ . This is due to the fact in the current car robot applications, there are large variations in the viewpoint particularly in the horizontal direction, and setting the parameter to a small value  $\text{crop.w}=0.5$  allows the robot to adaptively learn the size and pose of the subimages according to those variations.

One of key properties of the proposed BoBB framework is that the BoBB image descriptor is strongly dependent on the choice of the reference image that is used for common pattern discovery. We are particularly interested in understanding the impact of the choice of the reference image on the retrieval performance. Thus, we further conducted series of independent retrieval experiments using 15 different reference images, which is created from 8 reference images with the same size shown as “1”-“8” in Fig.3 by appending horizontally a pair of images (e.g. “1234” is an append of a pair of “12” and “34”), as shown in Fig.3. The graph in Fig.4 reports the ANR performance for each of the 15 reference images, where the vertical axis is the normalized rank [%] and the horizontal axis is the sorted query ID [%]. It can be seen that the proposed BoBB framework is

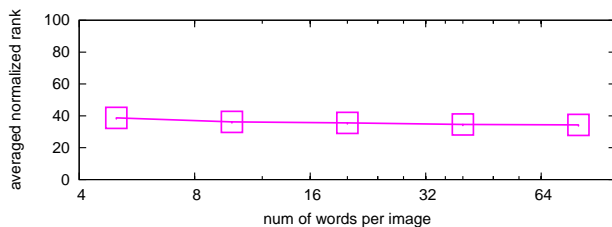


Fig. 5. ANR performance vs. num of words per image.

stable and successful for almost all the reference images used in this study. However, since our algorithm is designed to represent an input image by a pool of cropped reference images, our algorithm would not be suitable for general cases where whole regions of the input image is dissimilar from the reference image, e.g. obviously not suited for the case where the reference image is indoor. In the future we shall study a way for automatically choosing the reference images adaptively for a given set of database images.

To investigate the relationship between the number of words per image and the retrieval performance, we conducted additional retrieval experiments, using different number of words per image, 5, 10, 20, 40, and 80. For each case, we got the ANR performance 38.76, 36.19, 35.58, 34.61, 34.33, as summarized in Fig.5. The large number of words per image was used, the better was the ANR performance. However, increasing the number of words per image requires larger number of entries per image and decrease the compactness of the image database, thus there is a tradeoff between compactness and retrieval performance. In future, we would like to explore methods to improve this tradeoff.

## V. CONCLUSIONS

We proposed a novel BoW approach, bag-of-bounding-boxes (BoBB), to build and retrieve a compact view image database, which is characterized by (1) *semantic* object-level image characterization, (2) *unsupervised* scene modeling, (3) *compact* view image descriptor, (4) *efficient* indexing and retrieval, and (5) the state-of-the-art CPD techniques. Experiments on challenging outdoor datasets show that our framework is insensitive to system parameters and robust to variations in the viewpoint, contaminations of images by noise, color and partial occlusions. Future work will explore the optimization and adaptive learning of dictionary image for unseen environments and compact the description of view images which are enabled by the proposed bag-of-bounding-boxes framework.

## REFERENCES

- [1] Mark Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics: Science and Systems*, 2009.
- [2] William P. Maddern, Michael Milford, and Gordon Wyeth. Capping computation time and storage requirements for appearance-based localization with cat-slam. In *ICRA*, pages 822–827, 2012.
- [3] Andreas Wendel, Michael Maurer, Gottfried Graber, Thomas Pock, and Horst Bischof. Dense reconstruction on-the-fly. In *CVPR*, pages 1450–1457, 2012.
- [4] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, pages 127–144, 2006.

- [5] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.
- [6] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *CVPR*, pages 257–263, 2003.
- [7] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.
- [8] Hung-Khoon Tan and Chong-Wah Ngo. Common pattern discovery using earth mover’s distance and local flow maximization. In *ICCV*, pages 1222–1229, 2005.
- [9] Dorit S. Hochbaum and Vikas Singh. An efficient algorithm for cosegmentation. In *ICCV*, pages 269–276, 2009.
- [10] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.
- [11] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Randomized visual phrases for object search. In *CVPR*, pages 3100–3107, 2012.
- [12] Junsong Yuan and Ying Wu. Spatial random partition for common visual pattern discovery. In *ICCV*, pages 1–8, 2010.
- [13] Matthieu Guillaumin and Vittorio Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, pages 3202–3209, 2012.
- [14] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.
- [15] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.
- [16] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008.
- [17] Li-Jia Li, Hao Su, Eric P. Xing, and Fei-Fei Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010.
- [18] Herve Jegou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *CVPR*, pages 1169–1176, 2009.
- [19] Ondrej Chum, James Philbin, Michael Isard, and Andrew Zisserman. Scalable near identical image and shot detection. In *CIVR*, pages 549–556, 2007.
- [20] Ondrej Chum, James Philbin, and Andrew Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008.
- [21] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Packing bag-of-features. In *ICCV*, pages 2357–2364, 2009.
- [22] Kanji Tanaka and Kensuke Kondo. Multi-scale bag-of-features for scalable map retrieval. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, pages 793–799, 2013.
- [23] Javier Civera, Dorian Gálvez-López, Luis Riazuelo, Juan D. Tardós, and J. M. M. Montiel. Towards semantic slam using a monocular camera. In *IROS*, pages 1277–1284, 2011.
- [24] Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard, and Sven Behnke. Metric localization with scale-invariant visual features using a single perspective camera. In *EUROS*, pages 195–209, 2006.
- [25] Maurice F. Fallon, Hordur Johannsson, and John J. Leonard. Efficient scene simulation for robust monte carlo localization using an rgb-d camera. In *ICRA*, pages 1663–1670, 2012.
- [26] Kanji Tanaka and Eiji Kondo. A scalable algorithm for monte carlo localization using an incremental e<sup>2</sup>lsh-database of high dimensional features. In *ICRA*, pages 2784–2791, 2008.
- [27] Kouichirou Ikeda and Kanji Tanaka. Visual robot localization using compact binary landmarks. In *ICRA*, pages 4397–4403, 2010.
- [28] Tomomi Nagasaka and Kanji Tanaka. An incremental scheme for dictionary-based compressive slam. In *IROS*, pages 872–879, 2011.
- [29] Gunhee Kim, Eric P. Xing, Fei-Fei Li, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, pages 169–176, 2011.
- [30] Jan Cech, Jiri Matas, and Michal Perdoch. Efficient sequential correspondence selection by cosegmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1568–1581, 2010.
- [31] Yuuto Chokushi, Kanji Tanaka, and Masatoshi Ando. Common landmark discovery in urban scenes. *IAPR Int. Conf. Machine Vision Applications*, 2013.

# Cart-O-matic project\* : autonomous and collaborative multi-robot localization, exploration and mapping.

Antoine Bautin<sup>1</sup>, Philippe Lucidarme<sup>2</sup>, Remy Guyonneau<sup>2</sup>, Olivier Simonin<sup>1</sup>,  
Sebastien Lagrange<sup>2</sup>, Nicolas Delanoue<sup>2</sup> and Francois Charpillet<sup>1</sup>

**Abstract**—The aim of the Cart-O-matic project was to design and build a multi-robot system able to autonomously map an unknown building. This work has been done in the framework of a French robotics contest called Defi CAROTTE organized by the General Delegation for Armaments (DGA) and the French National Research Agency (ANR). The scientific issues of this project deal with Simultaneous Localization And Mapping (SLAM), multi-robot collaboration and object recognition. In this paper, we will mainly focussed on the two first topics : after a general introduction, we will briefly describe the innovative simultaneous localization and mapping algorithm used during the competition. We will next explain how this algorithm can deal with multi-robots systems and 3D mapping. The next part of the paper will be dedicated to the multi-robot path-planning and exploration strategy. The last section will illustrate the results with 2D and 3D maps, collaborative exploration strategies and example of planned trajectories.

## I. INTRODUCTION

Localization and mapping become the basis of many mobile robotics systems. Vacuum cleaners and mowers are great illustrations in tune with the times. Such systems may also be of prime interest for defence, military applications and rescue [9]. In 2008, the french research agency (ANR) and the General Delegation for Armaments (DGA) launched a robotics challenge called CAROTTE (CARTographie par un ROBoT d'un TEritoire - Autonomous mapping of an area with a robot). Five teams ([15], [5] and [12]) have been selected and founded to participate in this challenge organized as a robotics competition similar to [13]. Each team had to design and build an autonomous grounded robotics system able to map a planar stage of a building in less than 30 minutes. The system must output at the end of the run the following data :

- a 2D map of the building,
- a 3D map of the building,
- a topological map of the building,
- location and type of walls,
- location and classification of objects.

Three events were organized in 2010, 2011 and 2012 and the results of the five teams were scientifically measured and compared. Unfortunately the results of the comparison

\*This work was partially supported by the French National Research Agency (ANR) and General Delegation for Armaments (DGA) through the Cart-O-matic project in the CAROTTE challenge.

<sup>1</sup>MAIA Group, INRIA Lorraine, LORIA, Campus scientifique, BP 239, 54506 Vandoeuvre-les-Nancy Cedex, France. `firstname.lastname@loria.fr`

<sup>2</sup>LISA - University of Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France `firstname.lastname@univ-angers.fr`

stay confidential but the rank of each team was published. As the reader probably understood, we were one of the team engaged in the competition. Our system reached the first overall rank during the last evaluation (2012, June) and the aim of this paper is to present and share our solution. Each selected team was specialized in a given topic and the characteristic of our team was to proposed a multi-robot solution (the reader can refer to [8] for previous works). The philosophy behind this approach is the reliability (if a robot encounters a failure, it does not compromise the mission) and the speed improvement of the mission (sharing the area between several robots decreases the exploration time). The first part of this paper is mainly focussed on localization and mapping, the next section will present the multi-robot exploration strategy. An overview of the experimental sets and results will be described and a general conclusion ends the paper.

## II. SLAM-O-MATIC

For such exploration missions, localization and mapping are clearly key items of the development of the architecture. We proposed a novel SLAM algorithm based on scan matching called Slam-O-matic [11]. This algorithm is odometry-free and only requires LIDAR data. It is based on scan matching: the key idea is to find the transformation (two translations ( $\Delta_y$  and  $\Delta_\psi$ ) and one rotation ( $\Delta_\psi$ ) for 2D SLAM) that offers the best matching between the LIDAR data and the known map. The principle of Slam-O-matic is similar to the one used for Hector Slam [10] which is based on the computation of the map derivatives and use the Gauss-Newton algorithm to maximize the matching between scan data and the map. Slam-O-matic does not require the computation of the derivatives and uses the Nelder and Mead algorithm for minimizing the distance between scans and known map ( $C_d$  on Equation 1). Nelder and Mead is a derivative-free optimization algorithm.

SLAM is thus reformulated as an optimization problem where  $\Delta_x$ ,  $\Delta_y$  and  $\Delta_\psi$  are the parameters to optimize. The objective function is the sum of absolute distances between each end-point of the scan and each obstacle of the map is the scalar to minimize:

$$Cd(\Delta_x, \Delta_y, \Delta_\psi) = \sum_{i=1}^n \sqrt{(X_s^i - X_m^i)^2 + (Y_s^i - Y_m^i)^2} \quad (1)$$

where :



- $X_s^i, Y_s^i$  are the coordinates of the  $i^{st}$  point of the scan data according to  $\Delta_x, \Delta_y, \Delta_\psi$ .
- $X_m^i, Y_m^i$  are the coordinates of the closest occupied cell of the map in regard with the  $i^{st}$  point of the scan data.

The environment is represented as a grid map [16], i.e. a value associated to each cell of the map is representing the current estimation of chances of having an obstacle. When the occupancy reaches a given threshold (half of the maximum value in practice) the cell is considered as an obstacle otherwise it is considered as a free space. For each cell of the map considered as an obstacle, the distance to the closest occupied cell is computed as shown on Figure 1. Unlike existing algorithm [1] where the distance is approximated, the Euclidean distance is here pre-computed thanks to a Look-Up-Table.

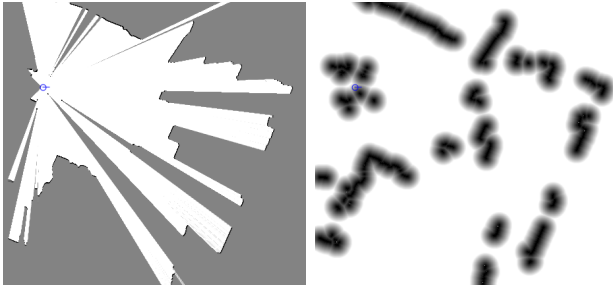


Fig. 1. Occupancy grid map (Left) and associated map with locally computed distances (Right)

When a new LIDAR scan is available the location of each end-point of the scan is located in the maps (occupancy and distances map) based on the assumption that the previous estimated pose is the best known. It becomes thus very simple and fast to compute the cumulated distance between the new scan and the known map (Figure 2). This principle allows a fast estimation of the cumulated distances for any given transformation ( $\Delta_x, \Delta_y$  and  $\Delta_\psi$ ). In other words, this provides a quick numerical evaluation of the mathematical function  $Cd(\Delta_x, \Delta_y, \Delta_\psi)$ .

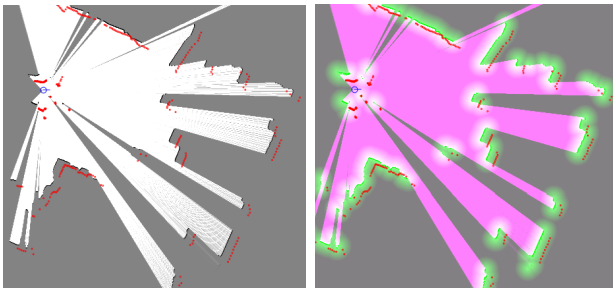


Fig. 2. New scan located in the map (Left) and surimposed maps: occupancy, distances and new scan (Right)

The first intuitive idea is to take advantage of the distance map (that can be seen as a gradient) to find the attractive direction of the transformation as it is done in [1]. Unfortunately, such gradient descent approach needs parameters tuning (size of the steps for example) that may prevent

the algorithm to converge and may be sensitive to initial conditions. We preferred the Nelder and Mead algorithm (also called downhill simplex method) that is based on iterative transformations of a simplex defined in the search space. A map illustration is presented on Figure 3.



Fig. 3. Illustration of a map (63m long building) created without loop closure.

### III. MULTI-ROBOT SLAM

As explained in the introduction our objective is multi-robot exploration that necessarily involves multi-robot SLAM. In 2011, we experimented the following strategy : each robot locally computes the best location for its own current scan data and send to the other robots the result of the optimization (best computed location and scan data). The other robots have no more computation to perform while the optimization has been previously done by the involved robot. They just have to update their maps with the received information. This strategy takes advantage of the multi-robot to perform distributed computation. It also ensures (if we assume there is no communication failure) that all the robots have the same map. Unfortunately, in practice, this solution appeared to be unsatisfactory due to communication problems. With a complete sharing of the information of each robot, the wireless bandwidth quickly saturated due to the amount of sent data.

In 2012, based on our experience, the global map was not computed on line: each robot computes its own local map and stores scan data in memory. At the end of the mission, raw data were sent via a wired network to a central laptop for computing a single map for the whole system. This acts exactly as the solution used in 2011, except that scan data are gathered on a central computer to avoid communication failure.

### IV. RGB-3D MAPPING

As explain in the introduction, a 3D map of the building was requested. Each robot was equipped with a RGB-D sensor that provide RGB images and letter "D" stands for depth image. This combination makes the acquisition of a 3D



colored image as illustrated in Figure 4. A popular solution for realization of such a device is the KINECT™.



Fig. 4. Raw output of the RGB-D camera, and illustration of 3D image forged by combination of depth and color information.

We knew, according to the rules, that the floor was planar in the explored building. Once the robot are located in the 2D map it becomes easy to build a 3D map since the RGB-D sensor of each robot is calibrated before the mission. Calibration consists in estimating the transformation between the LIDAR and the KINECT. First step consists in computing roll and pitch while the robot is resting on a planar floor (ground is used as a reference). Next step consists in estimating yaw in front of a wall : the wall is simultaneously observed by the LIDAR and the RGB-D sensor. A line and a plane are extracted respectively from the scan and 3D data that provides the yaw angle between the LIDAR and the RGB-D sensor. Similar operations are performed for estimating translation parameters. Once sensors are calibrated, 3D data can easily be located according to the 2D map. Figure 13 shows an illustration of the 3D map.

## V. MULTI-ROBOT EXPLORATION

Our multi-robot exploration strategy is frontier-based [17] i.e. the targets assigned to robots are borders between known and unknown cells. The problem consists in assigning a frontier to each robot during the exploration process. The originality of the approach is to favour the distribution of robots among the frontier *directions*. For this purpose, we do not only take into account the distance between robots and frontiers, but we also consider the notion of rank of a robot towards a frontier, by counting how many robots are closer to the frontier than the considered one. By reasoning on ranks instead of distances, two close robots will be assigned on frontiers having distinct directions where they will be in first position whatever the distances. Such an approach tends to separate robots on different directions favouring a well balanced assignation on frontiers.

To cooperate, each robot broadcasts periodically its location and a sub-sampled map of the environment. Each robot autonomously decides its next target when it has reached the previous one. This decision is based on the robot current available information.

To formally define the algorithm, let's introduce the following notations :

- $\mathcal{R}$  the set of robots,  $\mathcal{R} : \{\mathcal{R}_1 \dots \mathcal{R}_n\}$  with  $n = |\mathcal{R}|$  the total number of robots,
- $\mathcal{F}$  the set of frontiers,  $\mathcal{F} : \{\mathcal{F}_1 \dots \mathcal{F}_m\}$  with  $m = |\mathcal{F}|$  the number of frontiers,
- $\mathcal{C}$  a cost matrix with  $\mathcal{C}_{ij}$  the path distance from robot  $\mathcal{R}_i$  to frontier  $\mathcal{F}_j$ ,

- $\mathcal{A}$  an assignment matrix with  $\alpha_{ij} \in [0, 1]$  defined as follows :

$$\alpha_{ij} = \begin{cases} 1 & \text{if robot } \mathcal{R}_i \text{ is assigned to } \mathcal{F}_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let  $RK_{ij}$  be the rank of the robot  $\mathcal{R}_i$  towards the frontier  $\mathcal{F}_j$ .  $RK_{ij}$  is equal to the number of robots which are closer to the frontier than the robot  $\mathcal{R}_i$ . Algorithm 1 formally defines the algorithm, named *MinPos*, processed by each robot for computing its assignment.

---

### Algorithm 1: *MinPos*

---

**Input:**  $\mathcal{C}$  cost matrix

**Output:**  $\alpha_{ij}$  assignment of robot  $\mathcal{R}_i$

**foreach**  $\mathcal{F}_j \in \mathcal{F}$  **do**

$RK_{ij} = \text{Card}(\tilde{\mathcal{R}})$  with  $\tilde{\mathcal{R}} = \{\forall \mathcal{R}_k \in \mathcal{R} \mid \mathcal{C}_{kj} < \mathcal{C}_{ij}\}$

**end**

$j = \text{argmin}_{j \in \mathcal{F}} RK_{ij}$  (If several  $RK_{ij}$  are minimum then choose the one with lowest cost  $\mathcal{C}_{ij}$ )

$\alpha_{ij} = 1$

---

Figure 5 illustrates the exploration with 3 robots in a  $35m^2$  rooms environment. The trajectories of each robot demonstrate the validity and efficiency of the proposed approach, indeed each robot explored a different part of the environment.

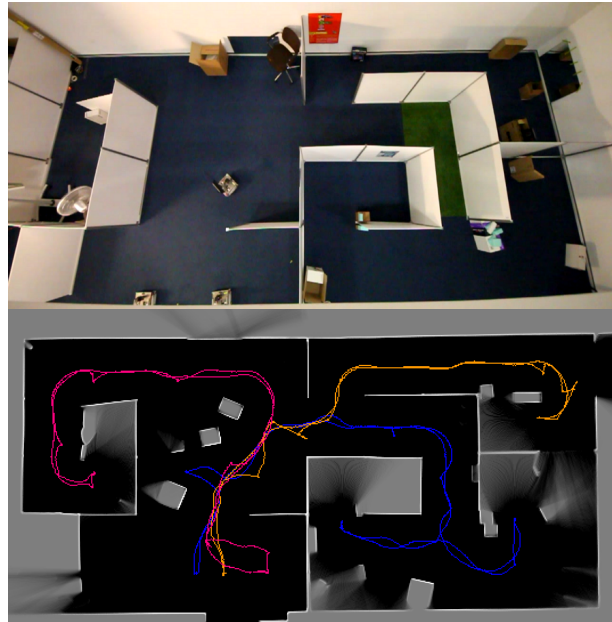


Fig. 5. Photo of the environment and map with trajectories resulting from an exploration with 3 robots.

Simulation results demonstrated that our *MinPos* algorithm outperforms the nearest frontier algorithm [18]. Depending on the environment topology and the number of robots, our algorithm outperforms or gives similar results than

utility greedy algorithm [4]. However, our approach has a lower computational complexity ( $O(nm)$ ) than the greedy algorithm ( $O(n^2m)$ ).

Figure 6 compares the exploration times given in simulation steps of different methods, while varying the number of robots. The methods compared are the nearest frontier algorithm [18], the Burgard et al. greedy-based algorithm [4] and our *MinPos* algorithm, on an hospital section environment. Results shown are an average of 60 runs of each algorithm with a given robot count. We observe that the Burgard et al. and *MinPos* algorithms are more efficient improving by 13% on average the number exploration steps required to fully explore the environment. We improve the greedy approach when the number of robots is low, as *MinPos* forces a well balanced spatial distribution. Details can be found in [3].

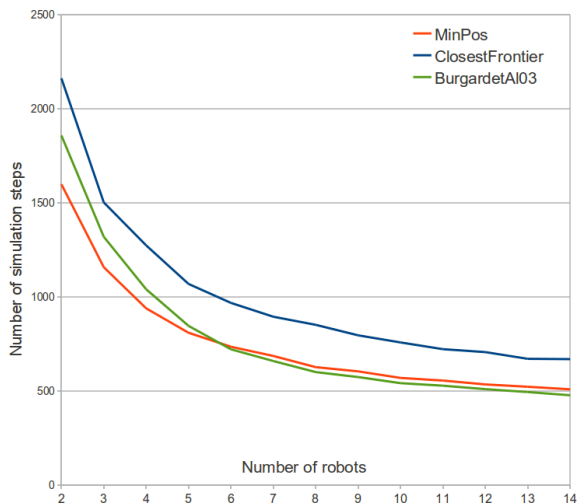


Fig. 6. Results from the exploration of the hospital environment when varying the number of robots

The computation of this novel rank criteria depends on the information of the cost matrix. To compute this matrix, distances are evaluated using a wavefront propagation algorithm [2] on a discrete environment representation. A wavefront computes path distances incrementally around a source. Here, the idea is to propagate a wavefront from each frontier. For a robot assignment, the propagation of a wavefront is stopped when it encounters its location. Thus, it gives the shortest paths (on the grid) to the frontier from all points closer than the robot's location. This is sufficient to know the distances from the other robot's location useful to compute the robot rank. This approach is computationally efficient especially when the number of robot is large, compared to computing the path distance using an A\* algorithm from the frontier to every robot. Such a wavefront propagation is illustrated in Figure 7.

## VI. TRAJECTORY-PLANNING

The wavefront propagation used to compute the robots assignment also provides paths that could be used for the robot navigation. However, it does not take into account the dynamic and nonholonomic constraints.



Fig. 7. Illustration . Wavefronts are stopped on the encounter of the yellow robot computing its assignment. Color code : white=explored, gray=unknown, black=walls, red-green-blue=frontier and gradient wavefront propagation result, only the wavefront closest to the frontier is shown where waves are superimposed.

To tackle this issue, we perform an A\* algorithm in 4D ( $x, y, \text{orientation}, \text{speed}$ ) using the wavefront distances, previously computed, as heuristics. The computational time of A\* depends on the heuristics. Using the euclidean distance, as heuristic, is computationally costly. The originality of our trajectory planning is that we use the already computed 2D wavefront propagation as heuristic (introduced in section V).

Trajectory planning is quite efficient. However, as it uses the almost-shortest path the robot tends to graze obstacles. We therefore added a penalty to nodes close to obstacle with a value inversely proportional to its distance to the closest obstacle. This generates smooth and safe trajectories. Figure 8 illustrates such a planned trajectory.

To evaluate our technique we randomly draw 500 points and compute a trajectory passing by all these points in the order they were generated in. On average, the trajectory planning in an office environment (14 rooms along a corridor in a 1 million pixels image) takes :

- 264.2 ms with no heuristics,
- 20.6 ms with euclidean distance heuristics,
- 14.6 ms with the potential field heuristics (11.7 ms for the A\* and 2.8 ms for the wavefront propagation).



Fig. 8. Example of a planned trajectory keeping away from obstacles

## VII. RESULTS

### A. MiniRex

MiniRex (MINI Robot for EXploration) is a robot, dedicated to the project, designed and build in our laboratory. The main specification was to design a low cost, reliable and small robot. The robot has a square shape (0.25 x 0.25m width) by 0.5m height. It is a tracked robot actuated by two DC geared motors (Faulhaber 2657 012 CR). A PC (Kontron pITX-SP - Intel Atom Z530 1.6GHz) and a real-time dedicated processor (ATmega2560) are embedded and powered by two Lithium-Polymer batteries (22.2V 3300mAh). Several sensors provide internal and external information : ultrasonic ranging and proximity sensors, voltage battery sensors, inclinometer, track encoders, KINECT™ and an actuated LIDAR (Hokuyo UTM-30LX). Figures 9 and 10 show details and illustration of the robot.

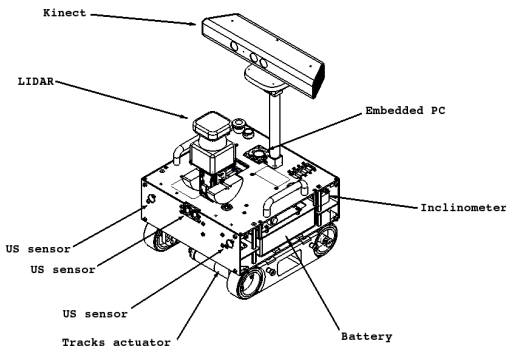


Fig. 9. Specifications of the MiniRex robot.



Fig. 10. Minirex exploring a test area in our laboratory.

Seven robots were built. According to the area size (about  $120m^2$ ), we decided to engage only five robots in the exploration to prevent congestion during the exploration. The sixth robot was kept as a spare robot and the seventh for spare parts.

### B. Results

This section presents the results of the final run during the last year competition. Figure 11 shows the global map and trajectories of the robots. The exploration task was clearly distributed between the robots and each area (not to say each room) was explored by a robot. During the mission, one robot got stuck in the gravel and was not able to reach its

starting point (on the bottom of the area located in the left of the map). Despite the fact that a failure occurred during the mission, the other robots successfully end their task and the failure didn't compromise the whole mission. Figure 13 shows the 3D map built thanks to the RGB-D sensor.

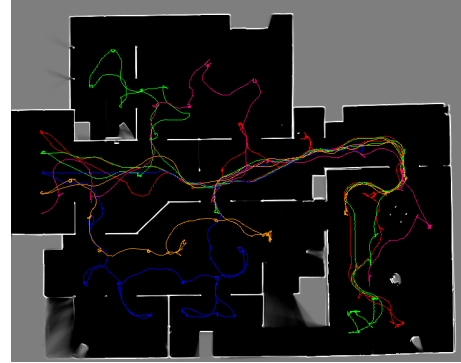


Fig. 11. Map and trajectories of the robots.

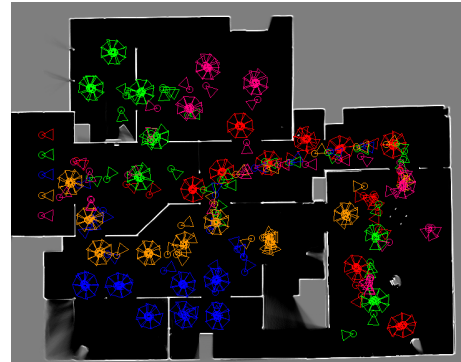


Fig. 12. Map with the location of the RGB-D captures.

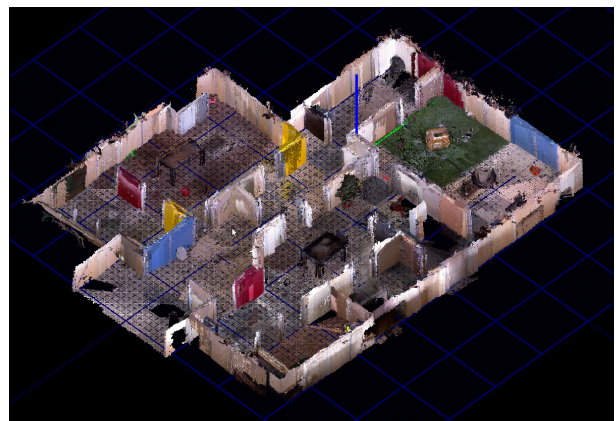


Fig. 13. 3D map of the building.

For a global overview of the mission, object recognition has been illustrated on Figure 14 although this topic is not the aim of the present paper. For more information about object recognition, the reader is referred to [14].



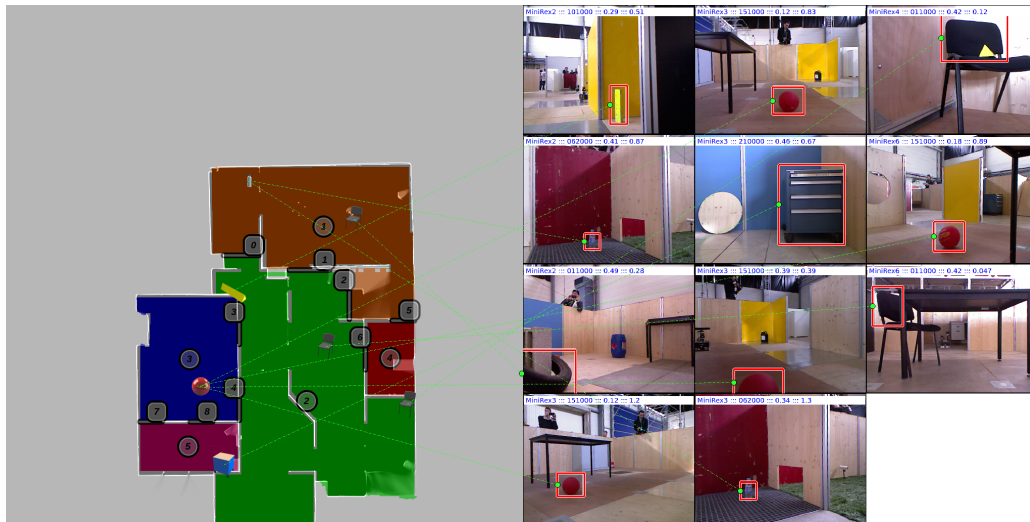


Fig. 14. Object recognition and localization.

### VIII. CONCLUSIONS

This paper described an overview of the software and hardware architecture of the project Cart-O-matic. We presented a SLAM algorithm called Slam-O-matic with a quick overview of the performances. During the competition, a comparison of the map produced by each team has been performed. Detailed results are confidential, but we know that Slam-O-matic reached the first rank in term of accuracy. However, a comparison with similar algorithm (Gmapping [6], Hector Slam [10], ICP [7] ...) would be interesting to compare computation time, memory space and reliability.

We presented the *MinPos* algorithm for multi-robot exploration strategy, which uses a novel criteria ensuring a well balanced distribution of robots among different directions. Results in simulation and with MiniRex robots demonstrated the efficiency in exploration time of this algorithm. More generally, our multi-robot approach showed good robustness and efficiency during the French national robotics challenge 'Carotte', that we won in 2012. We now aim to extend this work to the exploration and mapping of dynamic environments.

### REFERENCES

- [1] Steux B. and H.E. Oussama. tinslam: A slam algorithm in less than 200 lines c-language program. In *ICARCV*, pages 1975–1979, 2010.
- [2] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1012–1017 vol.2, jun 1991.
- [3] A. Bautin, O. Simonin, and F. Charpillet. Minpos : A novel frontier allocation algorithm for multi-robot exploration. In Chun-Yi Su, Subhash Rakheja, and Honghai Liu, editors, *Intelligent Robotics and Applications*, volume 7507 of *Lecture Notes in Computer Science*, pages 496–508. Springer Berlin Heidelberg, 2012.
- [4] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, june 2005.
- [5] D. Filliat and al. Rgb object recognition and visual texture classification for indoor semantic mapping. In *Proceedings of the 4th International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 127–132, 2012.
- [6] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.
- [7] D. Holz and S. Behnke. Sancta simplicitas - on the efficiency and achievable results of slam using icp-based incremental registration. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1380–1387, 2010.
- [8] Andrew Howard, Lynne E Parker, and Gaurav S Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6):431–447, 2006.
- [9] S. Noda I. Matsubara H. Takahashi T. Shinjou A. Kitano, H. Tadokoro and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS*, pages 739–746. IEEE Computer Society, 1999.
- [10] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.
- [11] P. Lucidarme and S. Lagrange. Slam-o-matic : Slam algorithm based on global search of local minima. Brevet num. FR1155625, June 2011.
- [12] A.-I Matignon, L. Mouaddib and L. Jenapierre. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processe. In *International Conference on Advanced Artificial Intelligence (AAAI)*, 2012.
- [13] Edwin Olson, Johannes Strom, Ryan Morton, Andrew Richardson, Pradeep Ranganathan, Robert Goeddel, Mihai Bulic, Jacob Crossman, and Bob Marinier. Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792, 2012.
- [14] Saïd. Gholami Shahbandi and Philippe Lucidarme. Object recognition based on radial basis function neural networks: experiments with rgb-d camera embedded on mobile robots. In *ICSCS*, 2012.
- [15] L. Thorel S. Steux, B. Bouraoui and L. Benazet. CoreBot M : Le robot de la Team CoreBots préparé pour l'édition 2011 du défi Carotte. In *6th National Conference on Control Architectures of Robots*, Grenoble, France, May 2011. INRIA Grenoble Rhône-Alpes. 3 pages.
- [16] W. Thrun, S. Burgard and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [17] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., IEEE International Symposium on*, pages 146–151, Washington, DC, USA, jul 1997. IEEE Computer Society.
- [18] B. Yamauchi. Frontier-based exploration using multiple robots. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 47–53, New York, NY, USA, 1998. ACM.

# Driving Intention Assistance for Front-wheel-drive Personal Electric Vehicle

Satoshi Fujimoto<sup>†</sup>, Zhencheng Hu<sup>†</sup>, Claude Aynaud<sup>‡</sup> and Roland Chapuis<sup>‡</sup>

<sup>†</sup>*The Graduate School of Science and Engineering, Kumamoto University, Japan*

<sup>‡</sup>*Institut Pascal, Clermont Ferrand, France*

satoshi@its.cs.kumamoto-u.ac.jp, hu@cs.kumamoto-u.ac.jp, caynaud@gmail.com, roland.chapuis@univ-bpclermont.fr

**Abstract**—Indoor personal electric vehicle “STAVi” was developed to reduce the burden of moving for elderly people in the progress of the aging population in Japan, in order to improve their quality of life. The STAVi is a front-wheel-drive EV which is operated through an 8-directional joystick by the driver. However, the over-steering caused by two rear caster wheels leads to unstable vehicle dynamics and difficult to control in some driving scenarios. This paper presents a novel Lidar SLAM based driving intention assistance algorithm which employs Line Segment Matching SLAM technique for fast SLAM matching for indoor scenario. Line Segment Matching provides more accurate result than the conventional corner-based Scan Matching. Model Error Compensator (MEC) is used in our feedback controller to assist STAVi moving correctly by driving intention. Real indoor experimental results show the effectiveness of the proposed algorithm.

**Keywords**—personal mobility; STAVi; SLAM; Driving Assistance System; MEC; Autonomous robot;

## I. INTRODUCTION

Japan’s elderly population has rapidly grown to 24.1% of the total population in 2012 and with projections to 33.4% in 2035 [1]. Personal mobility systems and electric wheel chairs are used by elderly to reduce their burden of everyday transportation. Personal vehicle “STAVi” was developed by Sanwa-Hitech Co. Ltd [2] (Fig.1). The STAVi has some characteristics that can be used in the field of welfare. Elderly and handicapped people can easily access (to get on, to get off) the STAVi and the seat can be shifted up and down to provide better eye-line and to reach higher places easily. With the help of this personal mobility tool, elderly and handicapped people are able to greatly improve their quality of life.

Our goal is to build an intelligent driving assistance system based on the STAVi platform to help driver drive safely and smoothly. To achieve this goal, several sensors and controllers are installed on STAVi, including a Hokuyo Lidar range finder (LRF) in the lower frontal bumper, a Kinect 2.5D image sensor on the top of frontal chassis and two ultrasonic sensors in the rear bumper positions. LRF is used for building the mid-range environment map and collision avoidance. Kinect is employed to detect, recognize and track the specified target

like a pedestrian, leading STAVi or docking station. Rear ultrasonic sensors are used to avoid rear collision.

The STAVi is a front-wheel-drive EV which is operated through an 8-directional joystick by the driver. However, the over-steering caused by two rear caster wheels leads to unstable vehicle dynamics and difficult to control in some driving scenarios. This paper presents a novel Lidar SLAM based driving intention assistance algorithm which employs Line Segment Matching SLAM technique for fast SLAM matching for indoor scenario. Line Segment Matching provides more accurate result than the conventional corner-based Scan Matching. Model Error Compensator (MEC) is used in our feedback controller to assist STAVi moving correctly by driving intention. In MEC, instead of initial sensor, we use the yaw rate and velocity data estimated from SLAM as feedback to control the movement. Real indoor experimental results show the effectiveness of the proposed algorithm.

The paper is organized as follows: Section II quickly reviews STAVi and gives the over-steering characters; MEC controller design concept and our driving intention assistance control strategy are described in Section III. Section IV provides previous works in Lidar SLAM and our Line Segment Matching algorithm. Section V shows implementation details and experimental results.



Figure 1. Personal electric vehicle “STAVi”

## II. CHARACTERISTICS OF PERSONAL VEHICLE “STAVi”

The front-wheel-drive STAVi is designed for elderly and handicapped people. It uses two rear free caster wheels to make a flat rear deck. This design is considered that driver can easily access from bed or wheel chair. STAVi also has a movable seat that can be shifted up and down. A driver controls the STAVi through an 8-directional joystick.



However, the over-steering caused by two rear caster wheels leads to unstable vehicle dynamics and difficult to control in some driving scenarios [5], for example, driver always needs to adjust the joystick direction even when going straight. Characteristic of over steering on flat floor is shown in Fig. 2.

To overcome the over-steering problem of STAVi, Model Error Compensator (MEC) is used in our feedback controller to assist STAVi moving correctly by driving intention.

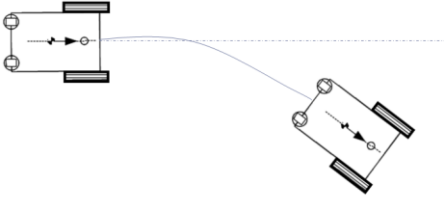


Figure 2. Over steering characteristic

### III. CONTROL BLOCK WITH MEC

As described above, STAVi has the characteristic of over-steering. Since STAVi's self weight is only 250 lb, the vehicle dynamics is variable when loading different drivers, which makes the vehicle difficult to operate, even on a flat road. Therefore the traditional feedback controller like PID controller cannot provide a good performance. Instead of attempting to minimize the effects of the disturbance as in the robust filters or to decouple the disturbance as in the unknown input observers, it is proposed to estimate the disturbance estimation is used to reduce the model error and thus to improve the state estimation. This technique is denoted as model error compensator (MEC) [8].

#### A. Controller interface

STAVi is operated through an 8-directional joystick as shown in Fig. 3. Moving the stick along X-axis controls rotation angle and moving along Y-axis controls vehicle speed. For example, pushing the stick forward will make the vehicle go straight forward and pushing the stick toward left side will make the vehicle turn left in the same location. Joystick position will be converted to the input voltages in both speed and direction to the controller box.

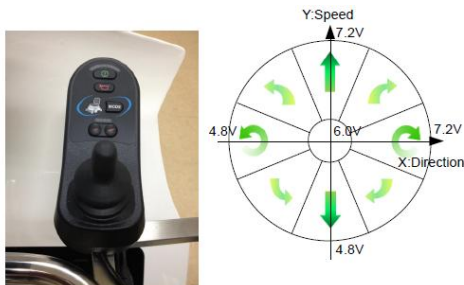


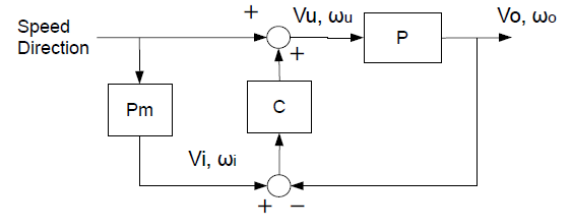
Figure 3. Joystick controller

#### B. MEC controller design

As described in Section II, STAVi has the over-steering characteristic and need to assist even for moving straight on

a flat surface. However, since STAVi system dynamics is variable with different drivers, we need a simple and powerful controller to reduce the model error. MEC is an ideal controller for this purpose. With the input of plant output and ideal model output, MEC is able to control the system overall output to get closer to the ideal output.

To compensate the dynamic model change and disturbance of input observers, a novel MEC feedback controller is proposed. Figure 4 shows the controller block with MEC. Regarding to MEC concept details, please refer to [8].



*Speed, Direction* : input voltage from joystick operation  
*Vi, ωi* : control to STAVi motor  
*P* : plant - STAVi  
*Vo, ωo* : observation of velocity and angular velocity with SLAM  
*C* : PI controller  
*Vi, ωi* : ideal velocity and angular from ideal model.  
*Pm* : ideal dynamic model

Figure 4. MEC Feedback Controller

*Pm* is the ideal dynamic model which is estimated by experiments of driving on a flat floor. Real velocity and angular velocity are subtracted from the ideal velocity and angular velocity. The subtracted value is inputted to PI controller *C*. The controlled value is added to input value of speed and direction.

Status equation is shown like follows:

$$V_u = K_{vp} \left( V_i \left( 1 - \exp \left( -\frac{t}{\tau_v} \right) \right) - V_o \right) + K_{vi} \int \left( V_i \left( 1 - \exp \left( -\frac{t}{\tau_v} \right) \right) - V_o \right) dt + S$$

$$\omega_u = K_{op} \left( \omega_i \left( 1 - \exp \left( -\frac{t}{\tau_\omega} \right) \right) - \omega_o \right) + K_{oi} \int \left( \omega_i \left( 1 - \exp \left( -\frac{t}{\tau_\omega} \right) \right) - \omega_o \right) dt + D$$
(1)

Where, *S* and *D* are joystick input value for speed and direction control.  $\tau_v$ ,  $\tau_\omega$  is integration delay time for PI control.  $V_o$ ,  $\omega_o$  are estimated by SLAM shown in the next section.

### IV. LINE SEGMENT MATCHING SLAM

In the previous section, observation of velocity and angular velocity in the MEC feedback controller block diagram could use sensor output from the odometer and gyroscope or yaw rate sensor. However, all these local measurements need to be integrated in order to obtain the position and track. Large accumulated error cannot be avoided while driving a longer distance or continuous turning.

Simultaneous localization and mapping (SLAM) is an alternate solution which uses Lidar, vision or fused sensor to obtain a continuous obstacle map as well as own localization. Many approaches have been proposed for the last several decades [3][4][6][7]. To estimate the vehicle position, an internal sensor is generally used also known as dead reckoning.

However, using only odometers to estimate the position of the vehicle causes a stack of error due to a slipped tire on a slope and rough road.

To reduce this stack of error, external sensor such as the LRF enables to estimate SLAM accurately. On the contrary, if the external sensor extracts very few landmarks, ambiguity of matching positions between continuous frames will lead to a big error. In indoor environment, we have confirmed that line-based scan matching approach is more efficient than corner-based one. Because the corner-based scan matching is difficult to extract the feature point. Line-based scan matching algorithm is shown in Figure 5.

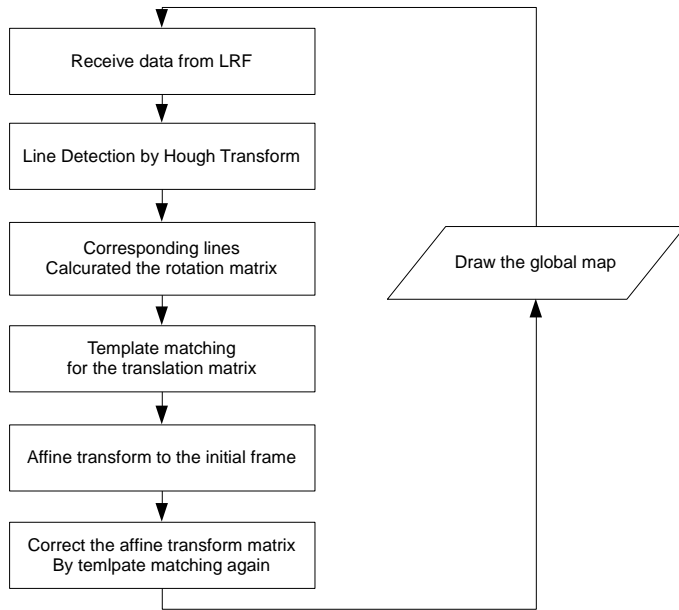


Figure 5. Line-Matching SLAM

#### 1) Line Detection:

The equipped LRF, Hokuyo UTM 30-LX, has the maximum detection distance of 30m, scanning angle from  $-135^\circ$  to  $135^\circ$  with the resolution of  $0.25^\circ$ . The received range data will be converted to scan image at 20mm/pixel resolution. Line segments are extracted from scan data by Hough transform. Each segment has its property descriptions like length, orientation, center position, end point.

#### 2) Line Segment Matching for Rotation Calculation

Center position, orientation and length are used to match these line segments between two consequential frames. Figure 7 shows the detected and corresponded lines in frames. Vehicle rotation between two consequential frames is calculated by the average of orientation differences between matched line segment pairs.

#### 3) Template matching for Translation Calculation

To calculate the translation matrix, we use template matching technique to find the best matching between the rotated frame and the previous frame. Searching region size depends on vehicle speed and orientation. We use a simple

Kalman filter to predict vehicle's position and its variance to act as start position and searching region for template matching.

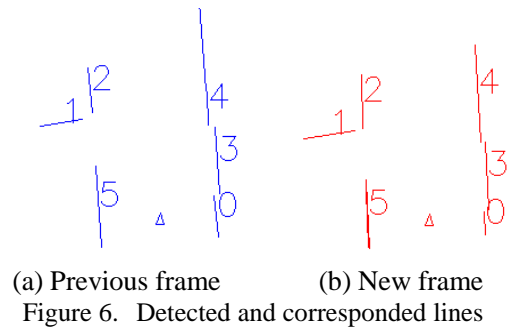
#### 4) Affine Transform for Merging Maps

After calculating rotation and translation matrix, the new frame will be merged to the previous frame to build a continuous map through Affine transform. Figure 7 shows the merged map result.

#### 5) Building Global Map

To avoid accumulated Affine transform error, the merged map in step 4) will be matched with the previous global map by template matching and the rotation and translation matrix are refined in this step. An example of global SLAM map result is shown in Figure 8.

In this way, the proposed SLAM algorithm generates a global obstacle map and its own track, meanwhile, the position and attitude angle are also estimated for  $V_0$ ,  $\omega_0$  of MEC controller feedback control described in Section III.



(a) Previous frame (b) New frame  
Figure 6. Detected and corresponded lines

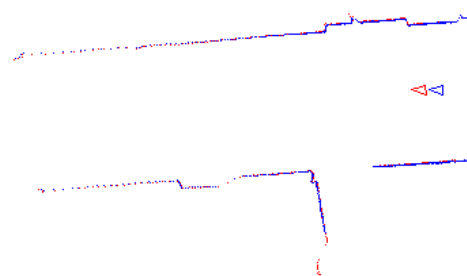


Figure 7. Merged map by template matching where blue points are previous frame data and red ones are new frame data

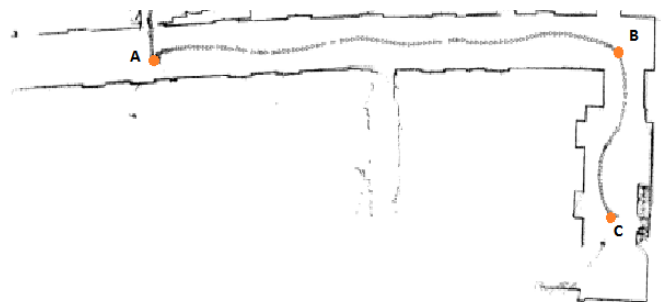


Figure 8. Global SLAM map result example

V. EXPERIMENTAL RESULTS

A. Vehicle Control System Description

STAVi control system block diagram is shown in Figure 9. Joystick module is connected to the R-Net Input port and R-NET Outputs control signal to the R-Net power module which generates the driving force to STAVi's left/right motors. A switch is used to switch manual/autonomous operation mode. Autonomous mode connects a tablet PC to the R-Net IOM through RS-232 port.

B. Line Segment Matching SLAM Result

To compare the proposed Line Segment Matching SLAM algorithm with the corner-based SLAM algorithm, several indoor experiments were carried out. Example result is shown in Figure 10. Since indoor environment does not provide enough corners and false corners by the occlusion problem, Corner-based scan matching suffers from the mismatching problem. Therefore, the proposed line segment matching SLAM algorithm shows its advantages in indoor environments.

To evaluate the quantitative measurement error of proposed line segment matching SLAM, we compared the measurement between manually measured result with our SLAM output for both translation and rotation matrix. In Figure 8, distance from point A to B is 24 m by manually measurement and SLAM result is 23.91 m, which gives the measurement error is 9cm (0.305%). From point B to C is 5.6m by manual measurement and our SLAM result is 5.55 m, therefore the measurement error is 5cm (0.169%). In Figure 11, the rotation measurement error is 1.5 ~ 3 degrees.

In Figure 12, the translation measurement error happens because of the two parallel of long line segments on the corridor that is disabled to make a distinction of moving forward or backward.

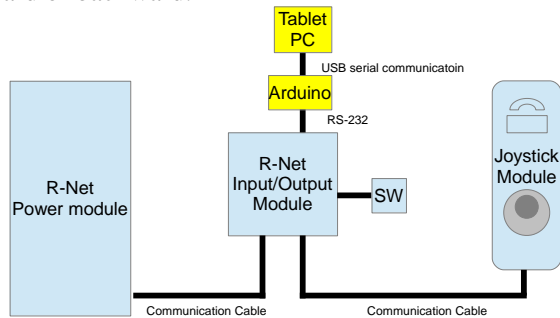


Figure 9. The experimental apparatus

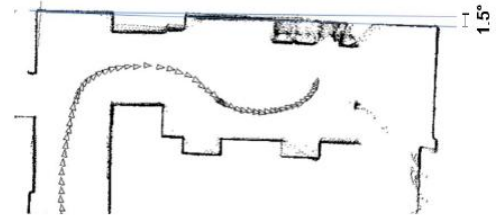


Figure 11. Line Segment Matching SLAM rotation error

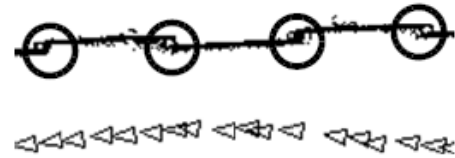


Figure 12. Line Segment Matching SLAM translation error

C. MEC Controller Design

In order to obtain the ideal response between joystick input and motor output, we measured the STAVi transient response value (speed and angular velocity) based on joystick's 11 steps of directional input (from -50 degrees to 50 degrees) and 11 steps speed input (from 10 to 110). Test results are shown in Table 1 (STAVi speed output) and Table 2 (STAVi angular velocity output). Delay time constants are also measured and the average time constant is adopted for the design of ideal first order lag system. All measurements are taken by our SLAM system.

In Table 1 (real angular velocity), when the input value is 60 of the Direction against the Speed, the angular velocity is not zero. From the real dynamics (Table 1), we create an ideal dynamic model (see Fig. 13.). The error between the real and ideal model is added to the input.

Table 1. Transient response of STAVi velocity

Direction \ Speed	-50	-40	-30	-20	-10	0	10	20	30	40	50	Average
	80	20.2	-	13.7	8.6	10.2	9.1	7.8	11.2	-	-	
90	26.8	23.7	19.2	17.6	17.0	15.5	15.5	16.6	17.0	21.1	25.8	19.6
100	24.4	29.8	29.9	23.6	25.3	21.6	23.9	25.5	23.5	22.0	-	24.9
110	26.2	32.9	35.9	33.4	33.5	32.6	30.1	32.3	30.6	32.7	33.2	32.1

Table 2. Transient response of STAVi angular velocity

Direction \ Speed	-50	-40	-30	-20	-10	0	10	20	30	40	50
	80	57.7	-	32.9	12.8	4.1	0.1	-1.9	-12.9	-	-
90	80.5	63.5	44.3	23.1	8.0	2.8	-4.6	-14.4	-34.8	-54.0	-64.7
100	57.8	60.1	57.7	31.7	11.9	1.4	-5.1	-23.0	-34.5	-63.0	-
110	62.0	60.5	60.4	38.4	24.9	3.6	-10.4	-21.7	-40.0	-61.9	-62.9
Average	64.5	61.4	48.8	26.5	12.2	2.0	-5.5	-18.0	-36.4	-59.6	-62.3

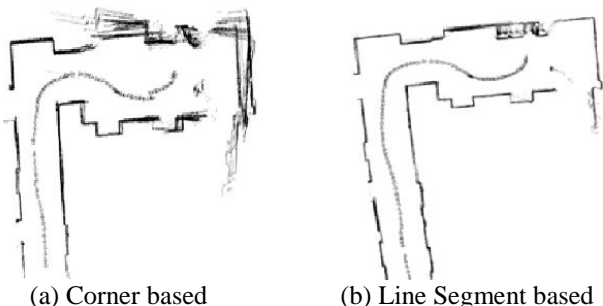


Figure 10. Comparing SLAM result

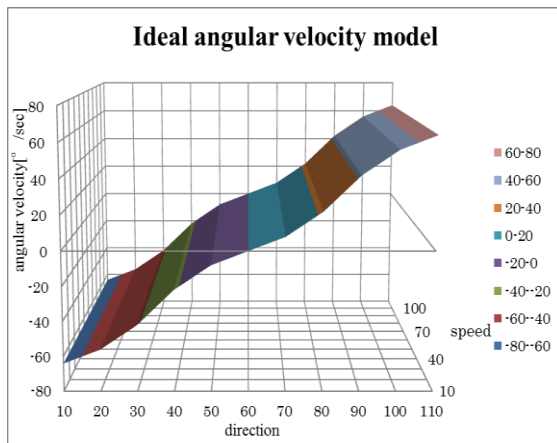
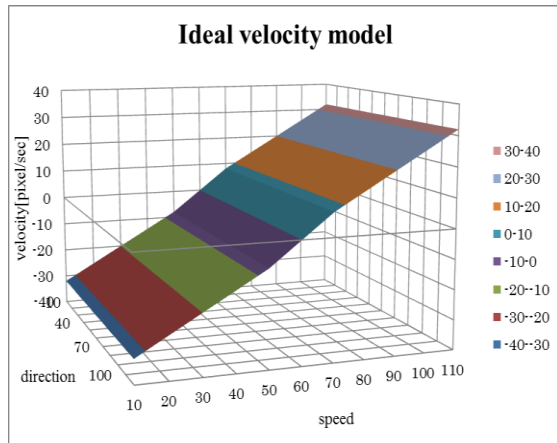


Figure 13. Ideal dynamic model

In PI controller, the proportional gain  $K_{vp}$  and  $K_{op}$  and integral gain  $K_{vi}$  and  $K_{oi}$  for input driving force are given by the following.

$$\begin{aligned} K_{vp} &= 2.0 \\ K_{vi} &= 0.7 \\ K_{op} &= 2.0 \\ K_{oi} &= 0.7 \end{aligned} \quad (2)$$

The optimal value is calculated by experiments. In the next section, the results by using this feedback controller are shown.

#### D. Driving Intention Assistance results by SLAM

Figure 14 Shows the trajectory of STAVi movement along a straight line on a flat floor. On the left side, it is with no control and right side is with control. As you can see, with the control there is an improved straightness trajectory by comparison with no control. The significant smoothness and efficiency helps driver driving with their intention.

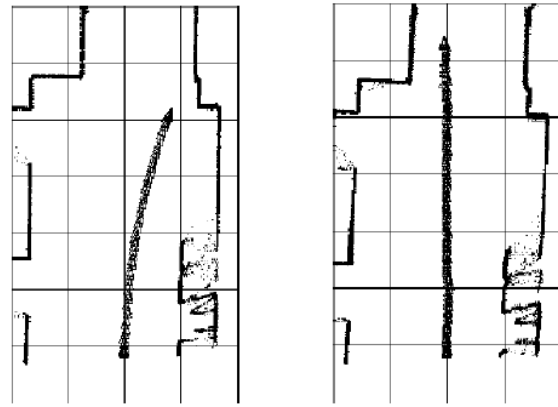


Figure 14. Without MEC controller and with MEC (grid interval : 1m)

## VI. Conclusion and Future work

Feedback from SLAM is proposed to control for indoor personal vehicle STAVi with our novel MEC controller. The Line matching SLAM approach is more efficient than corner matching SLAM for indoor environment. We developed the ideal dynamic by referenced experimental real dynamic. The STAVi becomes ideal dynamic model which has no over steering characteristic. It enables driver to control easily and no need to adjust the direction when going straight.

In the future work, the autonomous vehicle will be produced using SLAM. Adding, we are developing path planning and recognizing obstacles and avoiding system. Furthermore, to be more intelligent vehicle we try to build a 3D map using Microsoft Kinect which enables STAVi to understand the environment accurately for precise control.

## References

- [1] Cabinet Office, Government of Japan, "annual report on the aging society: 2013", p1, Jun 2013. <http://www8.cao.go.jp/kourei/whitepaper/w-2013/zenbun/pdf/1s1s.pdf>
- [2] F-CITE, "STAVi", <http://f-cite.com/>, 2013
- [3] Masaaki OHKITA, "Traveling control of the Autonomous Mobile Wheel-Char DREAM-3 Considering Correction of the Initial Position", The 47<sup>th</sup> IEEE International Midwest Symposium on Circuits and Systems, 2004
- [4] Konishi Ryouzuke, "Study on the development of next generation type electric wheel chair", Tottori University research result repository, 2010.
- [5] Yutaro Maruno, Aydin Tarik Zengin, Hiroshi Okajima, Nobumoto Matsunaga, Norihito Nakamura, "Driving Experiment of Front Drive Type Electric Wheelchair using Yaw-rate Control", p1410, SICE Annual Conference 2012, August 2012.
- [6] Young-Ho Choi, Tae-Kyeong Lee, Se-Young Oh, "A line feature based SLAM with low grade range sensors using geometric constrains and active exploration for mobile robot", Auton Robot, 2008
- [7] Zheyuan Lu, Zhencheng Hu, Keiichi Uchimura, "SLAM Estimation in Dynamic Outdoor Environments: A Review", Intelligent Robotics and Applications, pp255-267, 2009
- [8] Jay F. Tu & Jeffrey L. Stein, Model error compensation for

observer design, *International Journal of Control*, Volume 69,  
Issue 2, pages 329-345, 1998



# Ad-hoc heterogeneous (MAV-UGV) formations stabilized under a top-view relative localization

Martin Saska and Vojtěch Vonásek and Tomáš Báča and Libor Přeučil

**Abstract**—A stabilization and navigation technique for ad-hoc formations of autonomous ground and aerial robots is investigated in this paper. The algorithm, which enables a composing of heterogeneous teams via consequence splitting and decoupling, is aimed at deployment of micro-scale robots in environments without any precise global localization system. The proposed approach is designed for utilization of an on-board visual navigation and a top-view relative localization of team members. The leader-follower formation driving method is based on a novel avoidance function, in which the entire 3D formation is represented by a convex hull projected along a desired path to be followed by the groups. This representation of the formation shape is crucial to ensure that the direct visibility between the team members in environments with obstacles is kept, which is the key requirement of the top-view relative localization. A Receding Horizon Control (RHC) concept is employed to integrate this avoidance function. The RHC scheme enables fluent splitting and decoupling of formations and responding to dynamic environment and team members' failures. All these abilities are verified in simulations and experiments, which prove the possibility of formation driving based on the visual navigation and top-view relative localization.

## I. INTRODUCTION

Micro Aerial Vehicles (MAVs) may provide numerous new possibilities in applications that are strictly addressed to Unmanned Ground Vehicles (UGVs) recently. MAVs can be employed in locations that are hardly reachable by UGVs. They enable measurement and mapping in 3D environment. In reconnaissance and surveillance missions, they provide a top-view, which is important for a global overview of the scene. Besides, the top-view from MAVs could be efficient for a relative localization of team members in multi-robot applications. The aim of this paper is to investigate possibilities of utilization of such a visual top-view localization for stabilization of heterogeneous MAV-UGV formations. This approach may act as an enabling technique for deployment of fleets of micro unmanned vehicles outside laboratories equipped with a global localization system, which is usually used for stabilization of robotic groups in a compact formation.

The work presented in this paper is motivated by a scenario of multi-robot surveillance. In the illustrative mission, an autonomous formation of mobile robots with surveillance cameras has to repeatedly follow a predefined path in a wide phalanx to cover a large operating space. The desired path can be splitted into several branches to inspect smaller areas simultaneously by sub-formations created ad-hoc from

the larger group. The heterogeneous MAV-UGV formations then can provide surveillance in large areas by spreading into a wide searching phalanx, where MAVs and UGVs give view from a different perspective and can visit locations of different types. In large areas under surveillance, there usually cannot be pre-installed a precise global localization infrastructure and public available systems (as GPS) lack sufficient precision for stabilization of compact formations. Therefore, we propose the formation driving technique, which is designed for the top-view visual relative localization and for a simple vision based navigation. Both these methods rely only on on-board sensory and computational resources of micro-scale robots. The relative localization uses simple light-weight cameras mounted on all MAVs and identification patterns placed on UGVs and MAVs, where the distance between the vehicles is available due to the known size of the patterns. Details on the visual based relative localization together with description of its precision and reliability is provided in [1]. The navigation approach (referred to as GeNav) uses image features detected by a monocular camera carried by a robot of the formation. It enables to robustly navigate the group along a pre-learnt path consisting of a set of straight segments (a proof of stability of this method, where the necessity of piecewise straight path is shown, can be found in [2]).

## II. STATE-OF-THE-ART AND PROGRESS BEYOND

In up-to-date literature, one can find works aimed at both aspects investigated in this paper, the formation stabilization [3], [4] and the path following by a formation [5], [6], [7]. The mentioned approaches rely on utilization of robots under a precise external global localization system (e.g. VICON system in [4], [6]) or only theoretical solutions verified by simulations are provided [3], [5], [7]. Our work goes beyond these approaches by strict utilization of on-board systems for robots' localization and navigation, which are inherently included in the essence of the formation driving approach. We rely on the Receding Horizon Control (RHC) to be able to involve the requirements of available robust localization and navigation techniques into the formation driving. In particular, constraints imposed by the inter vehicle relations (shape of the formation feasible for the top-view relative localization) and by the GeNav technique employed for the navigation of the entire group along straight line segments of the desired path are included. This paper extends our previous publication [8] with the description and verification of the algorithm that provides the ability of the formations

Authors are with Department of Cybernetics at the Czech Technical University in Prague. {saska,vonasek,preucil}@labe.felk.cvut.cz

merging and splitting to be able to inspect smaller areas simultaneously.

RHC is usually employed in the formation driving approaches due to its ability to respond to changes in dynamic environment [7], [5], [6]. In [5] and [6] it was shown, that the computational power of microprocessors available on-board of unmanned helicopters enables to employ RHC techniques also for the formation control of such a high dynamic system, similarly as it is proposed here. Again, we go beyond these papers mainly in the aspect of the formation stabilization with included requirements of the top-view relative localization, which could be an enabling technique for deployment of heterogeneous MAVs-UGVs teams outside the laboratories without any global localization. We present a novel dynamic obstacle avoidance function with a simple and effective representation of the 3D formation as a convex hull. Besides, our formation driving method is designed for the purpose of simple yet stable visual navigation [2], which is well suited for the surveillance missions being our target applications. Finally, our method is well suited for creating of ad-hoc formations via merging and splitting under the RHC stabilization.

### III. PRELIMINARY NOTES

The problem of following desired paths by  $n_F$  compact UGV-MAV formations of given shapes is tackled in this paper. Let us assume that the environment contains  $n_0$  of compact static (in a known map) or dynamic and unknown (detected by on-board sensors) obstacles. For the global localization of each group, we assume that a robot of the group, called GeNav leader, is equipped with the navigation system based on the features detection. The GeNav system is suited for guidance of robots along a path that consists of set of straight segments. Beside the GeNav leader, each formation consists of MAV followers (quadrotors) and may consist also of UGV followers (robots without any localisation system available on-board). MAVs are equipped with a bottom camera and the system for visual relative localization between the camera and centres of identification patterns carried by all UGVs and MAVs (except MAV flying in the highest altitude).

For the formation driving description, let  $\psi_j(t) = \{x_j(t), y_j(t), z_j(t), \varphi_j(t)\}$ , where  $j \in \{GL, VL, 1, \dots, n_r\}$ , denote configurations of the GeNav leader  $GL$ , a virtual leader  $VL$ , and  $n_r$  followers of each formation at time  $t$ .  $GL$  is positioned in front of each formation and it is used as a reference point for the coordinate system using the top-view relative localization.  $VL$  is initially placed in the same position and orientation as the GeNav leader and it acts as a reference point for the proposed formation driving technique. Using the presented trajectory following approach (Section IV-B), it keeps the same position as  $GL$  except the deviation caused by obstacles that could brake the top-view localization or to cause collisions. Besides,  $VL$  is crucial for merging of sub-formations into a compact formation, where the relative error in position has to be diminished.

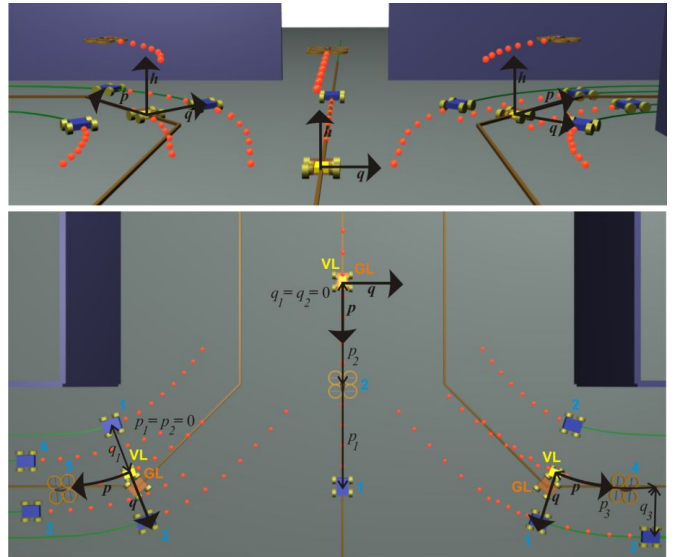


Fig. 1. Curvilinear coordinates of three formations going into the merging point.

The Cartesian coordinates  $x_j(t)$ ,  $y_j(t)$  and  $z_j(t)$  define positions  $\bar{p}_j(t)$  of all robots (leaders and followers) and  $\varphi_j(t)$  denotes their heading. Both platforms, MAVs and UGVs, (except the robots assigned as the GeNav leaders) are denoted as followers in this notation. For the MAVs, the heading  $\varphi_j(t)$  becomes directly the yaw. Roll together with pitch do not need to be included in the kinematic model employed in RHC, but they depend on the type of utilized MAVs as we have shown for a quadrotor in [9].

The kinematics for any robot  $j$  in 3D is described by the simple nonholonomic kinematic model:  $\dot{x}_j(t) = v_j(t) \cos \varphi_j(t)$ ,  $\dot{y}_j(t) = v_j(t) \sin \varphi_j(t)$ ,  $\dot{z}_j(t) = w_j$  and  $\dot{\varphi}_j(t) = K_j(t)v_j(t)$ , where feed-forward velocity  $v_j(t)$ , curvature  $K_j(t)$  and ascent velocity  $w_j(t)$  represent control inputs denoted as  $\bar{u}_j(t) = \{v_j(t), K_j(t), w_j(t)\}$ . We assume that UGVs operate in a flat surface and that  $z_j(\cdot) = 0$  and  $w_j(\cdot) = 0$  for each of the UGVs. In case of MAVs,  $v_j(\cdot)$ ,  $K_j(\cdot)$  and  $w_j(\cdot)$  values are inputs for the low level controller, as shown in [9].

Let us now define a time interval  $[t_0, t_{end}]$  that consists of a sequence of elements of increasing times  $\{t_0, t_1, \dots, t_{end-1}, t_{end}\}$ , such that  $t_0 < t_1 < \dots < t_{end-1} < t_{end}$ . We will refer to  $t_k$  using its index  $k$  in this paper. The inputs of the receding horizon control are held constant over each time interval  $[t_k, t_{k+1})$ , where  $k \in \{0, \dots, end\}$ . We will call the points at which the control inputs change as *transition points* and we will refer to them with index  $k$ .  $\Delta t$  will be a sampling time, which is uniform in the whole interval  $[t_0, t_{end}]$ . The control inputs  $v_j(k+1)$ ,  $K_j(k+1)$  and  $w_j(k+1)$  are constant between transition points with index  $k$  and  $k+1$ .

We propose to maintain the shape of each heterogeneous formation using the leader-follower technique with the notation visualized in Fig.1. In this method, both types of followers, MAVs and UGVs, follow the trajectory of the virtual leader in distances defined in  $\{p, q, h\}$  curvilinear co-

ordinate system. The position of each follower  $i$  is uniquely determined by states  $\psi_{VL}(t_{p_i})$  in *travelled distance*  $p_i$  from the actual position of the virtual leader along the virtual leader's trajectory, by *offset distance*  $q_i$  from the trajectory in perpendicular direction and by elevation  $h_i$  above the trajectory.  $t_{p_i}$  is the time when the virtual leader was at the *travelled distance*  $p_i$  behind its actual position.

#### IV. DESCRIPTION OF THE FORMATION DRIVING METHOD

##### A. Overview of the formation driving method

The stabilization of each MAV-UGV formation is realized separately in a decentralized manner, where only the desired paths and shapes for each formation are distributed within the teams by a *coordination unite*. The formation control algorithm is divided into three main blocks (see Fig. 2). The first block, *GL* leader, is responsible for navigation of the entire formation in the environment. It provides control inputs for the GeNav leader based on image features gained by its on-board camera. The GeNav method enables to navigate a robot or a group of robots along a pre-learned path consisting of straight segments.

Beside the GeNav leader steering, the output of the *GL* module is a prediction of GeNav leader's states. The predicted trajectory consists of  $n$  states derived with constant sampling time  $\Delta t$  and it acts as an input of the *VL* block. This part is important for avoidance of obstacles and it enables to follow the GeNav leader in connections of the line segments of the desired path. In the *VL* part, the *Trajectory Following* block provides control inputs for the virtual leader, which respects the requirements of the top-view relative localization through the model of the formation. In the straight segments of the desired path, the trajectory found by the *Trajectory Following* block follows the desired trajectory with a minimal deviation. A significant deviation arises mainly due to appearing obstacles or near to line segment connections. Besides, it is important to diminish the position error in case of the sub-formations merging. Details on the trajectory following mechanism with emphasis on incorporation of the 3D heterogeneous formation stabilized under the top-view localization are presented in Section IV-B.

The trajectory obtained in the *Trajectory Following* block is described by a sequence of configurations of the virtual leader  $\psi_{VL}(k)$ , where  $k \in \{1, \dots, N\}$ , and by constant control inputs applied in between the transition points. According to the RHC concept, only a portion of the computed control actions is applied on the interval  $\langle t_0, t_0 + n\Delta t \rangle$ , known as the receding step. This process is then repeated on the interval  $\langle t_0 + n\Delta t, t_0 + N\Delta t + n\Delta t \rangle$  as the finite horizon moves by *time steps*  $n\Delta t$ , yielding a state feedback control scheme strategy. The unused part of the trajectory can be employed for re-initialization of the planning process in each planning step, since the plan of the formation between two consequent steps is usually changed only slightly. To summarize this,  $n$  is number of transition points in the part of the planning horizon, which is realized by robots in each

planning step and  $N$  is the total number of transition points in the planning horizon.

In the proposed formation driving system, the trajectory obtained in the *Trajectory Following* block is used as an input for the *Formation Driving* module, where the transition points of the trajectory are shifted for each of the follower  $i$  by the vector  $V(t_{p_i})$ . The core of the third main block, which is multiplied for MAVs and UGVs followers, is also the *Trajectory Following* module. This part is responsible for avoiding impending collisions with obstacles or team members and it corrects deviations from the desired trajectory provided by the virtual leader.

The physical communication via WiFi is required only between the *GL* leader and particular followers. It is assumed that the *GL* and *VL* modules as well as the *Coordination Unite* are realized on the same vehicle. Also the data from the relative localisation processes are stored there. Therefore, the communication between the *GL* leader and followers is limited to sending the desired trajectory and actual data from the visual relative localization.

Finally, let us remark that the trajectories of *VL* leader and followers are given in the local frame of the *GL* leader, since all members of the formation know its relative position provided by the top-view localization.

The ability of the system to ensure 3D formation stabilization under the top-view visual relative localization in environments with dynamic obstacles requires to integrate an obstacle avoidance function into the trajectory following methods (introduced in the previous subsection). The proposed avoidance function is based on a representation of the entire formation, which incorporates the requirement on the direct visibility between the robots into the formation stabilization process.

In the method, the 3D formation is represented by a convex hull of positions of followers projected into a plane  $\mathcal{P}_{VL}$ , which is orthogonal to the trajectory of the virtual leader in its actual position. The convex hull of the set of projected points is an appropriate representation of the 3D formation under the top-view relative localization by two reasons: 1) Each follower  $i$  intersects the plane  $\mathcal{P}_{VL}$  at the projected point in future. 2) The convex hull of such a set of points denotes borders of the area, which should stay obstacle free. This ensures that the direct visibility between MAVs and UGVs, which is crucial for the presented top-view visual localization, is satisfied.

Moreover for the obstacle avoidance function presented in Section IV-B, the convex hull needs to be dilated by a detection boundary radius  $r_s$  to keep obstacles in a desired distance from followers. Only obstacles that are closer to the convex hull than  $r_s$  are considered in the avoidance function. In the trajectory following process applied for the followers' control, the dilated convex hull is reduced to a circle with radius equal to  $r_s$  to represent a single robot.

##### B. RHC trajectory following

The aim of the formation stabilization mechanism with the obstacle avoidance function is to find a control sequence that

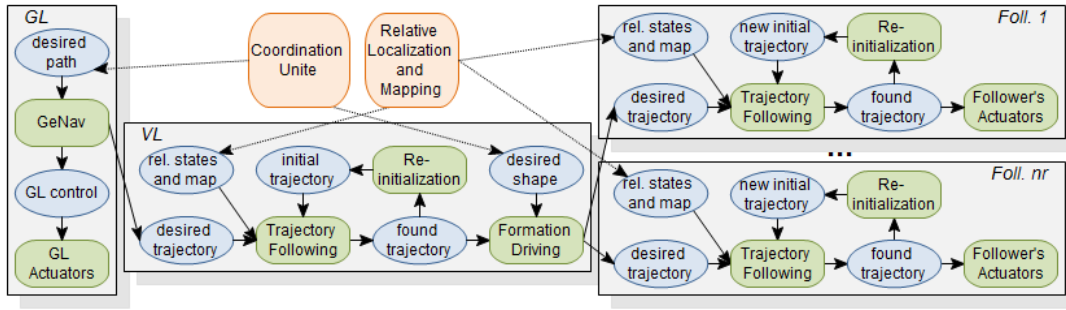


Fig. 2. Relation between modules of the formation stabilization system.

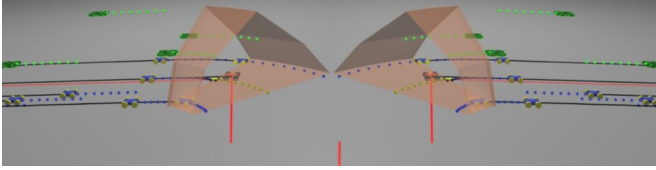


Fig. 3. The dilated convex hull projected along the planned trajectory of virtual leaders leading formations into a merging point.

steers the virtual leader along the desired path followed by the GeNav leader and consequently to find control sequences that stabilize the followers behind the virtual leader in desired relative positions. The intention of the method is to keep the virtual leader as close as possible to the GeNav leader and followers as close as possible to their desired position behind the virtual leader, while the requirements given by the non-collision formation driving and the top-view relative localization are satisfied.

To define the trajectory planning problem in a compact form, we need to gather states  $\psi_j(k)$ , where  $k \in \{1, \dots, N\}$  and  $j \in \{VL, 1, \dots, n_r\}$ , into vectors  $\Psi_j \in \mathbb{R}^{4N}$  and the control inputs  $\bar{u}_j(k)$  into vectors  $\mathcal{U}_j \in \mathbb{R}^{3N}$  for each of the formation. All variables describing the trajectory of the virtual leader or a follower can be collected in a single optimization vector:  $\Omega_j = [\Psi_j, \mathcal{U}_j] \in \mathbb{R}^{7N}$ . Then, the trajectory planning can be transformed to minimization of a cost function  $J_j(\Omega_j)$ ,  $j \in \{VL, 1, \dots, n_r\}$ , subject to sets of equality constraints  $h_j(k) = 0, \forall k \in \{0, \dots, N-1\}$ , and inequality constraints  $g_j(k) \leq 0, \forall k \in \{1, \dots, N\}$ . The cost function consists of three parts as described in details in [8].

Solutions with states deviated from the desired states  $\bar{p}_{d,j}(k)$ , where  $k \in \{1, \dots, N\}$ , are penalised in the first part. The desired states are obtained by the prediction of the movement of the GeNav leader in the virtual leader's trajectory tracking. In the followers' trajectory planning, the desired states are derived from the result of the virtual leader's trajectory tracking using the formation driving concept for each of the followers.

The second term of  $J_j(\Omega_j)$  contributes to the final cost when an obstacle is inside the projection of the dilated convex hull along the planned trajectory. The convex hull represents the entire formation in case of the virtual leader's trajectory planning or a single robot in case of the followers'

trajectory planning. Examples of the projected convex hull are shown in Fig. 3. The value of the second term of  $J_j(\Omega_j)$  will be increasing as the obstacle is approaching to the centre of the convex hull.

The third part of the cost function  $J_j(\Omega_j)$  is crucial for the failure tolerance of the system. This term is a sum of avoidance functions in which the other members of the team are considered also as dynamic obstacles if they are leaving their desired position within the formation.

The equality constraints  $h(k)$  represent the discretized kinematic model for all  $k \in \{0, \dots, N-1\}$ , which ensures that the obtained trajectory stays feasible for the utilized robots. The sets of inequality constraints  $g(k)$  characterize bounds on control inputs  $\bar{u}_j(k)$  for all  $k \in \{1, \dots, N\}$ . The control inputs are limited by vehicle mechanical capabilities (i.e., chassis and engine) as  $v_{min,i} \leq v_i(k) \leq v_{max,i}$ ,  $|K_i(k)| \leq K_{max,i}$  for all followers. For MAVs also constraints  $w_{min,j} \leq w_j(k) \leq w_{max,j}$  have to be satisfied. These limits are extended for the virtual leader planning, since the trajectory of the virtual leader must be feasible for all followers in their desired positions. For the virtual leader, the admissible control set can be determined using the leader-follower approach as  $\max_{i=1, \dots, n_r} \left( \frac{-K_{max,i}}{1-q_i K_{max,i}} \right) \leq K_{VL}(k) \leq \min_{i=1, \dots, n_r} \left( \frac{K_{max,i}}{1+q_i K_{max,i}} \right)$  and  $\max_{i=1, \dots, n_r} \left( \frac{v_{min,i}}{1+q_i K_L(t)} \right) \leq v_{VL}(k) \leq \min_{i=1, \dots, n_r} \left( \frac{v_{max,i}}{1+q_i K_L(t)} \right)$ . These restrictions must be applied to respect different values of curvature and speed of robots in different positions within the guided formation. Intuitively, e.g. the robot following the inner track during a turning movement goes slower but with a bigger curvature than the robot further from the center of the turning.

### C. Splitting and merging

The formation splitting and merging process is realized fully autonomously using the RHC stabilization method presented in this paper. Firstly, let us analyse in which place before the crossroad of desired paths to split the formation. Two opposite requirements have to be satisfied. 1) The point of splitting needs to be postponed to as late as possible, since the robots connected to a single team better avoid collisions within the formation and with obstacles. Then, the coordination of robots may be ensured by the proposed



formation driving approach. 2) The formation have to be splitted under the control of independent virtual leaders once the planning horizon reaches the crossroad. From this point, the planning horizons have to follow different directions of the desired roads. Therefore, the splitting point is placed in distance  $l_{spl}$  ahead of the center of the crossroad.  $l_{spl}$  is an upper bound of the length of the planning horizon:  $l_{spl} = N\Delta t \max_{\tau \in (t; t+N\Delta t)}(v_{max,L}(\tau))$ . In the switching process, the virtual leader agent leading the old formation is killed and new virtual leaders for each arising formations are created. Dedicated robots (former followers) equipped as GeNav leaders switch on the GeNav navigation algorithm and the old GeNav leader becomes a follower if it is not employed to lead one of the new formations.

The place of the formation merging is also restricted by two antagonistic requirements: 1) again the sub-groups should be merged as soon as possible to enable the cooperative movement and 2) the virtual leaders of sub-formations have to follow parallel desired paths. Therefore, the formations are merged if the positions of virtual leaders of all formations are behind the crossroad of their desired paths. The merging process is begun once all the sub-formations are waiting in the merging position. Reversely to the splitting, the redundant GeNav leaders become followers, the old virtual leaders processes are killed and a new virtual leader is created for leading the arising formation. The formations are linked through the visual relative localization, which means that the coordinate systems of the separate groups are unified via new links between MAV cameras and identification patterns on an MAV or UGV robot. Possible deviations in positions of particular groups that are caused by positioning error of the visual navigation are compensated in the next few steps of the periodical RHC replanning.

## V. VERIFICATION EXPERIMENTS

Results presented in this section have been obtained using the proposed algorithm with parameters:  $n = 2$ ,  $N = 8$  and  $\Delta t = 0.25s$ . We have employed the Sequential Quadratic Programming (SQP) method [10] for solving the optimization problems used in the virtual leader trajectory tracking and for the stabilization and obstacle avoidance of followers. This solver provided the best performance from the tested available algorithms. Nevertheless, one can use any optimization method, which is capable to solve such an optimization problem.

The performance of the proposed approach in a complex mission with static and dynamic obstacles is shown in the video available on-line at [11] and reported in [8]. In the experiment, the formation driving technique is employed in a scenario with a heterogeneous team of 4 MAV followers and 8 UGV followers led by 1 UGV GeNav leader and 1 virtual leader. The formation is periodically moving through three rooms connected by a corridor. Three MAVs are positioned in a lower altitude to be able to relatively localize the ground robots and the fourth MAV is flying above to provide relative positions of the lower MAVs. The objective of the mission is to follow a given path and to keep a desired shape of the

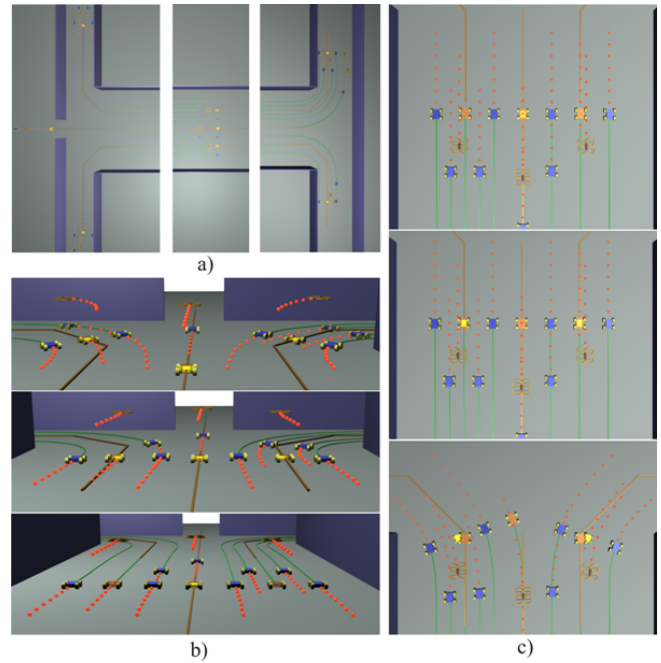


Fig. 4. Formation splitting and merging. a) Overview of the scene with depicted 3 single, 1 merged and 2 splitted formations. b) The merging process. c) The splitting process.

formation (the shape can be autonomously changed only due to an obstacle avoidance).

During the experiment, performance of the formation driving resulting from the presented concept is shown. The formation is temporarily shrunk to pass a narrow passage. Then, it is avoiding overhead obstacles that are sufficiently high to be passed under by all robots except the MAV flying in the highest altitude. The GeNav leader can be navigated without any influence of the obstacle, but the rest of the formation has to move away from the desired path to keep the constraints given by the relative localization, which results in the deviation of the virtual leader from position of the GeNav leader. This enables to avoid the obstacle in a way that the obstacle is always situated outside the dilated convex hull of the formation. Besides, the turning in connections of path segments of the desired path is demonstrated. The virtual leader and the followers are always waiting for the GeNav leader, which is turning on the spot. The formation is deviated from the path to be able to smoothly continue without any complicated manoeuvring. A failure tolerance (steering of a follower is blocked) of the system is presented with highlighted responses of other robots to predictions of possible collisions. Finally, manoeuvres for avoiding unknown and dynamic obstacles are presented. The first obstacle is avoided using the virtual leader's obstacle avoidance function at the price of temporarily leaving the desired path. The second dynamic obstacle cannot be avoided by the virtual leader's re-planning, since it was detected too late by followers. Therefore, the shape of the formation has to be temporarily changed (by the follower's re-planning) to keep the obstacle outside the dilated convex hull.



In the second simulation, the ability of the formation merging from smaller separate teams (Fig. 4 b)) and the consequence splitting back into independent units (Fig. 4 c)) is shown. In the first snapshot in Fig. 4 b), the smaller formation consisting of GeNav leader and 2 followers (MAV and UGV) is waiting in the merging point for the two formations. The first one consists of the GeNav leader, 1 MAV follower and 3 UGV followers. The second one consists of GeNav leader, 1 MAV follower and 3 UGV followers. Once the merging point is reached, the three virtual leaders leading the separate formations are switched off and a new virtual agent is created in the position of the middle robot equipped as the GeNav leader. The two remaining GeNav leaders in the former outer formations become followers and the whole group continues led by one shared GeNav leader and one virtual leader into the splitting point at the end of the wide corridor. In this point, the formation is divided into two new sub-formations, each led by own virtual and GeNav leaders. The GeNav leader employed for navigation of the large formation becomes a follower.

The ability of the obstacle avoidance by temporary shrinking of the formation is shown also in the hardware experiment in Fig. 5. The Cameleon robot from ECA company has been employed as the leader of the formation carrying the localization tags for the system of visual relative localisation on-board of MAVs. Two MikroKopter quad-rotors have been used to the formation stabilization and the UGV following. In Fig. 5 beside the pictures from the experiment, one can see visualisation of plans of the robots found by the presented approach. An experiment of the formation movement in connections of path segments can be found in the report in [8] and in video record of the experiment in [11]. In the experiment, the Pioneer 3-AT robotic platform is employed as the GeNav leader and two MMP5 platforms and the Ar.Drone MAV act as followers. To be able to follow the proposed approach, the MAV is equipped with a bottom monocular camera and with a vision system [12] being able to identify location and size of color dresses of UGVs in the image. This information is used for the relative localization of all members of the formation. Beside the pictures of the formation movement, images used for the GeNav visual navigation and for the top-view relative localization are shown in [8].

## VI. CONCLUSION

A novel approach for stabilization and navigation of 3D UGV-MAV formations with splitting and merging abilities was presented in this paper. The proposed formation driving approach is based on visual navigation and relative localization techniques using simple on-board sensors. The method aims to enable utilization of teams of closely cooperating micro-scale robots in environment without any pre-installed global localization system.

## VII. ACKNOWLEDGEMENTS

This work was supported by GAČR under M. Saska's postdoc grant no. P10312/P756 and by MŠMT under project

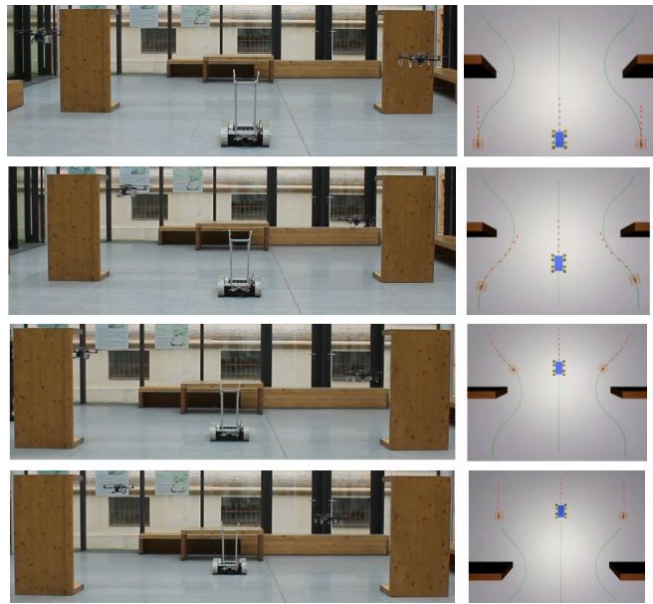


Fig. 5. Formation stabilization based on the visual relative localization.

Kontakt II no. LH11053.

## REFERENCES

- [1] J. Faigl, T. Krajník, J. Chudoba, L. Preucil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *Proc. of IEEE International Conference on Robotics and Automation*, 2013.
- [2] T. Krajník, J. Faigl, M. Vonásek, V. Kulich, K. Košnar, and L. Přeučil, "Simple yet stable bearing-only navigation," *J. Field Robot.*, 2010.
- [3] Y. Liu and Y. Jia, "An iterative learning approach to formation control of multi-agent systems," *Systems & Control Letters*, vol. 61, no. 1, pp. 148 – 154, 2012.
- [4] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," *Auton. Robots*, vol. 33, no. 1-2, pp. 143–156, 2012.
- [5] M. Saffarian and F. Fahimi, "Non-iterative nonlinear model predictive approach applied to the control of helicopters group formation," *Robotics and Auton. Syst.*, vol. 57, no. 67, pp. 749 – 757, 2009.
- [6] C. Liu, W.-H. Chen, and J. Andrews, "Piecewise constant model predictive control for autonomous helicopters," *Robotics and Autonomous Systems*, vol. 59, no. 78, pp. 571 – 579, 2011.
- [7] Z. Chao, S.-L. Zhou, L. Ming, and W.-G. Zhang, "Uav formation flight based on nonlinear model predictive control," *Mathematical Problems in Engineering*, vol. 2012, no. 1, pp. 1–16, 2012.
- [8] M. Saska, T. Krajník, V. Vonásek, P. Vanek, and L. Preucil, "Navigation, localization and stabilization of formations of unmanned aerial and ground vehicles," in *ICUAS*, 2013.
- [9] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl, "AR-Drone as a Platform for Robotic Research and Education," in *Research and Education in Robotics: EUROBOT 2011*. Heidelberg: Springer, 2011.
- [10] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.
- [11] Movie, "Movie of hw experiment and simulation presented in this paper [online]. <http://imr.felk.cvut.cz/formations/> [cit. 2013-7-9]," 2013.
- [12] M. Saska, T. Krajník, and L. Přeučil, "Cooperative micro uav-ugv autonomous indoor surveillance," in *IEEE SSD*, 2012.

# Toward Smooth and Stable Reactive Mobile Robot Navigation using On-line Control Set-points

Lounis Adouane

Institut Pascal, UBP/IFMA UMR CNRS 6602, France

e-mail: Lounis.Adouane@univ-bpclermont.fr

**Abstract**—This paper deals with the challenging issue of on-line mobile robot navigation in cluttered environment. Indeed, it is considered in this work, a mobile robot discovering the environment during its navigation, it should thus, to react to unexpected events (e.g., obstacles to avoid) while guaranteeing to reach its objective. Nevertheless, in addition to avoid safely and on-line these obstacles, it is proposed to enhance the smoothness of the obtained robot trajectories. Otherwise, to quantify this smoothness, suitable indicators were used. Specifically, this paper proposes to appropriately link on-line set-points defined using elliptic limit-cycle trajectories with a multi-controller architecture which guarantees the stability (according to Lyapunov synthesis) and the smoothness of the switch between controllers. Moreover, a comparison between fully reactive mode (the aim of this paper) and planned mode is given through the proposed control architecture which could exhibit the two aspects. Many simulations in cluttered environments permit to confirm the reliability and the robustness of the overall proposed reactive control.

## I. INTRODUCTION

Among the main challenges to obtain a fully autonomous mobile robot navigation, is the ability for the robot to react on-line to unpredictable events encountered in its environment. The asked question is thus *how to navigate toward a goal in a cluttered environment when obstacles are discovered in real time?* [13]. Nevertheless, it is not sufficient to avoid these obstacles. In fact, robot should also guarantee a smooth navigation [7] for the comfort, for example of the passengers. In [8], the author characterizes this smooth navigation while using a cost functional  $J$  that reflects the trade-off between the travel time and the integral of acceleration (which characterizes the amount of jerking of angular and linear robot velocity). All these criterion are concatenated in one and modulated by weights which give thus the priority for each one.

To obtain on-line, accurate, flexible and reliable navigation, one part of the literature in this domain considers that the robot is fully actuated with no control bound and focuses the attention on path planning and re-planning. Voronoï diagrams and visibility graphs [12], navigation functions [18] or planning based grid checking and trajectory generation [17] are among these road-map-based methods. However, the other part of the literature considers that to control a robot with the above criterion, it is essential to accurately take into

account: robot structural constraints (e.g., nonholonomy); avoid command discontinuities and set-point jerk, etc. Our proposed control architecture is linked to this last approach, thus where the control stability is rigorously demonstrated.

It is commonly used in the literature a pre-planned reference trajectory, which means that it was appropriately planned or selected before robot movement [14]. However, in real motion conditions where the environment can to be very cluttered and with high dynamic, these methods could not be very efficient due, among others, to time consuming to obtain the new re-planned trajectory [13]. Otherwise, a large class of model-based techniques use optimization to choose between a set of admissible trajectories [5], [16]. In the proposed paper, it is defined a fully reactive mobile robot navigation. Indeed, at each sample time, the robot should follows defined set-points, according to local robot perceptions and objectives.

To guarantee multi-objective criteria, control architectures can be elaborated in a modular and bottom-up way as introduced in [6] and so-called behavioral architectures [3]. These techniques are based on the concept that a robot can achieve a complex global task while using only the coordination of several elementary behaviors. In fact, to tackle this complexity, behavioral control architecture decompose the global controller into a set of elementary behavior/controller (e.g., attraction to the objective, obstacle avoidance, trajectory following, etc.) to master better the overall robot behavior. Moreover, it is considered in a lot of studies the investigation of the potentialities of the hybrid systems controllers [21] to provide a formal framework to demonstrate the robustness and the stability of such architecture. In their most simple description, hybrid systems are dynamical systems modeled as a finite state automaton. These states correspond to a continuous dynamic evolution, and the transitions can be enabled by particular conditions reached by the continuous part. This formalism permits a rigorous automatic control analysis of the performances of the control architecture [4].

Among controllers which can make up a behavioral control architecture, obstacle avoidance controllers play a large role to achieve autonomously and safely the navigation of mobile robots in a cluttered and unstructured environments. An interesting overview of obstacle avoidance methods is accurately given in [13]. The proposed control architecture integrates obstacle avoidance method which uses limit-cycle vector field [10], [11], [1]. Moreover, it introduces an adap-

\*This work was supported by the French National Research Agency (ANR) through the Safeplatoon project.

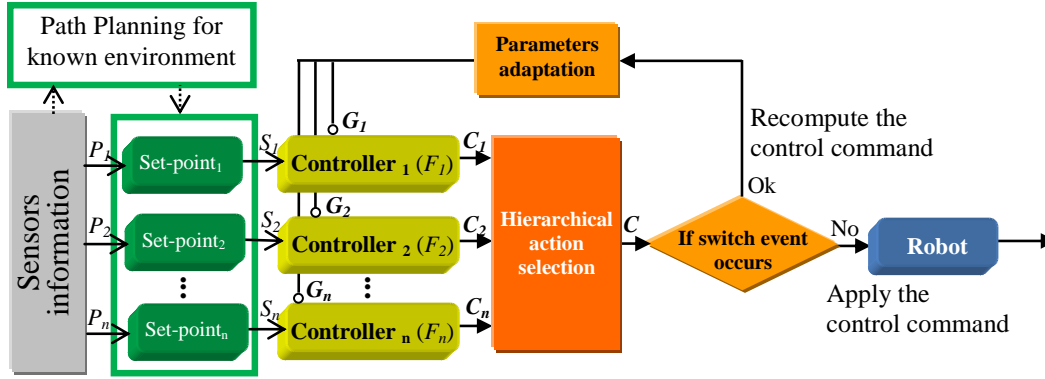


Fig. 1. The proposed hybrid control architecture for mobile robot navigation

tive and flexible mechanism of control which guarantees the stability and the smoothness of the switch between controllers.

The rest of the paper is organized as follows. Section II gives the specificities of the proposed control architecture. In section III, the control architecture is applied to the task of navigation in the presence of obstacles. It presents the model of the considered robot and the different modules constituting the proposed control architecture. Section IV deals with safety mode mechanism. Section V is devoted to the description and analysis of the simulation results. This paper ends with some conclusions and further work.

## II. CONTROL ARCHITECTURE

The proposed control architecture (cf. Figure 1) is dedicated for the general framework of the navigation of mobile robots in cluttered environments. It permits to manage the interactions between different elementary controllers while guaranteeing the stability and the smoothness of the overall control. Moreover, a specific “safety mode” is proposed in section IV to avoid undesirable robot behaviors (oscillations, abrupt movement, etc.). The robot can therefore have very smooth trajectories while guaranteeing safe obstacle avoidance. The specific blocks composing this generic overall control architecture are detailed below. In section III a concrete control architecture applied for a real task is given.

### A. Path planning for known environment

This path planner (and re-planner) block is activated only if the entire mission is well known or when the navigation is achieved in relatively low dynamic environment. The aim of the proposed paper is to make the focus only around reactive mobile robot navigation (where the environment is discovered on-line). Therefore, the used path planner and its interaction with low level control are not addressed here. This part of the control was studied in [15] and will be the subject of a future developments.

### B. Set-points blocks

These blocks, which have as input the perceptions  $P_i$ , are responsible to give for each dedicated controller block (e.g., obstacle avoidance, target to reach, etc.) the set-points useful

for its working (e.g., for attraction to target controller, the relative position of the target to reach).

### C. Controllers blocks

Every controller  $F_i$  is characterized by a stable nominal law which is represented by the function:

$$F_i(S_i, t) = \eta_i(S_i, t) \quad (1)$$

with  $S_i$  is the set-point sent to the controller “ $i$ ”. Otherwise, in order to avoid the important controls jumps at the time for example of the switch between controllers (e.g., from the controller “ $j$ ” toward the controller “ $i$ ” at the instant  $t_0$ ), an adaptation of the nominal law is proposed,  $F_i$  becomes thus:

$$F_i(S_i, t) = \eta_i(S_i, t) + G_i(S_i, t) \quad (2)$$

with  $G_i(S_i, t)$  (cf. Equation 3) is a strictly monotonous function that tends to zero after a certain amount of time “ $T = H_i(P_i, S_i)$ ”. The value of this time depends on the criticality of the controller $_i$  to join as quickly as possible the nominal law  $\eta_i(S_i, t)$ . It constitutes thus the controller safety mode (cf. Section IV for a specific example for obstacle avoidance controller).

$$G_i(S_i, t_0) = F_j(S_j, t_0 - \Delta t) - \eta_i(S_i, t_0) \quad (3)$$

where  $\Delta t$  represents the sampling time between two control set-points and  $t_0$  is the time of abrupt change in  $S_i$ .

The definition of  $G_i(S_i, t)$  allows to guarantee that the control law (cf. Equation 2) tends toward the nominal control law after a certain time  $T$ , thus:

$$G_i(S_i, T) = \varepsilon \quad (4)$$

Where  $\varepsilon$  very small constant value  $\approx 0$ . The adaptive function  $G_i(S_i, t)$  is updated by the “Parameters adaptation” block every time a hard control switch concerning the “ $i$ ” controller occurs (cf. Section II-D) (cf. Figure 1). The main challenge introduced by this kind of control is to guarantee the stability of the updated control law (cf. Equation 2) even during the period where  $|G_i(S_i, t)| \gg \varepsilon$ .

#### D. Parameters adaptation block

This block has as input the “conditional block” (cf. Figure 1) that verifies if specific control switch event occurs. So, if it is the case then it must update “Adaptive Function” corresponding to the future active controller (cf. Equation 3). The different configurations which need the activation of parameters adaptation block are given below:

- 1) When a controller which should be active at the current “ $t$ ” instant is different than the one which was active at the “ $t-\Delta t$ ” instant,
- 2) When an abrupt transition in the set-points  $S_i$  of the controller $_i$  is encountered.

### III. NAVIGATION IN PRESENCE OF OBSTACLES TASK

The navigation in a cluttered environment aims here to lead the robot to reach a target-position while avoiding obstacles (cf. Figure 2). The robot movement needs to be fast and smooth while avoiding statical and dynamical obstacles which could have different shapes.

One supposes in the setup that robot and obstacles are surrounded by respectively cylindrical and elliptical boxes (cf. Figure 2). The cylindrical box (the robot) is characterized by  $R_R$  radius and elliptical boxes (obstacles) are given by:

$$a(x-h)^2 + b(y-k)^2 + c(x-h)(y-k) = 1 \quad (5)$$

With:

- $h, k \in \mathbb{R}$ , give the coordinate of the center of the ellipse,
- $a \in \mathbb{R}^+$ , permits to give the half length  $A = 1/\sqrt{a}$  of the longer side (major axis) of the ellipse,
- $b \in \mathbb{R}^+$ , permits to give the half length  $B = 1/\sqrt{b}$  of the shorter side (minor axis) of the ellipse (thus  $b > a$ ),
- $c \in \mathbb{R}$ , permits to give the ellipse orientation  $\Omega = 0.5\arctan(c/(b-a))$  (cf. Figure 2). When  $a = b$  equation 5 becomes a circle equation ( $\Omega$  will do not gives thus any more information).

The choice of ellipse box rather than circle as used in [11], [9] or [1] is to have one more generic and flexible mean to surround and fit accurately different kind of obstacles shapes (specifically longitudinal shapes [2]).

The surrounded ellipse parameters ( $h, k, A, B$  and  $\Omega$ ) (cf. equation 5 and figure 2) can be obtained on-line, while using an appropriate weighted least square method on the data range given by the robot infrared sensors [19]. An extension of this approach while using Extended Kalman Filter and an appropriate heuristic is given in [20].

#### A. Mobile robot model

Before proposing appropriate elementary controllers to achieve the considered task, it is important to know the robot model. Its model is given by the well known kinetic model of a unicycle robot (cf. Figure 2):

$$\dot{\xi} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (6)$$

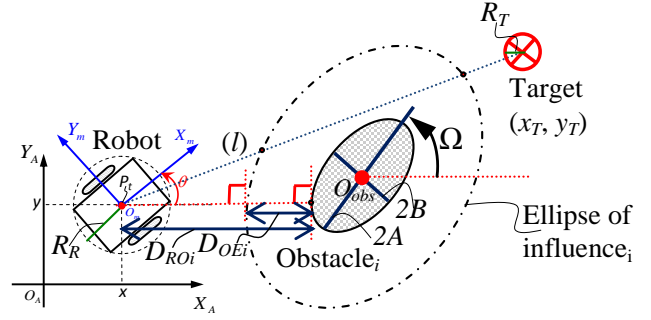


Fig. 2. Robot pose and the used perceptions for the navigation.

With  $x, y, \theta$  correspond to configuration state of the unicycle and  $v$  and  $w$  correspond respectively to linear and angular velocity of the robot at the point “ $P_t$ ”.

Knowing the model of the robot as well as the task to achieve, we present below the controller of *Attraction to the target* and the *Obstacle avoidance* which are necessary to the mobile robot navigation in cluttered environment. In section III-C the control law used for the two controllers is presented.

#### B. The used controllers

1) *Attraction to the target controller*: This controller leads the robot toward the target to reach. This target is represented by a circle of  $(x_T, y_T)$  center and  $R_T$  radius (cf. Figure 2).

2) *Obstacle avoidance controller*: The objective of this controller is to avoid obstacles which hinder the robot movement toward the target. In what follows we will give only few details about the overall obstacle avoidance algorithm in order to make more the focus on the proposed mechanisms of control which can guarantee at the same time: the stability and the smoothness of the switch between controllers. More details about the proposed obstacle avoidance algorithm are given in [1] and [2].

To implement the obstacle avoidance behavior, limit-cycles was used [10], [1]. The differential equations giving elliptic limit-cycles are:

- For the clockwise trajectory motion (cf. Figure 3(a)):

$$\begin{aligned} \dot{x}_s &= y_s + x_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_s y_s) \\ \dot{y}_s &= -x_s + y_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_s y_s) \end{aligned} \quad (7)$$

- For the counter-clockwise trajectory motion (cf. Figure 3(b)):

$$\begin{aligned} \dot{x}_s &= -y_s + x_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_s y_s) \\ \dot{y}_s &= x_s + y_s(1 - x_s^2/A_{lc}^2 - y_s^2/B_{lc}^2 - cx_s y_s) \end{aligned} \quad (8)$$

where  $(x_s, y_s)$  corresponds to the position of the robot according to the center of the ellipse;  $A_{lc}$  and  $B_{lc}$  characterize respectively major and minor elliptic axis (cf. Figure 2);  $c$  if  $\neq 0$  gives the  $\Omega$  ellipse angle (cf. Section III).

Figure 3 shows that the ellipse of a major axis =  $2A_{lc} = 4$  and of minor axis =  $2B_{lc} = 2$  is a periodic orbit. This periodic orbit is called a limit-cycle [10]. Figure 3(a) and 3(b) show the shape of equations (7) and (8) respectively.



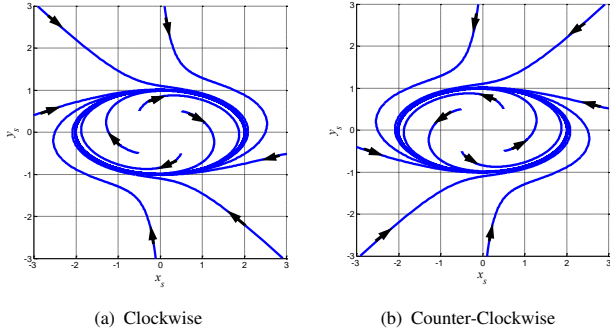


Fig. 3. Shape possibilities for the used elliptic limit-cycles for different initial conditions  $(x_0, y_0)$ .

They show the direction of trajectories (clockwise or counter-clockwise) according to  $(x_s, y_s)$  axis. The trajectories from all points  $(x_s, y_s)$  of  $X, Y$  reference frame, including inside the ellipse, move towards the ellipse.

Summarily, the obstacle avoidance algorithm used in the paper follow these steps [2]:

- Detect the most disturbing obstacle which avoids the robot to reach the target, i.e., it is enough here to know if it exists an intersect points between the line “ $l$ ” and the *Ellipse of influence* (cf. Figure 2). In fact, it is defined for each perceived obstacle an *Ellipse of influence* which has the following features:
    - The same center  $(h, k)$  and tilt angle  $\Omega$  as the ellipse which surround the obstacle,
    - The value of its major axis is  $2A_{lc}$  with  $A_{lc} = A + R_R + Margin$ ,
    - The value of its minor axis is  $2B_{lc}$  with  $B_{lc} = B + R_R + Margin$ .
- Where *Margin* corresponds to a safety tolerance which includes: perception uncertainty, control reliability and accuracy, etc.
- According to the relative position of the robot with regard to the disturbing obstacle and to the target to reach, the direction of avoidance (clockwise or counter-clockwise) is taken,
  - The robot passes after by two steps, go into the orbit of the obstacle <sub>$i$</sub>  to avoid (*Attractive phase*) and after go out the orbit of the obstacle <sub>$i$</sub>  (*Repulsive phase*).

### C. The used control law

To make a focus specifically around the efficiency of the proposed adaptive control mechanism a simple control law is used:

$$v = v_{\max} e^{-K_v/d} \cos(\theta_e) \quad (9a)$$

$$w = \dot{\theta}_d + K_p \theta_e \quad (9b)$$

where  $v_{\max}$  is the robot maximum linear velocity,  $K_v$  and  $K_p$  are constant values  $\in \mathbb{R}^+$ , and  $d$  is the distance between the robot and the target when the *attraction to the target* controller is activated, and  $d$  is equal to  $D_{ROi}$  (cf. Figure 2) if the *obstacle avoidance* is activated. The robot reaches

the target when  $0 < d \leq R_T$  (cf. Figure 2).  $\theta_e$  is the angular error given by:

$$\theta_e = \theta_d - \theta \quad (10)$$

The desired robot orientation  $\theta_d$  is given according the following two cases:

1)

$$\theta_d = \arctan\left(\frac{y_T - y}{x_T - x}\right) \quad (11)$$

Where  $(x, y)$  and  $(x_T, y_T)$  correspond respectively to the position of the robot and the target (cf. Figure 2) in the case of the activation of *attraction to the target* controller.

2) and it is equal to:

$$\theta_d = \arctan\left(\frac{\dot{y}_s}{\dot{x}_s}\right) \quad (12)$$

Where  $\dot{x}_s$  and  $\dot{y}_s$  are given by differential equation of the limit-cycle (7) or (8)) in the case of the activation of *obstacle avoidance* controller.

It is interesting to note that only one control law is applied to the robot even if its control architecture contains two (or more) different controllers. Only the set-points change according to the applied controller.

In what follows, a study is given to use the adaptive control mechanism on the nominal angular control law (9b). While using (9b), it is straightforward to demonstrate that the evolution of  $\theta_e$  will be given by:

$$\dot{\theta}_e = -K_p \theta_e \quad (13)$$

To guarantee the right transition between controllers as described in section (II-C), the modification of the controller law must be done, it becomes thus:

$$w = \dot{\theta}_d + K_p \theta_e + G(t) \quad (14)$$

where  $G(t)$  the adaptive function.  $\dot{\theta}_e = \dot{\theta}_d - \dot{\theta}$  will be given now by:

$$\dot{\theta}_e = -K_p \theta_e - G(t) \quad (15)$$

Let's consider the following Lyapunov function

$$V = \frac{1}{2} \theta_e^2 \quad (16)$$

$\dot{V}$  is equal then to  $\theta_e \dot{\theta}_e = -K_p \theta_e^2 - G(t) \theta_e$ . To guarantee that the proposed controller is asymptotically stable, we must always have  $\dot{V} < 0$ , thus:

$$K_p > -\frac{G(t)}{\theta_e} \quad (17)$$

Where  $G(t)$  is a function chosen with respect to the constraints given in sections (II-C and II-D) and to the fact that it decreases more quickly to zero than  $\theta_e$ .



#### D. Hierarchical action selection block

The proposed control architecture uses a hierarchical action selection mechanism to manage the switch, between two or even more controllers, according to environment perception. Obstacle avoidance strategy is integrated in a more global multi-controller architecture. Otherwise, the controllers' activations are achieved in a reactive way as in [6]. The proposed algorithm 1 activates the obstacle avoidance controller as soon as it exists at least one obstacle which can obstruct the future robot movement toward its target.

**if** *It exists at least one constrained obstacle*  
*{i.e., it exists at least one intersect point between the line "l" and the ellipse of influence (cf. Figure 2)}* **then**  
 | Activate *Obstacle avoidance* controller  
**else**  
 | Activate *Attraction to the target* controller  
**end**

**Algorithm 1:** Hierarchical action selection

#### E. Parameters adaptation block

In the applied navigation task, the "conditional" block activate the "parameters adaptation" block (cf. Figure 1) when at least one of the following switch events occurs:

- the "Hierarchical action selection" block chose to switch from one controller to another,
- the "obstacle avoidance" algorithm chose another obstacle to avoid,
- the "obstacle avoidance" controller switch from attractive phase to the repulsive phase (cf. Section III-B.2).

#### IV. OBSTACLE AVOIDANCE SAFETY MODE

The adaptive function  $G(t)$  (cf. Equation 14) permits mainly to obtain smooth control when a switch event occurs. However, during " $T$ " time (cf. Section II-C) the obstacle avoidance controller is far from its nominal law (given when  $G(t) \neq 0$ ) and the robot can collide with obstacles. Therefore, to insure the smoothness of the control without neglecting the robot safety,  $G$  will be parameterized according to the robot-obstacle distance " $d = D_{RO_i}$ " (cf. Figure 2),  $G$  becomes thus:

$$G(t, d) = Ae^{Bt} \quad (18)$$

Where:

- $A$  value of the control difference between the control at the instants " $t - \Delta t$ " and " $t$ " (cf. Equation 3),
- $B = \log\left(\frac{\varepsilon}{|A|}\right)/T(d)$  with:
  - $\varepsilon$  very small constant value  $\approx 0$  (cf. Equation 4),
  - $\begin{cases} T(d) = T_{max} & \text{if } d > D_{OE_i} \\ T(d) = c.d + e & \text{if } D_{OE_i} \geq d \geq D_{OE_i} - p.Margin \\ T(d) = \varepsilon & \text{if } d < D_{OE_i} - p.Margin \end{cases}$

Where:

- $D_{OE_i}$  corresponds to the distance Obstacle-Ellipse of Influence (cf. Figure 2),

- $Margin$  defined in sub-section III-B.2,
- $p$  positive constant  $< 1$  which allows to adapt the maximum distance " $d$ " where the adaptive function must be resetting to zero. As small as  $p$  is, more the priority is given to the safety behavior instead to the smoothness of controllers switch,
- $c = T_{max}/p.Margin$
- $e = T_{max}(1 - D_{OE_i}/p.Margin)$

Therefore,  $T(d)$  goes from  $T_{max}$  until 0 while following a linear decrease. If the robot is out of  $D_{OE_i}$  than  $T = T_{max}$  and decrease linearly to become 0 when  $d < D_{OE_i} - (p.Margin)$ . This function permits thus, when  $d < D_{OE_i} - (p.Margin)$ , to remove completely the effect of adaptive control (which promote the smoothness of control) and insures thus the complete safety of the robot navigation.

#### V. SIMULATION RESULTS

In this section, many simulations on different robot configurations and cluttered environments will permits to confirm the reliability and the robustness of the proposed control architecture (cf. Figure 1). Figure 4 shows the smoothness of the obtained robot trajectories. It shows also the clockwise and counter-clockwise obstacle avoidance using on-line set-point based elliptic limit-cycle. In figure 4(a), it is showed the tracks of "limit-cycle planned path", which is not really followed by the robot, in fact, at each sample time, the robot computes the new control set-points given by equations (7) and (8). The showed planned track corresponds to the limit-cycle path obtained the first time that the robot see the obstacle to avoid, this trajectory do not take into account the robot constraints, e.g., its nonholonomy (cf. Equation 6).

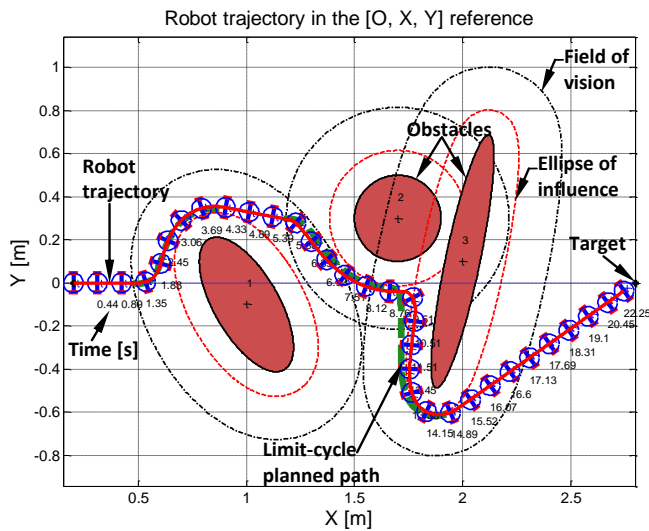
Figures 5 (c) and (d) shows respectively the progress of  $v$  and  $w$  robot velocities when the adaptive functions are used (cf. Equation 3). These controls are thus less abrupt and smoother than those obtained without adaptive functions (cf. Figures 5 (a) and (b)). In addition, to demonstrate the real smoothness enhancement of the obtained trajectories, a statistical survey was made while doing a large number of simulations in different cluttered environments and with for each one, a navigation with and without adaptive function is performed. We did specifically 1000 simulations with every time, 10 obstacles with different random positions in the environment (cf. figure 4(b) for an example of trajectory). Otherwise, to quantify the smoothness of the obtained robot trajectories [7], [8], it is proposed to use these two indicators:

$$I_v = \int_0^{T_T} |\dot{v}| dt \quad (19)$$

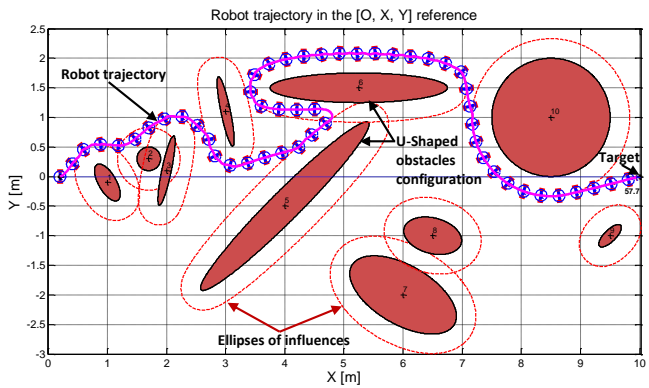
and

$$I_w = \int_0^{T_T} |\dot{w}| dt \quad (20)$$

Where  $\dot{v}$  and  $\dot{w}$  correspond respectively to linear and angular robot acceleration, and  $T_T$  is the necessary time, for



(a) Environment with 3 obstacles



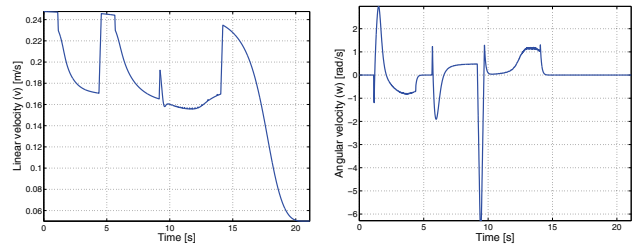
(b) Environment with 10 obstacles

Fig. 4. Some smooth robot trajectories obtained with the proposed on-line control architecture.

the robot, to reach the target. According to these indicators we can observe a significant gain in the smoothness of  $v$  and  $w$  controls which are equal respectively to 30% and 35%.

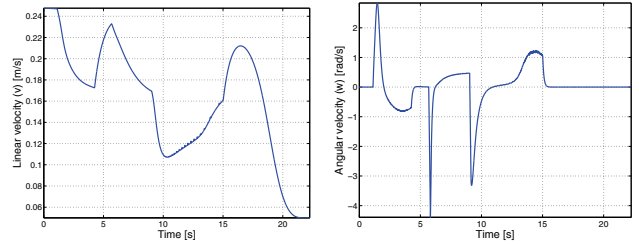
The second step of simulations permits to demonstrate the relevance of the proposed safety mode, especially when the robot navigates very close to obstacles. Figure 6 shows the case where obstacle avoidance controller apply or not the safety mode (cf. Section IV). When it do not apply it, the robot hit the obstacle (cf. Figure 6(a)). Figures 7 (a) and (b) give the evolution of adaptive functions when the safety mode is applied. We observe in these figures that the maximal time  $T_{max}$  to achieve the interpolation ( $\approx 3s$  in the simulation (cf. Section IV)) decreases every time that the robot moves dangerously closer to the obstacle (cf. Figure 6(b)). Figure 7(c) shows that the overall proposed structure of control is always stable even when the adaptive safety mode is applied.

The two peaks shown in Figure 7(c) correspond respectively to the phase of the attraction toward the elliptic limit-cycle and the repulsion from this one. The applied algorithm is accurately explained in [1] and [2].



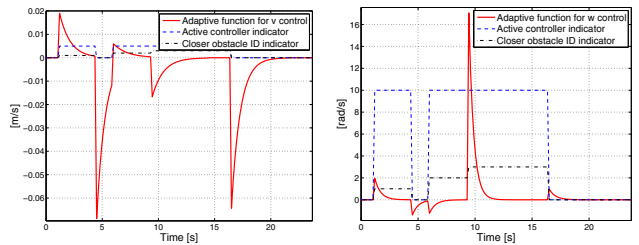
(a)  $v$  without AF

(b)  $w$  without AF



(c)  $v$  with AF

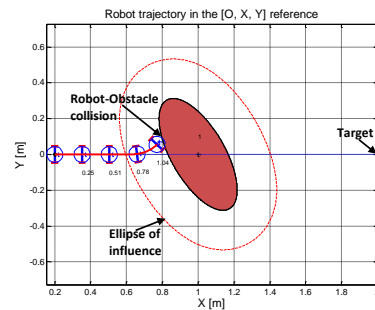
(d)  $w$  with AF



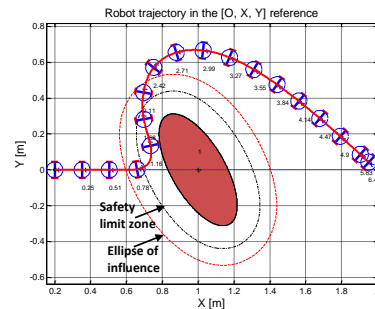
(e) AF evolution for  $v$

(f) AF evolution for  $w$

Fig. 5. Adaptive Function (AF) influence on the  $v$  and  $w$  robot velocities (cf. Section III-C).



(a) Without safety mode



(b) With safety mode

Fig. 6. Robot trajectories with and without safety mode.

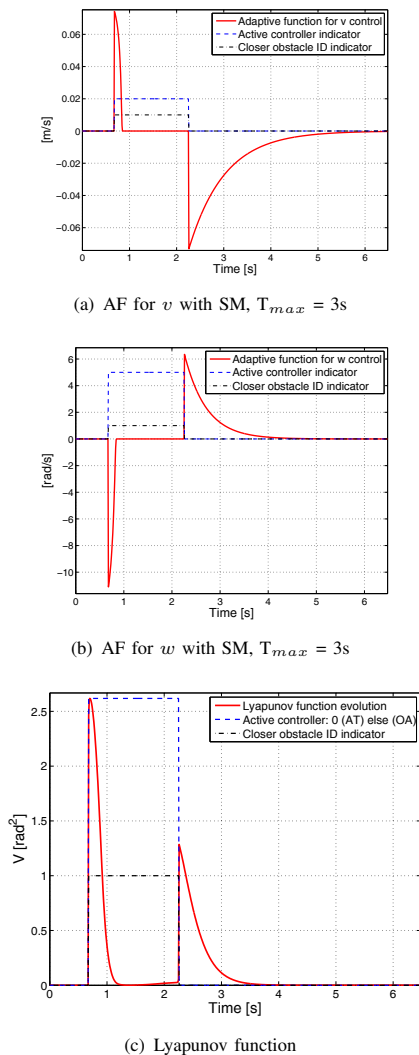


Fig. 7. Adaptive Functions (AF) with Safety Mode (SM) and Lyapunov function evolution.

## VI. CONCLUSION AND FURTHER WORK

This paper proposes to link, on-line control set-points using elliptic limit-cycle trajectories and a multi-mode control architecture which uses an adaptive mechanism to guarantee at the same time, the stability (according to Lyapunov synthesis) and the smoothness of the switch between controllers. Therefore, in addition to safe robot navigation, the robot trajectories become also smoother. Otherwise, appropriate indicators are used to quantify the trajectories smoothness. Moreover, to obtain safer robot navigation, an appropriate safety mode is proposed and experimented in cluttered environment. Many simulations confirm the reliability and the robustness of the proposed control architecture. Future works aim, first to apply this control architecture in real robots and secondly, to propose control architecture which find the right balance between reactive and cognitive aspects (planned).

## REFERENCES

- [1] Adouane, L.: Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation. In: 9th Conference on Autonomous Robot Systems and Competitions. Portugal (2009)
- [2] Adouane, L., Benzerrouk, A., Martinet, P.: Mobile robot navigation in cluttered environment using reactive elliptic trajectories. In: 18th IFAC World Congress. Italy (2011)
- [3] Arkin, R.C.: Behavior-Based Robotics. The MIT Press (1998)
- [4] Branicky, M.S.: Multiple lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Transaction on Automatic Control **43**(4), 475–482 (1998)
- [5] Brock, O., Khatib, O.: High-speed navigation using the global dynamic window approach. In: ICRA, pp. 341–346 (1999)
- [6] Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation **RA-2**, pp.14–23 (1986)
- [7] Fleury, S., Souères, P., Laumond, J.P., Chatila, R.: Primitives for smoothing mobile robot trajectories. In: ICRA (1), pp. 832–839 (1993)
- [8] Gulati, S.: A framework for characterization and planning of safe, comfortable, and customizable motion of assistive mobile robots. Ph.D. thesis, The University of Texas at Austin (2011)
- [9] Jie, M.S., Baek, J.H., Hong, Y.S., Lee, K.W.: Real time obstacle avoidance for mobile robot using limit-cycle and vector field method. Knowledge-Based Intelligent Information and Engineering Systems pp. 866–873 (2006)
- [10] Khalil, H.K.: Frequency domain analysis of feedback systems. Non-linear Systems: Chapter7, 3 edition (2002)
- [11] Kim, D.H., Kim, J.H.: A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer. Robotics and Autonomous Systems **42**(1), 17–30 (2003)
- [12] Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Boston, MA (1991)
- [13] Minguez, J., Lamiroux, F., Laumond, J.P.: Handbook of Robotics, chap. Motion Planning and Obstacle Avoidance, pp. pp.827–852 (2008)
- [14] Morin, P., Samson, C.: Control of nonholonomic mobile robots based on the transverse function approach. Trans. Rob. **25**, 1058–1073 (2009)
- [15] Mouad, M., Adouane, L., Khadraoui, D., Martinet, P.: Mobile robot navigation and obstacles avoidance based on planning and re-planning algorithm. In: 10th International IFAC Symposium on Robot Control (SYROCO12). Dubrovnik - Croatia (2012)
- [16] Ogren, P., Leonard, N.E.: A convergent dynamic window approach to obstacle avoidance. IEEE Transactions on Robotics **21**(2), 188–195 (2005)
- [17] Pivtoraiko, M., Kelly, A.: Fast and feasible deliberative motion planner for dynamic environments. In: International Conference on Robotics and Automation (2009)
- [18] Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential fields. IEEE Transactions on Robotics and Automation **8**(5), 501–518 (1992)
- [19] Vilca, M., Adouane, L., Mezouar, Y.: On-line obstacle detection using data range for reactive obstacle avoidance. In: 12th International Conference on Intelligent Autonomous System (IAS'12). Korea (2012)
- [20] Vilca, M., Adouane, L., Mezouar, Y.: Robust on-line obstacle detection using data range for reactive navigation. In: 10th International IFAC Symposium on Robot Control (SYROCO'12). Dubrovnik - Croatia (2012)
- [21] Zefran, M., Burdick, J.W.: Design of switching controllers for systems with changing dynamics. In: IEEE Conference on Decision and Control CDC'98, Tampa, FL, pp. 2113–2118 (1998)



## **Session IV**

### **Navigation, Control, Planning**

- **Keynote speaker: Tirthankar Bandyopadhyay (CSIRO, Australia)**  
**Title: Intention Aware Planning for Autonomous Vehicles**  
**Co-Authors: S. Brechtel, T. Gindele**
- **Title: Safe highways platooning with minimized inter-vehicle distances of the time headway policy**  
**Authors: Alan Ali, Gaetan Garcia, Philippe Martinet**
- **Title: Optical Flow Templates for Superpixel Labeling in Autonomous Robot Navigation**  
**Authors: Richard Roberts, Frank Dellaert**

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**





2013 IEEE/RSJ International Conference on Intelligent Robots and Systems

## Session IV

Keynote speaker: **Tirthankar Bandyopadhyay (CSIRO, Australia)**

### **Intention Aware Planning for Autonomous Vehicles**

**Abstract :** As robots venture into new application domains as autonomous vehicles on the road or as domestic helpers at home, they must recognize human intentions and behaviors in order to operate effectively. This generates a new class of motion planning, problems with uncertainty in human intention. Identifying human intentions is difficult because of the diversity, subtlety of human behaviors and the lack of a powerful "intention sensor". The intentions have to be inferred from observations about the person's behavior. This is especially true for many cases of interactions between autonomous vehicles and human agents on the road where explicit communication channels are not always available. In this talk, I will mention some of the work we have been doing in developing an intention aware planning framework for autonomous vehicles on the road. I will present the framework in the context of Partially Observable Markov Decision Processes (POMDPs) and show how the recent advances in the field make Intention Aware Planning practical on real systems.

**Biography:** Dr Tirthankar Bandyopadhyay is a Research Scientist at CSIRO Brisbane. Previously he was working with Prof. Emilio Frazzoli and Prof. Daniel Rus as a Research Scientist leading the Autonomy group at the Future Urban Mobility, (FM) at Singapore MIT Alliance for Research and Technology (SMART). He was investigating the utilization of autonomous vehicles in improving personal mobility in urban environments. They have developed a low cost autonomous platform, providing Mobility-on-Demand service in select locations of NUS campus. Prior to FM, he worked with Prof. Franz Hover and Prof. N. Patrikalakis in the Center for Environmental Sensing and Modeling (CENSAM) on developing robust navigation techniques for marine autonomous surface vehicles for Harbor like environments. They operated in Singapore waters to deploy and test our autonomous kayak. He did his Ph.D in *Motion Planning for Target Tracking* at the National University of Singapore under the supervision of Prof. David Hsu and Prof. Marcelo Ang. His research dealt with developing motion strategies in 2-D as well as in 3-D, to track and follow a target of unknown intentions in an unknown environment, using only local information. His research interest is in developing robotics for the real world.

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# Intention-Aware Planning for Autonomous Vehicles



Daniela Rus  
Emilio Frazzoli



Tirthankar Bandyopadhyay  
tirtha.bandy@csiro.au



David Hsu  
Marcelo Ang  
Wee Sun



# Real world has inherent uncertainty



An intersection in Jaipur (India)



Even structured and well regulated environments have inherent uncertainties.



# Real world has inherent uncertainty



First Robot-Robot vehicle collision at DUC

- Failure of perception ?
- Failure to plan for uncertain perception



Even when perception is perfect, the signal maybe wrong (unintended)

People sometimes fail to turn off the indicator.

**Autonomous Navigation needs to take into account such Uncertainty**



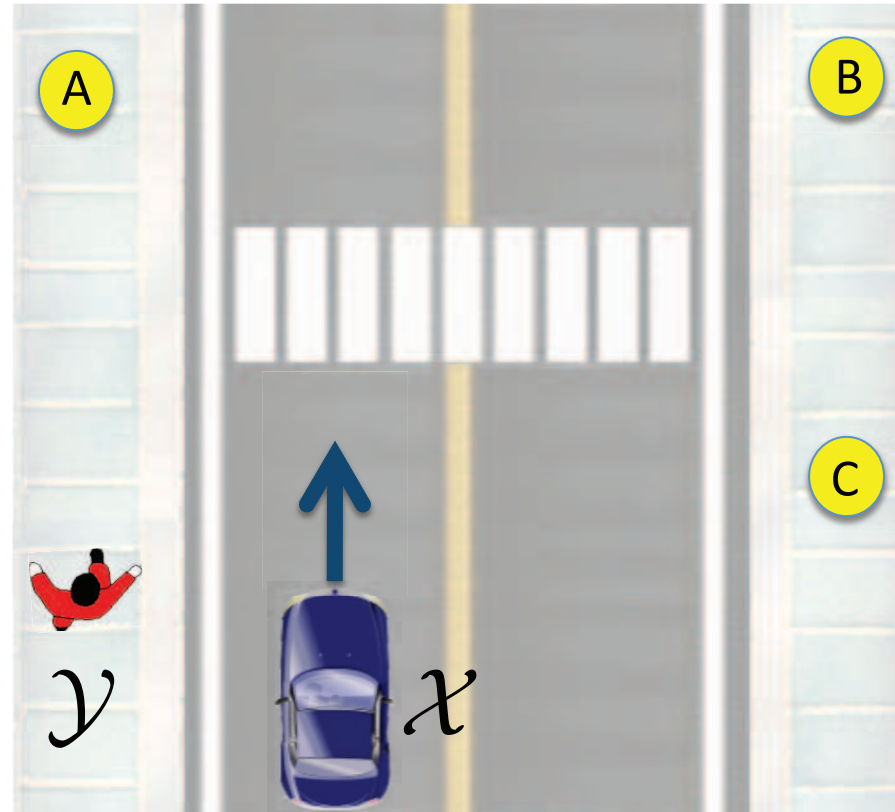
# The problem of pedestrian avoidance

$\mathcal{X}$  : Robot,  $\mathcal{Y}$  : Agent

$$T_y : f(x, y, \theta)$$

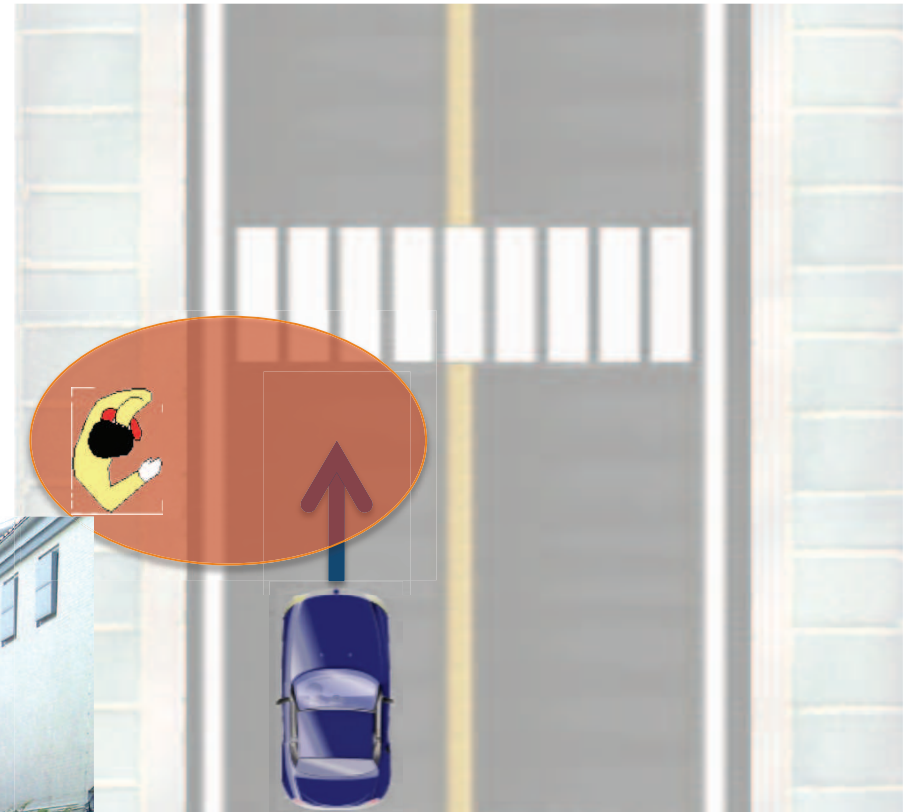
$\theta$  : unobservable

Motion planning under unknown system dynamics



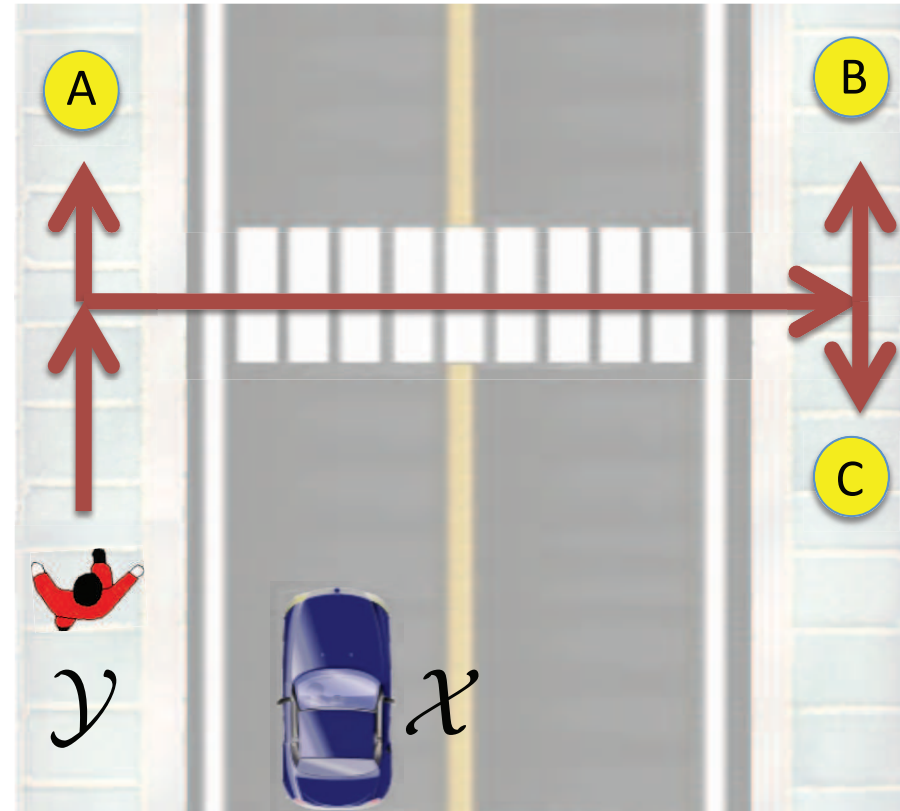
# Alternate approaches : Reactive

- Reactive control
  - high frequency worst case predictor
- Pro
  - Simple and Scalable
- Cons
  - Overly conservative

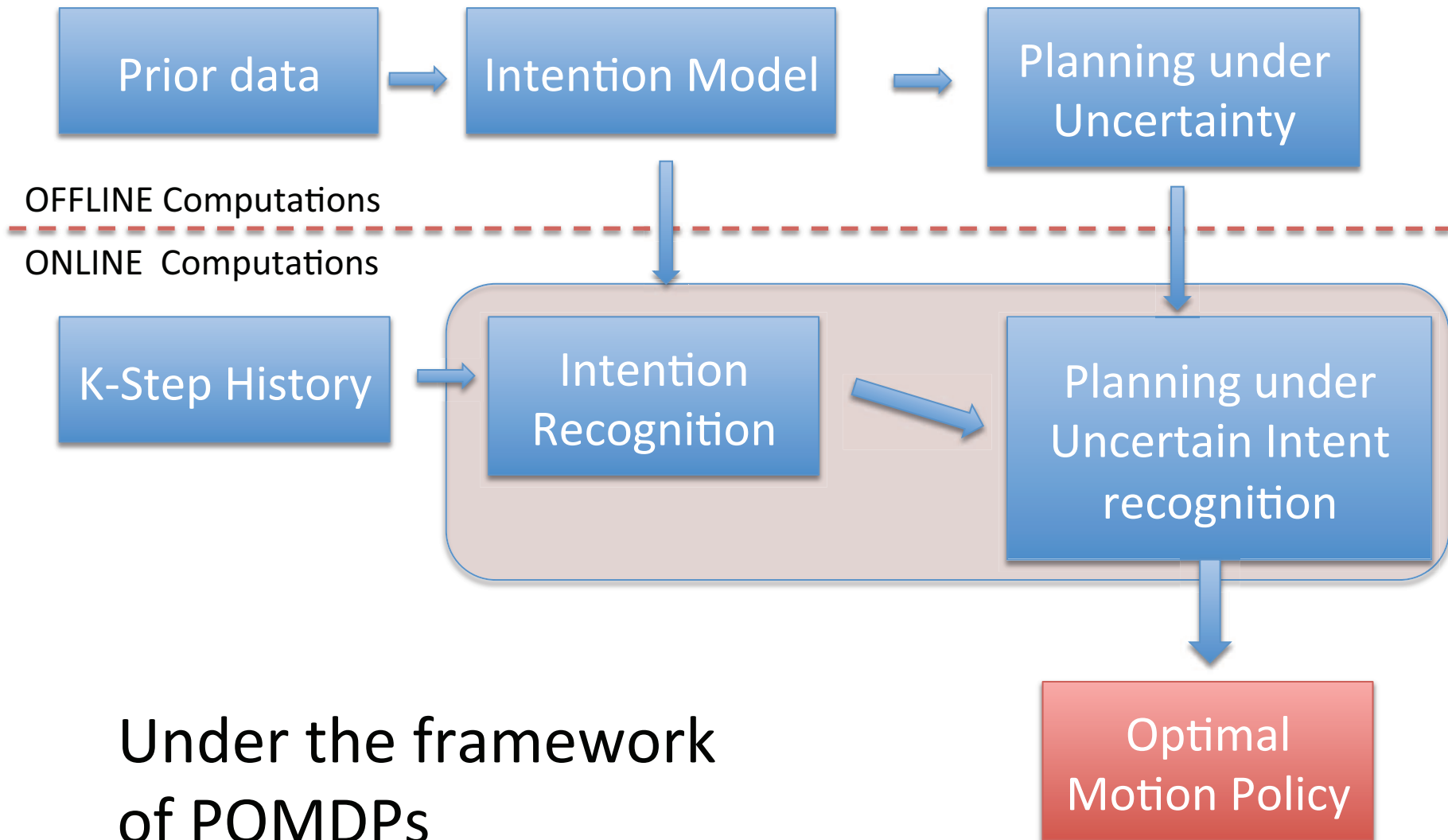


# Prior knowledge should be used when possible

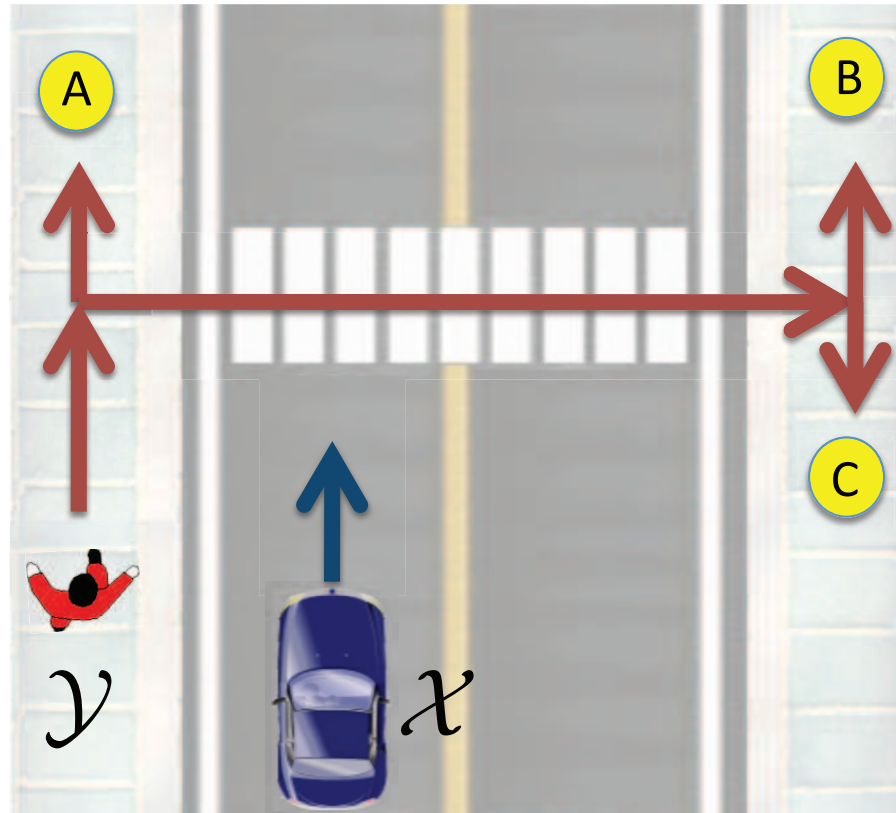
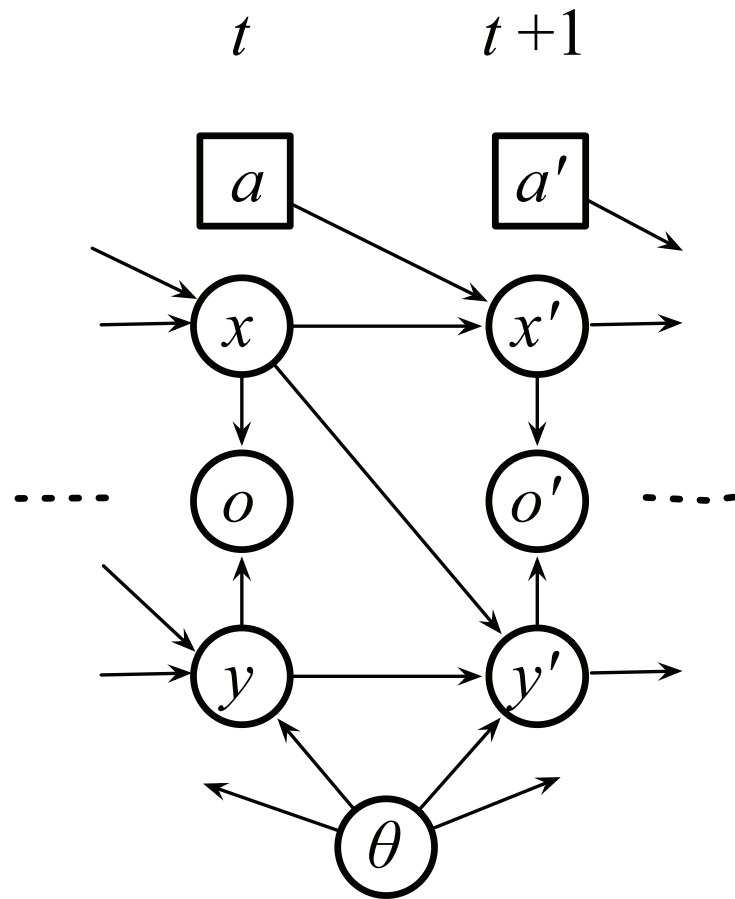
- Model based approaches
  - Learn the goals
  - Learn the conditional system transitions
- Predictors give a possible goal
  - Use any kind of predictor (HMMs, GPs, GP-RRT)
- Common approach
  - Identify most likely goal
  - Then choose suitable action
- Many cases uncertainty measure provided by predictors are ignored



# Our approach



# Pedestrian Avoidance as POMDP





# Pedestrian Avoidance as Factored POMDP (MOMDP)

$$\mathcal{M}_\Theta: (\mathcal{X}, \mathcal{Y}, \Theta, \mathcal{A}, \mathcal{O}, T_x, T_y, Z, R, \gamma)$$

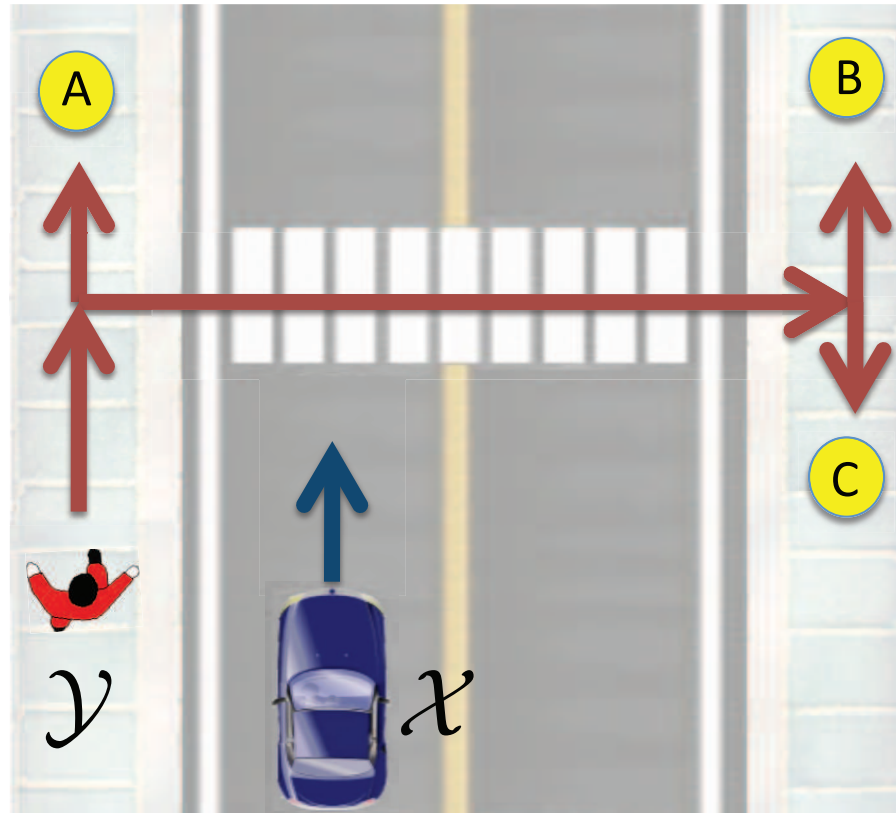
$\mathcal{X}$  : Robot {pos, vel}

$\mathcal{Y}$  : Pedestrian {pos}

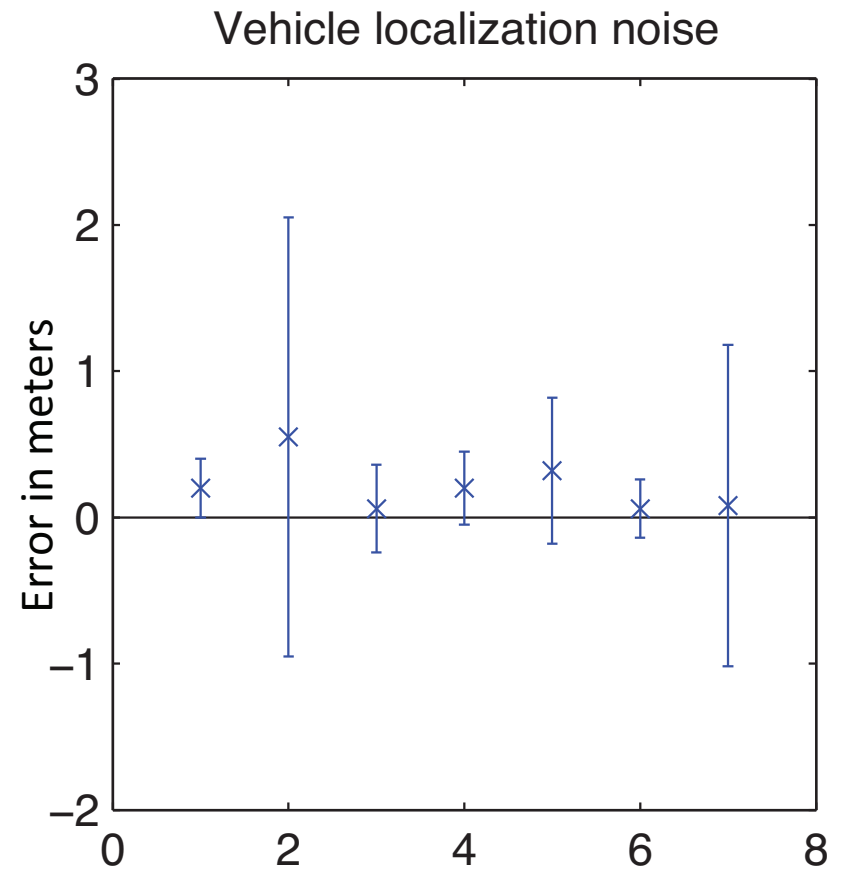
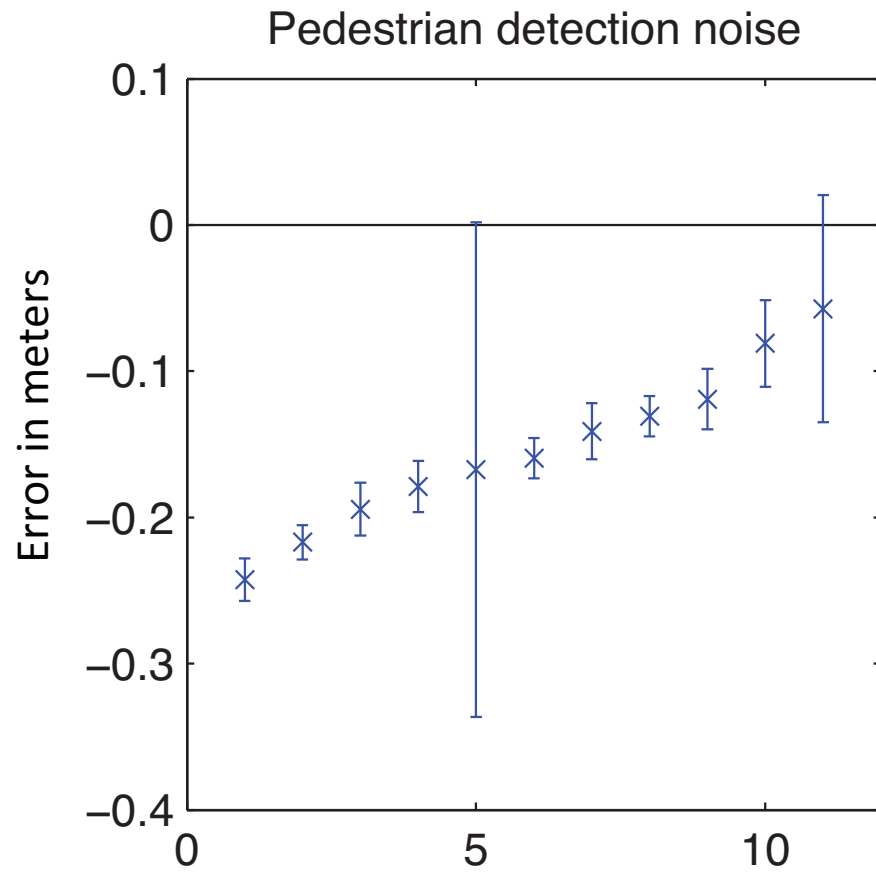
$\Theta$  : Intention {A, B, C}

$\mathcal{A}$  : Action {Acc., Decc.,  
Cruise}

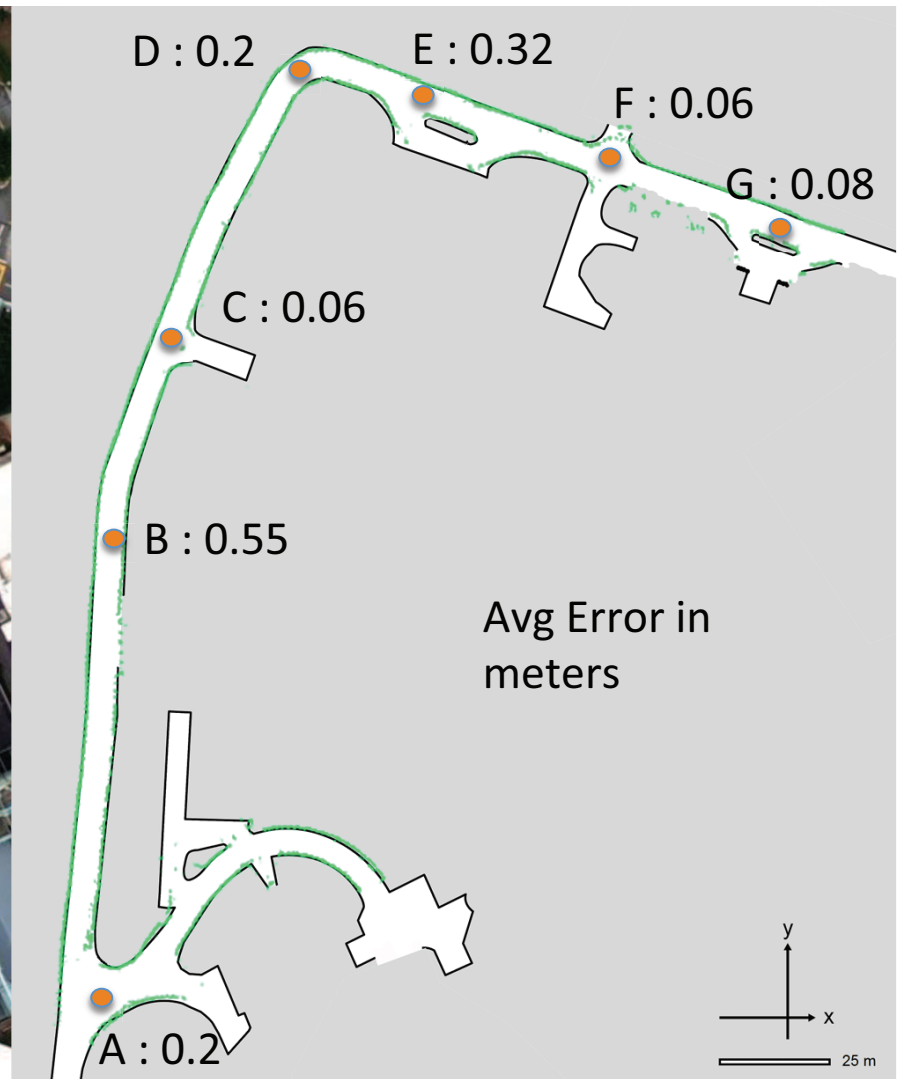
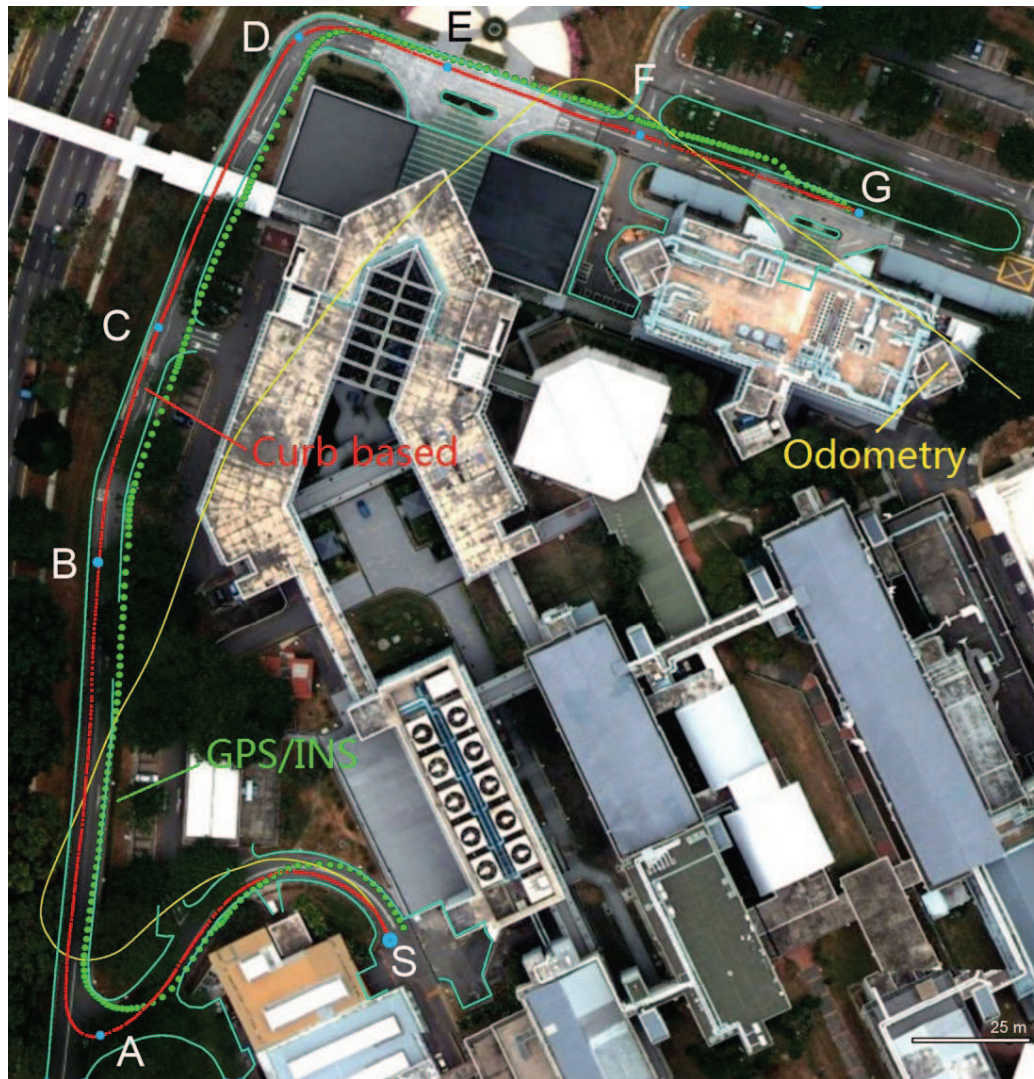
$R$  : Reward {Collision,  
Goal}



# Validation of Modeling



# Results



# Pedestrian Avoidance as MOMDP

$$\mathcal{M}_\theta: (\mathcal{X}, \mathcal{Y}, \Theta, \mathcal{A}, \mathcal{O}, T_x, T_y, Z, R, \gamma)$$

$T_X$  : Robot Motion Model

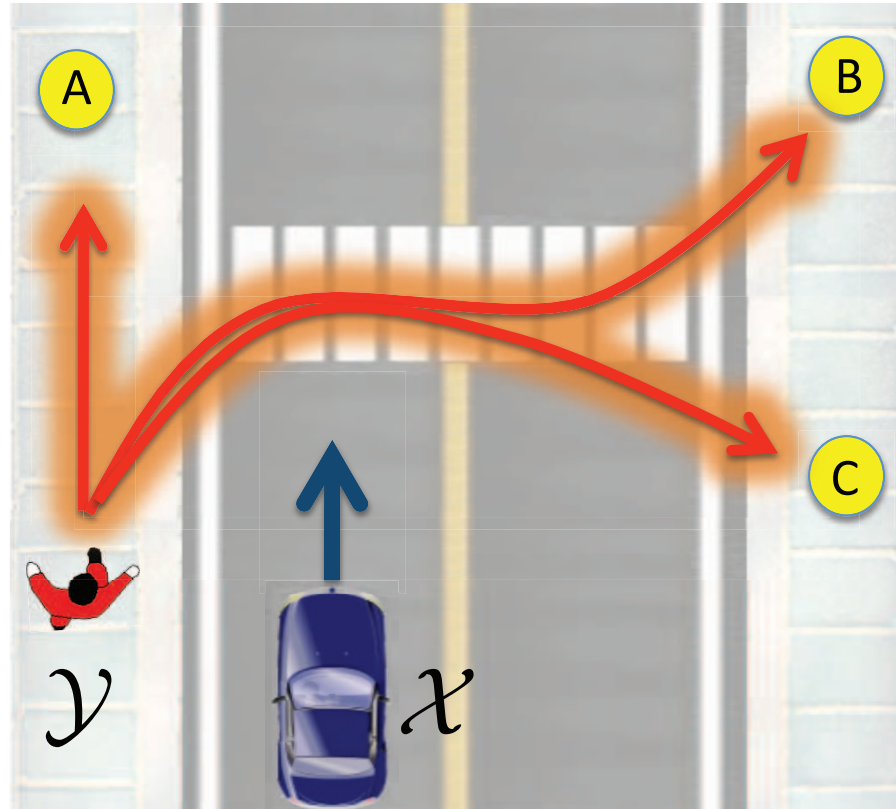
$$x'_r = x_v \Delta T + x_r^e$$

$$x_v = a \Delta T + x_v^e$$

$T_Y$  : Ped. Motion Model

$$y'(\theta) = y_v(\theta) \Delta T + y^e$$

$$y_v = \alpha e^{-d(\theta)/\beta} \mathbf{n}_\theta$$



# Overview of the solver

## Belief Structure

$$b = (x, y, \theta)$$

$$\mathcal{B} = \bigcup_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathcal{B}_\theta(x, y)$$

$$|\mathcal{B}| = |\mathcal{X}| |\mathcal{Y}| |\Theta| - 1$$

$$|\mathcal{B}_\theta(x, y)| = |\Theta| - 1$$

Huge computational  
reduction

## Offline Policy learning

$$V(b) = V(x, y, \theta)$$

$$V(x, y, \theta) : \{\Gamma_\theta(x, y)\}$$

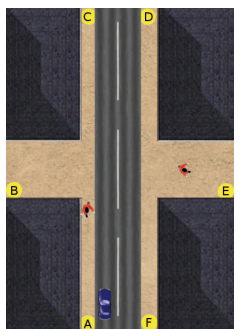
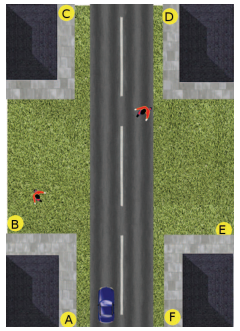
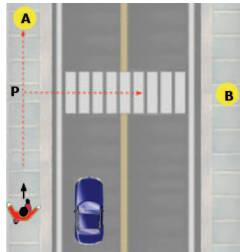
## Online Policy execution

$$a : \arg \max_{\alpha \in \Gamma_\Theta(x, y)} \{\alpha \cdot b_\theta\}$$

$$b_\theta^{new} = \eta T_\gamma(x, y, \theta, y^{new}) b_\theta$$



# Simulation Results



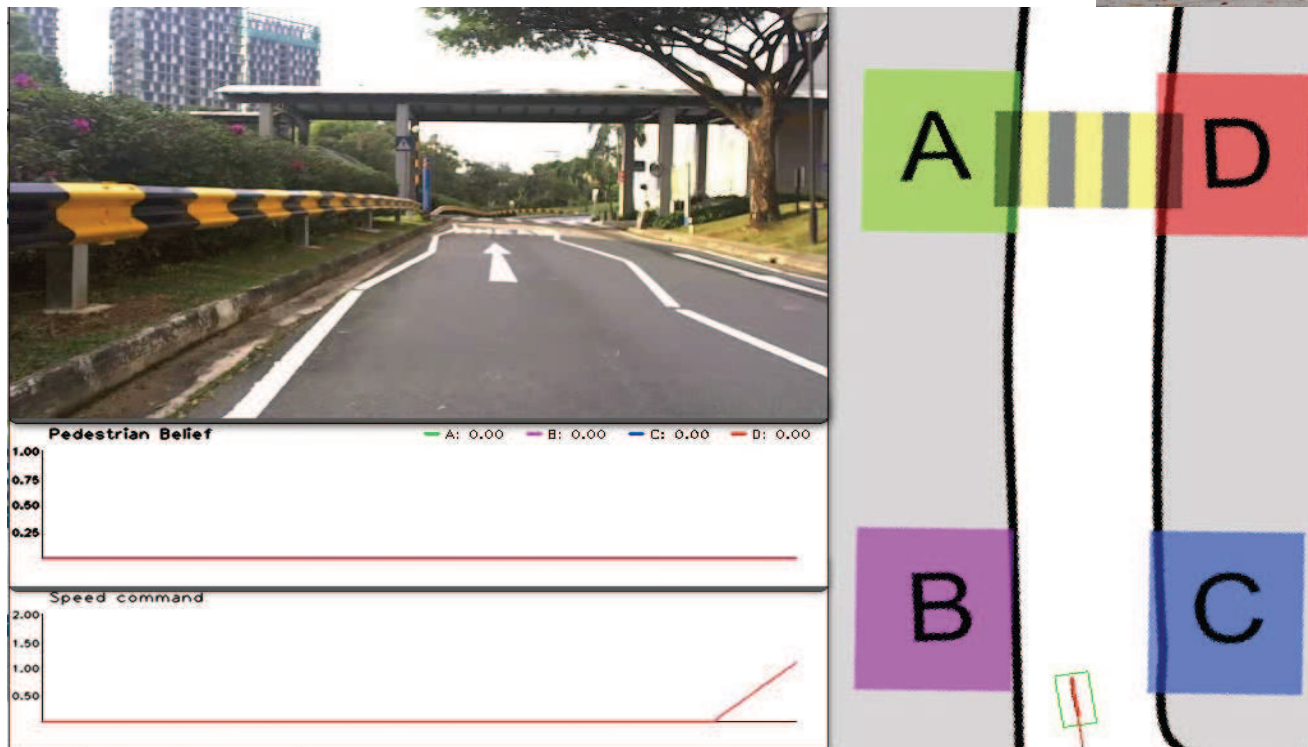
Environment	Noise	Bayes-ML		MOMDP	
		Time	Accident	Time	Accident
zebra crossing	zero	14.8 (0.4)	4.8%	14.7 (0.5)	0%
	high	17.0 (3.6)	2.8%	16.8 (3.5)	2.1%
lane	zero	7.6 (1.0)	2.5%	7.7 (1.2)	0.3%
	low	9.1 (1.8)	3.1%	9.1 (2.2)	1.9%
	med	8.6 (1.8)	6.1%	8.6 (1.8)	5.9%
	high	10.1 (4.0)	5.8%	10.7 (4.4)	5.0%
	open	zero	11.3 (0.9)	0.8%	11.4 (1.1)
open	low	13.5 (2.6)	1.7%	13.6 (2.6)	1.3%
	med	14.3 (3.8)	2.5%	14.5 (4.0)	2.2%
	high	14.6 (4.5)	3.0%	14.5 (4.4)	2.7%
	constrained	zero	11.3 (0.8)	1.6%	11.5 (1.2)
constrained	low	13.5 (2.5)	1.7%	13.7 (2.5)	0.9%
	med	14.6 (4.0)	3.6%	15.1 (4.3)	3.3%
	high	18.4 (10.4)	4.0%	21.9 (13.5)	3.2%



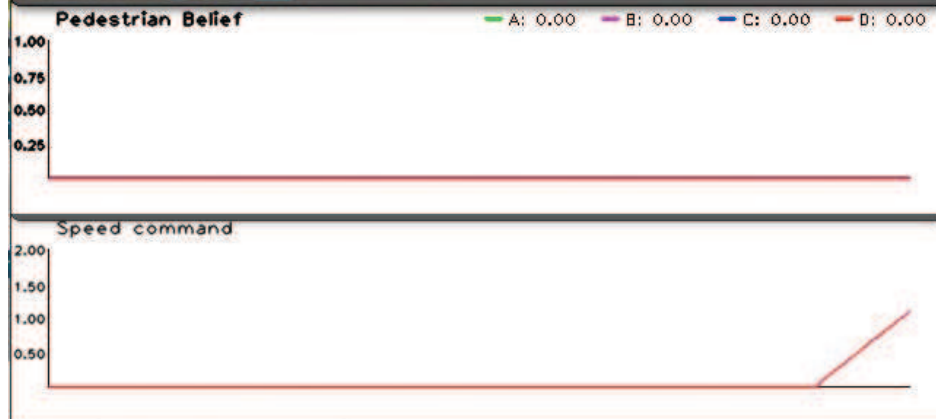
For the same time of navigation, IA-momdp gives safer statistics.

# Sample example

- Play video from playlist



# Stationary Pedestrian

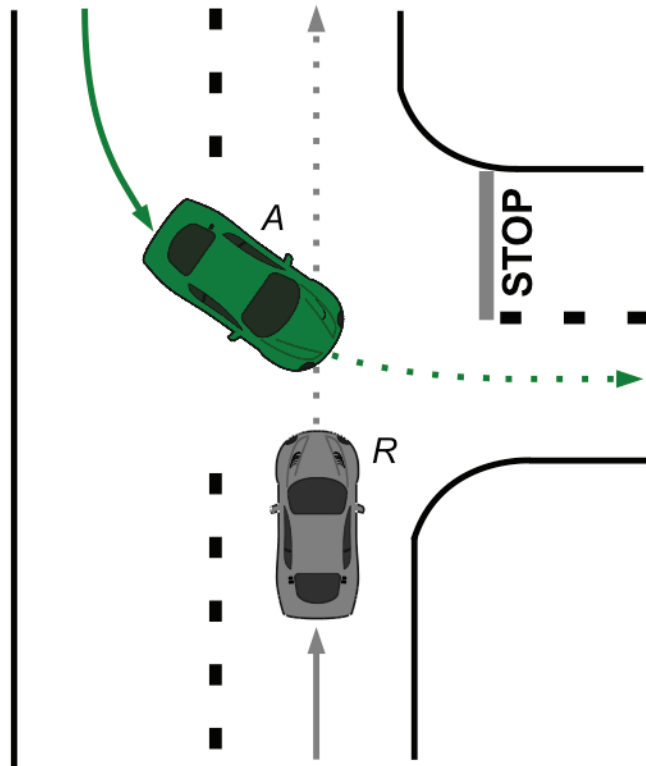




# Multiple Ped.



# Extension to vehicle interactions



Problem formulation,

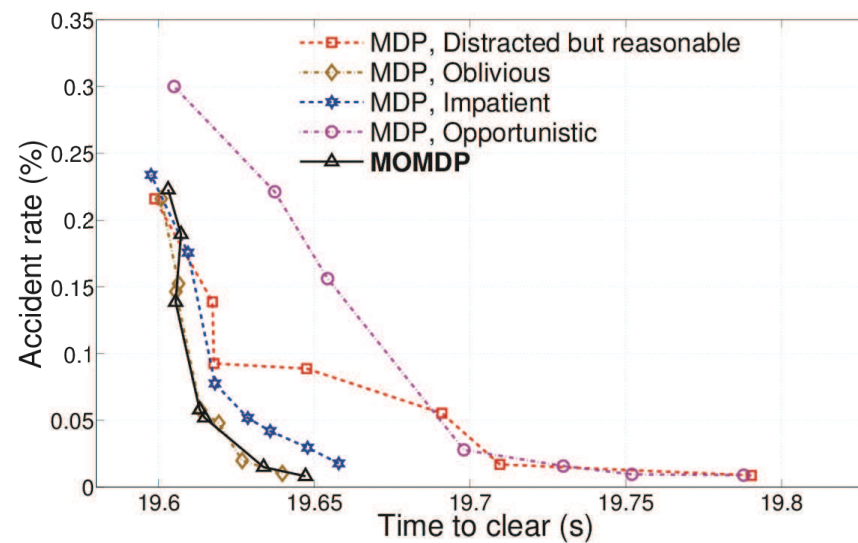
$\mathcal{X}$ : Robot position, robot velocity

$\mathcal{Y}$ : Vehicle position, vehicle velocity

$\Theta$ : Driver models

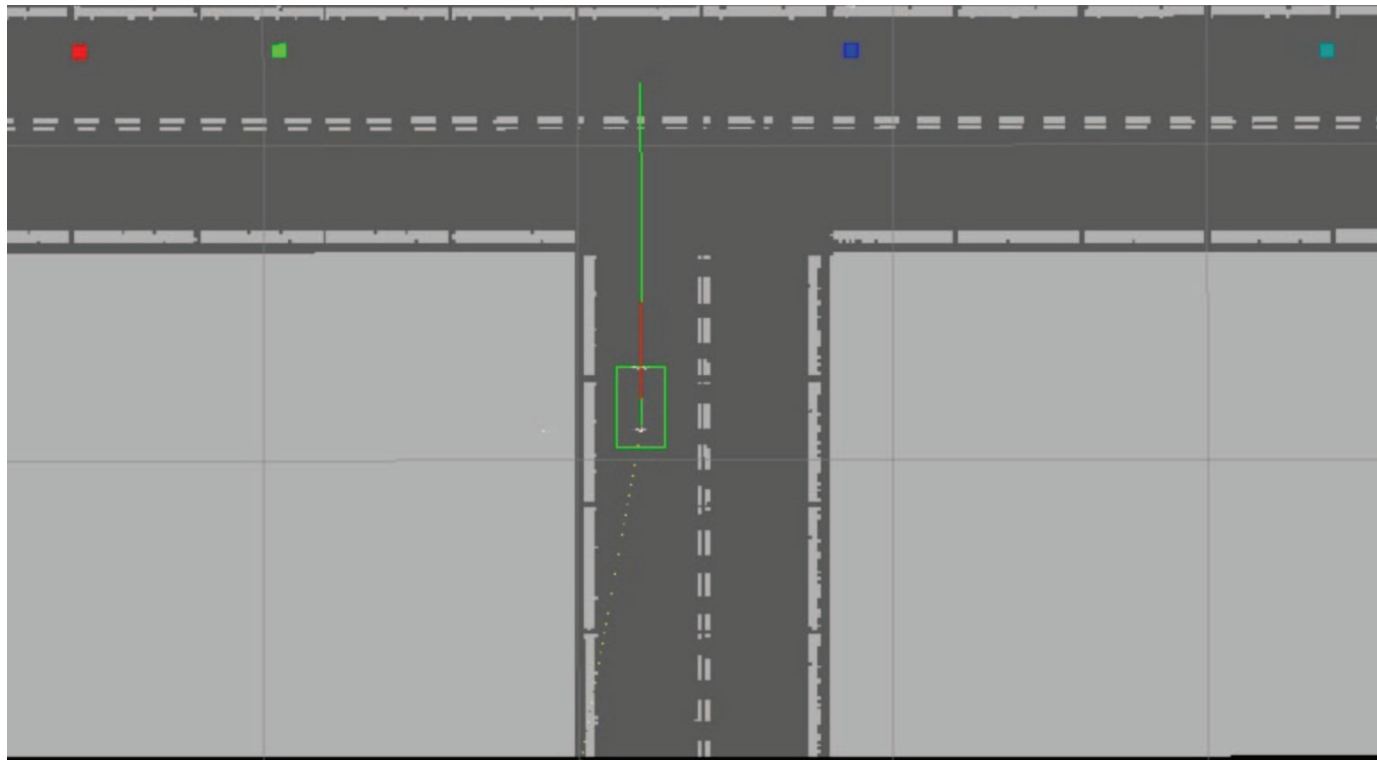
- ▶ Reasonable-distracted
- ▶ oblivious
- ▶ impatient
- ▶ opportunistic

$\mathcal{A}$ : Robot acceleration commands  
(accelerate, decelerate, cruise, emergency stop)

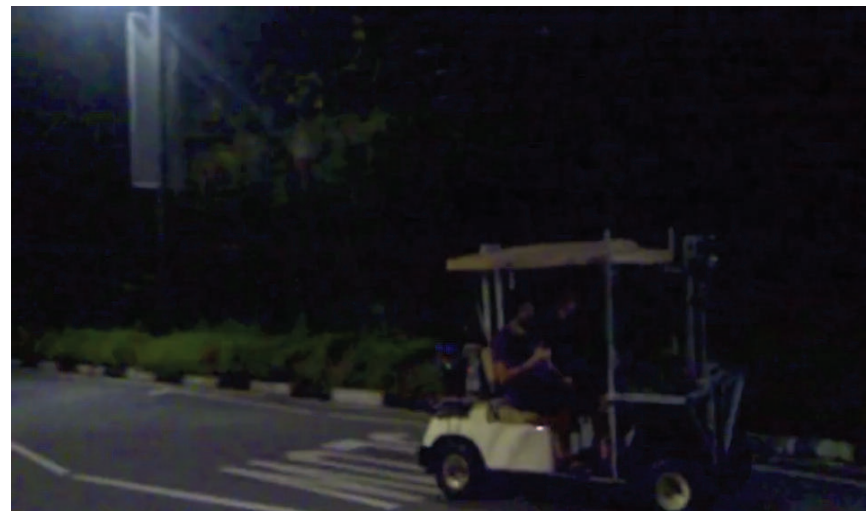
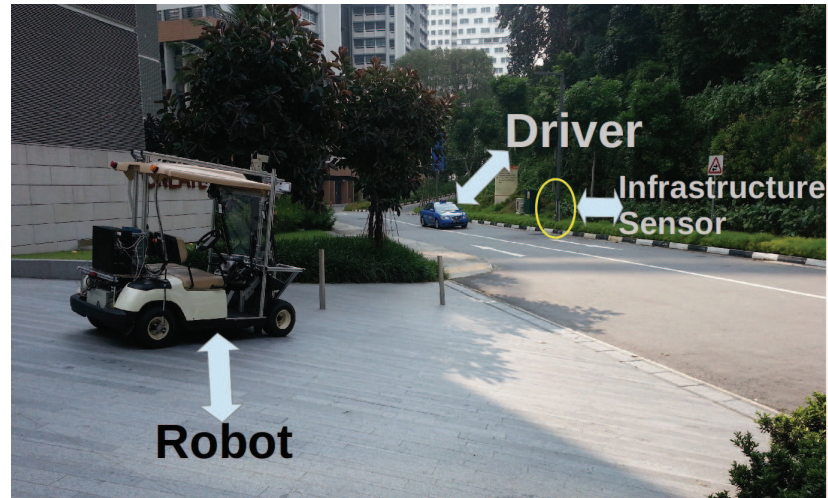
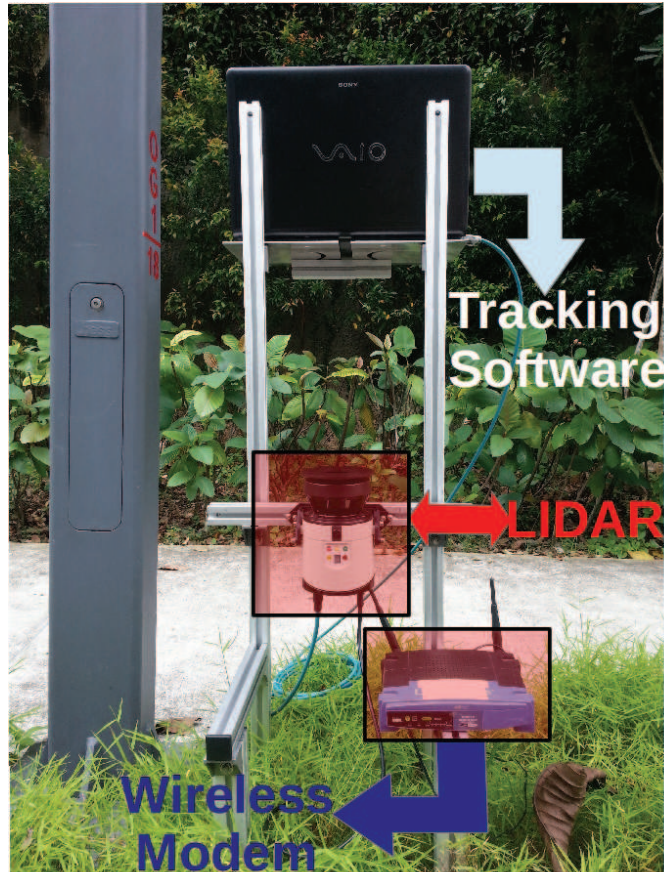




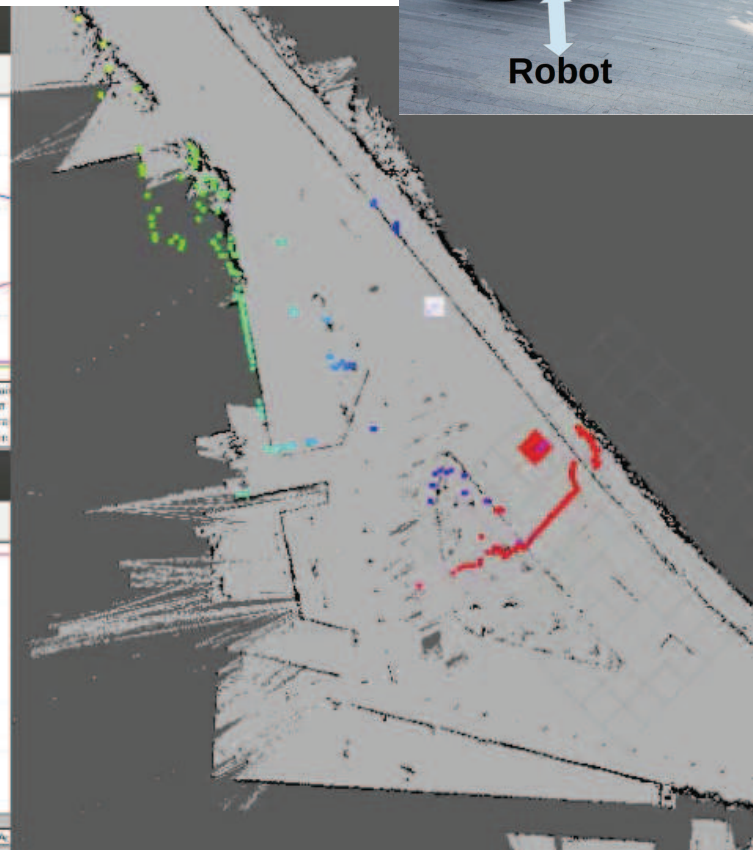
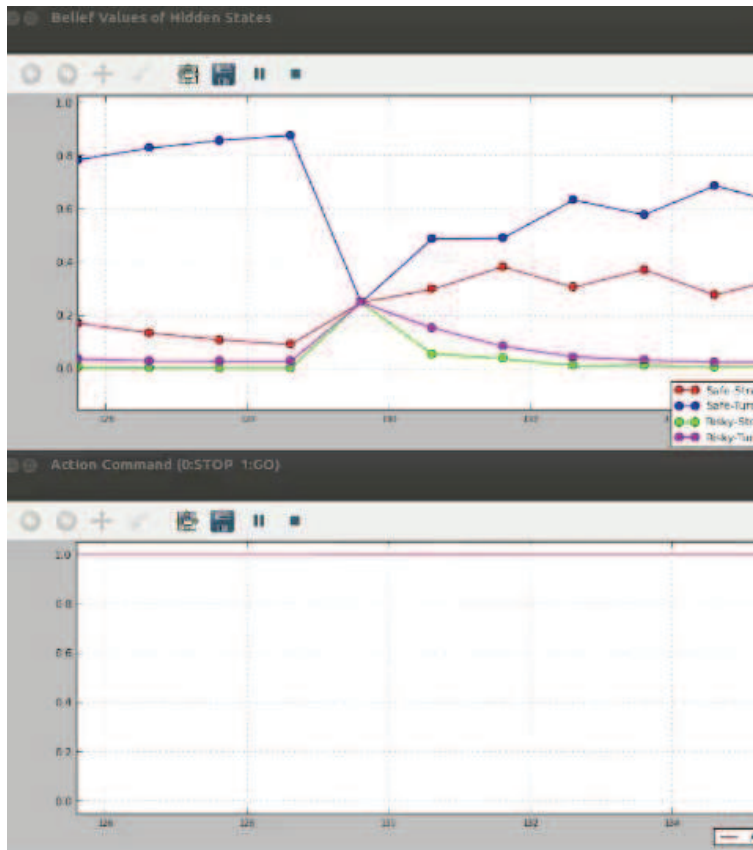
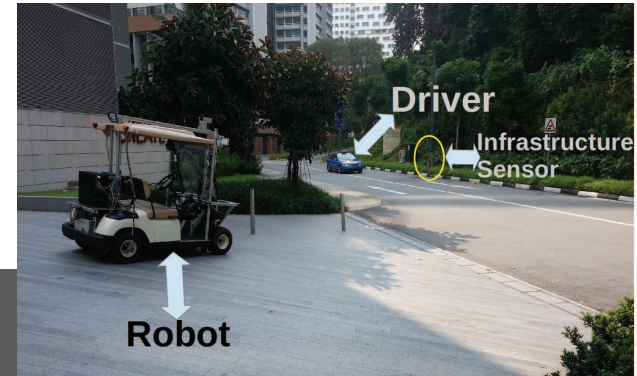
# Tjunction Merging



# Real system



# Intersection Navigation



# Conclusion

- Modeling Intentions improves performance
- We presented an Integrated Perception and Planning approach
  - Need based prediction is necessary
- Proper choice of variables makes POMDP tractable





## **Session IV**

### **Navigation, Control, Planning**

- **Title: Safe highways platooning with minimized inter-vehicle distances of the time headway policy**  
**Authors:** Alan Ali, Gaetan Garcia, Philippe Martinet
- **Title: Optical Flow Templates for Superpixel Labeling in Autonomous Robot Navigation**  
**Authors:** Richard Roberts, Frank Dellaert



**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

# Safe highways platooning with minimized inter-vehicle distances of the time headway policy

Alan ALI<sup>1</sup>, Gatan GARCIA<sup>2</sup> and Philippe MARTINET<sup>3</sup>

**Abstract**—Optimizing the inter-distances between vehicles is very important to reduce traffic congestion on highways.

Variable spacing and constant spacing are the two policies for the longitudinal control of platoon. Variable spacing doesn't require a lot of data (position, speed...) from other vehicles, and string stability using only on-board information is obtained. However, inter-vehicle distances are very large, and hence traffic density is low. Constant spacing can offer string stability with high traffic density, but it requires at least data from the leader.

In [1], we have proposed a modification of the constant time headway control law. This modification leads to inter-vehicle distances that are close to those obtained with constant spacing policies, while requiring only low rate information from the leader.

In this paper, the work done in [1] is extended by taking into account the model of the motor. This enables to reduce the distance between the vehicles to 1 meter, and it has been proved that the platoon is stable and safe in normal working mode. Simulation results are done using TORCS simulator environment.

## I. INTRODUCTION

The problems of traffic congestion, pollution, and people safety are becoming more and more important due to the increase in the number of cars.

Proposed solutions to these problems on highways differ from those in urban areas. On highways, road curvature is smaller and there are less obstacles. Under normal conditions, cars move faster than in urban areas.

Some proposed ideas require changes to the infrastructure (automatic speed limits, roads monitoring, reversible lanes...) Other ideas rely on automated vehicles to increase traffic density and to avoid the oscillation of the platoon. Driving in platoon has many advantages: it increases traffic density and safety, while simultaneously decreasing fuel consumption and driver tiredness [19].

From the modeling and control point of view, it is possible to decouple the longitudinal and lateral behaviors, when road curvature is assumed to be low, or by using techniques like chained systems theory [15]. Lateral control can be performed using different modalities like 3D laser (as used by the famous Google car), magnetic markers (PATH project), vision sensors [9]... So in a highway environment, it is

common to concentrate on longitudinal behavior, including modelling and control.

Platoon models can be found in [13], ranging from systems which do not include communication between the vehicles to systems which use full communications between them. Other authors have build physics-inspired models of the platoon: [2] considers the platoon as a multi agent system, in which the agents (vehicles) interact according to physical phenomena or mimick animal interaction behaviors, [17] represents the interactions as virtual spring-dumper systems, while [5] uses Newton forces.

In platooning applications, the desired behavior of a vehicle is generally defined by a desired distance to the previous vehicle in the platoon. Stability of the platoon control is very important. It uses the concept of String Stability, which requires that distance errors do not amplify as they propagate along the platoon, and have the same sign to avoid collisions. The definition is given in the time domain in [13] and in the frequency domain in [8].

Local control uses data from adjacent vehicles only, while global control depends on data from at least the leader. In local control, the car is totally autonomous: it does not require sophisticated sensors, and can be used in all environments, but trajectory tracking and inter-vehicle distances keeping are not very accurate. On the other hand, global control is more accurate, but it requires more sophisticated sensors, sometimes adaptation of the environment where it is used, and finally it requires very reliable communication systems.

Two policies are used to control the spacing between vehicles: constant spacing and variable spacing. Variable spacing usually doesn't require a lot of data from other vehicles. In addition, it can ensure string stability using on-board information only [4], but inter-vehicle distances vary with velocity and can be very large, hence traffic density is low. Constant spacing can achieve both string stability and high traffic density, at the cost of inter-vehicle communications.

Constant Time Headway (CTH) is the simplest and most common variable spacing policy [14], [17]. Variable time headway can vary linearly with the velocity, with relative velocity [18], or even with vehicle dynamics and road conditions [3].

In this paper, we will concentrate on the longitudinal control of platoons on highways. We will propose a modification to the time headway policy, develop the corresponding dynamic control law, study the stability of the platoon and demonstrate the effectiveness and safety of the novel approach for small inter-vehicle distances. The new

<sup>1</sup> A. ALI is with Institut de Recherche en Communications et Cyberntique de Nantes (IRCCYN), Ecole centrale de Nantes (ECN), 44300 Nantes, France

<sup>2</sup> G. GARCIA is with Ecole centrale de Nantes (ECN), 44300 Nantes, France

<sup>3</sup> P. MARTINET is with Institut de Recherche en Communications et Cyberntique de Nantes (IRCCYN), Ecole centrale de Nantes (ECN), 44300 Nantes, France <http://www.irccyn.ec-nantes.fr/martinet/>

control law is a mixture of local and global decentralized control. This work is a preliminary work to be generalized for platoons working in urban areas. Safety issues due to abnormal working conditions will not be discussed in this paper.

The paper is organized as follows. Section 2 describes the vehicle and platoon models. The control and string stability are presented in section 3. Section 4 explains the simulation results. Finally, section 5 discusses the most important advantages of the proposed approach, and compares it with other existing approaches.

## II. MODELING AND CONTROL

In the case of platooning on highways, where the road curvature is small, it is known that longitudinal and lateral controls can be considered as decoupled. In this paper, we also make this safe assumption, which allows us to consider only longitudinal control.

### A. Longitudinal Dynamic Model of the Vehicle

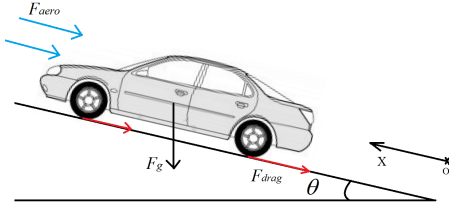


Fig. 1. The applied forces

According to Newton's law, we can write the dynamic equation [12] of the vehicle in the platoon shown in figure (1) as:

$$m \ddot{x} = F + F_g + F_{aero} + F_{drag}$$

$$m \ddot{x} = F - m g \sin(\theta) - \frac{\rho A C_d}{2} \dot{x}^2 \operatorname{sgn}(\dot{x}) - d_m \quad (1)$$

Since the vehicles are assumed to travel in the same direction at all times then we have  $\operatorname{sgn}(\dot{x}) = 1$

The engine of the vehicle is modeled as a first degree system, and is given by the following equation

$$\dot{F} = -\tau F + u \quad (2)$$

So the model of the vehicle can be represented in figure (2):

where:

- $x$ : Position of the vehicle along X axis.
- $F$ : Force produced by the vehicle engine.
- $\tau$ : The vehicle engine time constant.
- $u$ : The control input to the vehicle engine.
- $F_g, F_{aero}, F_{drag}$ : Gravitational, aero-dynamical and mechanical drag force respectively.
- $g$ : Acceleration of gravity.

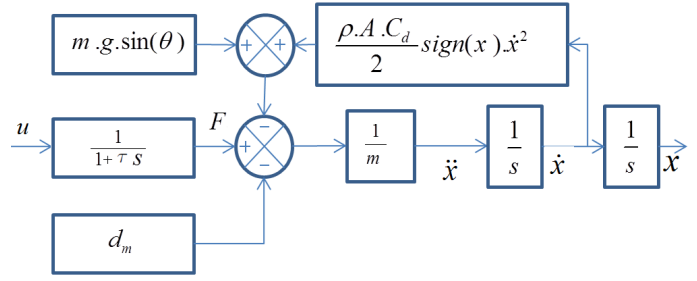


Fig. 2. Dynamic model of the car

- $\theta$ : Angle between the road surface and the horizontal plane.
- $\rho$ : Specific mass of air,
- $A, C_d$ : Cross-sectional area and drag coefficient of the vehicle.
- $d_m$ : The amplitude of the mechanical drag force.

By taking the derivative of (1) and substituting (2) in the resulting expression, we get the following:

$$m x^{(3)} = -\tau F - m g \cos(\theta) \dot{\theta} - \rho A C_d \dot{x} \ddot{x} + u \quad (3)$$

We can use exact linearization to linearize the previous system. We obtain a linear model of the longitudinal dynamics of the car by taking:

$$u = m w + \tau F + m g \cos(\theta) \dot{\theta} + \rho A C_d \dot{x} \ddot{x} \quad (4)$$

$F$  can be computed from (1).

Then, we get:

$$x^{(3)} = w \quad (5)$$

where  $w$  is the new control input for the linearized system shown in figure (3).

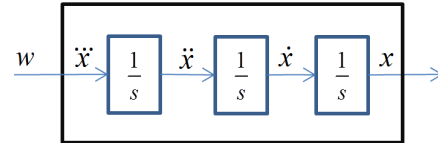


Fig. 3. Linearized car model

### B. Platoon definitions

Figure (4) shows a platoon which consists of  $N$  vehicles required to move at the same speed  $v_d$  with a desired inter distance  $L$  between two successive vehicles. The leader of the platoon can be driven by a human or autonomously. The followers are controlled to maintain a desired inter-distance.

We define the spacing error of the  $i$ -th vehicle assuming a point mass model for all vehicles :

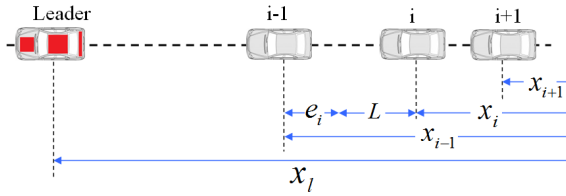


Fig. 4. A platoon

$$e_i = \Delta X_i - L \quad (6)$$

where :

- $\Delta X_i = x_{i-1} - x_i$ : real spacing between car number  $i$  and its predecessor, car number  $i - 1$ .
- $x_i$ : position of  $i$ -th vehicle.
- $L$ : desired inter-vehicle distance

The kinematic evolution of the spacing error is given by:

$$\dot{e}_i = \dot{x}_{i-1} - \dot{x}_i = v_{i-1} - v_i$$

where  $v_i$  represents the velocity of the  $i$ -th vehicle.

### III. PLATOON CONTROL AND STABILITY

#### A. Control Objectives

The main objectives of the control law are to:

- 1) Keep the inter-vehicle distance equal to  $L$ , and make all vehicles move at the same speed so  $\dot{e}_i = 0$ .
- 2) Assure the string stability of the platoon (the spacing error does not increase as it propagates through the platoon).
- 3) Increase the traffic density.
- 4) Keep the system stable in case of total loss of communication.

#### B. Control Law

In constant spacing control, the control law will make  $e_i \rightarrow 0$  so the inter-vehicle distance will become equal to  $L$ . But this requires, at least, information from the leader to assure the string stability of the platoon and robustness.

In time headway policy, a new term is added to the previous error, which will eliminate the need for communication with the leader and increase the string stability. A new spacing error is defined as:

$$\delta_i = e_i - h v_i = \Delta X_i - L - h v_i$$

In this case, the control law makes  $\delta_i \rightarrow 0$ , so the steady state of the inter-vehicle distance will be equal to  $\Delta X_i = L + h v_i$ , which is proportional to vehicle speed and can become very large when the vehicle travels at high speed.

Adding the time headway term ( $h v_i$ ) improves stability. This improvement is not due to enlarging inter-vehicle distance, but to the fact that it is a function of the velocity. So, the main idea of this paper is to propose a novel spacing

error, defining the time headway term as proportional to the difference between the velocity of the vehicle and some value  $V$  shared between all other vehicles in the platoon. We will discuss later how to set the parameter  $V$ . In this case, we define the novel error as:

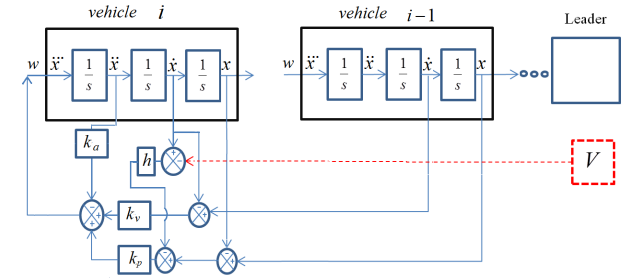
$$\delta_i = e_i - h (v_i - V) = \Delta X_i - L - h (v_i - V)$$

where  $V$  is a velocity value shared between all the vehicles at the same sampling time.

The new control law is defined by:

$$u_i = -k_a \ddot{x}_i + k_v \dot{e}_i + k_p \delta_i$$

which is represented in figure (5) for the  $i$ -th vehicle.


 Fig. 5. Control scheme of the  $i$ -th vehicle

To verify the effectiveness of the new law, the string stability of the platoon under this control law must be analyzed.

#### C. String Stability Analysis

The general string stability definition is given in [13]. In essence, it means that all the states are bounded if the initial states (position and velocity errors) are bounded and summable.

A sufficient condition for string stability is given in [8]:  $\|e_i\|_\infty \leq \|e_{i-1}\|_\infty$  which means that the spacing error must not increase as it propagates through the platoon. To verify this condition, the spacing error propagation transfer function is defined by:

$$G_i(s) = \frac{e_i(s)}{e_{i-1}(s)}$$

A sufficient condition for string stability is given by:

$$\|G_i(s)\|_\infty \leq 1 \quad \text{and} \quad g_i(t) > 0 \quad i = 1, 2, \dots, N \quad (7)$$

where  $g_i(t)$  is the error propagation impulse response of the  $i$ -th vehicle.

So, to verify the string stability of a platoon using the novel spacing error, the spacing error propagation transfer function  $G(s)$  must be calculated:

$$G_i(s) = \frac{k_v s + k_p}{s^3 + k_a s^2 + (k_v + h k_p) s + k_p} \quad (8)$$

So

$$\|G_i(\omega)\| = \sqrt{\frac{k_p^2 + k_v^2 \omega^2}{(k_p - k_a \omega^2)^2 + ((k_v + k_p h) \omega - \omega^3)^2}} \quad (9)$$

To ensure the stability we must verify the condition 7 so we get :

$$\omega^6 + (k_a^2 - 2(k_v + k_p h)) \omega^4 + (k_p^2 h^2 + 2 k_p (k_v h - k_a)) \omega^2 \geq 0 \quad (10)$$

To simplify we choose :  $k_v = k_a/h$ , which makes the coefficient of  $\omega^2$  always positive, then the stability conditions become:

$$\left\{ \begin{array}{l} h k_a \geq 2 \\ h k_a^2 - 2 k_a - 4 k_p h^2 \leq 0 \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{l} h k_a \leq 2 \\ h k_a^2 - 2 k_a - 4 k_p h^2 \geq 0 \end{array} \right\} \quad (11)$$

$$\text{or} \left\{ h k_a^2 - 2 k_a - 2 k_p h^2 \geq 0 \right\}$$

#### D. Maximum error amplitudes

In a stable platoon, the maximum error between vehicles is the error between the leader and the first vehicle. If we choose  $V_s = v_{leader}$  then the transfer function of the first error in the platoon is given by:

$$G_1(s) = \frac{e_1(s)}{w_{leader}(s)} = \frac{1}{s^3 + k_a s^2 + (k_v + h k_p) s + k_p} \quad (12)$$

The magnitude of this function is given by:

$$\|G_1(\omega)\| = \frac{1}{\sqrt{(k_p - k_a \omega^2)^2 + ((k_v + k_p h) \omega + \omega^3)^2}} \quad (13)$$

If the platoon is stable and by choosing  $k_p > 1$ , we get:

$$\|G_1\| < \|G_i\| \leq 1 \quad (14)$$

then

$$\|e_1\| < \|w_{leader}\| \quad (15)$$

So the maximum error in the platoon is bounded by the maximum control value of the leader (the jerk of the leader).

For passenger comfort [16] the maximum value of the jerk should not be bigger than  $0.5 - 0.6 \text{ m/s}^3$ , so it is clear that we can get a maximum error between vehicle much smaller than 1. So we have proved that we can get a stable platoon system with inter-vehicle distance equal to 1 without collision between vehicles.

#### IV. SIMULATIONS

The control law has been checked using TORCS, The Open Racing Car Simulator, a software which give us realistic results (as it takes many phenomena into account) and allows visual output when applying the novel spacing error.

TORCS is one of the most popular car racing simulators [7]. It is written in C++ and is available under GPL license from its web page. TORCS presents several advantages for academic purposes, namely:

- 1) It lies between advanced simulators, like recent commercial car racing games, and a fully customizable environment, like the ones typically used by computational intelligence researchers for benchmark purposes.
- 2) It features a sophisticated physics engine (aerodynamics, fuel consumption, traction...) as well as a 3D graphics engine for the visualization of the races.
- 3) It was not conceived as a free alternative to commercial racing games, but it was specifically designed to make it as easy as possible to develop your own controller.

All the simulations were done on nearly straight roads (small curvatures). The desired speed of the leader of the platoon is changed three times (see figure 6), to check the transit response and the stability of the platoon.

At the same time, a comparison between our control law and the classical CTH control law will be performed using the same parameters.

We consider 10 identical vehicles and we choose the following parameters values:  $h = 3, k_v = 1/3, k_p = 5, K_a = 1$ . The desired inter-vehicle distance (bumper-to bumper distance, so we omit all the cars lengths from all following figures) is fixed to  $L = 1 \text{ m}$ .

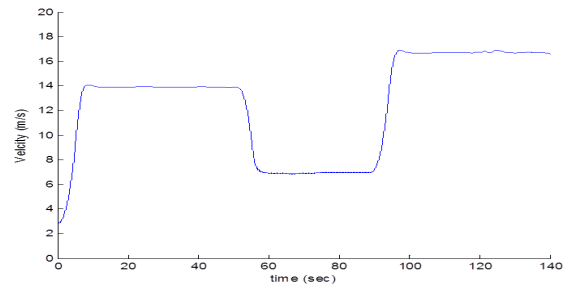


Fig. 6. Leader's velocity profile

We can see in figure (8) that the system is stable, as the errors are decreasing through the platoon. We can see also that the maximum error is smaller than  $L$ .

When comparing our control law and the classical CTH law we can see in figure (7) and figure (8) that the distances



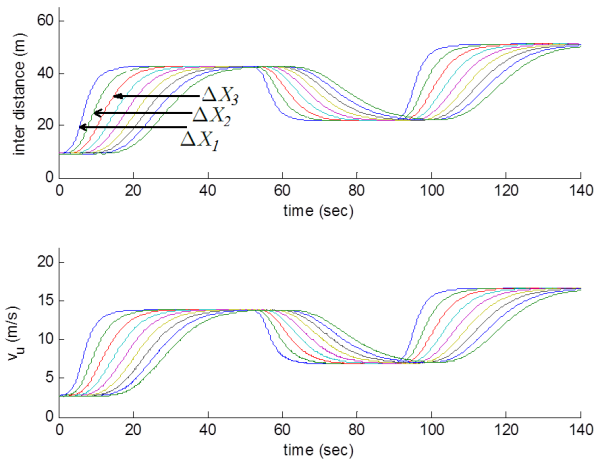


Fig. 7. Inter-vehicle distances and velocities using CTH law. Each curve represents a different vehicle of the platoon.

between vehicles have been reduced from the range [5-40] meters for CTH to the range [0.5-1.5] meters using our control law. In addition, we can see that the system becomes faster.

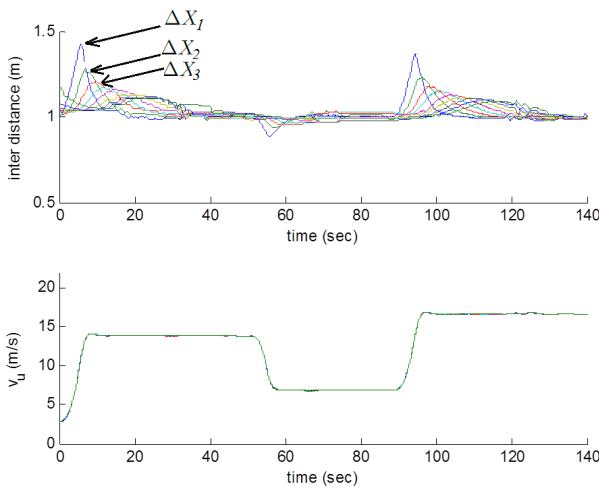


Fig. 8. Inter-vehicle distances and velocities using our control law

## V. DISCUSSION

The proposed approach greatly reduces inter-vehicle distances required, while assuring stability. This is obtained by making the distance proportional, not to velocity, but to the difference between the vehicle velocity and a common velocity value shared by all vehicles of the platoon.

### A. Advantages and comparison

Using the new spacing policy and the corresponding new control law, the advantages are the following:

**String stability:** The propagation function  $G(s)$ , corresponding to the new control law, is not related to  $V$ , so the value

of  $V$  will not affect the platoon stability. It can be noticed that it is exactly the same propagation transfer function as the classical time headway spacing policy, so with this modification the system remains string stable.

**Inter-vehicle distances:** The most important effect of the proposed modification is on the inter-vehicle distances. At equilibrium, if all velocities become equal to leader velocity  $v_L$  then  $\Delta X_i = L + h (v_L - V)$ . By choosing  $V = v_L$  the inter-vehicle distance becomes  $\Delta X_i = L$ , and during dynamic changes the inter-vehicle distance becomes  $\Delta X_i = L + h (v_i - V)$ .

The inter-vehicle distance has been decreased from  $\Delta X_i = L + h v_i$  (which might be very large at high speeds) in the case of the classical time headway policy [14], [6], to become  $\Delta X_i = L + h (v_i - V)$ , which is equal to  $L$  at equilibrium and slightly larger than  $L$  during transient phases. So during transient phases, the length of the platoon will be slightly different from the length of a platoon using constant spacing policy [17], [13].

Another important point is the effect of increasing parameter  $h$ , which has a positive effect on stability. In CTH it has a large negative effect on the inter-vehicle distance, as this distance increases proportionally to  $h$ , and hence the traffic density decreases. In our case, the inter distance is also proportional to  $h$  but with a smaller coefficient  $(v_i - V)$ , so the inter distance changes will be smaller than the changes in CTH.

**Collisions:** it is clear that the possibility of a collision between the vehicles is increased as the inter distance between them is reduced. The problem of collision can be addressed separately from the problem of stability by adding a new term for the safety, but we have proved in this article that the platoon is string stable and safe in normal working conditions with small inter-vehicle distances.

**Communication:** Adding  $V$  to the control law impose exchanging data between the vehicles. We have seen previously that stability is not related to  $V$ , so the rate of exchanged data between the vehicles can be reduced by updating the value of  $V$  every sample times according to the change rate of  $V$  as it will be discussed later.

**Stability without communication:** The string stability can be preserved even if the communication with the leader is totally lost, by switching to the classical time headway policy, which corresponds to setting  $V = 0$  (fully autonomous mode). In this case, there is no need to communicate with the leader. So this law can keep the platoon stable even if communication is lost. On the contrary, it has been proved that the constant spacing policy can not be string stable, for homogeneous platoon with homogeneous control (all the gains are equals), without using any information from other vehicles [10].

Hand shaking protocol, between the leader and other vehicles, is very important to detect any loss of communication. If any loss is detected, the leader will transmit an order to all vehicles to switch to full autonomous mode  $V = 0$ , while the vehicle which has lost communication, will automatically switch to this mode when it detects the communication loss.

**Simplicity and type of required data:** The new control has the same simplicity as CTH law. It uses the same variables as CTH, plus a low frequency updating of the common speed parameter  $V$  (which may be the leader or platoon speed). This last variable is the only difference with the classical time headway policy, while the constant spacing policy is always more complicated, as it may require the acceleration or other information, at least from the leader.

### B. Supervision of parameter $V$

As seen previously, the only condition to keep the platoon stable with the new control law is to make  $V$  identical for all the vehicles at any sample time. So, any value for  $V$  (e.g. leader's velocity, the medium velocity of the platoon or the minimum velocity in the platoon...) can be chosen.

To increase the safety and to prevent collisions, one can choose  $V = \min(v_{Leader}, v_1, v_2, \dots, v_N)$ . This will always make  $h.(v_i - V) > 0$ . In that case, the inter vehicle distance becomes  $\Delta X_i = L + h(v_i - V) > L$  but of course it will enlarge the inter-vehicle distance during velocity changes.

The rate of updating  $V$  define the rate of exchanged data between vehicles. Reducing this rate will improve our control law by avoiding the need for high rate communication and reducing the effects of transmission delays and data lose. The rate of changes of  $V$  is usually lower than the rate of the changes of  $v_i, a_i, i = 1 \dots N$ , so the update rate of  $V$  can be lower than the sampling rate of control laws of each vehicle. But, lowering the update rate may produce some jumps in  $V$ , which may have negative effects on the control and hence on the performance. So  $V$  must be interpolated to get smoother changes.

## VI. CONCLUSIONS

In this paper, the design of longitudinal control of platoons in highways has been addressed. We have improved the response of our control law proposed in [1] by taking into account the model of the motor of the vehicle. This enabled to reduce the distance between the vehicles down to 1 meter without losing the stability and the safety of the platoon when working in the normal conditions. We have proved also that most of the properties still correct for the model of third degree for the vehicle. All the provided results have been tested under TORCS to check the validity of the proposed approach.

## REFERENCES

- [1] Ali A., Garcia G., Martinet P., Minimizing the inter-vehicle distances of the time headway policy for platoons control in highways, the 10th International Conference on Informatics in Control, Automation and Robotics (ICINCO13), Reykjavik, Iceland, July 29-31, 2013.
- [2] Franck G., Vincent C., Francois C., A reactive multi-agent system for localization and tracking in mobile robotics, In 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004, pages 431-435, 2004
- [3] Huppe X., de Lafontaine J., Beauregard M., Michaud F., Guidance and control of a platoon of vehicles adapted to changing environment conditions. In IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pages 3091-3096, 2003
- [4] Ioannou P., Chien C., Autonomous intelligent cruise control, in IEEE Transactions on Vehicular Technology, 42(4):657 -672, 1993

- [5] Khatir M., Davison E., Decentralized control of a large platoon of vehicles using non-identical controllers, In Proceedings of the 2004 American Control Conference, vol. 3, pages 2769-2776, 2004
- [6] Xiao L., Gao F., Practical string stability of platoon of adaptive cruise control vehicles, in IEEE Transactions on Intelligent Transportation Systems, 12(4):1184 -1194, 2011
- [7] Onieva E., Pelta D., Alonso J., Milanés V., Perez J. A modular parametric architecture for the torcs racing engine. In IEEE Symposium on Computational Intelligence and Games, CIG 2009, pages 256-262, 2009
- [8] Rajamani R., Vehicle dynamics and control. Springer Science, 1 edition, 2006
- [9] Royer E., Bom J., Dhome M., Thuilot B., Lhuillier M., Marmoiton F., Outdoor autonomous navigation using monocular vision, IEEE/RSJ International Conference in Intelligent Robots and Systems (IROS 2005), pages 12531258, 2005
- [10] Seiler P., Pant A., Hedrick K., Disturbance propagation in vehicle strings, IEEE Transactions on Automatic Control, vol.49, no.10, pp.1835-1842, Oct. 2004
- [11] Sheikholeslam Shahab, Desoer C. A., Longitudinal control of a platoon of vehicles. ii, first and second order time derivatives of distance deviations. UC Berkeley: California Partners for Advanced Transit and Highways (PATH), 1989
- [12] Sheikholeslam S., Desoer C. A., Longitudinal control of a platoon of vehicles i: Linear model. Technical Report UCB/ERL M89/106, EECS Department, University of California, Berkeley, 1989
- [13] Swaroop D., String stability of interconnected systems: An application to platooning in automated highway systems, UC Berkeley: California Partners for Advanced Transit and Highways (PATH), 1997
- [14] Swaroop D., Rajagopal K., A review of constant time headway policy for automatic vehicle following, In Proceedings of IEEE Intelligent Transportation Systems, pages 65-69, 2001
- [15] Thuilot B., Bom J., Marmoiton F., Martinet P., guidance of an urban electric vehicle relying on a kinematic gps sensor, In 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV04), Lisboa (Portugal), July 2004.
- [16] Vuchic V. R., Specialized Technology Systems, in Urban Transit Systems and Technology, John Wiley Sons, Hoboken, NJ, USA, 2007
- [17] Yanakiev D., Kanellakopoulos I., Variable time headway for string stability of automated heavy-duty vehicles, In Proc 34th IEEE CDC, pages 4077-4081, 1995
- [18] Yanakiev D., Kanellakopoulos I., Variable Time Headway for String Stability of Automated Heavy-Duty Vehicles, Proc. 34th IEEE CDC 1995, 4077-4081, 1995
- [19] Ricardo (2009), Cars that drive themselves can become reality within ten years, <http://www.ricardo.com/en-GB/News-Media/Press-releases/News-releases1/2009/Cars-that-drive-themselves-can-become-reality-within-ten-years/>

# Optical Flow Templates for Superpixel Labeling in Autonomous Robot Navigation

Richard Roberts and Frank Dellaert  
Georgia Institute of Technology

**Abstract**—Instantaneous image motion in a camera on-board a mobile robot contains rich information about the structure of the environment. We present a new framework, *optical flow templates*, for capturing this information and an experimental proof-of-concept that labels superpixels using them. Optical flow templates encode the possible optical flow fields due to egomotion for a specific environment shape and robot attitude. We label optical flow in superpixels with the environment shape they image according to how consistent they are with each template. Specifically, in this paper we employ templates highly relevant to mobile robot navigation. Image regions consistent with *ground plane* and *distant structure* templates likely indicate free and traversable space, while image regions consistent with neither of these are likely to be nearby objects that are *obstacles*. We evaluate our method qualitatively and quantitatively in an urban driving scenario, labeling the ground plane, and obstacles such as passing cars, lamp posts, and parked cars. One key advantage of this framework is low computational complexity, and we demonstrate per-frame computation times of 20ms, excluding optical flow and superpixel calculation.

## I. INTRODUCTION

For a camera attached to a mobile robot, instantaneous image motion contains rich information about the robot’s egomotion and the structure of the environment. In this paper, our contribution is to present a new framework for capturing this information, in which we attempt to address some shortcomings of previous work, and present an experimental proof-of-concept of this framework.

This framework, which we call *optical flow templates*,

illustrated in Figure 1, permits semantically labeling image regions according to their observed optical flow and the predicted optical flow of several templates. Optical flow templates predict the optical flow due to robot egomotion, for a given robot velocity and attitude, and for a particular type of environment structure.

In this work, we use templates for *ground plane* and *distant structure*, and label regions that are not consistent with any other template as possible *obstacles*. Ground plane labels are likely to indicate fairly flat ground that can be driven upon. Far-away objects “at infinity” indicate line-of-sight directions that are likely free of obstacles. Image regions that are not consistent with either of these are likely to be nearby objects or other moving objects to be considered as obstacles.

The work to date towards using image motion for autonomous navigation has several drawbacks that limit its usefulness. While our framework does not completely solve the problem, it is a new way of looking at the problem that addresses some of the drawbacks.

In Section II we review related work and our position with respect to it, in Sections III and IV we explain optical flow templates and a method for labeling superpixels using them, and in Section V evaluate the method qualitatively, quantitatively, and in computational efficiency, in an urban driving scenario.

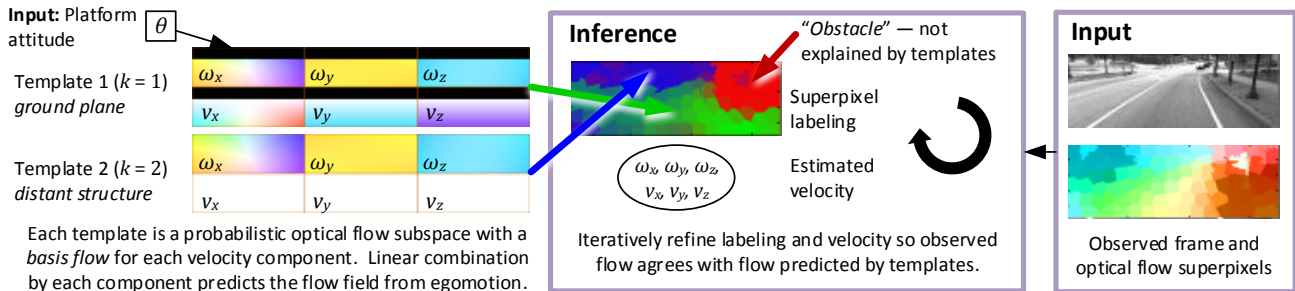


Fig. 1: *Optical flow templates* for superpixel labeling. Optical flow templates  $W^k(\theta)$ , one for each structure class *ground plane* ( $k=1$ ) and *distant structure* ( $k=2$ ), predict optical flow  $u_i$  at each  $i^{\text{th}}$  image location, given a platform velocity estimate  $\xi = [\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^T$  and attitude estimate  $\theta \in \mathbb{SO}(3)$ . Each template consists of 6 flow fields that combine linearly for each velocity component. For the optical flow color code, see Figure 3. Using observed optical flow, alternatively refine the labeling  $k_i$  for each  $i^{\text{th}}$  superpixel and the estimated velocity  $\xi$ . The special class  $k=0$  indicates optical flow pixels that cannot be explained by any template, which we label as *obstacle*.

## II. RELATED WORK

In autonomous driving, the current standard is to compute a 2D traversability map for path planning using information from 3D laser range scans, stereo correspondences, and structure from motion; for examples see [1], [2]. Becker *et al.* [3] accumulate optical flow information over short spans of time to infer a dense 3D reconstruction of the scene in front of the robot. The main drawback of these methods is the expensive computation required to calculate and then analyze large point clouds accumulated over many frames, both optimization problems with many variables. In comparison, our method uses very little computation because it directly estimates labels from optical flow data between pairs of frames, a single optimization problem with many fewer variables.

Instead of this standard method of measuring or computing 3D point clouds, other lines of research semantically label image regions using image appearance and machine learning [4], [5]. With applications outside of robotics as well, Hoiem *et al.* [6] also uses monocular features and machine learning to estimate 3D structure from single images. Our method, which uses optical flow information, is *complementary* to these methods that use image appearance. In this paper, our goal is to evaluate the utility of optical flow information alone, and to combine it with the appearance cues of this related work in future work.

Several lines of research combine 3D or image motion information with image appearance information using machine learning methods both to estimate semantic image labels and high-level semantic 3D structure. Brostow *et al.* [7] segment images into relevant regions such as street, sidewalk, car, *etc.* using structure-from-motion cues. Sturgess *et al.* [8] estimate similar segmentations using motion appearance and structure-from-motion information. These methods work well but require much labeled training data, and here we suggest that there is in fact a lot of information to be extracted from optical flow *before* having to use hand-labeled training data.

Geiger *et al.* [9] infer 3D street and traffic patterns from video from a moving platform, combining information from vehicle tracking, vanishing points, and image appearance. The information extracted is very rich, but comes at a high computational cost that makes it infeasible for small platforms.

Perhaps most closely related in method to this paper is that of Giachetti *et al.* [10], who use the differences between the observed optical flow and the flow predicted given motion on a ground plane to segment out other cars in the road. Additionally, Nourani-Vatani *et al.* [11] use optical flow for environment shape recognition with a discriminative learning method, matching flow fields to a database of locations using the flow field spatial statistics.

In contrast to planning over dense 2D or 3D maps, in which interpretation can be very difficult and computationally expensive, many researchers have investigated mobile robot control directly from optical flow. Inspired by research

into the role of optical flow in animal and human navigation [12], [13], Duchon *et al.* [14] evaluate control laws for chasing, escaping, and other behaviors. Srinivasan *et al.* review a number of other bio-inspired optical flow control strategies, developed both by their group and others [15]. The main drawback of these methods is that they use heuristics, such as left-right flow balancing, that make their behavior difficult to predict and result in systematic errors. For that reason, we explicitly leverage a geometric model.

Conroy *et al.* [16] and Hyslop and Humbert [17] develop methods for autonomous robot control using “wide-field optic flow integration”, which takes inner-products of the observed flow fields with a set of template flow fields. Using knowledge of possible coarse scene geometries such as walls and corridors, they develop templates and control laws. Our goal is different in that instead of inferring this coarse structure, we label finer structures and individual obstacles.

## III. OPTICAL FLOW TEMPLATES

An optical flow template encodes the space of possible optical flow fields corresponding to a certain environment structure, invariant to the platform velocity. In turn, optical flow templates also specify a linear mapping from platform velocity  $\xi = [\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^T$  to optical flow, for a given robot attitude, and for a particular environment structure class. Shown in Figure 1, in this paper we specifically work with 3 possible classes: *ground plane*, *distant structure*, and *obstacle*. Let  $W^k(\theta) \in \mathbb{R}^{2wh \times q}$  be the optical flow template for environment structure class  $k$ , where  $w$  and  $h$  are the image width and height, and  $q$  is the velocity dimension (in this paper  $q = 6$ ). Optical flow templates are nonlinear functions of the platform attitude  $\theta$ . Thus they predict a flow field for structure class  $k$  as  $u^k = W^k(\theta)\xi$ .

### A. Optical Flow as a Gaussian Mixture of Templates

Because each pixel may be generated from any template, we model the optical flow  $u_i$  at each pixel  $i$  as a Gaussian mixture of the flow predicted  $\hat{u}_i^k$  by each template,

$$p(u_i | \Lambda_i, \xi) = \mathcal{N}(0, \Sigma^0)^{\Lambda_i^0} \prod_{k=1}^{\kappa} \mathcal{N}(\hat{u}_i^k(\xi), \Sigma)^{\Lambda_i^k}, \quad (1)$$

where  $\Lambda_i^k \in \{0, 1\}$  is a binary indicator of 1 if pixel  $i$  takes the *discrete* label  $k$ , and 0 otherwise.  $\Sigma$  is the covariance of the Gaussian noise on the optical flow consistent with the template, and  $\Sigma^0$  is the covariance of zero-mean Gaussian noise on optical flow that is not consistent with any template, i.e. is labeled as *obstacle*.  $\hat{u}_i^k(v) = W_i^k(\theta)\xi$  is the optical flow predicted by template  $k$  at pixel  $i$ , and we assume that the platform attitude  $\theta$  is known, as explained in Section III-B where we calculate the templates. The notation  $W_i^k$  selects the pair of rows corresponding to pixel  $i$  from the optical flow template  $W^k$ . As is typical with notation in Gaussian mixtures, raising each term to the power of the indicator variable effectively makes only one term active for a given labeling.

Because the optical flow templates result in a linear relationship between velocity and optical flow, the measurement

likelihood in Eq. 1 is Gaussian, allowing inference to be fast and guaranteeing convergence. Each optical flow template is one instance of the optical flow subspace model introduced in our previous work [18].

### B. Calculating Optical Flow Templates

The optical flow field in a calibrated projective camera is (see e.g. [19] for derivation)

$$u_i = \begin{bmatrix} y_i & \frac{x_i y_i}{f} & -f - \frac{x_i^2}{f} & \frac{x_i}{z_i} & \frac{-f}{z_i} & 0 \\ -x_i & f + \frac{y_i^2}{f} & \frac{-x_i y_i}{f} & \frac{y_i}{z_i} & 0 & \frac{-f}{z_i} \end{bmatrix} \xi, \quad (2)$$

where  $x_i$  and  $y_i$  are the horizontal and vertical image coordinates (with the origin at the image center) at pixel  $i$ ,  $z_i$  is the depth at pixel  $i$ , and  $f$  is the focal length. The matrix is arranged such that  $\xi = [\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^T$  is the rotational and translational velocity in “robot coordinates”, where the  $x$ -axis points forwards and the  $z$ -axis points down.

For a flow field  $u$  comprised of the concatenated flow vectors  $u_i \in \mathbb{R}^2$ , the optical flow template  $W^k(\theta)$  is thus formed by stacking all of the matrices in Eq. 2. The depth  $z_i$  at each pixel depends on both the scene structure and the platform attitude  $\theta$ . In this paper, we consider templates corresponding to *ground plane* and *distant structure*. For *ground plane*, we compute  $z_i$  using plane-line intersection geometry. For brevity, we omit the derivation which is a straightforward application of the tools in Chapter 2 of [20]. For *distant structure*, we use  $z_i = \infty$  everywhere. In Figure 1, the ground plane template shows the camera is pitched slightly down – the boundary between the black and the colorful regions is the horizon.

In this paper, we assume the platform attitude  $\theta$  is known. In our experiments, we obtain it from an attitude/heading reference system (AHRS). On autonomous ground vehicles and aircraft, this is common equipment, and is typically implemented using measurements from an accelerometer and gyroscope [21].

## IV. SUPERPIXEL LABELING

Our method assigns a probability that each superpixel of optical flow in a frame of video belongs to each class, with each class represented by an *optical flow template*. Inference is a matter of labeling superpixels such that all superpixels assigned to the same template exhibit consistent optical flow within that template, and that all superpixels across all templates predict a consistent platform velocity. To simplify notation, we do not include frame numbers or time steps, but each frame is labeled independently.

Let  $\lambda_j \in \mathbb{R}^\kappa$  be the vector of probabilities that superpixel  $j$  belongs to each class, and  $\lambda_j^k \in \mathbb{R}$  be the  $k^{\text{th}}$  element, i.e. the probability that superpixel  $j$  belongs to class  $k$ . Let  $k \in \{0, 1, 2\}$ , where the special class  $k = 0$  corresponds to *obstacle*, and  $\kappa = 3$  is the number of classes (including *obstacle*). Both the superpixel labels  $\lambda_j$  and the velocity estimate  $\xi$  are alternatively refined, but for simplicity we omit iteration numbers from the notation.

### A. Superpixel Labeling Given a Velocity Estimate

Our end goal is to label superpixels according to which optical flow template they are consistent with. This requires knowing the optical flow templates themselves (as obtained in Section III-B). Although each optical flow template encodes an optical flow subspace that is invariant to platform velocity, we want to enforce that all superpixels predict a consistent platform velocity. Thus, we will iteratively estimate the platform velocity (as explained in Section IV-B), and in this section, take the current velocity estimate  $\xi$ .

While in the previous Section III we defined the density of optical flow at a pixel  $i$  predicted by an optical flow template, we now switch to superpixels. All of the inference here may equally be performed at the pixel level, but using superpixels greatly reduces the computational expense. We use the index  $j$  to denote a superpixel, and use  $u_j$  and  $\Lambda_j$  to denote the flow and labeling of a superpixel, and  $W_j^k(\theta)$  to denote the pair of rows corresponding to the pixel at the center of superpixel  $j$ .

We estimate the probability  $\lambda_j^k$  of each  $j^{\text{th}}$  superpixel belonging to each class. These probabilities define a multinomial distribution  $p(\Lambda_j = e_k) = \lambda_j^k$  over the labels (where  $e_k \in \mathbb{R}^\kappa$  is a vector with 1 in position  $k$  and 0 elsewhere, i.e. an assignment of the discrete indicator variables). Each assignment probability  $\lambda_j^k$  is the normalized likelihood that the superpixel is consistent with template  $k$ ,

$$\lambda_j^k = \frac{l(\Lambda_j = e_k | \tilde{u}_j, \xi)}{\sum_{\bar{k}} l(\Lambda_j = e_{\bar{k}} | \tilde{u}_j, \xi)}, \quad (3)$$

where  $l(\cdot) \propto p(\cdot)$  denotes a likelihood and  $\tilde{u}_j$  is the *average* measured optical flow in superpixel  $j$ . We calculate the likelihoods here using Bayes’ law,

$$l(\Lambda_j | \tilde{u}_j, \xi) = p(\tilde{u}_j | \Lambda_j, \xi) p(\Lambda_j), \quad (4)$$

where the measurement likelihood  $p(\tilde{u}_j | \Lambda_j, \xi)$  is as in Eq. 1, except that the predicted flow  $\hat{u}_i^k(\xi)$  is calculated by choosing pixel  $i$  to be at the center of superpixel  $j$ .  $p(\Lambda_j)$  is the class prior, which in our experiments is a simple multinomial distribution.

In the next section, we describe refining the velocity estimate  $\xi$  given the labeling estimated with Eq. 3 above. As mentioned before, the velocity and labeling are alternately refined until convergence.

### B. Refining the Velocity Estimate Given the Labeling

To refine the velocity  $\xi$ , we treat the labeling  $\Lambda_j$  as a hidden variable and update the velocity using expectation-maximization (EM),

$$\xi \leftarrow \arg \max_{\xi} \langle \mathcal{L}(\xi | \tilde{u}, \Lambda) \rangle, \quad (5)$$

where  $\mathcal{L}(\cdot) = \log l(\cdot)$  denotes a log-likelihood,  $\tilde{u}$  and  $\Lambda$  are the collections of optical flow vectors and indicator vectors for all superpixels, and the expectation  $\langle \cdot \rangle$  is with respect to  $p(\Lambda | \tilde{u}, \xi)$ . Using Bayes’ law and the linearity of the expectation, we have

$$\langle \mathcal{L}(\xi | \tilde{u}, \Lambda) \rangle = \mathcal{L}(\xi) + \sum_j \langle \mathcal{L}(\tilde{u}_j | \Lambda_j, \xi) \rangle, \quad (6)$$



where the expectation is now with respect to  $p(\Lambda_j | \tilde{u}_j, \xi)$ . To compute the expected log measurement likelihood  $\langle \mathcal{L}(\tilde{u}_j | \Lambda_j, \xi) \rangle$ , we note that the class probabilities  $\lambda_j^k$  calculated in the previous section comprise the *sufficient statistics* for this EM algorithm, giving everything we need to calculate

$$\langle \mathcal{L}(\tilde{u}_j | \Lambda_j, \xi) \rangle = \sum_k \lambda_j^k \mathcal{L}(\tilde{u}_j | \Lambda_j = e_k, \xi), \quad (7)$$

where the the log measurement likelihood is the log of Eq. 1. Note that the velocity  $\xi$  used in Eq. 1 is in fact the variable being optimized for in this section, and is not fixed here as it was in Section IV-A.

Because the measurement likelihood is Gaussian, the expected log-likelihood is a quadratic, and thus refining the velocity estimate with the maximization in Eq. 5 is a linear least-squares problem that is easily solved using direct methods.

### C. Summary and Implementation of Method

Given each of the components we have introduced in this section, we now summarize the steps that take place for each incoming video frame here in Algorithm 1:

---

#### Algorithm 1: Superpixel labeling for each frame.

---

**Input:** Current and previous video frames.

**Input:** Platform attitude  $\theta$  (e.g. from AHRS).

**Result:** Class label probabilities  $\lambda_j^k$  for each superpixel  $j$  and class  $k$ .

- 1 For current frame, compute SLIC superpixels [22].
  - 2 For previous and current frames, compute TV-L<sup>1</sup> [23], [24] optical flow  $\tilde{u}$  and average flow  $\tilde{u}_j$  in each superpixel.
  - 3 Compute basis flows  $W_j^k(\theta)$ , at image locations corresponding to superpixel centers.
  - 4 Initialize class label probabilities uniformly,  $\lambda_j^k = \frac{1}{\kappa}$ .
  - 5 **while** change in  $\langle \mathcal{L}(\xi | \tilde{u}, \Lambda) \rangle$  is greater than convergence threshold<sup>1</sup>  $\epsilon$  **do**
    - Update  $\xi \leftarrow \arg \max_{\xi} \langle \mathcal{L}(\xi | \tilde{u}, \Lambda) \rangle$  (Sec. IV-B)
    - Update  $\lambda_j^k \leftarrow \frac{l(\Lambda_j=e_k | \tilde{u}_j, \xi)}{\sum_{\bar{k}} l(\Lambda_j=e_{\bar{k}} | \tilde{u}_j, \xi)}$  (Sec. IV-A)
  - end**
  - 6 **return**  $\lambda_j^k$
- 

## V. EXPERIMENTS

Our experimental platform is a car equipped with a Point Grey Flea3 (gigabit ethernet version) camera running at  $1380 \times 480$  at 10Hz, and a MicroStrain 3DM-GX3-45 inertial navigation system (INS), from which we use only the AHRS attitude estimate. Both were connected to an Intel Core i7 laptop for data logging, and the computations described in this paper were performed on the logged data.

<sup>1</sup>This convergence criteria relies on the likelihood normalization constant remaining constant during optimization. To accomplish this, it is sufficient to include in the likelihood calculations the normalization constants from all Gaussians involving the velocity  $v$ .

We operated the car in an urban environment. In collecting the dataset, we took care not to follow any car in front too closely. We did not make any other special considerations in driving style while collecting the dataset.

The parameters  $\Sigma = \text{diag}(7.0)$  and  $\Sigma^0 = \text{diag}(15.0)$  were used for the noise covariances. These values were selected empirically: Too tight covariance causes too many superpixels to be labeled as *obstacle* due to slight noise in optical flow measurements, while too loose covariance causes all superpixels to be labeled only *ground plane* or *distant structure*. For the label priors, we used  $p(\Lambda = \text{ground plane}) = 0.4$  and  $p(\Lambda = \text{distant structure}) = 0.4$  for superpixels below the horizon, and  $p(\Lambda = \text{ground plane}) = 0$  and  $p(\Lambda = \text{distant structure}) = 2/3$  above the horizon (in each case the remaining probability is for *obstacle*). These values were chosen by manual estimation of the image portion occupied by each class in several typical frames, although the algorithm is not very sensitive to this prior.

Our implementation of the algorithms is in C++ using the GTSAM factor graph library [25]. The C++ classes are wrapped in MATLAB using the `wrap` utility included in GTSAM, and we perform data loading, scripting, and visualization in MATLAB. All source code and datasets will be made available online on the authors' web site by the time of publication (link will be provided in camera-ready version).

### A. Qualitative Analysis

In Figure 2, we present examples of the labeling produced by our method that we consider successful. These examples are successful because the information they contain is useful for autonomous navigation. For the most part the major structure above the ground plane and close to the camera is labeled as “*obstacle*” (red), and the *ground plane* (green) and *distant structure* (blue) are mostly correctly labeled. The *obstacle* structure above the ground plane is often obstacles, road boundaries, or independently-moving objects. Such structure could be avoided by a navigation method or passed on to higher-detail vision processing.

Sometimes the labeling produced by our method contains errors. One type of error arises because the optical flow estimate is incorrect. Most frequently, this occurs in regions with smooth or repetitive texture, as shown in Figure 4a. This type of error points us towards future work because the root cause of this problem is computing optical flow as an *input* that is unaware of the global optical flow constraints provided by our optical flow templates. We expand on this in the discussion. Another type of error occurs when the differences between the predicted optical flow from two templates becomes too small to distinguish from noise, as shown in Figure 4b. In this case, superpixels on the ground plane may be labeled with 50% probability of belonging to either the *ground plane* or *distant structure* classes, indicated in the figure by the green/blue or turquoise blended color; also, ground plane superpixels may be labeled as *distant structure* if small error in the velocity estimate of our method causes

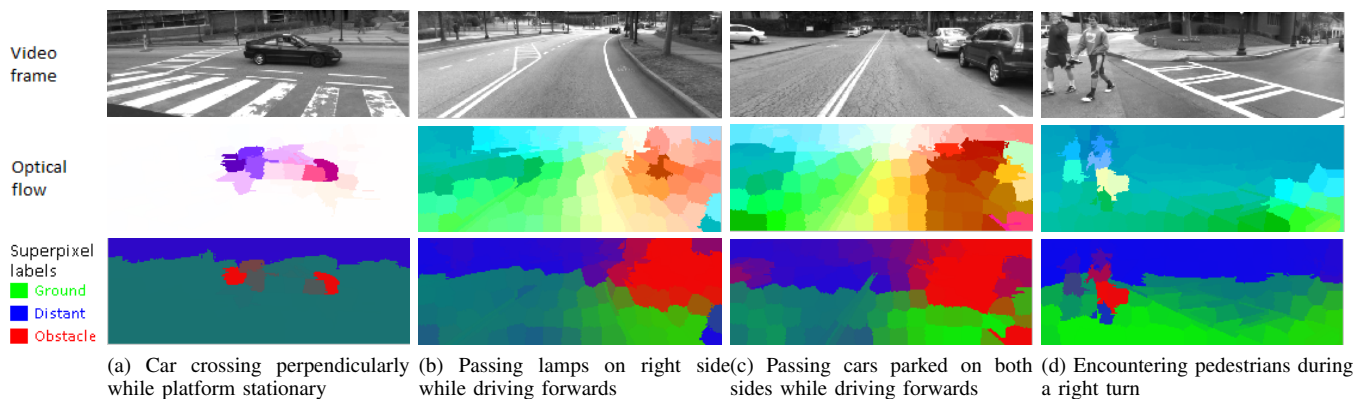


Fig. 2: Example video frames, superpixel optical flow fields, and superpixel labels by our method.

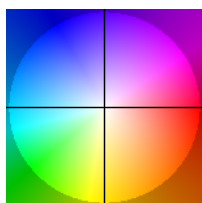
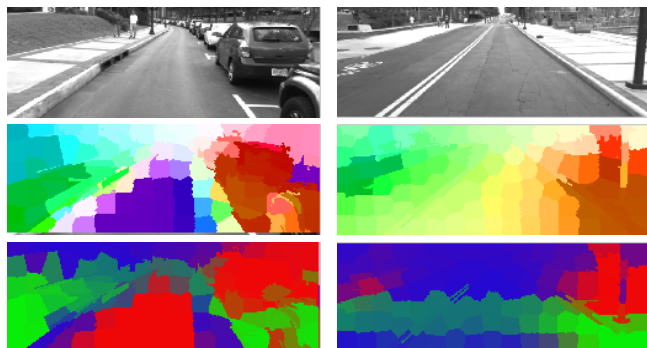


Fig. 3: The optical flow color code, as in [26]. Flow fields (e.g. Figure 1) are displayed with a color at each pixel. The hue indicates the direction of the flow and the saturation its magnitude. Here, the center of the black cross (color white) is zero flow. Yellow is downwards flow, red rightwards, etc.



(a) Error in labeling the ground as *obstacle* due to optical flow errors caused by smooth texture. (b) Error in labeling ground plane as *distant structure* due to similarity between rotational and translational basis flows.

Fig. 4: Examples demonstrating errors in labeling by our method.

the optical flow on the ground plane to agree more closely with rotational flow than with translational flow.

### B. Quantitative Analysis

We hand-evaluated 200 frames of video and labels produced by our method, the results of which are shown in Figure 5. In each frame, we determined the number of objects or regions of environment structure mislabeled by our method. “Extra obstacles” are image regions that our method labels in the *obstacle* class, yet there is no structure above

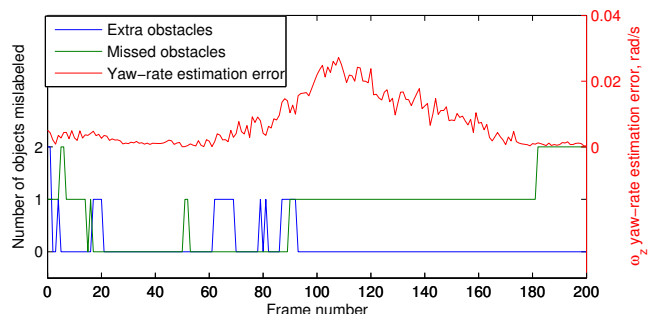


Fig. 5: Number of objects mislabeled by our method, evaluated by hand-inspecting 200 frames labeled by our method. We counted “extra obstacles”, which are image regions detected by our method as the *obstacle* class when no nearby structure was in fact present, and “missed obstacles”, where nearby structure was present but not detected.

the ground plane near the camera in that region. “Missed obstacles” are image regions containing structure above the ground plane near the camera but that our method does not label as *obstacle*. For example, in Figure 4a we would count the region mislabeled on the ground plane as one “extra obstacle”, and a missed car or pedestrian would count as one “missed obstacle”. In hand-labeling, we took into account the smoothing inherent in the optical flow calculation, so several objects close together, for example poles on the side of the road, are only counted as one object.

Because our method estimates superpixel labels jointly with platform velocity, we test whether errors in superpixel labels are coupled with errors in yaw-rate estimation. Figure 5 plots the yaw-rate error of our method as compared with the vehicle’s gyroscope. While gross yaw errors would certainly be coupled with many incorrect superpixel labels, the comparison demonstrates that there is no correlation between small yaw errors and errors in superpixel labeling.

### C. Timing

Our research implementation is single-threaded, and takes approximately 20 ms per frame ( $\pm 2$  ms) to infer superpixel labels and velocity, already having the optical flow and

superpixel segmentation available as input. For each frame, the combined time to calculate TV-L<sup>1</sup> optical flow and SLIC superpixels on the CPU is 500 ms per frame, however, there are GPU versions available of both algorithms that achieve real-time performance using CUDA. All timings were measured on an Intel Core i7 3.4 GHz desktop. The nature of the linear least-squares problem makes it adaptable to parallelization and GPU implementation.

## VI. SUMMARY AND FUTURE WORK

We have presented a new framework for interpreting optical flow observed by a mobile robot, called *optical flow templates*. Using templates for *ground plane* and *distant structure*, our method labels image regions whose flow is consistent with these templates, as well as labeling flow inconsistent with either as *obstacle*. This latter class comprises objects that occupy space above the ground plane near to the robot, and may be passed on for more detailed and computationally intensive processing.

The key aspects of the superpixel labeling method using this framework, in relation to previous work, include that computational complexity is very low, and geometric models of optical flow remove the need for hand-labeled training data or heuristics. We present an experimental proof-of-concept, labeling video in an urban driving scenario.

One direction of future work is to jointly infer optical flow along with the velocity and superpixel labels. We have observed, as shown in Section V-A, that many of the errors made by our method come from errors in optical flow estimation. These errors especially occur in regions of smooth texture, where the task of optical flow is underconstrained and ill-posed without a global optical flow model. Our optical flow templates in fact provide such a model, so joint inference should make much more accurate labels possible. Another benefit of joint inference will be sharper image segmentations of objects and boundaries between classes, which are currently blurred due to the smoothness terms necessary to compute dense optical flow.

## REFERENCES

- [1] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional lidar data for ground robot mobility," *Journal of Field Robotics*, vol. 23, no. 10, pp. 839–861, Oct. 2006.
- [2] A. Huertas, L. Matthies, and A. Rankin, "Stereo-Based Tree Traversability Analysis for Autonomous Off-Road Navigation," *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, pp. 210–217, Jan. 2005.
- [3] F. Becker and F. Lenzen, "Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences," *International Conference on Computer Vision*, 2011.
- [4] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning - ICML '05*. New York, New York, USA: ACM Press, 2005, pp. 593–600.
- [5] Y. N. Khan, P. Komma, and A. Zell, "High resolution visual terrain classification for outdoor robots," pp. 1014–1021, 2011.
- [6] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering Surface Layout from an Image," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 151–172, Feb. 2007.
- [7] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," *European Conference on Computer Vision*, 2008.
- [8] P. Sturges, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining Appearance and Structure from Motion Features for Road Scene Understanding," *Proceedings of the British Machine Vision Conference 2009*, pp. 62.1–62.11, 2009.
- [9] A. Geiger, M. Lauer, and R. Urtasun, "A generative model for 3D urban scene understanding from movable platforms," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Ieee, Jun. 2011, pp. 1945–1952.
- [10] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 34–48, 1998.
- [11] N. Nourani-Vatani, P. V. K. Borges, J. M. Roberts, and M. V. Srinivasan, "Topological localization using optical flow descriptors," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1030–1037, Nov. 2011.
- [12] J. J. Gibson, "Visually controlled locomotion and visual orientation in animals," *British journal of psychology*, vol. 49, pp. 182–194, Apr. 1958.
- [13] M. Lehrer, M. V. Srinivasan, S. W. Zhang, and G. A. Horridge, "Motion cues provide the bee's visual world with a third dimension," *Nature*, vol. 332, no. 6162, pp. 356–357, Mar. 1988.
- [14] A. Duchon, W. H. Warren, and L. P. Kaelbling, "Ecological Robotics: Controlling Behavior with Optic Flow," in *International Joint Conference on Artificial Intelligence*, 1995.
- [15] M. Srinivasan, S. Thurrowgood, and D. Soccol, "Competent Vision and Navigation Systems," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 59–71, Sep. 2009.
- [16] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *Autonomous Robots*, vol. 27, no. 3, pp. 189–198, Aug. 2009.
- [17] A. M. Hyslop and J. S. Humbert, "Autonomous Navigation in Three-Dimensional Urban Environments Using Wide-Field Integration of Optic Flow," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 147–159, Jan. 2010.
- [18] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [19] D. Heeger and A. Jepson, "Visual Perception of Three-Dimensional Motion," *Neural Computation*, vol. 2, pp. 127–137, 1990.
- [20] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [21] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008.
- [22] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–82, Nov. 2012.
- [23] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L 1 optical flow," *Ann. Symp. German Association Patt. Recogn.*, 2007.
- [24] J. Sánchez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, 2012.
- [25] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, 2012.
- [26] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, Nov. 2010.



## Session V

### Situation awareness & Risk Assessment

- **Keynote speaker: Christian Laugier (INRIA, Grenoble, France)**  
**Title: Road Scenes Understanding and Risk Assessment using Embedded Bayesian Perception**  
Co-authors: Mathias Perrollaz, Christopher Tay Meng Keat, Stéphanie Lefevre
- **Title: Detection of Unusual Behaviours for Estimation of Context Awareness at Road Intersections**  
**Authors:** Alexandre Armand, David Filliat, Javier Ibanez-Guzman
- **Title: Enhancing Mobile Object Classification Using Geo-referenced Maps and Evidential Grids**  
**Authors:** Marek Kurdej, Julien Moras, Veronique Cherfaoui, Philippe Bonnifait

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**



IROS'13

PPNIV'13

5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2013 IEEE/RSJ International Conference on Intelligent Robots and Systems

## Session V

Keynote speaker: **Christian Laugier (INRIA, Grenoble, France)**

### **Road Scenes Understanding and Risk Assessment using Embedded Bayesian Perception**

Co-authors: Mathias Perrollaz, Christopher Tay Meng Keat, Stéphanie Lefevre

**Abstract:** Robust analysis and understanding of dynamic scenes in road and urban traffic environments is needed to estimate and predict the collision risk level during vehicle driving. The risk estimation relies on the monitoring of the traffic environment of the vehicle either by means of on-board sensors, or by means of Vehicle-to-Vehicle (V2V) communications. In both cases, the collision risks are considered as stochastic variables. These variables are continuously evaluated and used by the vehicle embedded system to generate emergency warnings to the human driver, or to decide of the best driving action to be executed in the case of a fully autonomous vehicle. This talk addresses both the multi-modal embedded perception issue and the collision risk assessment problem. The perception issue is addressed using the new concept of "Bayesian Perception". The collision risk assessment problem has been solved using two complementary approaches leading to respectively use a "trajectory prediction" paradigm or a novel "intention/expectation" concept. In the first approach, the perception is performed using onboard sensors. Hidden Markov Model and Gaussian Processes are used to predict the likely behaviors of multiple dynamic agents in road scenes and to evaluate the related collision risks. This approach has been performed in collaboration with Toyota. In the second approach, V2V communication is used in the vicinity of road intersections for exchanging the dynamic states of the involved vehicles. In this context, we have shown that it is more efficient to identify dangerous situations by comparing "what drivers intend to do" with "what they are expected to do". What a driver intends to do is estimated from the motion of the vehicle, taking into account the layout of the intersection; what a driver is expected to do is derived from the current configuration of the vehicles and the traffic rules at the intersection. This approach has been developed in cooperation with Renault. Both approaches have been experimentally validated in simulation and on real experimental vehicles.

**Biography:** Dr. Christian Laugier is Research Director at INRIA and Scientific Leader of the *e-Motion* team-project. He is also responsible at the International Affairs Department of INRIA of the Scientific Relations with Asia & Oceania. He received the Ph.D degree in Computer Science from Grenoble University, France in 1976. His current research interests mainly lie in the areas of *Motion Autonomy*, *Intelligent Vehicles* and *Probabilistic Robotics*. He has co-edited several books in the field of Robotics and several special issues of scientific journals such as *IJRR*, *Advanced Robotics*, *JFR*, or *IEEE Trans on ITS*. In 1997, he was awarded the Nakamura Prize for his contributions to "Intelligent Robots and Systems". Dr. Christian Laugier is a member of several scientific committees such as the Steering/Advisory Committees of the IEEE/RSJ IROS, FSR, and ICARCV conferences. He is also co-Chair of the IEEE RAS Technical Committee on AGV & ITS. He has been General Chair, Program Chair or co-Chair of international conferences such as IEEE/RSJ IROS'97, IROS'02, IROS'08, IROS'10, IROS'12, or FSR'07. In addition to his research and teaching activities, he co-founded four start-up companies in the fields of Robotics, Computer Vision, Computer Graphics, and Bayesian Programming tools. He also served as Scientific Consultant for the ITMI, Aleph Technologies, and Probayes companies.

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**

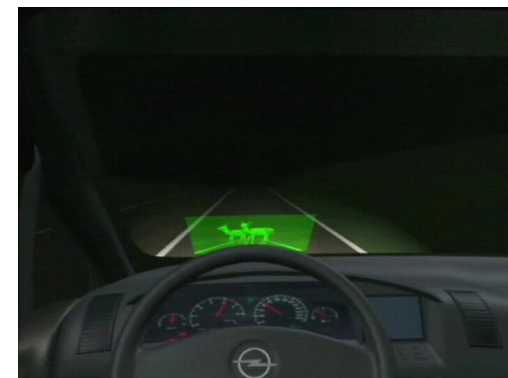
# Road Scenes Understanding & Risk Assessment using Embedded Bayesian Perception

Christian LAUGIER, First class Research Director at Inria

<http://emotion.inrialpes.fr/laugier>

*Contributions from*

*Mathias Perrollaz, Christopher Tay Meng Keat, Stephanie Lefevre*



*Keynote talk, PPNIV / IROS'13, Tokyo (November 2013)*

- **Context, State of the Art, New Challenges & Approach**
  
- **Bayesian Perception for Open & Dynamic Environments**
  - *Bayesian Perception paradigm*
  - *Embedded Perception & Bayesian Sensor Fusion*
  
- **Situation Awareness & Risk Assessment**
  - *Learn & Predict Paradigm*
  - *Trajectory Prediction & Probabilistic Collision Risk*
  - *Comparing Intentions & Expectations for Cooperative Safety*
  
- **Conclusion & Perspectives**



- Nowadays, Human Society is no more accepting the incredible socio-economic cost of traffic accidents !



***1.2 million fatalities / year in the world !!!!***

- *USA (2007) : Accident every 5s =>41 059 killed & 2.6 million injured*  
*.... Similar numbers in Europe*
- *France (2008): 37 million vehicles & 4443 fatalities (number reduced by 50% in the past years, thanks to both regulation & improved car technology).*  
*=> Human & financial cost estimated to 23 Md € for 2011 in France !*

- **Driving Safety** is now becoming a major issue for both governments (*regulations & supporting plans*) and automotive industry (*technology*)

- Thanks to the last decade advances in the field of *Robotics & ICT technologies, Smart Cars & ITS* are gradually becoming a reality

*=> Driving assistance & Autonomous driving, Passive & Active Safety systems, V2V & I2V communications, Green technologies ... and Sensors & Embedded Perception Systems*

- **Legal issue** has recently started to be addressed

*=> June 22, 2011: Law Authorizing Driverless Cars on Nevada roads ... and this law has also been adopted later on by California and some other states in USA*



# Governments plans for Robotics & IV Innovation

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

ENDING PAIN WITHOUT SIDE EFFECTS • THE MOUNTAINS THAT SANK

**SCIENTIFIC AMERICAN**

January 2007 \$4.99

If This Is a **PLANET**, Then Why Isn't Pluto?

**DAWN OF THE AGE OF ROBOTS**

Bill Gates writes that every home will soon have smart mobile devices

Evolution and Cancer

Can Ethanol Replace Gasoline?

Secret Controls for Genes

**President Obama announced Major Robotics Initiatives**

January 2007

**Bill Gates:**  
"The next hot field will be Robotics"

**President Obama announced Major Robotics Initiatives**

**France Robots Initiatives**

Mars 2013

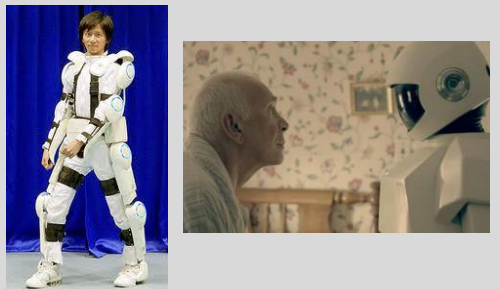
**+ 34 Industrial Plans (3.5 Md€), including**

- o Robotics (*Bruno Bonnell, Infogrames*)
- o Driverless Car (*Carlos Ghosn, Renault*)
- o Embedded Systems (*Eric Bantegnie, Esterel*)

MINISTÈRE DU REDRESSEMENT PRODUCTIF

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

**Japan**  
Next Generation Robots as one of the eight important areas promoted by the Government



**Taiwan**  
Intelligent Robotics Industry designated as the next-generation industry by the Government (expected to reach over NT\$ 90 billion in period 2009~13)

行政院國家科學委員會國際研究中心計畫  
National Science Council  
International Research Intensive Center of Excellence (IRIC)

**iCeIRA**

國立台灣大學智慧機器人及自動化國際研究中心  
NTU International Center of Excellence in Intelligent Robotics and Automation Research (NTU-iCeIRA)

**Korea**  
A 10 years Government plan (US\$ 316M) for developing Intelligent Robots

282

**"Korean robotics industry bursting into bloom"**  
*The Korea Herald/Asia News Network. Jul 20, 2011*

- ❑ **An EU driven concept since the 90's: “Cybercars”**
  - ✓ *Autonomous Self Service Urban & Green Vehicles at low speed*
  - ✓ *Numerous R&D projects in Europe during the past 20 years*
  - ✓ *Several European cities involved*
  - ✓ *Some commercial products already exist for protected areas (e.g. airports, amusement parks ...), e.g. Robosoft, 2GetThere , Induct...*
- ❑ **Several early large scale public experiments in Europe**



**Movie** : Floriade 2002, Amsterdam  
(2GetThere & Inria)



**Movie** : Shanghai public demo 2007  
(SJTU & Inria, EU FP7 project)



## □ Fully Autonomous Driving

- ✓ More than 25 years of research, for both Off-road & Road Vehicles
- ✓ Significant recent steps towards fully autonomous driving .... Partly pushed forward by events such as DARPA Grand & Urban Challenges ... and Google Car
- ✓ Fully Autonomous driving is gradually becoming a reality, for both the Technical & Legal point of views (e.g. Recent Nevada law for driverless cars)

## □ Results & Major events

Pioneer work at INRIA (mid 90's)



2007 Darpa Urban Challenge  
97 km, 50 manned & unmanned vehicles, 35 teams



2010 VIAC Intercontinental Autonomous Challenge  
13 000 km covered, 3 months race, leader + followers  
=> See Spring 2011 IEEE RAM issue



2011 Google Car project  
Fleet of 6 automated Toyota Prius  
140 000 miles covered on 284 California roads with occasional human interventions



Current Autonomous Vehicles are able to exhibit quite impressive skills .... BUT they are **not fully adapted to human environments** and they are often **Unsafe !**

## => DARPA Grand Challenge 2004

- ✓ Significant step towards Motion Autonomy
- ✓ But still some “Uncontrolled Behaviors” !!

## => URBAN Challenge 2007

- ✓ A large step towards road environments
- ✓ But still some accidents, even at low speed !!

## => Google Cars 2011 & Other projects in Europe

- ✓ Impressive results & fully autonomous driving capabilities
- ✓ But costly Sensors + Dense 3D mapping required + Human Factor weakly addressed !!



Some technologies are almost ready for use in some restricted or protected public areas

BUT

- ✓ Fully Open & Dynamic environments are still beyond the state of the art !
- ✓ Safety is still not guaranteed !
- ✓ Many costly onboard sensors & High computing power are still required !

## ❑ Cybercars : Some start-ups & first products



Cycab (Inria /Robosoft)



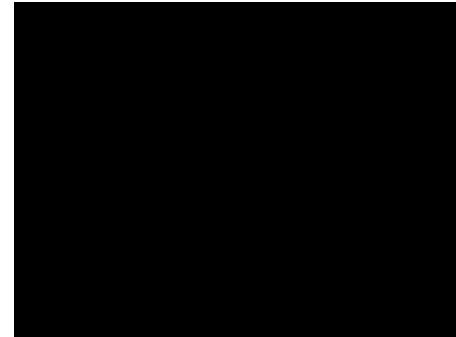
Cybergo (Induct)



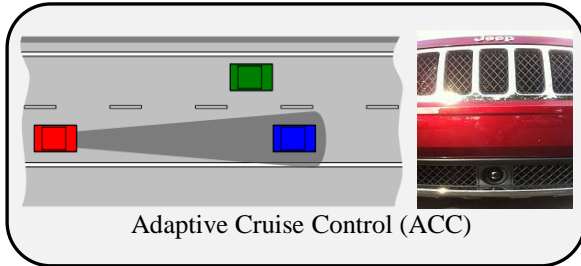
Amsterdam Schiphol Airport  
(2Get'Here, 1997-2004)



Cybus, La Rochelle 2012  
(CityMobil & Inria)



## ❑ ADAS : More and more equipped cars



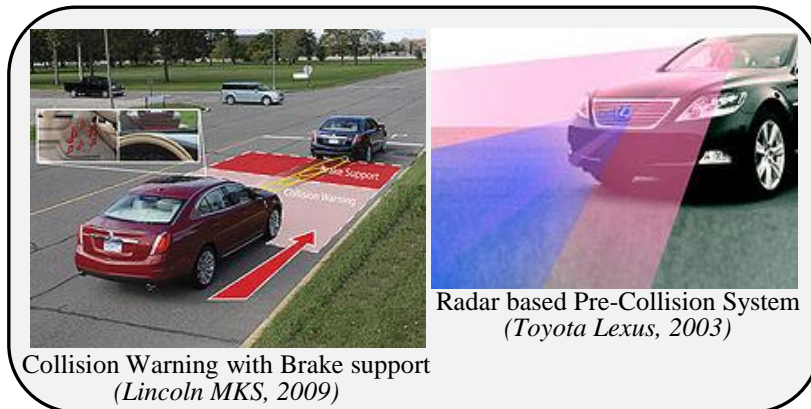
Adaptive Cruise Control (ACC)



Lane Guidance System (PCB and Camera sensor from Hyundai)



Night / Bad Weather Vision



Collision Warning with Brake support  
(Lincoln MKS, 2009)

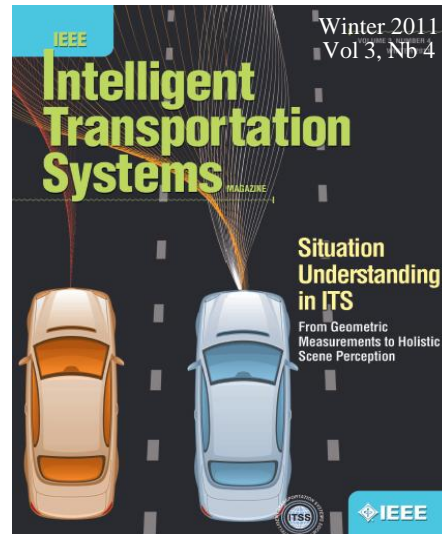
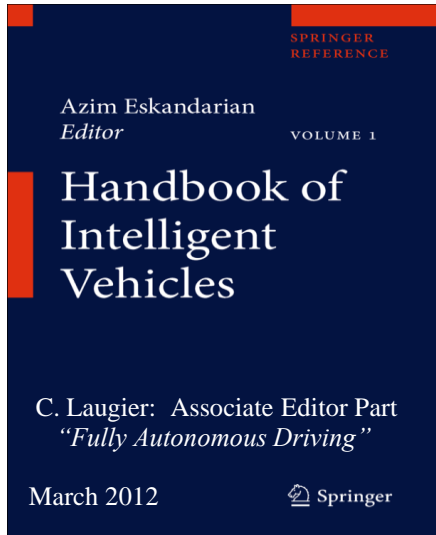


Radar based Pre-Collision System  
(Toyota Lexus, 2003)



Parallel Parking System  
(V1: Toyota Prius 2003 ; V2: Toyota Lexus 2006 & 2010)  
=> Inspired by Inria approach 1996





C. Laugier et al: “Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety”



Co-editors:  
C. laugier & J. Machan  
July 2013

Figure 1: All newly developed vehicles should be compatible with ITS guidelines and standards. Source: ETSI.

# Intelligent Cars & ITS - Towards Driverless Cars ?

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

## Horizon 2020-25 ?

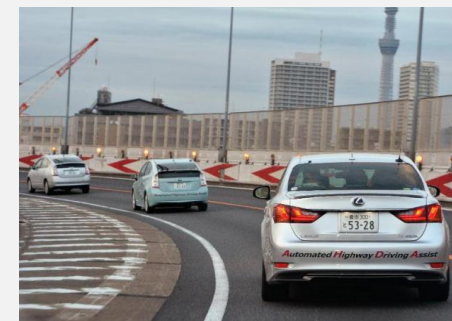
Nissan promises a driverless car for 2020

LE FIGARO

Date : 29/08/2013



Google Car 2011  
140 000 miles covered



Toyota

“Automated Highway Driving Assist”  
(Demo Tokyo 2013, Product 2015)

Voitures sans conducteur : Nissan va mettre un robot dans votre moteur !

Carlos Ghosn  
(Renault /Nissan)



LA TRIBUNE  
L'ESSENTIEL DE L'ACTUALITÉ ÉCONOMIQUE ET FINANCIÈRE



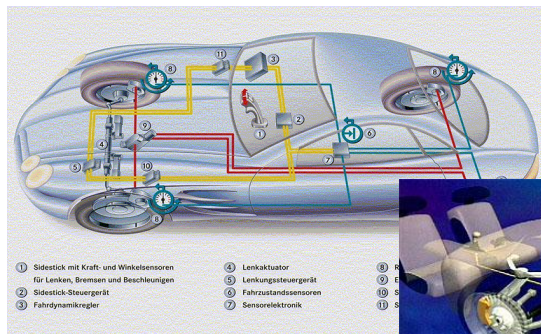
But also:  
Tesla (90% Autonomous, => 2016),  
Volvo, Mercedes Class S, BMW ...

Autonomous car: An industrial challenge for tomorrow !  
The French Minister of Industry Arnaud Montebourg promotes driverless car

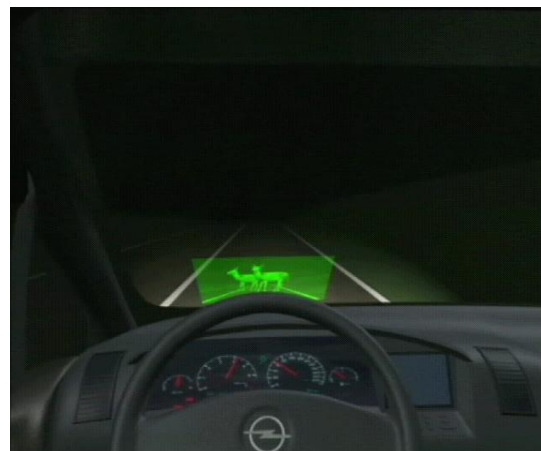
- ❖ Market Forecast : 8000 cars sold in 2020, about 95 millions in 2035
- ❖ Still some open questions: Why driverless cars ? Intelligent co-Pilot v/s Full Autonomy ? Acceptability ? Legal issue ? Driver / Co-Pilot Control transitions ?



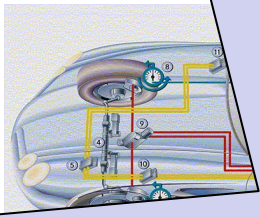
# Driving Assistance & Fully Autonomous Driving



*Steering by wire  
 Brake by wire  
 Shift by wire*



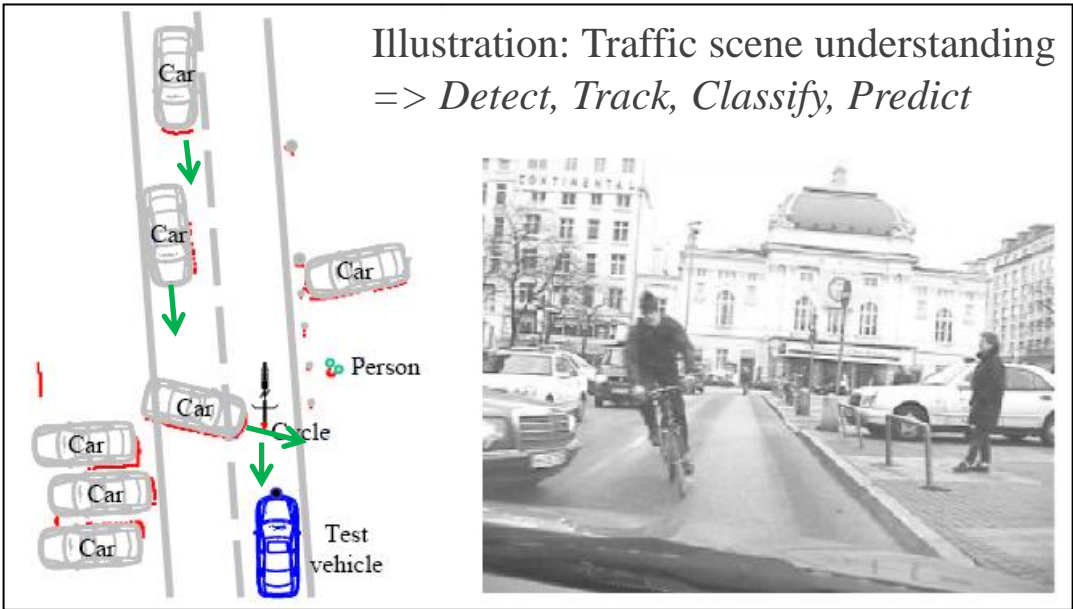
*Radar, Cameras, Night Vision, Multiple sensors ..... but also  
 "Sensor based Active Driving Assistance" (e.g. Automatic Parking )  
 => Cost decreasing & Efficiency increasing (future mass production,  
 embedded systems, SoC ...) !*



.... But a real deployment of *Advanced Technologies for ADAS & Autonomous Driving*, requires first to more deeply address two main technical issues:

- ✓ *Robust, Integrated, and Cheap enough “Embedded Perception Systems”*
- ✓ *Friendly Human – Vehicle Interactions*

ors, Parking e...  
easing (future mass  
ms ...) !!!!



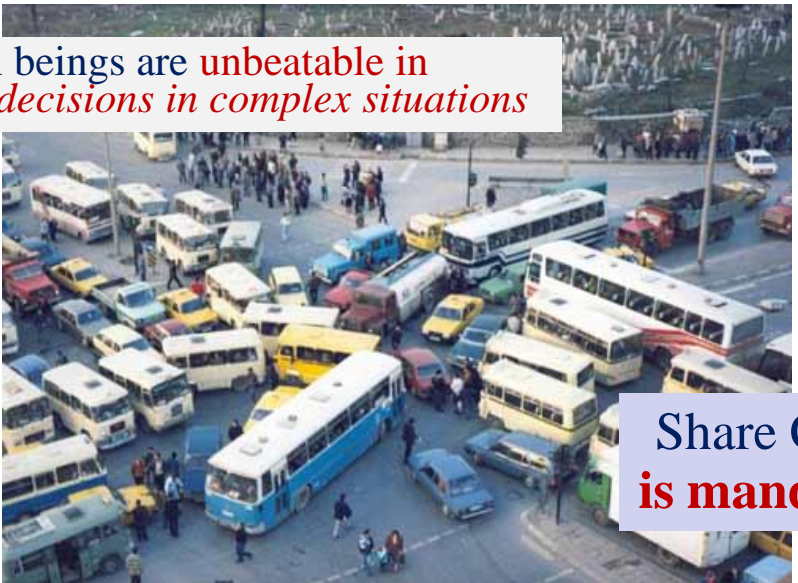
- ❑ **Dynamicity & Uncertainty**  
=> *Space & Time + Probabilities*
- ❑ **Interpretation ambiguities & Semantics**  
=> *History, context, prior knowledge + Sensor fusion*
- ❑ **Prediction of future states (recently addressed)**  
=> *Behaviors, prediction models*
- ❑ **Embedded Perception (necessary for deployment)**  
=> *Miniaturization & Software / Hardware integration*



# Challenge 2: Human Aware Navigation & Interaction

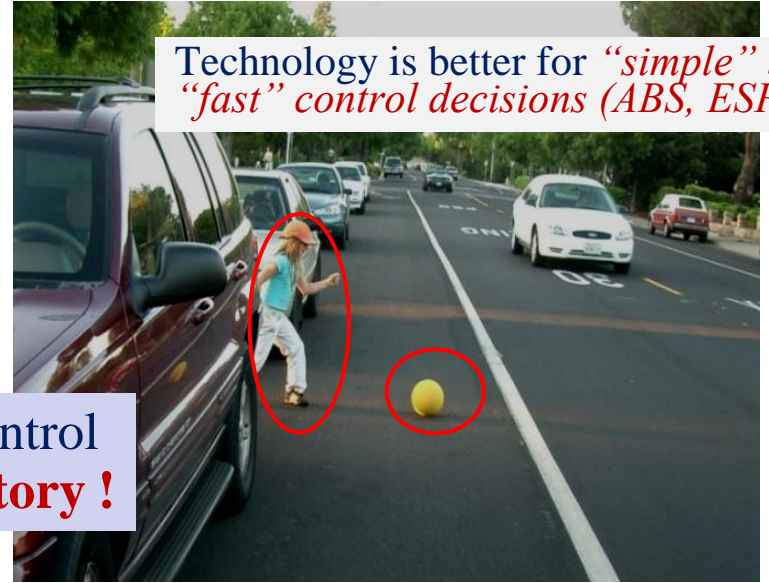
5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

Human beings are unbeatable in taking decisions in complex situations

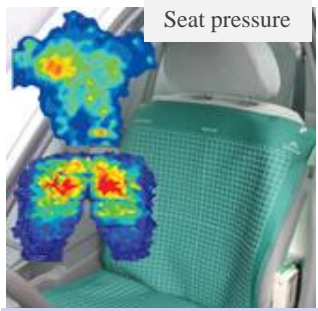


Share Control is mandatory !

Technology is better for "simple" but "fast" control decisions (ABS, ESP ...)



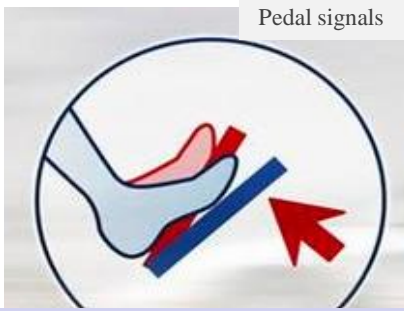
.... But Driver inattention is still a major cause of accident !



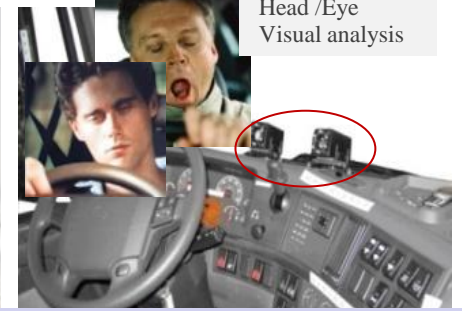
Seat pressure



Steering actions



Pedal signals



Head / Eye Visual analysis

**Driver Monitoring** (using on-board Perception)  
+  
**Safe & Socially Acceptable Human / Vehicle Interaction is necessary !**  
=> *"Mutual Driver / Vehicle understanding"*

# Key Technology 1: “*Bayesian Perception*”

- *Bayesian Perception paradigm (for Open & Dynamic Environments)*
- *Embedded Perception & Bayesian Sensor Fusion*

# Bayesian Perception Paradigm

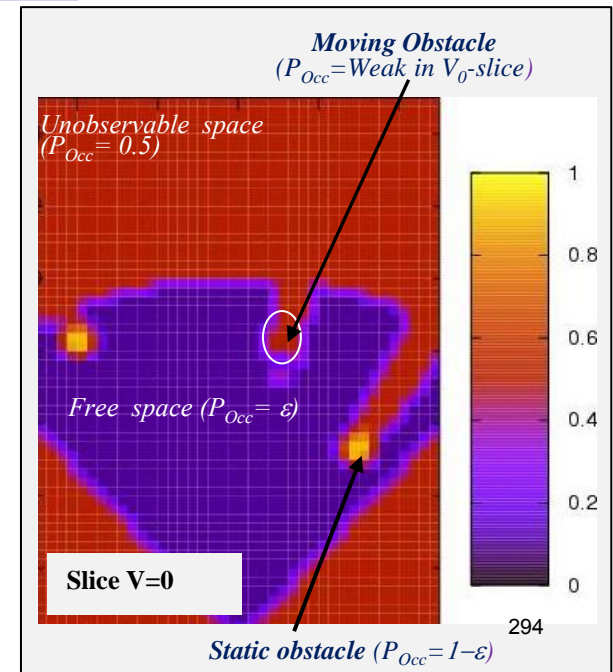
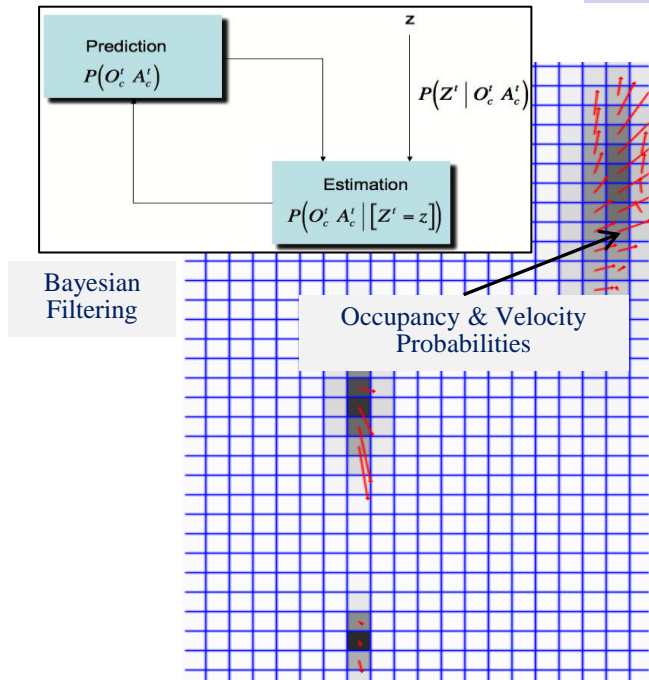
5th Workshop on Planning, Perception and Navigation of Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

## Bayesian Occupancy Filter (BOF)

- ❖ *Grid* approach based on *Bayesian Filtering* for **Dynamic Environments**
- ❖ Estimates at each time step the **Occupation & Velocity probabilities** for each cell in a “*Space-Velocity*” grid
- ❖ Bayesian Inferences performed using *Probabilistic Sensor & Dynamic models*
  - => *More robust to Sensing errors & Temporary occultation*
  - => *Designed for Sensor Fusion & Hardware implementation (GPU, Multi-core architectures, SoC)*

[Coué et al IJRR 05]

Patented by Inria & Probayes  
Commercialized by Probayes (2006)





## Temporary Space/Velocity Data Persistence & Bayesian Inferences => Application to Conservative Collision Anticipation

Autonomous  
Vehicle (Cycab)



Parked Vehicle  
(occultation)

[Coue et al IJRR 05]

Thanks to the prediction capability of the BOF technology, the Autonomous Vehicle “anticipates” the C.I behavior of the pedestrian and brakes (even if the pedestrian is temporarily hidden by the parked vehicle<sup>295</sup>)  
Perception” --- Keynote talk, WS PPNIV'13, IROS'13, Tokyo (Nov. 2013)

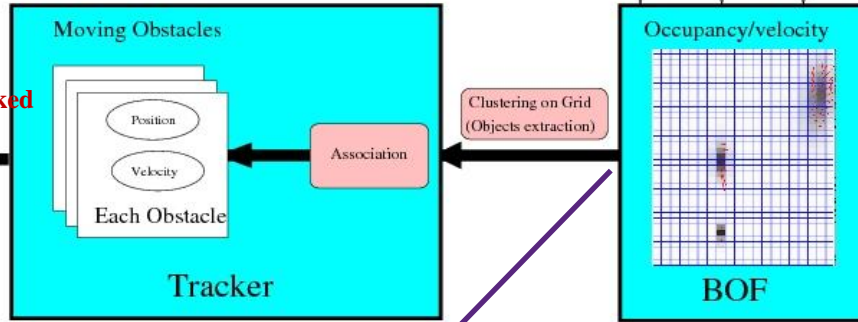
# Bayesian Perception for Dynamic Environments

Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

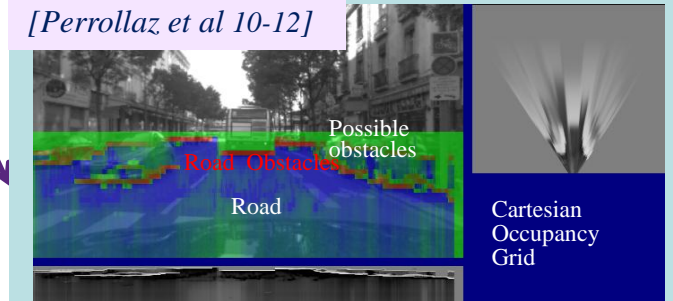
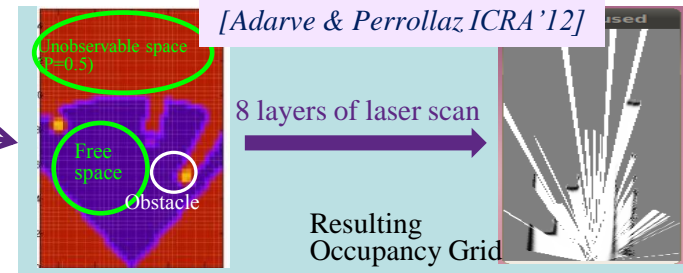
## Bayesian Sensor Fusion + Detection & Tracking

- Data association is performed as lately as possible
- More robust to Perception errors & Temporary occlusions

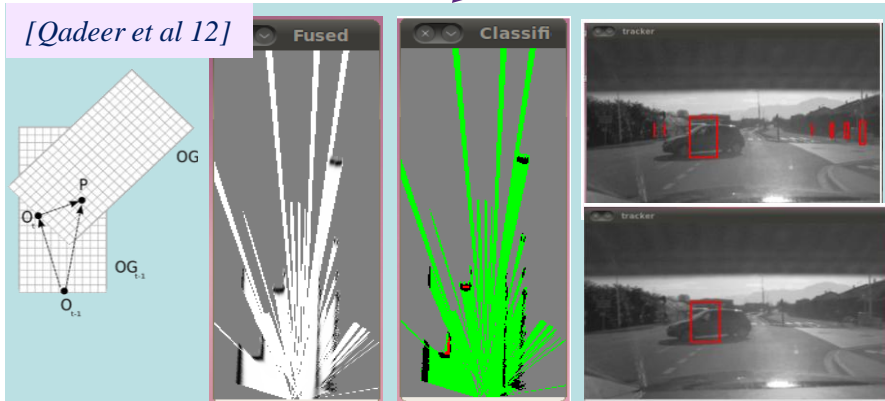
### Fast Clustering and Tracking Algorithm (FCTA)



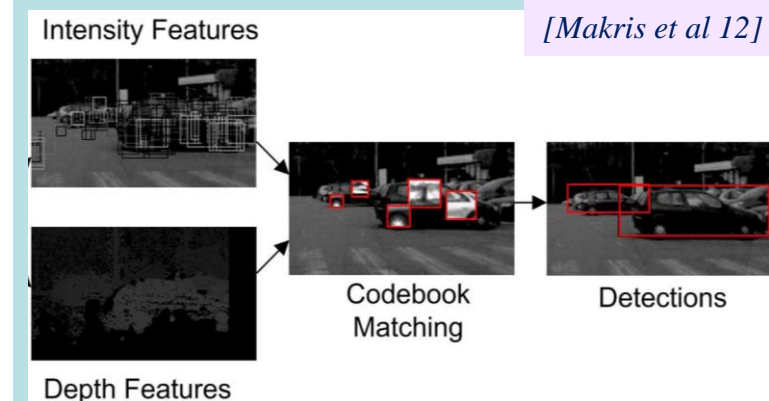
[Mekhnacha 09, Laugier et al ITSM'11]



U-disparity Occupancy Grid, with Road & Obstacle classification (superimposed on the camera image)



**Reducing False detection:** Static / Dynamic classification using Motion data (IMU) & Sliding window & Counting detection occurrences



Vehicle Detection & Recognition using Intensity & Depth features and codebook for characteristic parts of objects => more robust to partial occlusions



# Embedded Perception System (Lexus)

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (November 3rd, 2013, Tokyo, Japan)

CPU+GPU+ROS / Stereo + 2 Lidars + GPS + IMU



PC + GPU + ROS  
Inertial sensor & GPS (Xsens Mti-G)

Stereo camera TYZX

2 Lidars IBEO Lux

GPS track example  
(Using Open Street Map & GPS & IMU & Odometry)

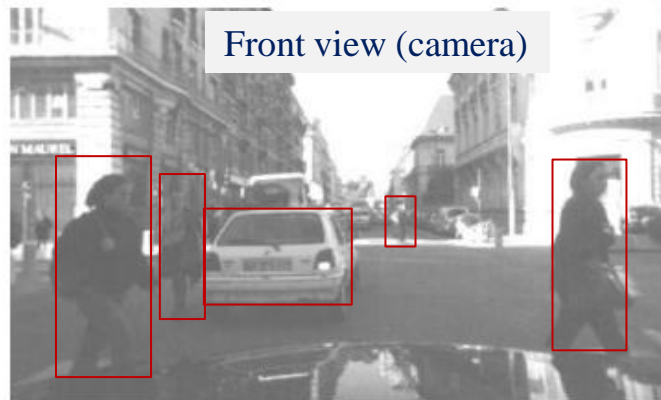


Navigable space & Collision risk

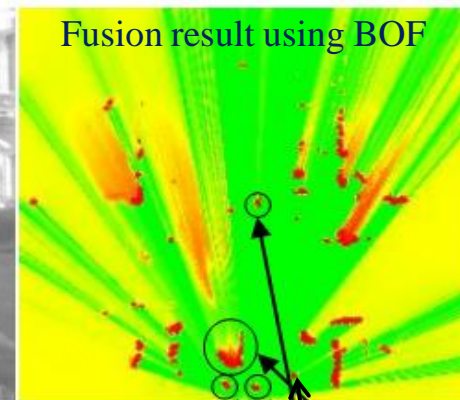


[Perrollaz et al 10] [Laugier et al ITSM 11]  
Iros Harashima Award 2012

Front view (camera)



Fusion result using BOF



a

OG from left Lidar



OG from right Lidar



OG from Stereo

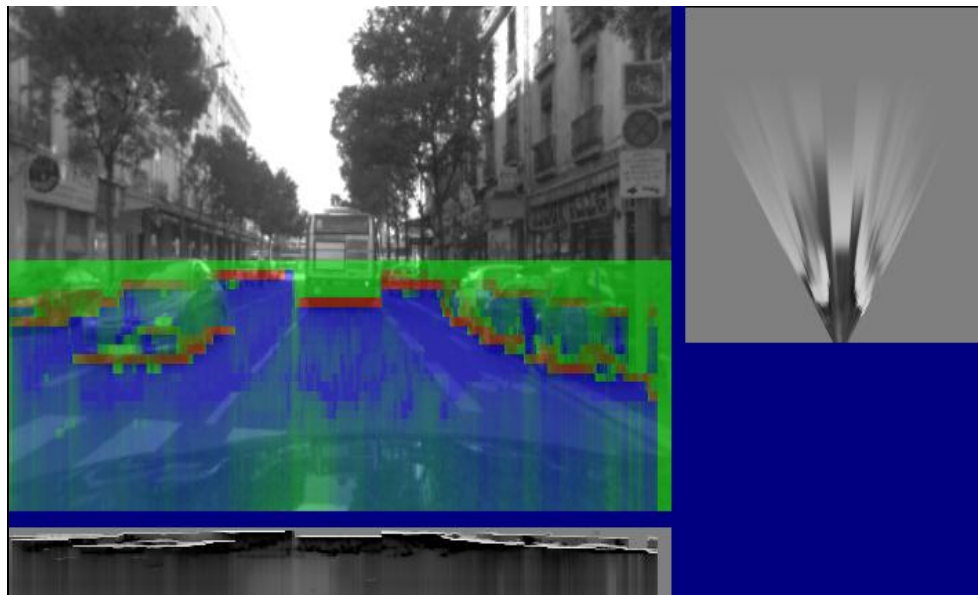


Car

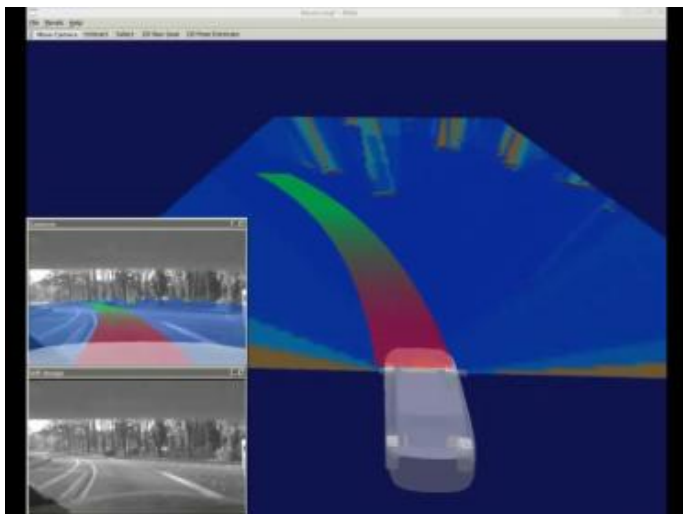
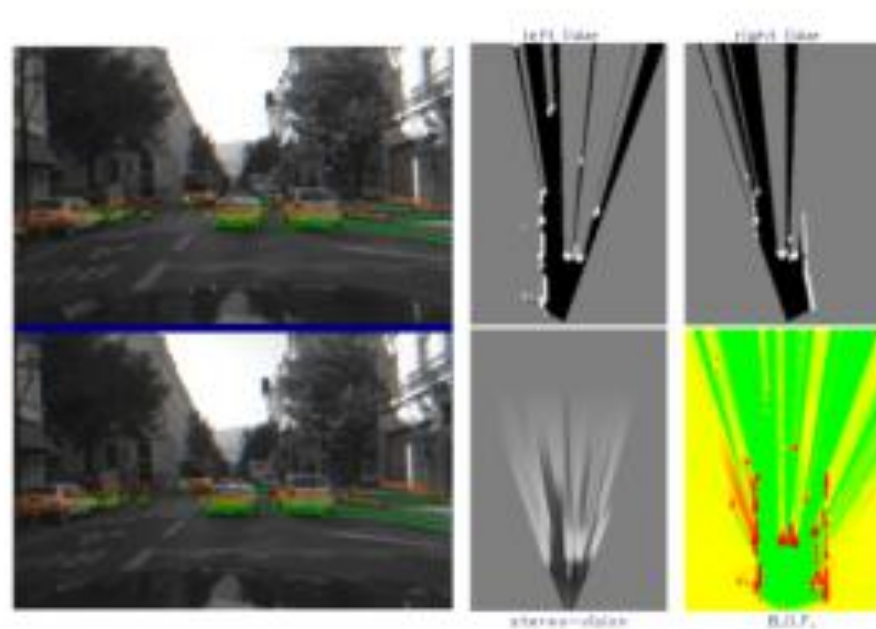
297 Pedestrians

# Bayesian Perception – Some experimental results

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan



Embedded perception on Lexus (cooperation Toyota)



Navigable Space & Risk



People Detection & Tracking using Fixed Cameras  
Inria & Probayes

# **Key Technology 2:** ***Situation Awareness & Risk Assessment***

- ❑ *Learn & Predict paradigm*
- ❑ *Trajectory Prediction & Probabilistic Risk Assessment*
- ❑ *Comparing Intention & Expectation for Cooperative Safety*

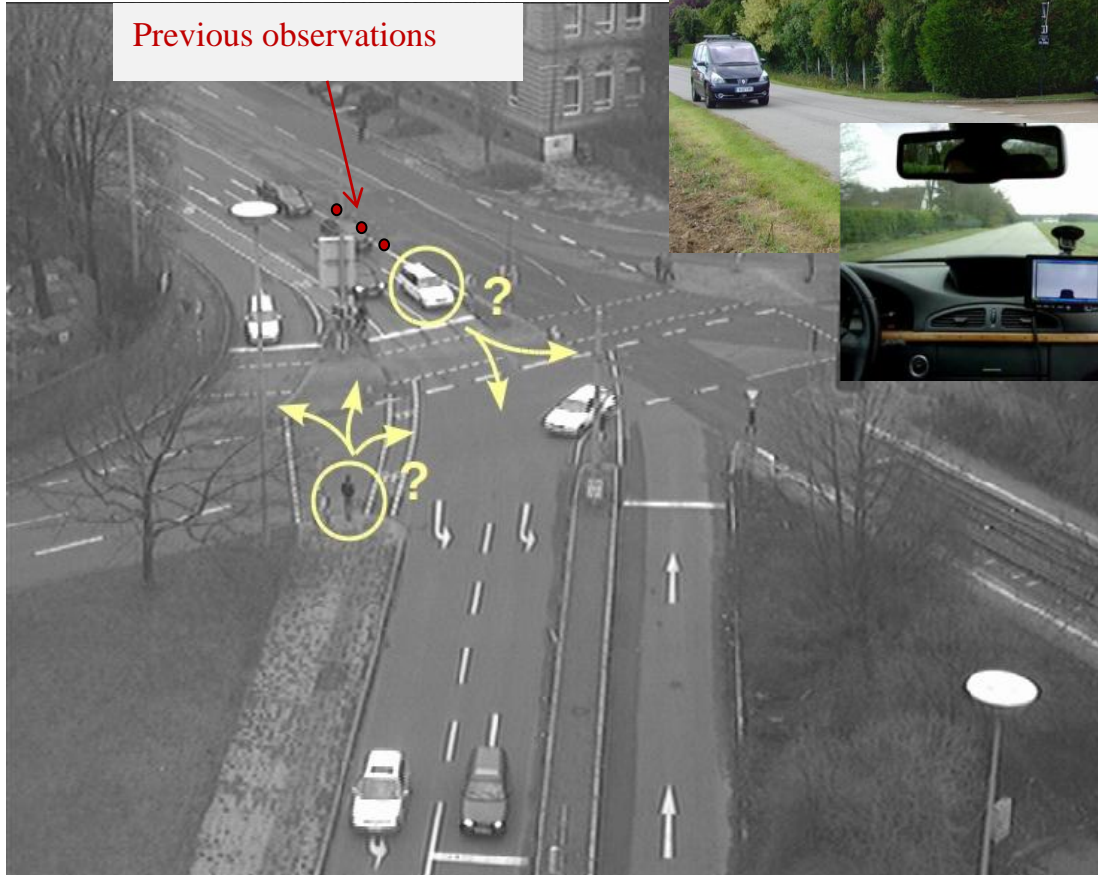


# Situation Awareness – Problem statement

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

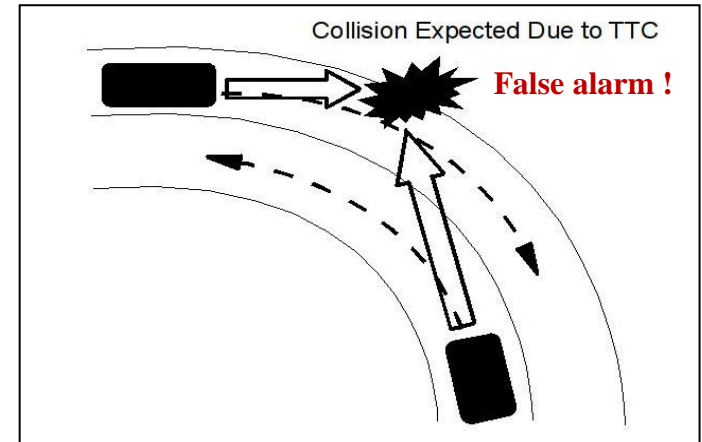
## Behavior Prediction + Probabilistic Risk Assessment

=> Understand the current situation & its likely evolution



Previous observations

Illustration using a road scene  
(Highly structured environment + Strict traffic rules)



Conservative TTC-based crash warning is  
**not sufficient !**



Previous observ.

=> **Consistent Prediction & Risk Assessment** requires to reason about:

- ✓ **History of obstacles Positions & Velocities**  
=> *Perception (Datmo) or V2V Communications*
- ✓ **Obstacles expected Behaviors**  
=> *Moving straight, turning, crossing, overtaking, stopping ...*
- ✓ **Space geometry / topology**  
=> *Road lanes, curves, intersections ...*
- ✓ **Traffic rules**

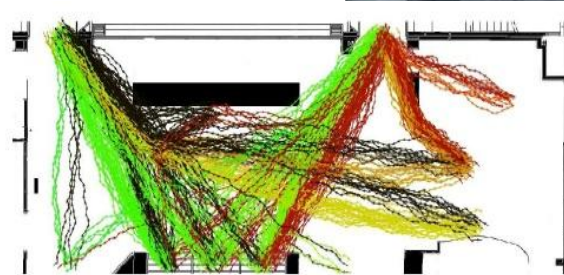
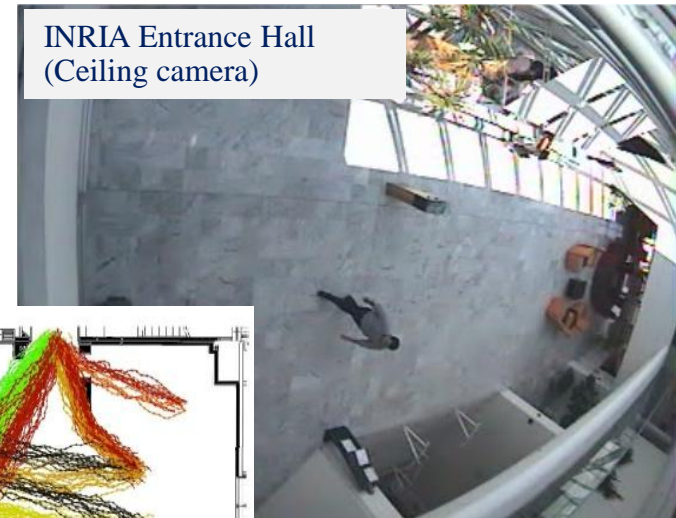


## Learn & Predict approach

[Vasquez & Laugier 07]

EU Euron PhD Thesis Award 09

- Concept of “intentional motion”
- Observe & Learn “typical motions & goals”
- Continuously “Learn & Predict”
  - ✓ Learn => GHMM & Topological maps (SON)
  - ✓ Predict => Exact inference, linear complexity



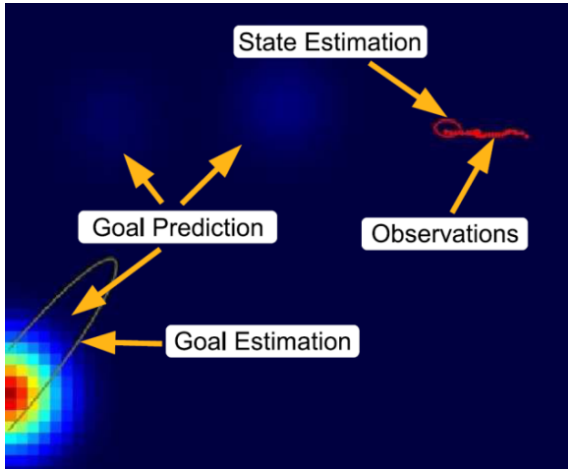
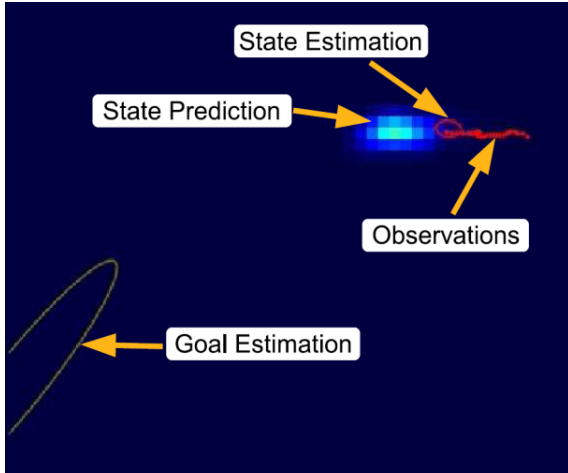
Learn



Predict



[Vasquez et al 07]



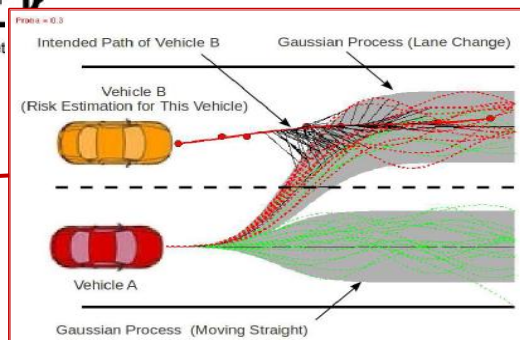
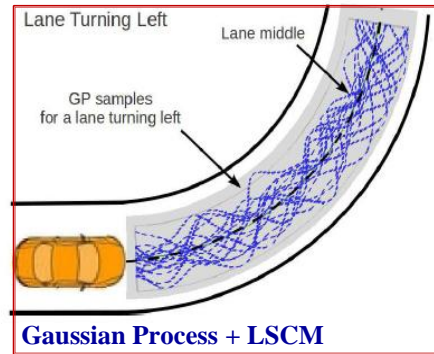
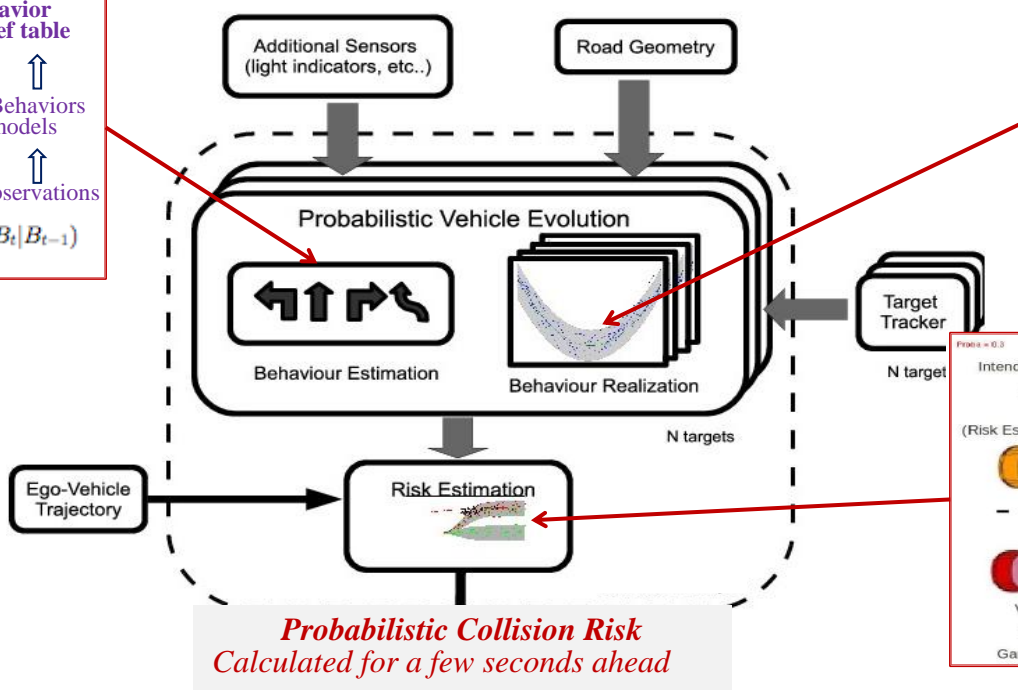
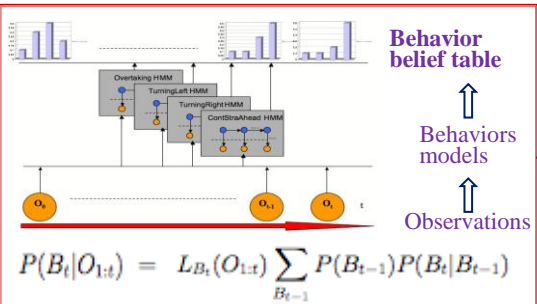
Experiments using Leeds University parking data

# Trajectory Prediction & Probabilistic Collision Risk

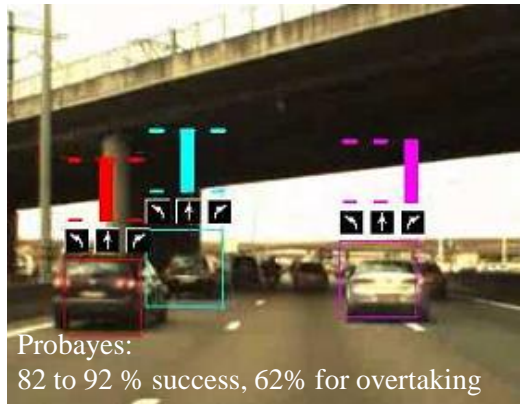
5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

[Tay 09] [Laugier et al 11]

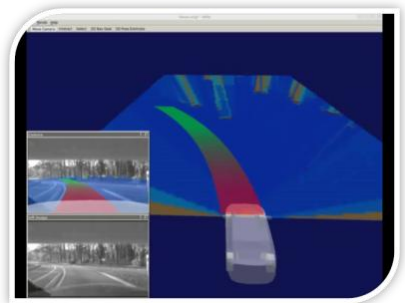
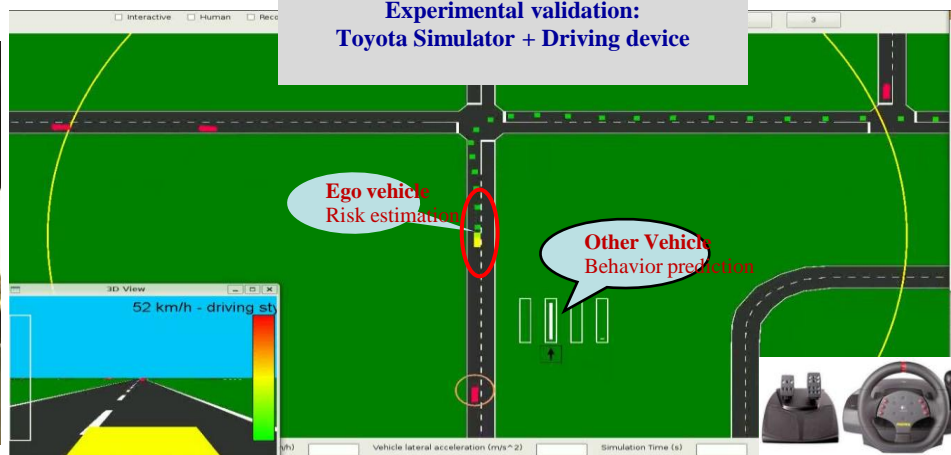
Patent INRIA & Toyota & Probayes 2010



Predicted 3s ahead



Experimental validation:  
Toyota Simulator + Driving device





# Risk = Comparing Drivers Intentions & Expectations

*Cooperation Stanford & Berkeley & Renault*

*[Lefevre & Laugier IV'12, Best student paper]  
Patent Inria & Renault*



## Intersection safety: a challenge

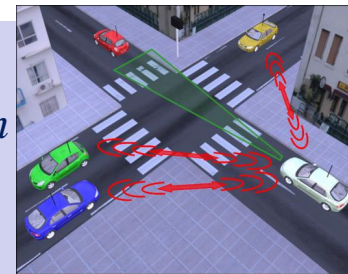
- ✓ 40-60% of road accidents in most countries
- ✓ 5 accidents out of 7 in DARPA Urban Challenge

## Risk assessment much more difficult !

- ✓ Complex Geometry & Traffic context
- ✓ Large number of Vehicles & Possible Maneuvers
- ✓ Vehicle behaviors are *Interdependent*
- ✓ *Human Drivers in the loop !*

## Our approach: A Human-like reasoning paradigm

- ✓ Exchanging vehicle states information (V2V communication and/or Perception)
- ✓ Estimating “*Drivers Intentions*” from Vehicles States Observations
- ✓ Inferring “*Behaviors Expectations*” from Drivers Intentions & Traffic rules
- ✓ Risk = Comparing Maneuvers *Intention & Expectation* (Dynamic Bayesian Network)
  - => Taking traffic context into account (Topology, Geometry, Priority rules, Vehicles states)
  - => Digital map obtained using “Open Street Map”



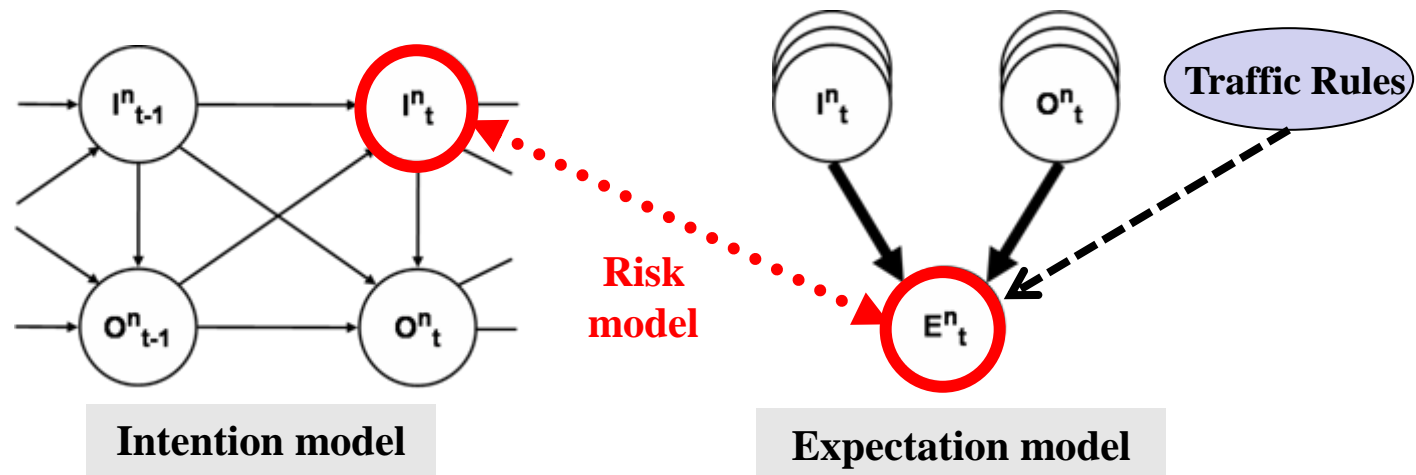
□ Human-like reasoning at a semantic level, based on a *Dynamic Bayesian Network*

□ **Main idea:** Detect dangerous situations by comparing « *what drivers intend to do* » with « *what drivers are expected to do* »

✓ **Intention:** Estimated from the successive states observations ( $X, Y, \theta, S, TS$ )

✓ **Expectation:** Estimated from Drivers Intentions & Traffic rules

✓ **Risk:** Based on  $P([I_t^n = 0][E_t^n = 1] | O_{0:t})$

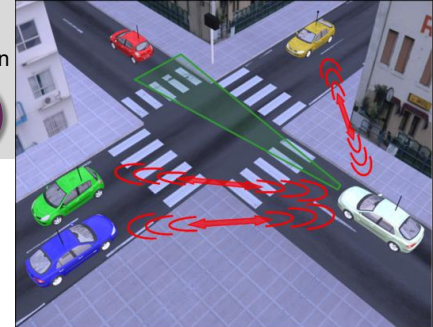




# Cooperative Roads Intersection Safety

5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, November 3rd, 2013, Tokyo, Japan

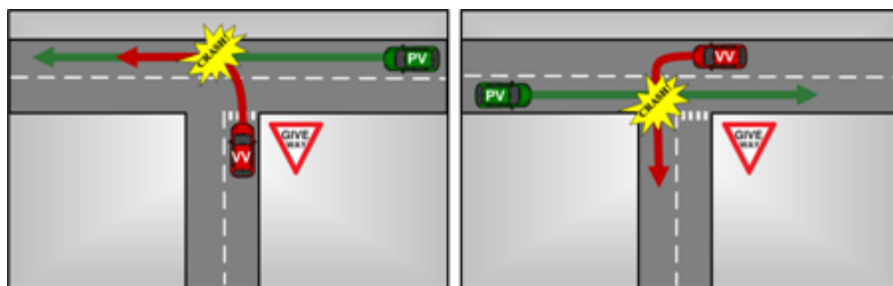
## Experimental Results (Field Trials)



### ❑ Two Renault passenger vehicles

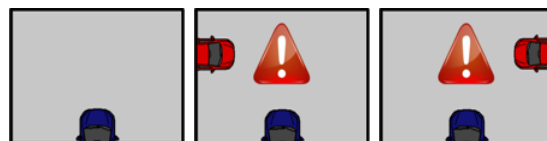
- ✓ Equipped with off-the-shelf V2V modems
- ✓ Sharing position, heading, speed, turn signal information at 10 Hz

### ❑ Collision scenarios (Blind rural intersection near Paris)



### ❑ Danger defined as $\exists n \in N : P([I_t^n = 0] | [E_t^n = 1] | O_{0:t}) > \lambda$

=> Audio-visual warnings



Video



### ❑ 90 instances, 9 Drivers (No accident)

- Thanks to recent advances in the field of **Robotics & ICT** technologies, **Intelligent Cars** are gradually becoming a reality



Parking Assistant (2004)



Volvo Pedestrian avoidance system (2011)



Fully Autonomous Driving (2020 -25 ?)

- Embedded Perception & Situation Awareness** are two key functionalities for improving Driving Safety. For addressing these issues, we have proposed and implemented four main technologies:

- ❖ *The “Embedded Bayesian Perception” approach for dealing with Open & Dynamic Environments*

- ❖ *3 complementary approaches for “Risk Assessment & Decision Making”*

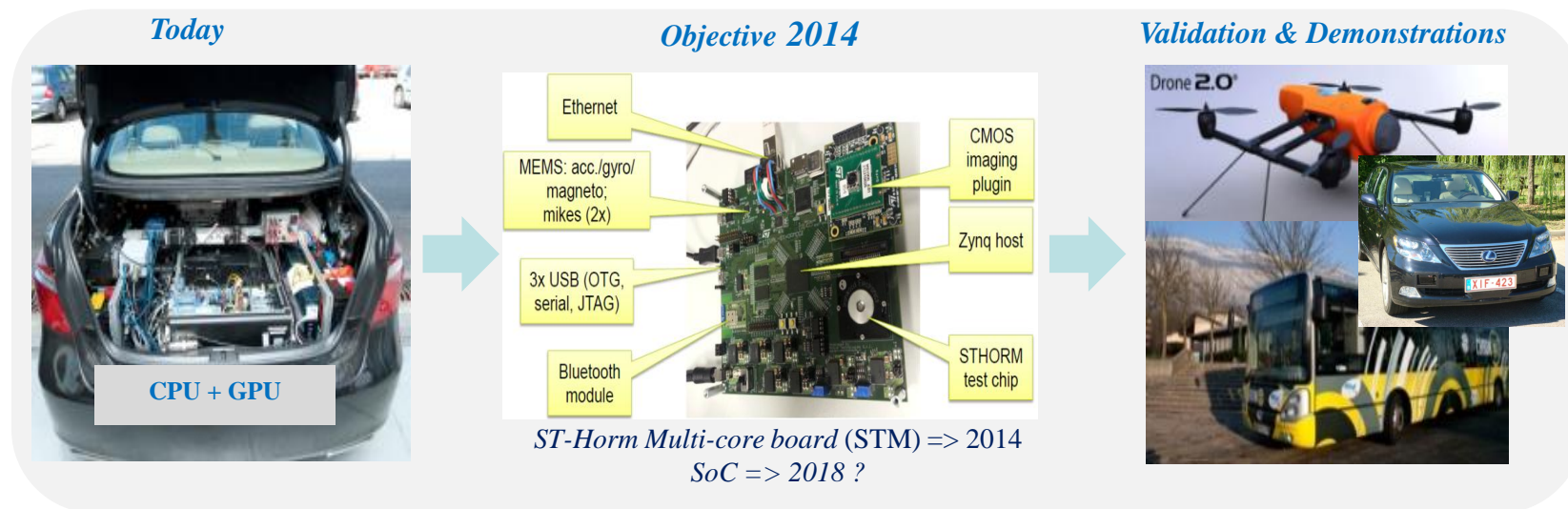
- *Learn & Predict paradigm*

- *Trajectories prediction + Probabilistic collision check*

- *Comparing Intention & Expectation for cooperative safety*

## □ Miniaturization & Increased efficiency (*Software & Hardware integration*)

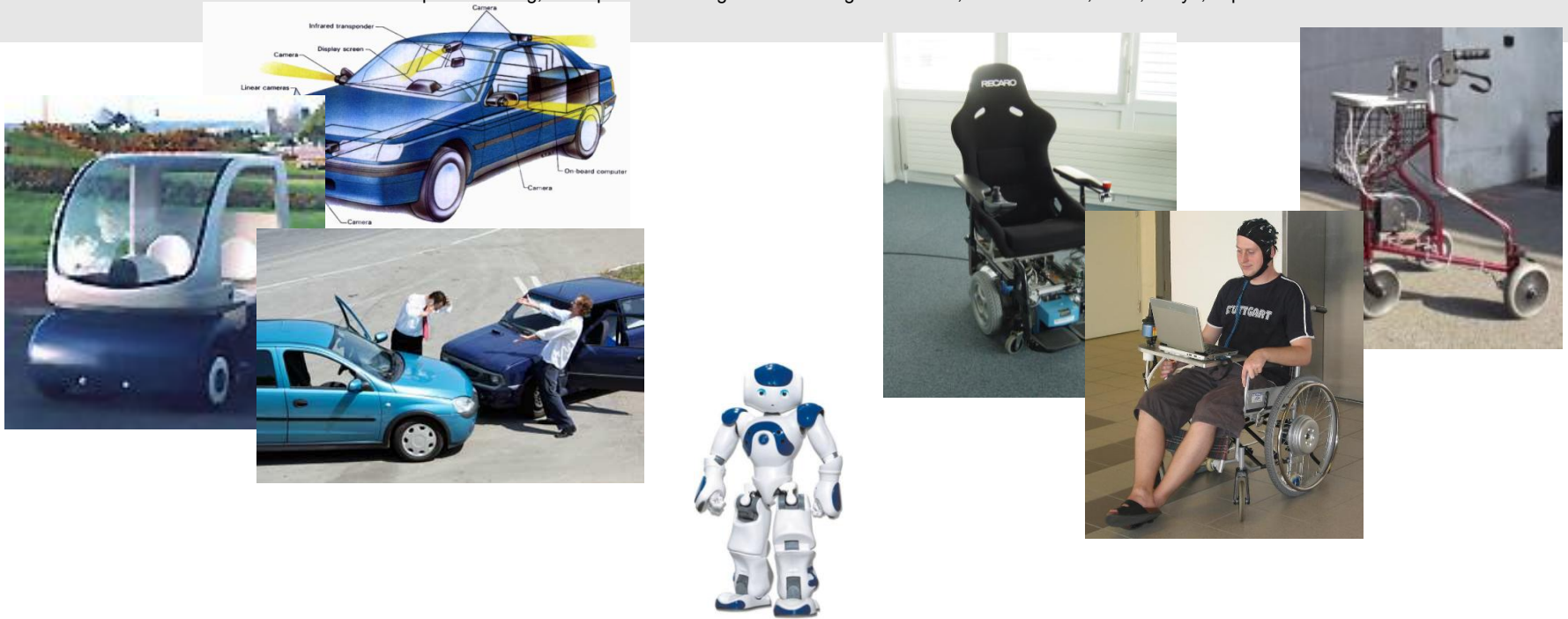
- ✓ Reduce drastically Size, Weight, Energy consumption, Cost ... while improving Efficiency
- ✓ Cooperation CEA (French Nuclear Energy Institute) & ST Microelectronics



## □ Embedded Perception & Decision

- ✓ Integrating Risk Assessment & Decision => Paper IROS 2013 + Patent Inria & Berkeley 2013
- ✓ Cooperation Berkeley & Renault





# Thank you for your attention Any questions ?

<http://emotion.inrialpes.fr/laugier>  
[christian.laugier@inrialpes.fr](mailto:christian.laugier@inrialpes.fr)

**IROS'13**

**PPNIV'13**



**5<sup>th</sup> Workshop on Planning, Perception and Navigation for Intelligent Vehicles**

**2013 IEEE/RSJ International Conference on Intelligent Robots and Systems**





## Session V

### Perception & Situation awareness

- **Title: Detection of Unusual Behaviours for Estimation of Context Awareness at Road Intersections**  
**Authors:** Alexandre Armand, David Filliat, Javier Ibanez-Guzman
- **Title: Enhancing Mobile Object Classification Using Geo-referenced Maps and Evidential Grids**  
**Authors:** Marek Kurdej, Julien Moras, Veronique Cherfaoui, Philippe Bonnifait

# Detection of Unusual Behaviours for Estimation of Context Awareness at Road Intersections

Alexandre Armand<sup>1,2</sup>, David Filliat<sup>1</sup>, Javier Ibanez-Guzman<sup>2</sup>

**Abstract**— In general, Advanced Driving Assistance Systems (ADAS) warn drivers once a high risk situation has been inferred. This is made under the assumption that all drivers react in the same manner. However, it is not the case as drivers react as a function of their own driving style. This paper proposes a framework which allows the estimation of the degree of awareness with regard to the focus object of the context that is governing the vehicle behaviour (e.g. the arrival to an intersection). The framework learns the manner in which individual drivers behave for a given context, and then detects whether or not the driver is behaving differently under similar conditions. In this paper the principles of the framework are applied to a fundamental use-case, the arrival to a stop intersection. Results from experiments under controlled conditions are included. They show that the formulation allows for a coherent estimation of the driver awareness while approaching to such intersections.

## I. INTRODUCTION

Statistics have shown that most road accidents are due to human errors, inferred by factors such as distraction, tiredness, over speeds, etc [16]. These result in bad situation understanding which often leads to abnormal and dangerous situations.

Road intersections represent complex environments where over 40% of collisions and 20% of fatalities occur [12]. Further, most of those involved in such collisions are young inexperienced drivers and the elderly. Given the complexity that exists at road intersections, namely the convergence of various entities to the same area, intersections represent a major challenge for ADAS.

An underlying framework for the estimation of unusual behaviours with regard to the road context and the driver individualities is presented. The tenet is that the vehicle evolves within a context which is built of contextual elements. These impose constraints to the subject vehicle, and usually one object has more influence than the rest. The framework takes into consideration this contextual object, and also the usual vehicle response (driver pattern) as it interacts with this object. This is then compared with the actual behaviour. If the driver actual behaviour differs much from the expected one, it is considered as unusual, which could be synonym of context misunderstanding and thus a source of risk. This concept is exploited in a simple scenario,

road intersections. The aim of the framework is not to warn the driver when the situation becomes dangerous, but to make sure that the driver has all necessary information for coherent decision making.

The remainder of the paper is organized as follows. Section II includes results of the state of the art review for risk estimation at road intersections, followed by the problem formulation. In Section III, the proposed model of the framework is described, and Section IV presents preliminary results from experiments applied to road intersections. Section V concludes the paper.

## II. RELATED WORK AND PROBLEM STATEMENT

### A. Related Work

Road intersection safety is of much concern. Some risk reduction has been achieved with the introduction of roundabouts instead of classic intersections. Another long term solution is to use communication technologies [6]. Currently, increasing driver awareness before arriving to the intersection remains a challenge.

One of the most intuitive approaches consists in using rules associated to the context. The set of rules, function of contextual inputs (e.g. the vehicle state, the maximum velocity, etc.) define the situations which can be considered as risk situations. In [6], rules are set to define when the velocity is not safe when a vehicle is reaching an intersection. The main problem of these approaches is the difficulty to take uncertainties into account. In addition, when such systems become complex, rules become interleaved and hence difficult to trace.

Several of the algorithms available in the literature are mainly based on the estimation of the so called Time To Collision (TTC) [9], [7]. This indication estimates the time remaining before a collision between two objects. Alerts are usually given as soon as the TTC becomes lower than a threshold. However no conclusion can be drawn before the situation gets critical.

Other approaches include the driver as part of the system to infer driver manoeuvres. Given a context, by observing the differences between the driver intention and the expected behaviour, risky situations can be detected. In [8], this risk is inferred within a Dynamic Bayesian Network (DBN) implemented in cooperative vehicles. In [3], it is proposed to decompose manoeuvres into a series of consecutive actions which are then represented as Hidden Markov Models (HMM). A framework based on Support Vector

<sup>1</sup> ENSTA ParisTech/ INRIA FLOWERS team, 828 Boulevard des Maréchaux, 91762 Palaiseau Cedex, France. alexandre.armand@ensta-paristech.fr, david.filliat@ensta-paristech.fr.

<sup>2</sup> Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. javier.ibanez-guzman@renault.com.

Machines (SVM) coupled with HMM to determine the driver's behaviour is presented in [1]. The system proposed in [13] also decomposes the manoeuvres into a sequence of elementary states, and a multilayer perceptron is used to learn the mapping between the current situation and the future vehicle states. All these systems generate warning only when the situation becomes dangerous. In addition, except driver actuations, no information about the driver is used to improve performances in term of reactivity.

In addition, several driver centric systems have been studied. Most of them are using either physiological sensors [5] or vision technologies [17] to look at the driver and get some vigilance information. Whilst progress has been achieved, reliability remains a problem. Moreover, using this kind of technologies requires more sensors in the vehicle which is not compatible with vehicle OEM constraints.

### B. Problem Statement

The literature has shown that most of risk assessment systems estimate the risk only when the situation becomes dangerous and thus when an accident is imminent. These systems can be called curative systems. Some studies have highlighted the negative effects that ADAS can sometimes have on drivers, for instance in term of emotions, and time to react [11], [10]. Surrounding vehicles may also be directly impacted by the consequence of an alert lately or not well interpreted by the driver. In addition, these studies have shown that early warnings improve the efficiency of the alerts. Thus, the main challenge of risk assessment systems remains the responsiveness of the system and the integrity of generated information, so that such systems can become preventive instead of remedial.

In addition, to our knowledge, there is no related work that takes advantage of drivers' individualities. Though, some highlight the differences of behaviour between different drivers in similar contexts [4], [2]. For example, it is unlikely that a driver used to decelerate smoothly decides intentionally to decelerate much harder than usual.

In this paper, it is proposed to take advantage of driver patterns with regard to road contexts, to detect unusual behaviours which might be signs of incomplete situation awareness. Multiple sources of data are used within the framework, with respect to the subject vehicle, the road context and the driver. An underlying architecture of the system is presented, followed by a concrete exploitation in a road intersection context. It is shown that the use of driver individualities may help detect unusual behaviours (i.e. indicators of situation unawareness) to provide early advices instead of late warnings.

## III. PROPOSED APPROACH

### A. Concept

The aim of the framework is not to generate alerts when the situation is getting dangerous, but rather to provide advices as a human copilot would do. For instance, a fellow

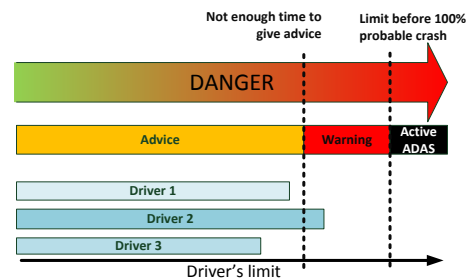


Fig. 1. Solutions to avoid accident: advice, warning and active ADAS. Drivers react more or less rapidly to the context, depending on their habits. For some of them, early advices can be given instead of warnings.

traveller who feels that the driver did not understand or perceive something would say “Have you seen ... ?” instead of waiting the last minute to say “Brake !”. A copilot usually knows the driver's practices, and can estimate the need to advice the driver in case of unusual behaviour.

Figure 1 illustrates the nuance between warnings and advices, such as:

- Active ADAS: the situation is critical, and the TTC is too small to let the driver react and brake. The vehicle takes the control.
- Warning: the situation is dangerous, however the driver has time to react and to avoid accidents. The system warns the driver.
- Advice: the situation is not yet dangerous, however it seems that the driver is not aware of a contextual object. The system gives a pertinent advice to make sure that the driver has all the required information for a coherent decision making.

Depending on the manner a driver is used to drive and to behave in particular contexts, advices can become relevant, or not. For instance, a sporty driver usually starts braking late at stop intersections. The situation becomes abnormal for him very late, and the situation can become dangerous very quickly. It is more relevant to warn the driver than giving him an advice. On the contrary, a relaxed driver who does not brake as early as usual can become suspect, and even if the situation is not yet dangerous an advice can be relevant for him.

### B. Framework

The framework uses inputs from different information sources, as illustrated in Figure 2:

- Environment & Context. The environment can be known through digital maps which store informations about the road network and infrastructure. On the other hand, dynamic objects which cannot be included in maps (vehicles, pedestrians, etc.) have to be perceived in real time by using sensors such as cameras or radars.
- Vehicle State. The position, speed and other parameters related to the subject vehicle are provided by localization devices (GNSS, etc.) and the vehicle CAN bus.
- Driver. Actuations of the driver can be directly provided by messages in the vehicle CAN bus. Driver patterns

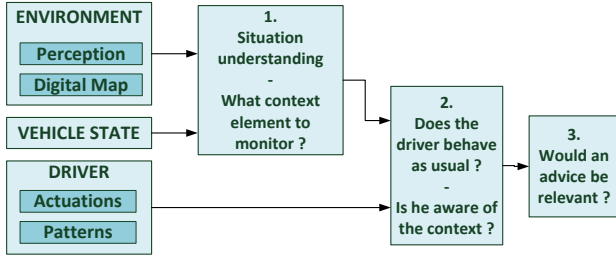


Fig. 2. Block Diagram of the proposed framework

(habits, in other words) have to be previously learnt. For example, [2] proposes a method to learn approaches to stop intersections.

The first step is to fuse environment data and vehicle state data to clearly understand the context, and the influence of every contextual object on the subject vehicle. Usually, one object has more influence than the others on the vehicle, and this influence leads to a particular vehicle parameter to monitor (speed, etc.). This step is not the object of this paper. In the rest of the paper, the most important contextual object is manually set. Some works propose methods for scene understanding, such as [15] and [18] which may be used within the framework.

The second step is to estimate if the behaviour of the driver matches with the behaviour expected in similar contexts. Driver patterns are compared with the current behaviour of the driver to estimate the awareness regarding the main contextual element. This step will be described in the rest of this paper.

### C. Detection of unusual behaviours and awareness estimation

This section aims at describing the box number 2 introduced in Figure 2 which allows detection of unusual behaviours and thus estimation of awareness. For this task, a Bayesian Network (BN) [14] has been developed. BNs offer a way to fuse different sources of information, taking uncertainties into consideration.

It is assumed that the context has been understood (box number 1) and that the parameter to monitor has been identified.

1) *Variables*: Variables are separated into two categories, depending on their observability:

a) *Observable variables*: These variables can be measured by the embedded sensors on a subject vehicle. They are defined as follows:

- $P_t \in \mathbb{R}$ , the parameter to be monitored, with regard to the contextual object considered by the box 1 (c.f. Figure 2). It may be the vehicle speed, interdistance, lateral position, etc.
- $R_t \in \{0, 1\}$ , the reaction of the driver. The driver can give an indication that he finally perceived/ took into consideration the most important contextual object. This variable is considered as a way to reduce the risk

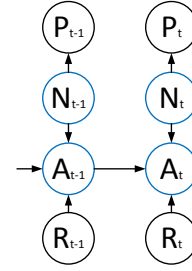


Fig. 3. Graphical Representation of the Bayesian Network

of non-relevant advice. It is related to the parameter to monitor, and may be an action on the brake pedal, or for instance an information provided by vision (c.f. [17]).

b) *Hidden variables*: These variables cannot be directly measured. However the DBN enables to estimate their values.

- $N_t \in \{0, 1\}$ , the estimation of the “Normality” of the driver’s behaviour. By “Normal behaviour”, it is understood a behaviour that matches with the behaviour that the driver usually has in a similar context.
- $A_t \in \{0, 1\}$ , the estimation of the awareness of the driver with regard to the contextual object taken into consideration by the box number 1 (c.f. Figure 2).

2) *Graphical Representation*: The structure of the proposed BN is shown in Figure 3; its corresponding joint distribution is given by Eq.(1).

$$P(P_t, N_t, A_t, R_t) = P(N_t) \times P(R_t) \times P(P_t|N_t) \times P(A_t|R_t, N_t, A_{t-1}) \quad (1)$$

The relationship between all the nodes has to be understood as follows:

- A behaviour considered as Normal means that the observed Parameter matches with the driver’s patterns, and that the driver seems Aware of the main contextual object.
- The Awareness of the driver (with regard to the main contextual object) is inferred by the estimation of the Normality of the driver’s behaviour, and also by a Reaction of the driver.

3) *Conditional probabilities*: A description of the parametric form of the conditional probabilities is presented in this section.

a) *The Parameter to monitor,  $P_t$* : It is considered that  $P_t$  follows the normal law such as:

$$P(P_t|N_t) = \mathcal{N}(p_{mean}, \sigma_s)$$

It means that whatever method can be used to provide the usual driver’s behaviour, the only constraint is that the provided value has to be composed by mean and variance.

Table I gives the value of  $P_t$  given the Normality of the behaviour  $N_t$ :

$N_t$	$P_t$
0	$\mathcal{N}(p_{Abnormal}, \sigma_{Abnormal})$
1	$\mathcal{N}(p_{Normal}, \sigma_{Normal})$

TABLE I

CONDITIONAL PROBABILITY OF THE PARAMETER  $P_t$ 

#	Cond.			Prob.
	$N_t$	$A_{t-1}$	$R_t$	$P(A_t = 1   N_t, A_{t-1}, R_t)$
1	0	0	0	$\alpha$
2	1	0	0	$\alpha$
3	0	1	0	$\alpha$
4	1	1	0	$\beta$
5	0	0	1	$\beta$
6	1	0	1	$\beta$
7	0	1	1	$\beta$
8	1	1	1	$\beta$

TABLE II

CONDITIONAL PROBABILITIES OF THE AWARENESS  $A_t$ 

b) *The Normality of the behaviour,  $N_t$* : The probability that abnormal behaviours occur is low, it is defined as follows:

$$P(N_t = 0) = \gamma$$

c) *The Reaction of the driver,  $R_t$* : The probability that the driver reacts because of the presence of a contextual object is set as follows:

$$P(R_t = 1) = 0.5$$

d) *The Awareness,  $A_t$* : It is assumed that there is continuity in the driver awareness. The conditional probabilities of the Awareness node are defined in Table II.

It is considered that the driver may be not aware of the main contextual object (i.e.  $\alpha$  is small) if:

- At time  $t - 1$ , the driver is not considered as aware of the context, and does not show any reaction at time  $t$ .
- Even if at time  $t - 1$  the driver was considered as aware of the context, if the behaviour turns abnormal and no reaction is perceived.

On the contrary, it is considered that the driver seems aware of the main contextual object (i.e.  $\beta$  is high) if:

- A reaction is perceived at time  $t$ .
- At time  $t - 1$  it was estimated that the driver was aware of the context, and his behaviour seems normal at time  $t$ .

4) *Behaviour Normality and Context awareness estimation*: The model is used to estimate the behaviour normality and the degree of awareness regarding the contextual object that is supposed to have the biggest influence on the subject vehicle. This awareness is estimated by Eq. 2:

$$P([N_t = 0], [A_t = 0] | P_t, R_t, A_{t-1}) \quad (2)$$

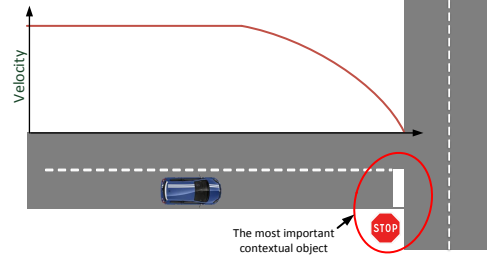


Fig. 4. The use case: a typical stop intersection vs expected velocity profile

## IV. PRELIMINARY EVALUATION AND DISCUSSION

### A. Use Case

A simple use case has been chosen to run a first evaluation of the framework proposed in the previous section. A subject vehicle is moving on a road that leads to a stop intersection. There is no lead vehicle moving in front of the subject vehicle, no pedestrian and no infrastructure such as speed bumpers or crossing. Thus the intersection is the only contextual object that has influence on the vehicle, and the only vehicle parameter to monitor is the velocity. Figure 4 illustrates the use case.

### B. Bayesian Network adaptation

The observable nodes of the DBN described in Section III-C have to be adapted to the given use case:

1) *The parameter  $P_t$* : This parameter to monitor is the vehicle velocity. This velocity depends on the distance to the stop intersection, as illustrated in Figure 4.

It is considered that a driver has an unusual behaviour when he does not decelerate before the intersection.

The usual behaviour of the driver while he is approaching to a stop intersection has to be learnt. In [2], it is proposed to learn the customized velocity profile of a driver at the approach to stop intersections. Gaussian Processes are used. It has been shown that this method is well adapted for this task, since it allows to model accurately the driver patterns taking into account uncertainties which might exist due to the driver and the quality of the on-board sensors. In addition, the outputs follow the normal law and are composed by mean and variance.

From a dozen of approaches recorded on real roads, the framework described in [2] allows to provide learnt patterns as the one shown in Figure 5. At any position before the intersection, Gaussian Processes allow to compute the velocity (mean and variance) at which the vehicle is usually moving at the same position, in similar contexts.

2) *The Reaction  $R_t$* : Usually, when approaching to a road intersection, a driver decelerates or brakes. In the case of stop intersections, a sign that the driver understood the presence of the contextual object is that he pushes (more or less) the brake pedal. For the proposed use case, the framework uses the brake pedal state (0 or 1) as a reaction of the driver.



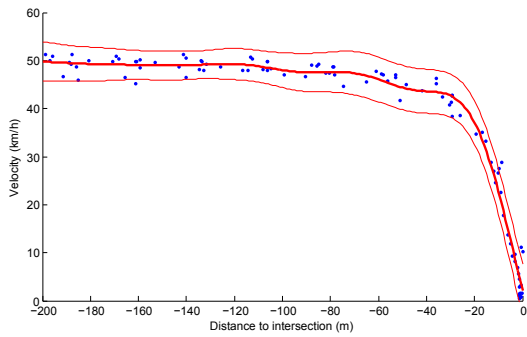


Fig. 5. Customized velocity profile provided by Gaussian Processes

3) *Parameters  $\alpha$ ,  $\beta$  and  $\gamma$*  : The value of these Bayesian Network parameters are set manually such as  $\alpha = \gamma = 0.1$  and  $\beta = 0.9$ .

### C. Results

Preliminary evaluations of the framework have been realized, using real data recorded on open roads. Acquisitions were accomplished with the same protocol as the one described in [2].

Three scenarios have been chosen for the evaluation:

- 1) Scenario 1: Normal behaviour.
- 2) Scenario 2: Unusual late deceleration.
- 3) Scenario 3: No reaction, comparison with the use of a generic profile.

1) *Scenario 1*: In this scenario, the driver behaves as he usually behaves while approaching to a stop intersection. The Figure 6 illustrates the behaviour of the DNB for this normal behaviour (red curves). It is noticeable that:

- The velocity stays inside the individual envelope defined by the customized driver pattern, and thus seems to be adapted to the context.
- Since the velocity of the vehicle matches with the driver pattern, the action of the driver on the brake pedal does not have influence on the model.
- The model considers that the driver is aware of the stop intersection.

2) *Scenario 2*: In this scenario, the driver does not approach to the intersection with an usual behaviour, and reacts lately. The Figure 6 illustrates the behaviour of the DNB for this abnormal behaviour (green curves). It is noticeable that:

- The velocity leaves the envelope defined by the personalized speed profile.
- As soon as the behaviour (i.e. the velocity) turns unusual, the risk that the driver did not consider the stop intersection starts increasing.
- When the driver starts pushing the brake pedal, the system considers that he took the stop intersection into consideration, and thus that he is aware of this contextual object. The risk decreases close to 0.

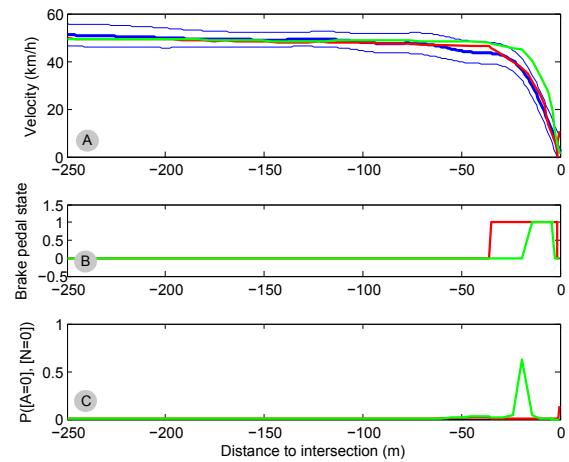


Fig. 6. Scenarios 1&2 : Normal behaviour and late reaction. In red, data related to normal behaviour, in green data related to late reaction. In windows A, the learnt pattern is in blue (mean and 95% confidence).

3) *Scenario 3*: In this scenario, it is simulated that the driver does not decelerate and does not react at all while approaching to a stop intersection. Figure 7 illustrates the behaviour of the system for this scenario. The behaviour of the DNB is tested using a profile customized for a rather relaxed driver (in blue), and with an average generic profile (in green). The generic profile shows a  $2.4m/s^2$  deceleration rate which is an average rate at  $50km/h$ , as indicated in [19]. In addition, a  $-9m/s^2$  deceleration curve is drawn. This curve represents an average maximum deceleration rate for emergency braking. It is noticeable that:

- As soon as the velocity leaves the envelopes (learnt and generic envelopes) without any reaction (on the brake pedal), the probability that the driver did not take the stop intersection into account increases up to 0.9.
- This example highlights the advantage of using customized patterns. For a relaxed driver (in blue), the system detects a risk of context unawareness about  $19m$  before the estimation of a risk with the generic profile. Moving at  $50km/h$ ,  $19m$  are travelled in  $1.35s$  which may represent a high average reaction time for a driver.
- With the learnt pattern, the estimated probability that the driver is not aware of the stop intersection reaches a value of 0.9  $34m$  before the maximum emergency deceleration curve. Moving at  $50km/h$ , this distance is travelled in  $2.42s$ . This is more than enough for the driver to react to an advice (for example: “Have you seen the stop intersection ?”) and to brake much smoother than an emergency braking.

### D. Discussion

According to the preliminary evaluation presented, the proposed framework provides a coherent estimation of the risk that a driver does not take into account the main contextual object. However, a quantitative evaluation of the system have to be done with a significant amount of data recorded in real conditions.

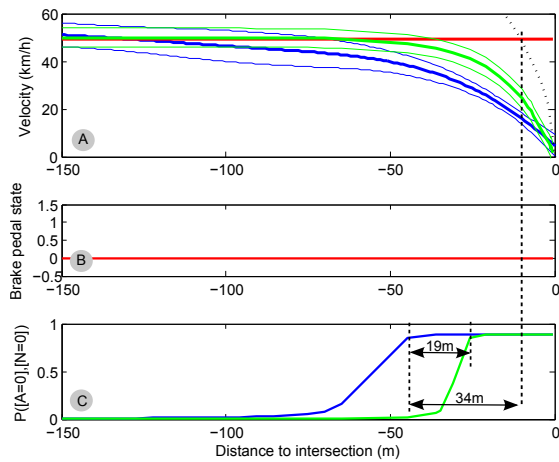


Fig. 7. Scenario 3. 1 run (in red) tested with 1 customized profile (in blue) and with a generic profile (in green). The black dot curve is the  $9\text{m/s}^2$  deceleration curve.

In addition, the use of customized driver patterns enhances the integrity of the generated information. Whilst uncertainties are taken into account by the personalized driver model, small uncertainties on measurements will lead to better estimation of risk.

For drivers used to drive sportingly, the framework allows also an estimation of unawareness. Nevertheless, this information can come too late to have time to generate an advice. In this case, it is better to generate warnings instead of advices. Further works have to be done to estimate when time is no longer sufficient to give an advice.

Finally, the simplicity of the use case enables to know in a straightforward manner that the stop intersection had to be monitored (c.f. Box 1 in Figure 2). In more complex contexts, it is not that simple. Thus, further works have to be done to automatically interpret the road context, and to detect contextual objects which interfere with the subject vehicle.

## V. CONCLUSION

An underlying framework for the estimation of driver awareness with regard to a particular contextual object has been presented. It takes into account that all drivers have different driver patterns, thus it learns how drivers behave under different contextual situations. Then, it infers if drivers are behaving differently as they approach similar situations. The model has been used within a simple use case (stop road intersection) to evaluate its relevance using a single observed variable, the vehicle velocity profile as it approaches the stop line. For this task, real data and customized driver patterns have been used. This preliminary evaluation has shown that the model provides a coherent estimation of context awareness (with regard to the focus object), and would make it possible to produce early advices to the driver.

The proposed framework will be extended by including other contextual objects, namely the presence of a lead

vehicle between the subject vehicle and the next road intersection. This use case requires the use of other observation variables.

## REFERENCES

- [1] G.S. Aoude, V.R. Desaraju, L.H. Stephens, and J.P. How. Driver behavior classification at intersections and validation on large naturalistic data set. *in Proceedings of the IEEE Intelligent Transportation Systems Conference*, 13(2):724–736, june 2012.
- [2] A. Armand, D. Filliat, and J. Ibanez-Guzman. Modelling stop intersection approaches using gaussian processes. *Accepted for IEEE Intelligent Transport Systems Conference 2013*, 2013.
- [3] H. Berndt, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. *in Proceedings of the IEEE Intelligent Transportation Systems Conf.*, pages 1189–1194, oct. 2008.
- [4] H. Berndt, S. Wender, and K. Dietmayer. Driver braking behavior during intersection approaches and implications for warning strategies for driver assistant systems. *in Intelligent Vehicles Symposium, 2007 IEEE*, pages 245–251, 2007.
- [5] K. Hayashi, Y. Kojima, K. Abe, and K. Oguri. Prediction of stopping maneuver considering driver's state. *in Proceedings of the IEEE Intelligent Transportation Systems Conf.*, pages 1191–1196, sept. 2006.
- [6] J. Ibanez-Guzman, S. Lefevre, A. Mokkadem, and S. Rodhaim. Vehicle to vehicle communications applied to road intersection safety, field results. *in Proceedings of Intelligent Transportation Systems (ITSC)*, pages 192–197, 2010.
- [7] R. Labayrade, C. Royere, and D. Aubert. A collision mitigation system using laser scanner and stereovision fusion and its assessment. pages 441–446, 2005.
- [8] S. Lefevre, C. Laugier, and J. Ibanez-Guzman. Risk assessment at road intersections: Comparing intention and expectation. *in Proceedings of the IEEE Intelligent Vehicles Symp.*, pages 165–171, june 2012.
- [9] M. Liebner, M. Baumann, F. Klanner, and C. Stiller. Driver intent inference at urban intersections using the intelligent driver model. *in Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1162–1167, june 2012.
- [10] A. Lindgren and F. Chen. State of the art analysis: An overview of advanced driver assistance systems (adas) and possible human factors issues. *Human Factors and Economic Aspects on Safety*, page 38, 2006.
- [11] C. Maag, D. Muhlbacher, C. Mark, and H-P Kruger. Studying effects of advanced driver assistance systems (adas) on individual and group level using multi-driver simulation. *Intelligent Transportation Systems Magazine, IEEE*, 4(3):45–54, 2012.
- [12] European Road Safety Observatory. Traffic safety basic facts 2010 junctions. *SafetyNet, Project co-financed by the European Commission*, 2010.
- [13] M.G. Ortiz, J. Fritsch, F. Kummert, and A. Gepperth. Behavior prediction at multiple time-scales in inner-city scenarios. *in Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1068–1073, june 2011.
- [14] J. Pearl. Bayesian networks: a model of self-activated: memory for evidential reasoning. 1985.
- [15] M. Platho, H-M Groß, and J. Eggert. Traffic situation assessment by recognizing interrelated road users. *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1339–1344, 2012.
- [16] TRACE. Analyzing human factors in road accidents. Technical report, 2008.
- [17] C. Tran and M.M. Trivedi. Vision for driver assistance: Looking at people in a vehicle. *Visual Analysis of Humans*, pages 597–614, 2011.
- [18] S. Vacek, T. Gindele, JM Zollner, and R. Dillmann. Situation classification for cognitive automobiles using case-based reasoning. *Intelligent Vehicles Symposium, 2007 IEEE*, pages 704–709, 2007.
- [19] J. Wang, K. Dixon, H. Li, and J. Ogle. Normal deceleration behaviour of passenger vehicles starting from rest at all way stop controlled intersections. *Transportation Research Record*, pages 158–166, 2005.

# Enhancing Mobile Object Classification Using Geo-referenced Maps and Evidential Grids

Marek Kurdej,

Julien Moras,

Véronique Cherfaoui,

Philippe Bonnifait

**Abstract**—Evidential grids have recently shown interesting properties for mobile object perception. Evidential grids are a generalisation of Bayesian occupancy grids using Dempster–Shafer theory. In particular, these grids can handle efficiently partial information. The novelty of this article is to propose a perception scheme enhanced by geo-referenced maps used as an additional source of information, which is fused with a sensor grid. The paper presents the key stages of such a data fusion process. An adaptation of conjunctive combination rule is presented to refine the analysis of the conflicting information. The method uses temporal accumulation to make the distinction between stationary and mobile objects, and applies contextual discounting for modelling information obsolescence. As a result, the method is able to better characterise the occupied cells by differentiating, for instance, moving objects, parked cars, urban infrastructure and buildings. Experiments carried out on real-world data illustrate the benefits of such an approach.

**Index Terms**—dynamic fusion, geo-referenced maps, mobile perception, prior knowledge, evidential occupancy grid, autonomous vehicle

## I. INTRODUCTION

Autonomous driving has been an important challenge in recent years. Navigation and precise localisation aside, environment perception is an important on-board system of a self-driven vehicle. The level of difficulty in autonomous driving increases in urban environments, where a good scene understanding makes the perception subsystem crucial. There are several reasons that make cities a demanding environment. Poor satellite visibility deteriorates the precision of GPS positioning. Vehicle trajectories are hard to predict due to high variation in speed and direction. Also, the sheer number of mobile objects poses a problem, e.g. for tracking algorithms.

On the other hand, more and more detailed and precise geographic databases become available. This source of information has not been well examined yet, hence our approach of incorporating prior knowledge from digital maps in order to improve perception scheme. A substantial amount of research has focused on the mapping problem for autonomous vehicles, e.g. Simultaneous Localisation and Mapping (SLAM) approach [1], but the use of maps for perception is still understudied.

In this article, we propose a new perception scheme for intelligent vehicles. The information fusion method is based on Dempster–Shafer theory of evidence [2]. The principal innovation of the method is the use of meta-knowledge obtained from a digital map. The map is considered as

an additional source of information on a par with other sources, e.g. sensors. We show the advantage of including prior knowledge into an embedded perception system of an autonomous car. To model the vehicle environment, our approach uses multiple 2D evidential occupancy grids described in [3]. Originally, occupancy grids containing probabilistic information were proposed in [4].

Our method aims to model complex vehicle environment, so that it can be used as a robust world representation for other systems, such as navigation. We want to detect mobile and static objects and distinguish stopped and moving objects. The objective of the proposed scheme is to model the free and navigable space as well.

This paper describes a robust and unified approach to a variety of problems in spatial representation using the Dempster–Shafer theory of evidence. The theory of evidence was not combined with occupancy grids until recently to build environment maps for robot perception [3]. Only recent works take advantage of the theory of evidence in the context of mobile perception [5]. There is also some research on efficient probabilistic and 3-dimensional occupancy grids [6]. Some authors have also used a laser range scanner as an exteroceptive source of information [5]. Some works use 3D city model as a source of prior knowledge for localisation and vision-based perception [7], whereas our method uses maps for scene understanding. Geodata are also successfully used for mobile navigation [8].

This article is organised as follows. Section II gives necessary theoretical background of the Dempster–Shafer theory of evidence. In section III, we describe the details of the proposed method, starting with the description of needed data and the purpose of each grid. Further, details on the information fusion are given. Data-dependent computation which are not in the heart of the method are described in section IV. Section V presents the results obtained with real-world data. Finally, section VI concludes the paper and presents ideas for future work.

## II. DEMPSTER–SHAFFER THEORY OF EVIDENCE

The Dempster–Shafer theory (DST) is a mathematical theory specially adapted to model the uncertainty and the lack of information introduced by Dempster and further developed by Shafer [2]. DST generalises the theory of probability, the theory of possibilities and the theory of fuzzy sets. In the Dempster–Shafer theory (DST), a set  $\Omega = \omega_1, \dots, \omega_n$  of mutually exclusive propositions is called the frame of discernment (FOD). In case of closed-world hypothesis, the FOD presents also an exhaustive set. Main difference in

\* Authors are with UMR CNRS 7253 Heudiasyc University of Technology of Compiègne, France. E-mail: firstname.surname@hds.utc.fr

comparison to the theory of probability is the fact that the mass of evidence is attributed not only to single hypotheses (singletons), but to any subset of the FOD, including an empty set.

As stated in the previous paragraph, beliefs about some piece of evidence are modelled by the attribution of mass to the corresponding set. This attribution of mass, called a basic belief assignment (bba), or a mass function, is defined as a mapping:

$$m(\cdot) : 2^\Omega \mapsto [0, 1] \quad (1)$$

$$\sum_{A \subseteq \Omega} m(A) = 1 \quad (2)$$

$$m(\emptyset) = 0 \quad (3)$$

In order to combine various information sources in the DST, there are many rules of combination. Combined mass functions have to be defined on the same FOD  $\Omega$  or transform to a common frame using refining functions. A *refining* is defined as a one-to-many mapping from  $\Omega_1$  to  $\Omega_2$ .

$$r : 2^{\Omega_1} \mapsto 2^{\Omega_2} \setminus \emptyset \quad (4)$$

$$r(\omega) \neq \emptyset \quad \forall \omega \in \Omega_1 \quad (5)$$

$$\bigcup_{\omega \in \Omega_1} r(\omega) = \Omega_2 \quad (6)$$

$$r(A) = \bigcup_{\omega \in A} r(\omega) \quad (7)$$

The frame of discernment  $\Omega_2$  is then called the *refinement* of  $\Omega_1$ , and  $\Omega_1$  is the *coarsening* of the  $\Omega_2$ .

When combined pieces of evidence expressed by bbas are independent and both are reliable, then the conjunctive rule and Dempster's combination rule are commonly used. In the case when the sources are independent, but only one of them is judged reliable, a disjunctive rule is used.

In the following, let us suppose that  $m_1, m_2$  are bbas. Then, the conjunctive rule of combination denoted by  $\odot$  is defined as follows:

$$(m_1 \odot m_2)(A) = \sum_{A=B \cap C} m_1(B) \cdot m_2(C) \quad (8)$$

The combination using the conjunctive rule can generate the mass on the empty set  $m(\emptyset)$ . This mass can be interpreted as the conflict measure between the combined sources. Therefore, a normalised version of conjunctive rule, called Dempster's conjunctive rule and noted  $\oplus$  was defined:

$$(m_1 \oplus m_2)(A) = \frac{(m_1 \odot m_2)(A)}{1 - K} \quad (9)$$

$$(m_1 \oplus m_2)(\emptyset) = 0 \quad (10)$$

$$K = (m_1 \odot m_2)(\emptyset) \quad (11)$$

The disjunctive rule of combination, noted  $\oslash$  is defined as follows:

$$(m_1 \oslash m_2)(A) = \sum_{A=B \cup C} m_1(B) \cdot m_2(C) \quad (12)$$

	$\emptyset$	$a$	$b$	$\Omega = \{a, b\}$
$m_1$	0	0.2	0.6	0.2
$m_2$	0	0.7	0.1	0.2
$m_1 \odot m_2$	0.44	0.34	0.18	0.04
$m_1 \oplus m_2$	0	0.61	0.32	0.07
$m_1 \oslash m_2$	0	0.14	0.06	0.8
${}^\alpha m_1$	0	0.18	0.54	0.28
betP <sub>1</sub>	0	0.3	0.7	1

TABLE I

EXAMPLE OF FUSION RULES, DISCOUNTING WITH  $\alpha = 0.1$  AND PIGNISTIC PROBABILITY.

In the DST, a discounting operation is used in order to, e.g. model information ageing. Discounting in its basic form requires to set a discounting factor  $\alpha$  and is defined as:

$${}^\alpha m(A) = (1 - \alpha) \cdot m(A) \quad \forall A \subsetneq \Omega \quad (13)$$

$${}^\alpha m(\Omega) = (1 - \alpha) \cdot m(\Omega) + \alpha \quad (14)$$

Decision making in DST creates sometimes a necessity of transforming a mass function into a probability function [9]. Smets and Kennes proposed so called *pignistic transformation* in [10]. Pignistic probability betP has been defined as:

$$\text{betP}(B) = \sum_{A \in \Omega} m(A) \cdot \frac{|B \cap A|}{|A|} \quad (15)$$

where  $|A|$  is the cardinality of the set  $A$ .

Table I presents an example of different combination rules, pignistic transform and discounting operation.

### III. MULTI-GRID FUSION APPROACH

This section presents the proposed perception schemes. We use three evidential occupancy grids to model prior information, sensor acquisition and perception result. The grid construction method is described in section III-B. We detail all data processing steps in section III-D. Figure 1 presents a general overview of our approach. Following sections correspond to different blocks of this diagram.

#### A. Heterogeneous data sources

There are three sources in our perception system: vehicle pose, exteroceptive acquisition data and vector maps. Figure 1 illustrates all system inputs. The proposed approach is based on the hypothesis that all these information sources are available. Other hypotheses on the input data are done. Firstly, a globally referenced vehicle pose is needed to situate the system in the environment. The pose provided by a proprioceptive sensor should be reliable, integrate and as precise as possible. It is assumed that the pose reflects closely the real state of the vehicle. Secondly, an exteroceptive sensor supplies a partial view of the environment. This sensor should be able to at least distinguish free and occupied space, and model it in 2D  $x, y$  or 3D  $x, y, z$  coordinates. The coordinates can be globally referenced or relative to the vehicle. A typical exteroceptive sensor capable of satisfying this assumption is a Lidar (laser range scanner), radar, or a

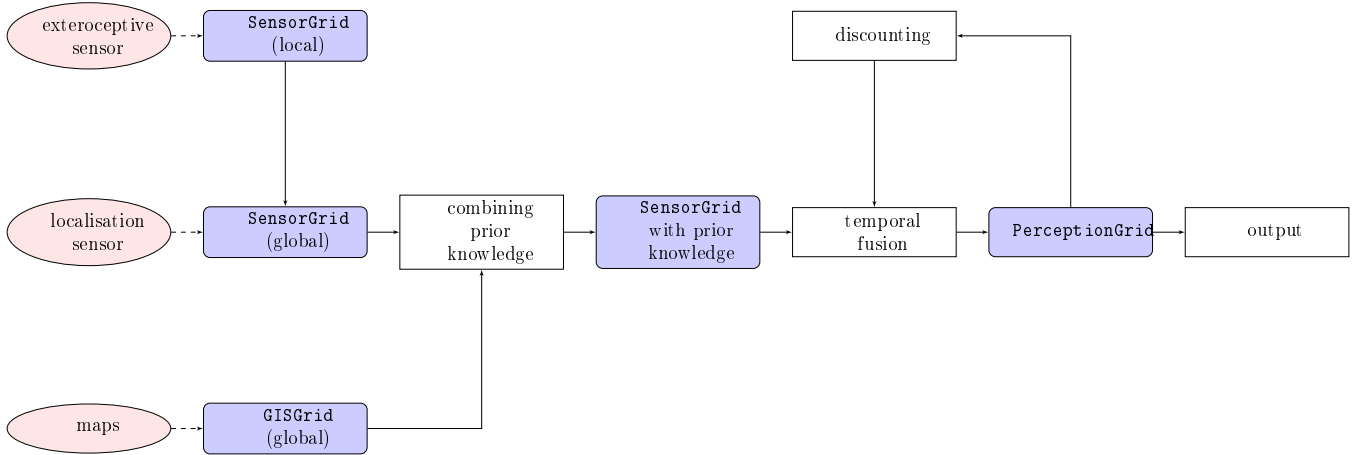


Fig. 1. Method overview.

stereo camera system. Lastly, our method tries to exploit at large the information contained in vector maps, so we assume that the maps are sufficiently rich and contain valuable accurate data. Typically, map data should contain information on the location of buildings and the model of road surface.

### B. Occupancy grids

An occupancy grid models the world using a tessellated representation of spatial information. In general, it is a multi-dimensional spatial lattice with cells storing some stochastic information. In our case, each cell representing a box (a part of environment)  $X \times Y$  where  $X = [x_-, x_+]$ ,  $Y = [y_-, y_+]$  stores a mass function.

1) *SensorGrid (SG)*: In order to process the exteroceptive sensor data, an evidential occupancy grid is computed when a new acquisition arrives, this grid is called *SensorGrid*. Each cell of this grid stores a mass function on the FOD  $\Omega_{SG} = \{F, O\}$ , where  $F$  refers to the free space and  $O$  – to the occupied space. The basic belief assignment reflects the sensor model.

2) *PerceptionGrid (PG)*: To store the results of information fusion, an occupancy grid PG has been introduced with a FOD  $\Omega_{PG} = \{F, I, U, S, M\}$ . The choice of such a FOD is directly coupled with the objectives that we try to achieve. Respective classes represent: free space  $F$ , mapped infrastructure (buildings)  $I$ , unmapped infrastructure  $U$ , temporarily stopped objects  $S$  and mobile moving  $M$  objects.  $\Omega_{PG}$  is a common frame used for information fusion. By using PG as a cumulative information storage, we are not obliged to store preceding *SensorGrids*.

3) *GISGrid (GG)*: This grid allows us to perform a contextual information fusion incorporating some meta-knowledge about the environment. *GISGrid* uses the same frame of discernment  $\Omega_{PG}$  as *PerceptionGrid*. The grid can be obtained, for instance, by projection of map data, buildings and roads, onto a 2D grid with global coordinates. However, the exact method of creating the GG depends on available GIS information. Section IV-B presents how the GG was constructed.

### C. Combining prior knowledge

In our method, prior information contained in maps serves to ameliorate the perception scheme. We have chosen to combine the prior knowledge with the sensor data of the *SensorGrid*. However, the Dempster–Shafer theory does not allow to combine sources with different frames of discernment. The frame of discernment  $\Omega_{SG}$  is distinct from  $\Omega_{PG}$  used in *GISGrid*. Hence, we are obliged to find a common frame for both sources. In order to enable the fusion of *SensorGrid* (SG) and *GISGrid* (GG), we define a refining:

$$r_{SG} : 2^{\Omega_{SG}} \mapsto 2^{\Omega_{PG}} \quad (16)$$

$$r_{SG}(\{F\}) = \{F\} \quad (17)$$

$$r_{SG}(\{O\}) = \{I, U, S, M\} \quad (18)$$

$$r_{SG}(A) = \bigcup_{\theta \in A} r_{SG}(\theta) \quad (19)$$

Refining  $r$  allows us to combine prior knowledge included in *GISGrid* with instantaneous grid obtained from sensor(s).

The refined mass function can be expressed as:

$$m_{SG}^{\Omega_{PG}}(r_{SG}(A)) = m_{SG}^{\Omega_{SG}}(A) \quad \forall A \subseteq \Omega_{SG} \quad (20)$$

Then, Dempster’s rule described in section II is applied in order to exploit the prior information included in GG:

$$m'_{SG,t}{}^{\Omega_{PG}} = m_{SG,t}^{\Omega_{PG}} \oplus m_{GG}^{\Omega_{PG}} \quad (21)$$

We have chosen to use the Dempster’s rule of combination, since the GIS data and the sensor data are independent. Besides, we suppose that both sources are reliable, even if errors are possible. In the end of this stage, we obtain a grid being combination of the sensor data, *SensorGrid*, with the prior knowledge from *GISGrid*.

### D. Temporal fusion

The role of the fusion operation is to combine current sensor acquisition with preceding perception result. The



sensor acquisition input is already combined with prior information as described in preceding paragraphs. We propose to exploit dynamic characteristics of the scene by analysing produced conflict masses. As the preceding perception result `PerceptionGrid` is partially out-of-date at the moment of fusion, the contextual discounting operation is employed to model this phenomena. Moreover, a counter of occupancy has been introduced and a mass function specialisation is performed to distinguish mobile, but temporarily stopped objects.

1) *Computing conflict masses*: To distinguish between two types of conflict which arise from the fact that the environment is dynamic, the idea from [11] is used.  $\emptyset_{FO}$  denotes the conflict induced when a free cell in PG is fused with an occupied cell in SG. Similarly,  $\emptyset_{OF}$  indicates the conflicted mass caused by an occupied cell in PG fused with a free cell in SG.

Conflict masses are calculated using the formulas:

$$m_{PG,t}(\emptyset_{OF}) = m_{PG,t-1}(O) \cdot m_{SG,t}(F) \quad (22)$$

$$m_{PG,t}(\emptyset_{FO}) = m_{PG,t-1}(F) \cdot m_{SG,t}(O) \quad (23)$$

where  $m(O) = \sum_A m(A)$ ,  $\forall A \subseteq \{I, U, S, M\}$ . In an error-free case, these conflicts represent, respectively, the disappearance and the appearance of an object.

2) *PerceptionGrid specialisation using an accumulator*: Mobile object detection is an important issue in dynamic environments. We propose the introduction of an accumulator  $\zeta$  in each cell in order to include temporal information on the cell occupancy. For this purpose, incrementation and decrementation steps  $\delta_{inc} \in [0, 1]$ ,  $\delta_{dec} \in [0, 1]$ , as well as threshold values  $\gamma_O$ ,  $\gamma_\emptyset$  have been defined.

$$\zeta^{(t)} = \min\left(1, \zeta^{(t-1)} + \delta_{inc}\right) \quad (24)$$

$$\text{if } m_{PG}(O) \geq \gamma_O \\ \text{and } m_{PG}(\emptyset_{FO}) + m_{PG}(\emptyset_{OF}) \leq \gamma_\emptyset$$

$$\zeta^{(t)} = \max\left(0, \zeta^{(t-1)} - \delta_{dec}\right) \quad (25)$$

$$\text{if } m_{PG}(\emptyset_{FO}) + m_{PG}(\emptyset_{OF}) > \gamma_\emptyset$$

$$\zeta^{(t)} = \zeta^{(t-1)} \quad (26)$$

$$\text{otherwise} \quad (27)$$

Using  $\zeta$  values, we impose a specialisation of mass functions in PG using the equation:

$$m'_{PG,t}(A) = S(A, B) \cdot m_{PG,t}(B) \quad (28)$$

where specialisation matrix  $S(\cdot, \cdot)$  is defined as:

$$\begin{aligned} S(A \setminus \{M\}, A) &= \zeta & \forall A \subseteq \Omega_{PG} \text{ and } \{M\} \in A \\ S(A, A) &= 1 - \zeta & \forall A \subseteq \Omega_{PG} \text{ and } \{M\} \in A \\ S(A, A) &= 1 & \forall A \subseteq \Omega_{PG} \text{ and } \{M\} \notin A \\ S(\cdot, \cdot) &= 0 & \text{otherwise} \end{aligned} \quad (29)$$

The idea behind the specialisation matrix and the accumulator is that the mass attributed to set  $N, S, M$  or  $S, M$  will be transferred to set  $N, S$  or  $S$ , respectively. The transferred mass value is proportional to the time that the cell stayed occupied. In this way, moving objects are differentiated from static or stopped objects.

3) *Fusion rule*: An important part of the method consists in performing the fusion operation of a discounted and specialized `PerceptionGrid` from preceding epoch  ${}^\alpha m'_{PG,t-1}$  with a SG combined with prior knowledge from current epoch  $m'_{SG,t}$ . The discounting operation is presented in section II and the specialisation is described in the preceding paragraph. In the section III-C, combination of prior knowledge with the `SensorGrid` is demonstrated.

$$m_{PG,t} = {}^\alpha m'_{PG,t-1} \otimes m'_{SG,t} \quad (30)$$

The fusion rule  $\otimes$  is a modified conjunctive rule adapted to mobile object detection. There are of course many different rules that could be used, but in order to distinguish between moving and stationary objects some modifications had to be performed. These modifications consist in transferring the mass corresponding to a newly appeared object  $\emptyset_{FO}$  to the class of moving objects  $M$  as described by the equation 31. Symbol  $\oplus$  denotes the conjunctive fusion rule.

$$(m_1 \otimes m_2)(A) = (m_1 \oplus m_2)(A)$$

$$\forall A \subsetneq \Omega \wedge A \neq M$$

$$(m_1 \otimes m_2)(M) = (m_1 \oplus m_2)(M) + (m_1 \oplus m_2)(\emptyset_{FO})$$

$$(m_1 \otimes m_2)(\Omega) = (m_1 \oplus m_2)(\Omega) + (m_1 \oplus m_2)(\emptyset_{OF})$$

$$(m_1 \otimes m_2)(\emptyset_{FO}) = 0$$

$$(m_1 \otimes m_2)(\emptyset_{OF}) = 0 \quad (31)$$

All the above steps allow the construction of a PG containing reach information on the environment state, including the knowledge on mobile and static objects.

#### E. Fusion rule behaviour

Proposed fusion scheme behaves differently depending on the context. In this section, we describe briefly the behaviour of the fusion rule. For an in-depth analysis, the reader is invited to read [12]. *Context* stands for prior knowledge information contained in `GISGrid`. To demonstrate the effect of the fusion operator, we have chosen two particular cases, which clearly represent different contexts.

*Building context*: In the building context, i.e. when  $m(F) + m(I) + m(\Omega) \approx 1$ , our fusion operator is roughly equivalent to the Yager's rule. The sum of conflict masses distinguished by the proposed rule is equal to the conflict mass in a regular fusion scheme without conflict management. This behaviour is relevant, since it is assumed that no mobile obstacles are present in this context. Therefore, only free space and infrastructure is to be distinguished.

*Road and intermediate space*: The conflict management adapted to the perception scheme direct mass attribution to moving obstacles (class  $M$ ). The introduction of occupied space counter and `PerceptionGrid` specialisation (see

section III-D.2) permits to transfer a part of the mass from “moving or other” class to “other”, where other is context-dependent.

#### IV. EXPERIMENTAL SETUP

##### A. Dataset

The data set used for experiments was acquired in the 12th district of Paris. The overall length of the trajectory was about 9 kilometres. The vehicle pose comes from a system based on a PolarX II GPS and a NovAtel SPAN-CPT inertial measurement unit (IMU). The system is supposed to provide precise positioning with high confidence. Our main source of information about the environment is an IBEO Alaska XT lidar able to provide a cloud of about 800 points 10 times per second. The digital maps that we use were provided by the French National Geographic Institute (IGN) and contain 3D building models as well as the road surface. We also performed successful tests with freely available *OpenStreetMap* project 2D maps [13], but here we limited the use to building data. We assume the maps to be accurate and up-to-date.

##### B. GISGrid construction

The map data can be represented by two sets of polygons defining the 2D position of buildings and road surface by, respectively,

$$\mathcal{B} = \left\{ b_i = \begin{bmatrix} x_1 x_2 \dots x_{m_i} \\ y_1 y_2 \dots y_{m_i} \end{bmatrix}, i \in [0, n_B] \right\} \quad (32)$$

$$\mathcal{R} = \left\{ r_i = \begin{bmatrix} x_1 x_2 \dots x_{m_i} \\ y_1 y_2 \dots y_{m_i} \end{bmatrix}, i \in [0, n_R] \right\} \quad (33)$$

Our dataset satisfies the condition:  $\mathcal{B} \cap \mathcal{R} = \emptyset$ .

We note that  $B = \{I\}$ ,  $R = \{F, S, M\}$ ,  $T = \{F, U, S, M\}$  for convenience and readability only. Set  $A$  denotes then all other strict subsets of  $\Omega$ . These aliases characterise the meta-information inferred from geographic maps. For instance, on the road surface  $R$ , we encourage the existence of free space  $F$  as well as stopped  $S$  and moving  $M$  objects. Analogically, building information  $B$  fosters mass transfer to  $I$ . Lastly,  $T$  denotes the intermediate area, e.g. pavements, where mobile and stationary objects as well as small urban infrastructure can be present. Please note that neither buildings nor roads are present, so the existence of mapped infrastructure  $I$  can be excluded, but the presence of the other classes cannot. Also, a level of confidence  $\beta$  is defined for each map source, possibly different for each context. Let  $\tilde{x} = \frac{x_- + x_+}{2}$ ,  $\tilde{y} = \frac{y_- + y_+}{2}$ , then:

$$m_{GG}\{X, Y\}(B) = \begin{cases} \beta_B & \text{if } (\tilde{x}, \tilde{y}) \in b_i \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

$$\forall i \in [0, n_B]$$

$$m_{GG}\{X, Y\}(R) = \begin{cases} \beta_R & \text{if } (\tilde{x}, \tilde{y}) \in r_i \\ 0 & \text{otherwise} \end{cases} \quad (35)$$

$$\forall i \in [0, n_R]$$

$$m_{GG}\{X, Y\}(T) = \begin{cases} 0 & \text{if } (\tilde{x}, \tilde{y}) \in b_i \vee (\tilde{x}, \tilde{y}) \in r_j \\ \beta_T & \text{otherwise} \end{cases} \quad (36)$$

$$\forall i \in [0, n_B], \forall j \in [0, n_R]$$

$$m_{GG}\{X, Y\}(\Omega) = \begin{cases} 1 - \beta_B & \text{if } (\tilde{x}, \tilde{y}) \in b_i \\ 1 - \beta_R & \text{if } (\tilde{x}, \tilde{y}) \in r_i \\ 1 - \beta_T & \text{otherwise} \end{cases} \quad (37)$$

$$\forall i \in [0, n_B], \forall j \in [0, n_R]$$

$$m_{GG}\{X, Y\}(A) = 0 \quad (38)$$

$$\forall A \subsetneq \Omega \text{ and } A \notin \{B, R, T\}$$

##### C. Sensor model

This section describes the way in which the data obtained from the sensor are transformed into the *SensorGrid*. If another exteroceptive sensor is used, one has to define an appropriate model. The model used in the presented method is based on the one described in [5].

##### D. Parameters

The size of the grid cell in the occupancy grids was set to 0.5 m, which is sufficient to model a complex environment with mobile objects. We have defined the map confidence factor  $\beta$  by ourselves, but ideally, it should be given by the map provider.  $\beta$  describes data currentness (age), errors introduced by geometry simplification and spatial discretisation.  $\beta$  can also be used to depict the localisation accuracy. Other parameters, such as counter steps  $\delta_{inc}$ ,  $\delta_{dec}$  and thresholds  $\gamma_O$ ,  $\gamma_\theta$  used for mobile object detection determine the sensitiveness of mobile object detection and were set by manual tuning. Parameters used for the construction of *SensorGrid*, were set to  $\mu_F = 0.7$ ,  $\mu_O = 0.8$ .

#### V. RESULTS

To assess the performance of our method, a comparison of perception results when prior knowledge from maps is present and when it is not available has been performed. In this way, we show the interest of using a map-aided approach to the perception problem.

The results for a particular instant of the approach tested on real-world data are presented on figure 2. The visualisation of the PG has been obtained by attributing to each class a colour proportional to the pignistic probability  $\text{betP}$  and calculating the mean colour [9]. The presented scene contains two moving cars (only one is visible in the camera image) going in the direction perpendicular to the test vehicle.

The principal advantage gained by using map knowledge is richer information on the detected objects. A clear difference between a moving object (red, car) and a stopped objects (blue) is visible. Also, stopped objects are distinct from infrastructure when prior map information is available (which is not highlighted on the figures). In addition, thanks to the prior knowledge, stationary objects such as infrastructure are distinguished from stopped objects on the road. Grids make noticeable the effect of discounting, as information on the environment behind the vehicle is being forgotten.

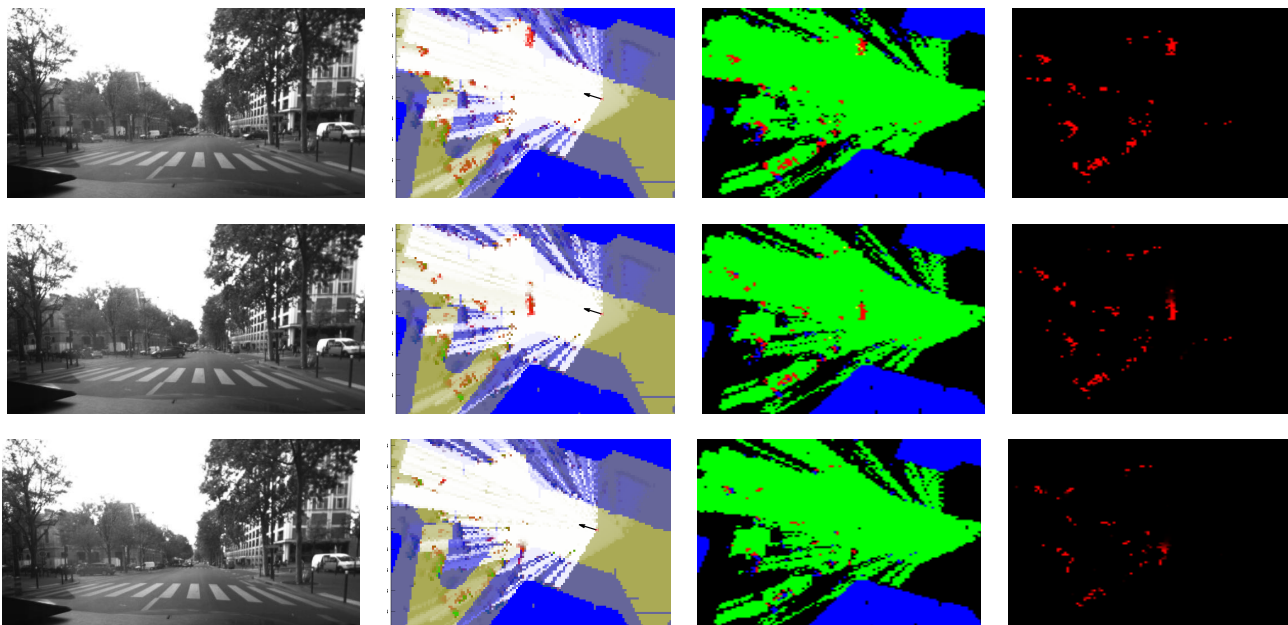


Fig. 2. From left to right: (1) scene capture, (2) PerceptionGrid pignistic probability, (3) simple decision rule to detect free space, moving and stopped obstacles, (4) trace of moving objects. Colour code for figures (3) and (4): green – free space, red – moving objects, blue – static objects (buildings, stopped objects), black – unknown space.

Figure 2 shows also the effect of the discounting which is particularly visible on the free space behind the vehicle. The grid cells get discounted, so the mass on the free class  $F$  diminishes gradually.

## VI. CONCLUSION AND PERSPECTIVES

A new mobile perception scheme based on prior map knowledge has been introduced. Geographic information is exploited to reduce the number of possible hypotheses delivered by an exteroceptive source. A modified fusion rule taking into account the existence of mobile objects has been defined. Furthermore, the variation in information lifetime has been modelled by the introduction of contextual discounting.

In the future, we anticipate removing the hypothesis that the map is accurate. This approach will entail considerable work on creating appropriate error models for the data source. Moreover, we envision differentiating the free space class into two complementary classes to distinguish navigable and non-navigable space. This will be a step towards the use of our approach in autonomous navigation. Another perspective is the use of reference data to validate the results, choose the most appropriate fusion rule and learn algorithm parameters. We envision using map information to predict object movements. It rests also a future work to exploit fully the 3D map information.

## ACKNOWLEDGEMENTS

This work was supported by the French Ministry of Defence DGA (Direction Générale de l'Armement), with a Ph.D. grant delivered to Marek Kurdej.

## REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, Massachusetts, USA: MIT Press, 2005.
- [2] G. R. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [3] D. Pagac, E. M. Nebot, and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 623–629, 1998.
- [4] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer Journal*, vol. 22, no. 6, pp. 46–57, Jun. 1989.
- [5] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist Occupancy Grids for Vehicle Perception in Dynamic Environments," in *IEEE International Conference on Robotics and Automation*, Shanghai, May 2011, pp. 84–89.
- [6] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems," in *IEEE International Conference on Robotics and Automation Workshop*, May 2010. [Online]. Available: <http://octomap.sourceforge.net/>
- [7] M. Dawood, C. Cappelle, M. E. E. Najjar, M. Khalil, and D. Pomorski, "Vehicle geo-localization based on IMM-UKF data fusion using a GPS receiver, a video camera and a 3D city model," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 510–515.
- [8] M. Hentschel and B. Wagner, "Autonomous Robot Navigation Based on OpenStreetMap Geodata," in *International IEEE Annual Conference on Intelligent Transportation Systems*, Madeira Island, Sep. 2010, pp. 1645–1650.
- [9] P. Smets, "Decision Making in the TBM: the Necessity of the Pignistic Transformation," *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 133–147, 2005.
- [10] —, *What is Dempster-Shafer's model?*, F. M. Yager R.R. and K. J., Eds. John Wiley & Sons, 1994.
- [11] J. Moras, V. Cherfaoui, and P. Bonnifait, "Moving Objects Detection by Conflict Analysis in Evidential Grids," *IEEE Intelligent Vehicles Symposium*, pp. 1120–1125, Jun. 2011.
- [12] M. Kurdej, J. Moras, V. Cherfaoui, and P. Bonnifait, "Controlling Remanence in Evidential Grids Using Geodata for Dynamic Scene Perception," *International Journal of Approximate Reasoning*, vol. (accepted), Mar. 2013.
- [13] "OpenStreetMap," 2013. [Online]. Available: <http://www.openstreetmap.org>