IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS14 6th International workshop on

Planning, Perception and Navigation for Intelligent Vehicles

Full Day Workshop

September 14th, 2014 Chicago, USA

http://ppniv14.irccyn.ec-nantes.fr/

Organizers

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 : +33 240 376 934, Fax : +33 240 376 6930
Email: Philippe.Martinet@irccyn.ec-nantes.fr
Home page: http://www.irccyn.ec-nantes.fr/~martinet

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 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, Human-Robot Interaction, Behavior modeling and learning, Robust sensor-based 3D reconstruction, Modeling and Control of mobile robot.

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, the fourth edition PPNIV'12 was in Vilamoura (over 95 attendees) during IROS'12, and the fifth edition PPNIV'13 was in Vilamoura (over 135 attendees) during IROS'13.

In parallel, we have also organized SNODE'07 in San Diego during IROS'07 (around 80 attendees), MEPPC08 in Nice during IROS'08 (more than 60 registered people), SNODE'09 in Kobe during ICRA'09 (around 70 attendees), RITS'10 in Anchrorage during ICRA'10 (around 35 attendees), PNAVHE11 in San Francisco during the last IROS11 (around 50 attendees), and the last one WMEPC14 in Hong Kong during the last ICRA14 (around 65 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 & Situation awareness
- Session III: Interactive session
- Session IV: Navigation, Control, Planning
- Panel Session: Towards driverless vehicles?

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

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/).

Christian Laugier, Philippe Martinet, Urbano Nunes and Christoph stiller

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session I

Localization & Mapping

- **Title: Automated driving in urban environments: car sharing distribution system and Parking Valet as canonical use-cases**
  Keynote speaker: Fawzy Nashashibi (INRIA, Rocquencourt, France)

- **Title: An Iterative Graph Optimization Approach for 2D SLAM**
  Authors: *He Zhang, Guoliang Liu, and Zifeng Hou*

- **Title: Appearance-based Localization across Seasons in a Metric Map**
  Authors: *Chris Beall, Frank Dellaert*

- **Title: High Precision 6DOF Vehicle Navigation in Urban Environments using a Low-cost Single-frequency GPS Receiver**
  Authors: *Sheng Zhao; Yiming Chen, Jay A. Farrell*

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session I

Keynote speaker: **Fawzi Nashashibi**
(INRIA, Rocquencourt, France)

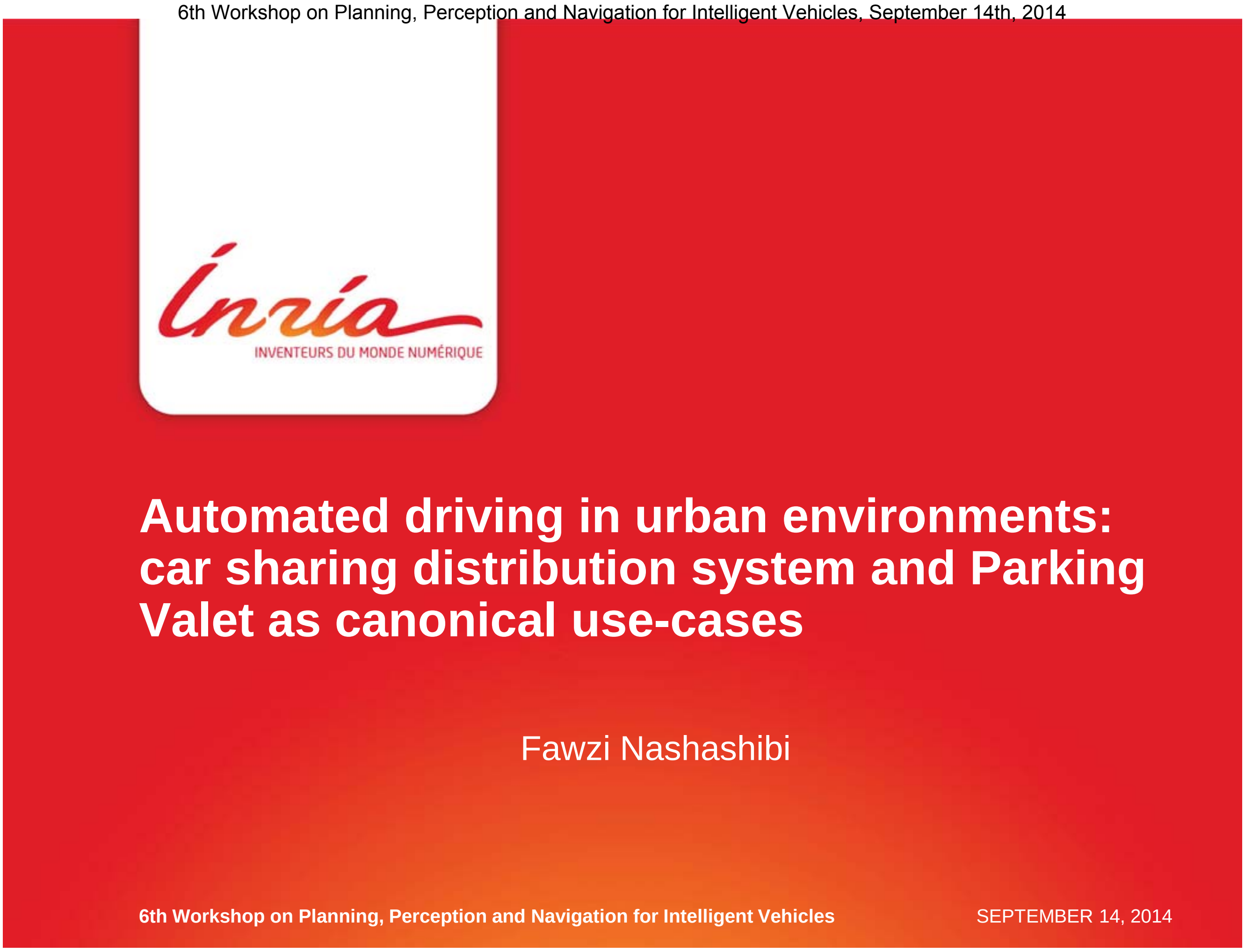
Automated driving in urban environments: car sharing distribution system and Parking Valet as canonical use-cases

Abstract : In September 2014, the French high authorities announced the creation of an ambitious plan to support the industrialization of France around 34 challenging industrial domains. Among these topics: “robotics” and “vehicles with automated driving”. This initiative is part of an international effort to develop automated driving before 2020. Automated driving in urban environments is particularly very challenging because of the technical challenges and legal limitations. In the meanwhile business models are limited due to these obstacles. In these conditions the automated distribution of a car-sharing system and a “Parking valet” system seem to be interesting use-cases because of the constrained environments in which vehicle navigation is performed while private areas as well as segregated lanes can offer a good solution to remove the legal barriers. From a pure technical point of view, “automated car-sharing distribution system” and “Parking valet” are interesting applications where different automated navigation functions and intelligent mobility concepts cohabit in order to provide a practical service to different end users. This talk will tackle the technical requirements to realize such services. Platooning, automated parking, accurate localization and environments mapping are among the automated functions to integrate. We will describe these advanced functions and their integration in this very specific framework. A first prototyping of such systems on automated vehicles will be presented as well as future developments and perspectives.

Biography: Dr. Fawzi Nashashibi, 48 years, is a senior researcher and the Program Manager of IMARA Team at INRIA (Paris-Rocquencourt) since 2010. He has been senior researcher and Program Manager in the robotics centre of the Ecole des Mines de Paris (Mines ParisTech) since 1994 and was an R&D engineer and a project manager at ARMINES since May 2000. He was previously a research engineer at PROMIP (working on mobile robotics perception dedicated to space exploration) and a technical manager at Light Co. where he led the developments of Virtal Reality/Augmented Reality applications. Fawzi Nashashibi has a Master's Degree in Automation, Industrial Engineering and Signal Processing (LAAS/CNRS), a PhD in Robotics from Toulouse University prepared in (LAAS/CNRS) laboratory, and a HDR Diploma (Accreditation to research supervision) from University of Pierre et Marie Curie (Paris 6). His main research topics are in environment perception and multi-sensor fusion, vehicle positioning and environment 3D modeling with main applications in Intelligent Transport Systems and Robotics. He played key roles in more than 50 European and national French projects such as Carsense, ARCOS, ABV, LOVe, HAVE-it, SPEEDCAM, PICAV, CityMobil... some of which he is coordinating. He is also involved in many collaborations with French and international academics and industrial

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

partners. He is author of numerous publications and patents in the field of ITS and ADAS systems. His current interest focuses on advanced urban mobility through the design and development of highly Automated Transportation Systems. This includes Highly Automated Unmanned Guided Vehicles (such as Cybercars) as well automated personal vehicles.



INVENTEURS DU MONDE NUMÉRIQUE

Automated driving in urban environments: car sharing distribution system and Parking Valet as canonical use-cases

Fawzi Nashashibi

6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles SEPTEMBER 14, 2014

INRIA: National Research Institute in Informatics and Automation

Inria PARIS - Rocquencourt

Inria LILLE Nord Europe

Inria NANCY Grand Est

Inria SACLAY Île-de-France

Inria RENNES Bretagne Atlantique

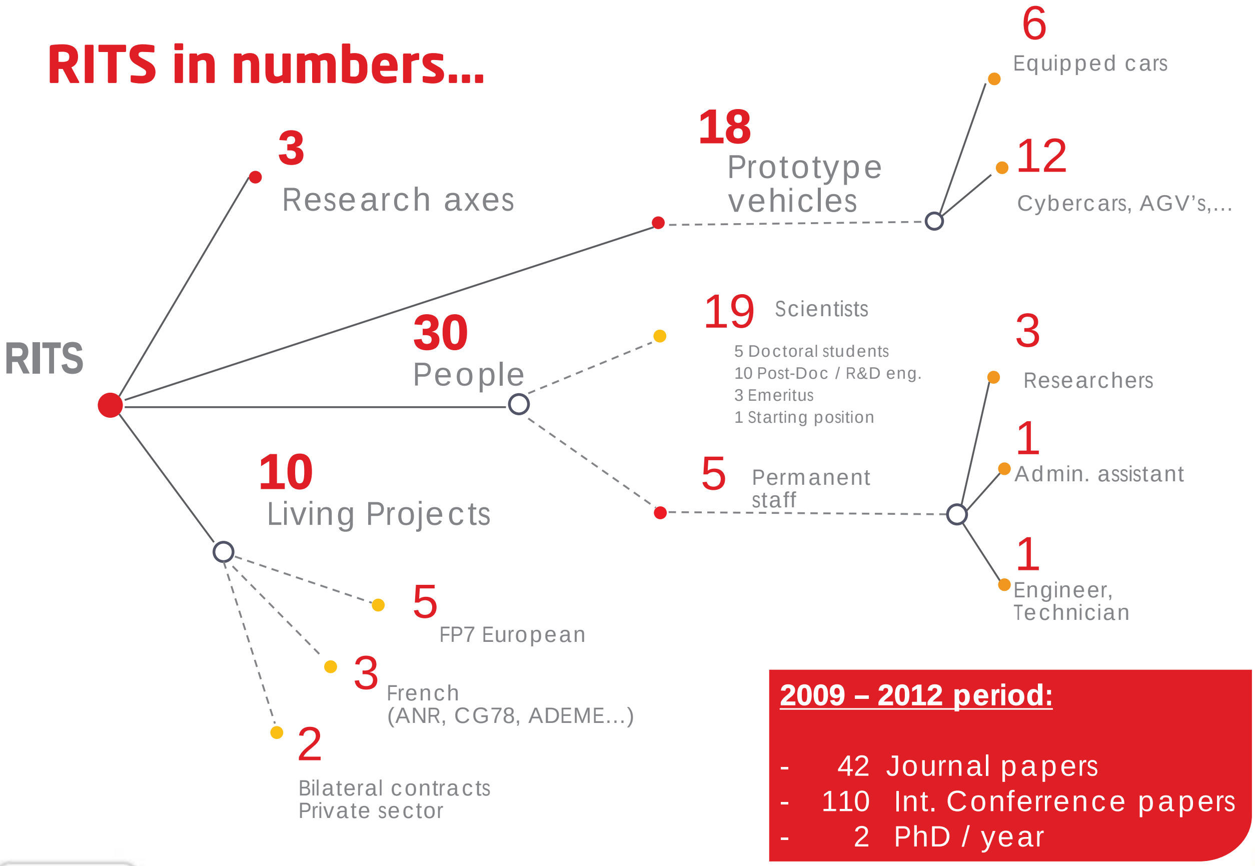
Inria GRENOBLE Rhône-Alpes

Inria BORDEAUX Sud-Ouest

Inria SOPHIA ANTIPOLIS Méditerranée

RITS team Robotics & ITS

RITS in numbers...



2009 – 2012 period:

- 42 Journal papers
- 110 Int. Conferrence papers
- 2 PhD / year

3 Main research topics

1 Robotics & ITS
Intelligent Transportation Systems

2 Modeling of large systems

3 Telecommunications & Networks

Road transport: figures

World: 7 Bn people -> 700 millions of cars

1910: 10% of earth’s population lived in cities
2010: 53%
☞ 2050: **75%**

Security:
World: 1,3 million road deaths / 25-50 million injuries
cost of road accidents: € 407 Bn

Environment
Tens of millions of tons of pollutants

Mobility
- Urban sprawl

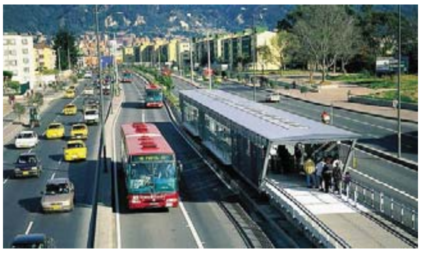
Urban Sprawl

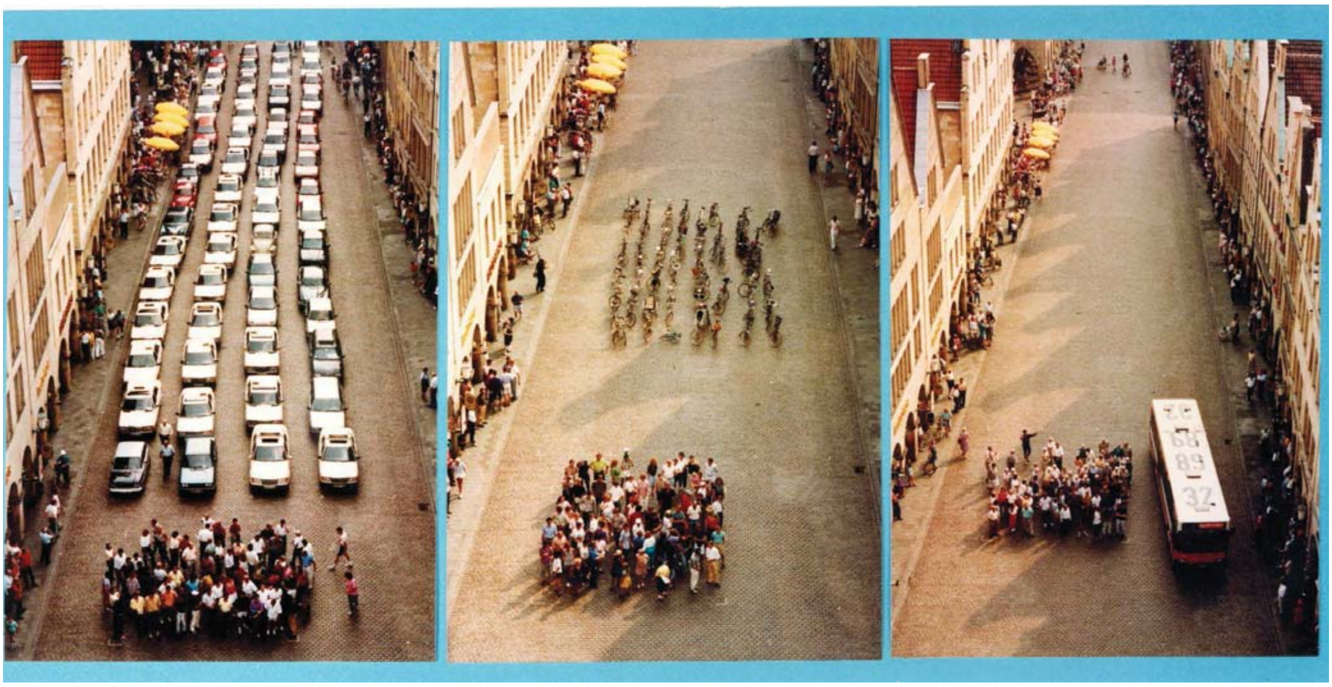
Parking

Space*Time Expenditure

- Moving :
  - Pedestrian: 1 m2*h
  - Bicycle: 1 m2*h
  - Moped: 2.0 m2*h
  - Car: 10 m2*h
  - Bus (30p): 0.3 m2*h
  - Tram (200p): 0.1 m2*h

- Parking (8h) :
- Pedestrian: 0
- Bicycle: 8 m2
- Moped: 12 m2
- Car (street): 80 m2
- Car (parking): 240 m2
- Bus/Tram: 0





ICT for sustainable & smart mobility

Transport: a societal and economic challenge

➔ **The use of ITS fits in the context of sustainable development :**

1. Offering a range of alternatives to the car : recommendation for the development of public transport."
2. New mobility concepts fitting with new cities organization
3. Development of the transport multi-modality
4. Improvement of tomorrow's vehicles: clean / "intelligent" / dedicated

« A smart city is a city that allows people to move freely and where they are informed about all possible mobility choices »

Which car for the future ?

- Private vs . shared ?
- Automated or Autonomous ?

- **Clean**: 2 l / 100 km
- **Light weight, Robust**
- **Full drive-by-wire**
- **Intelligent**
- **Connected**, communicant
- **Dual** (manual / automated)

AKKA – INRIA

Link & Go

Link & Go : autonomous driving

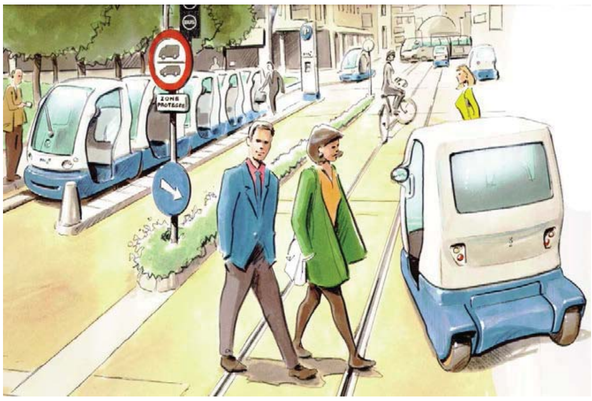
NEW CONCEPTS FOR URBAN MOBILITY

Bike Sharing

Car Sharing

Concept INRIA/INRETS (1991)

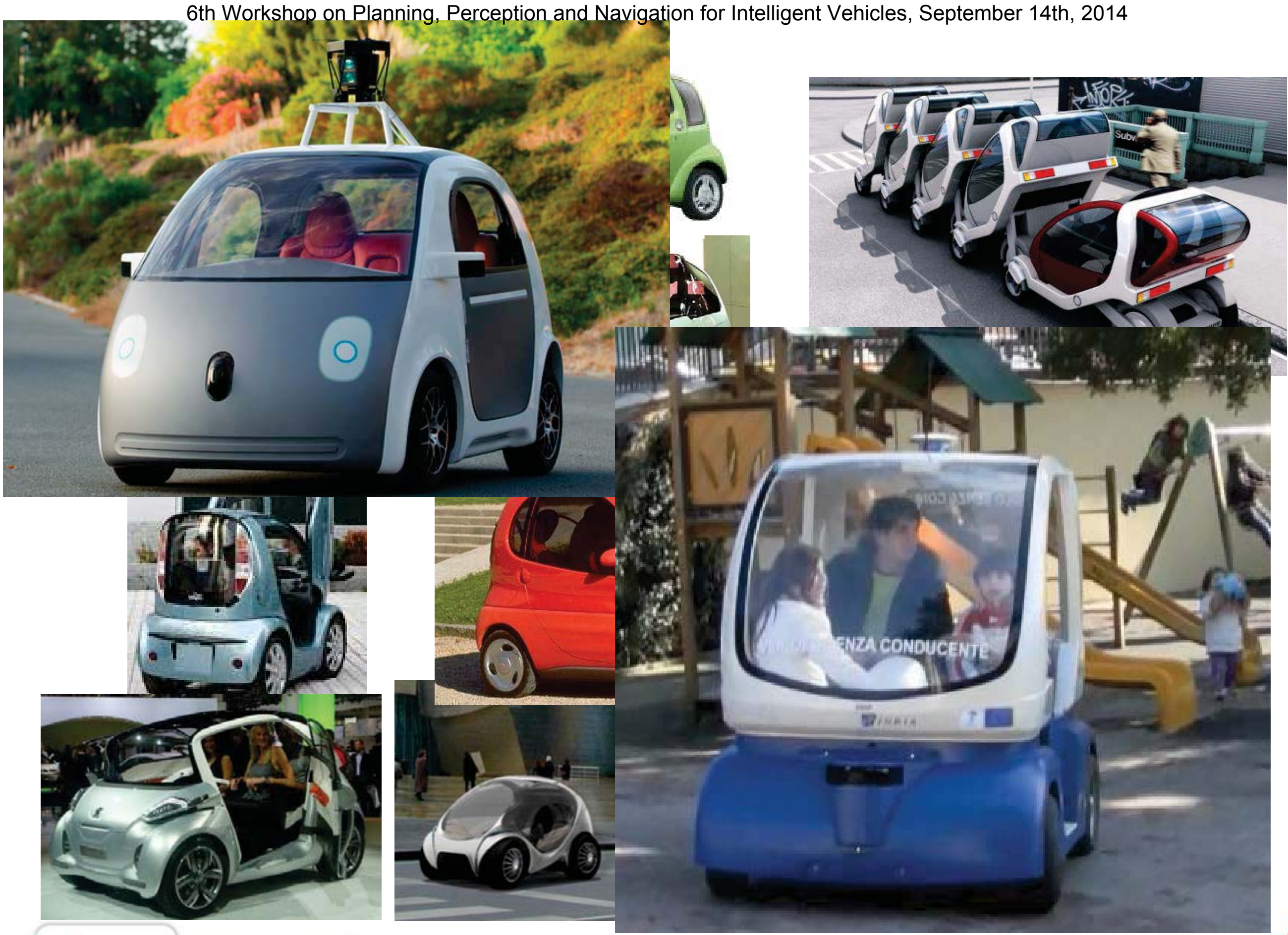
- Small Public Urban Vehicles
- Assited driving
- Platooning
- Automated Parking
- Automated tracks
- Complement to other modes



Praxitele (1993-1999)

MIT-GM SmartCity Project

Segway-GM (2010)



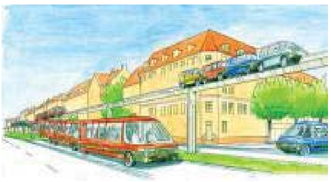
The European project CATS

Cybercars : a European Story

- First concepts in1990's (Serpentine, RUF, Dedale, Frog, ULTra,…)
- First prototypes en 1990
- First exploitation en 1997
- CyberCars/CyberMove en 2001
- Demonstrations at Antibes - 2004
- MobiVIP (2004)
- CyberC3 (2005)

- **CyberCars2 (2006)**
- **CityMobil (2006)**
- **Cristal (2007)**
- **CityNetMobil (2008)**
- **CATS (2009)**

- **CityMobil-2 (2012)**



CityMobil project: Heathrow's PRT

CityMobil project : Castellón automated bus

CityMobil project:
La Rochelle temporary demo

Demonstration site constraints

Intersections = cybercars priority crossing at limited speed

New concept of urban mobility

Some (real) questions !

1. What are the objectives of automation ?
2. What is the benefit of automation in cities ?
3. Automation or Autonomy ?

What are the objectives of automation ?!

SAFETY

- Reducing / annihilating accidents and their severity
- Improved safety for the driver and the surrounding elements

ADVANCED MOBILITY

- Efficiency: optimization/reduction of travel times
- Better use of the road infrastructure (repartition)
- Less traffic jams, congestion and bottlenecks ?

CIVIC TARGETS

- Reduce congestion
- Support economic viability
- Reduce accidents
- Reduce pollution : “clean” environment ?
- Support social needs ?

INDIVIDUAL TARGETS

- Less strain and fatigue while driving
- Less fines ?!
- More time for other tasks

Why automation in cities ?

- **Personal needs:**
  - For long journeys (> 20 min.)
  - Congestion management
  - Safety improvement: ego-vehicle, other vehicles, vulnerables
  - Parking
- **Private/commercial needs:**
  - Efficient car-sharing system -> vehicles redistribution
  - Automated taxi (e.g. from/to train stations)
  - Efficient « Parking Valet »
  - …
- **Solutions**:
  - Traffic management and high level planning
  - Global planning
  - On-board intelligence:
    - perception
    - Communication
    - Intelligent control

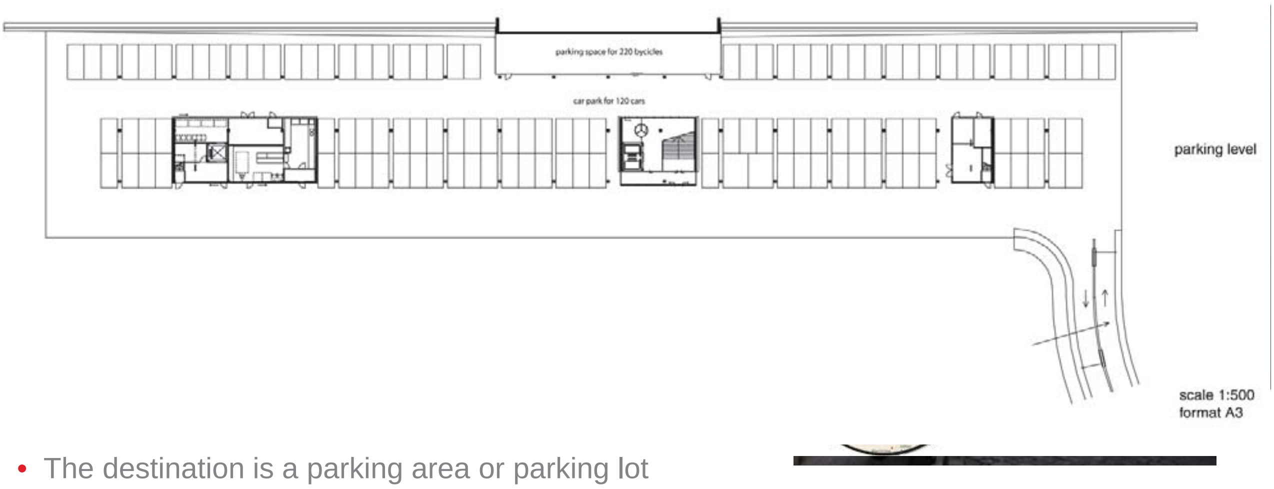
Automation in cities: a canonical case

- Some practical challenges:
- On-road…
- **Intersections: crossroads, roundabouts,…**
- **Need for dedicated infrastructure**
- **Need for automated functions:**
  - **Platooning**
  - **V2X communication**
  - **Optimal management**
- At destination…
- **Parking management**

☞ **Intelligent redistribution of car-sharing vehicles**

Intelligent redistribution of car-sharing vehicles

- Assumptions:
- Car sharing vehicles can be dropped anywhere BUT on pre-defined « authorized



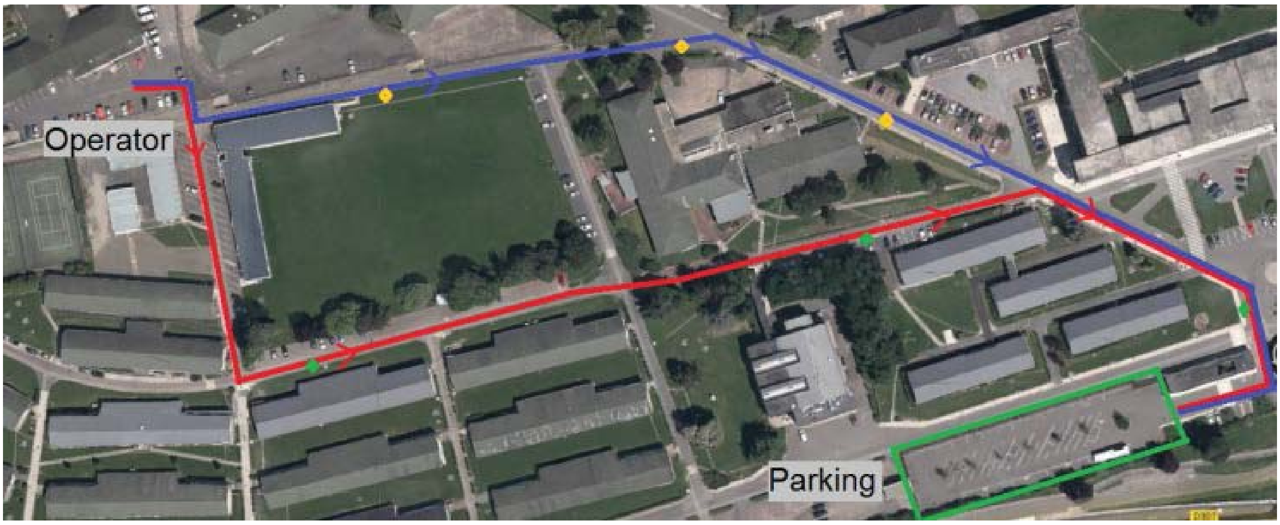
- The destination is a parking area or parking lot
  - The place is assigned
  - The parking's map is known

Typical scenario

First step:

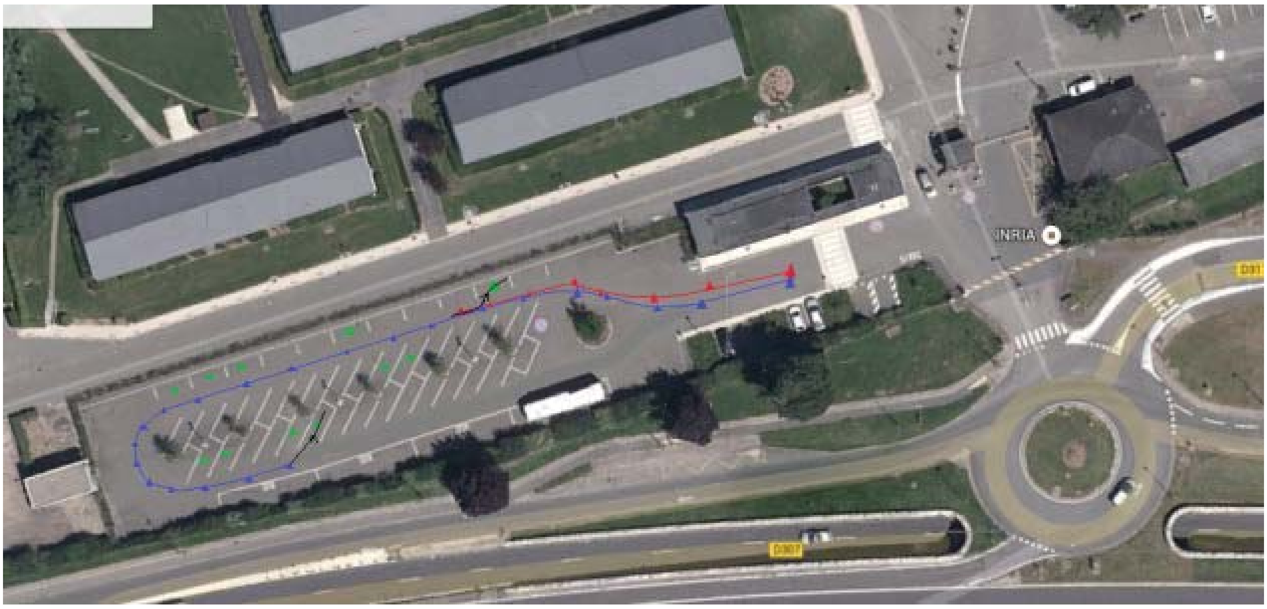
- Itineraries selection and optimization
- Paths are sent to each convoy head:

  number of vehicles, waypoints (vehicles geolocalized positions), destination



Platoons formations

- Second step: (for each Convoy)



- Automated parking: parallel, perpendicular and diagonal

Automated functions - Platooning

❑ Autonomous Platoon

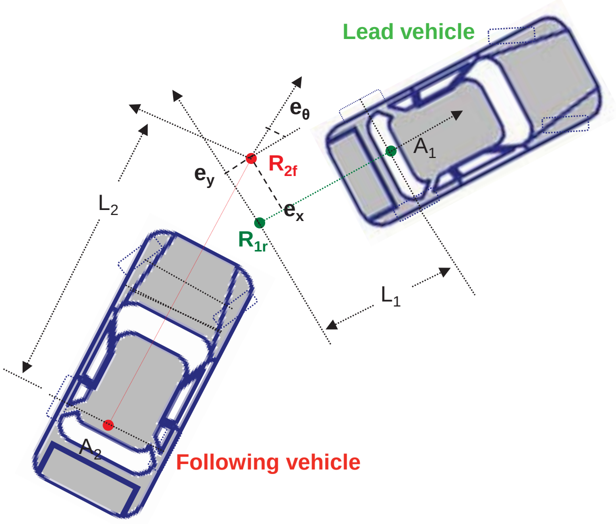
- Without the use of any road infrastructure or inter-vehicle communication;
- The current inter-vehicle position and orientation for feedback control of the following vehicle is obtained from onboard sensors (laser range finder);
- Estimation of the unknown linear and angular velocities of the vehicle ahead is achieved by using adaptive control;

❑ Objective

- Any following vehicle in the platoon has to track the trajectory of the vehicle ahead with prescribed distance.

❑ Approach

- Tracking a virtual reference point R_{1r} of the vehicle ahead with a reference point R_{2f} of the following vehicle.

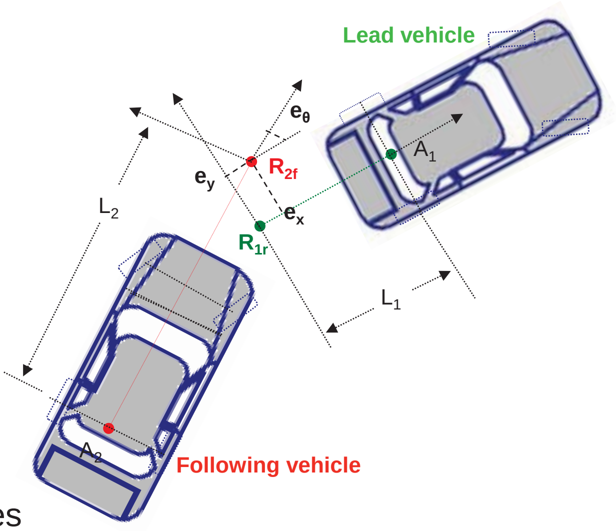


P. Petrov,M. Parent – "Nonlinear adaptive control for autonomous vehicle following" - 16th ITS Congress ,Stockholm 2009

Platooning

❑ Relative Kinematics

- Coordinates and orientation of the frame $R_{2f}xy$ in the coordinates frame $R_{1r}xy$



$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^F x_{R_{2f}} - {}^F x_{R_{1r}} \\ {}^F y_{R_{2f}} - {}^F y_{R_{1r}} \\ \theta_2 - \theta_1 \end{bmatrix}$$

$e = [e_x,\ e_y,\ e_\theta]^T \in \Re_3$ - error posture

- Inter-vehicle kinematics in error coordinates

$$\begin{bmatrix} \dot e_x \\ \dot e_y \\ \dot e_\theta \end{bmatrix} = \begin{bmatrix} \cos e_\theta & -L_2 \sin e_\theta \\ \sin e_\theta & L_2 \cos e_\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{A_2}v_{A_2 x_{A2}} \\ \omega_2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & -L_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{A_1}v_{A_1 x_{A1}} \\ \omega_1 \end{bmatrix} + \dot\theta_2 \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}$$

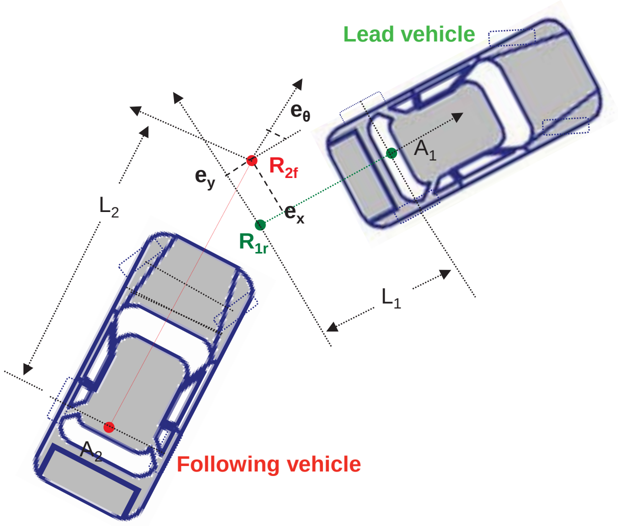
$\left({}^{A_1}v_{A_1 x_{A1}}, \omega_1\right)$ - linear and angular velocities of the lead vehicle

$\left({}^{A_2}v_{A_2 x_{A2}}, \omega_2\right)$ - linear and angular velocities of the following vehicle

Platooning

❑ Problem Formulation

Given the inter-vehicle kinematics in error coordinates, and assuming that the linear and angular velocities of the lead vehicle (v_{A1}, ω_1) are unknown constant parameters, the control objective is to asymptotically track the virtual reference point R_{1r} the lead vehicle with the reference point R_{2f} of the following vehicle.



❑ Adaptive Control

- Change of the input:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos e_\theta & -L_2 \sin e_\theta \\ \sin e_\theta & L_2 \cos e_\theta \end{bmatrix} \begin{bmatrix} v_{A2} \\ \omega_2 \end{bmatrix}$$

- Control law

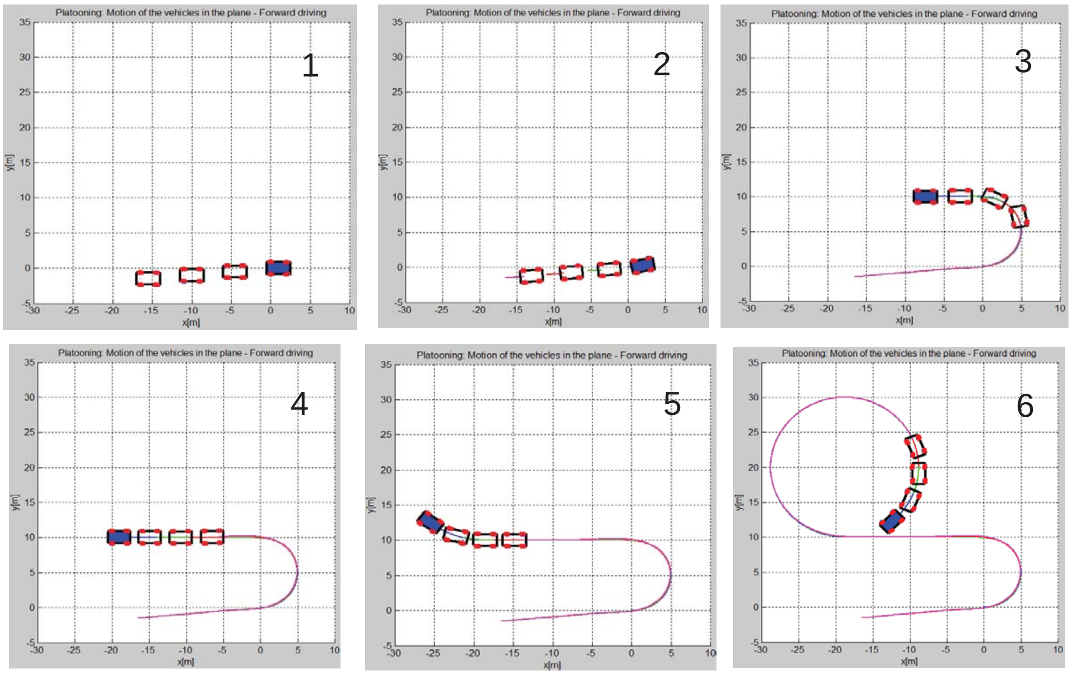
$$u_1 = -k_x e_x + \hat{v}_{A1} - \hat{\omega}_1 e_y$$
$$u_2 = -k_y e_y - (L_1 - e_x)\hat{\omega}_1$$

where the estimates ($\hat{v}_{A1}$ $\hat{\omega}_1$ are obtained form the adaptive update law:

$$\dot{\hat{v}}_{A1} = -\gamma_v e_x$$
$$\dot{\hat{\omega}}_1 = \gamma_\omega L_1 e_y$$

- Asymptotic stability of the closed loop system in error coordinates was proved.

- Simulation results with 4-vehicles platoon



EXPERIMENTS: TWO-VEHICLE PLATOON

EXPERIMENTS: THREE-VEHICLE PLATOON

Automated parking

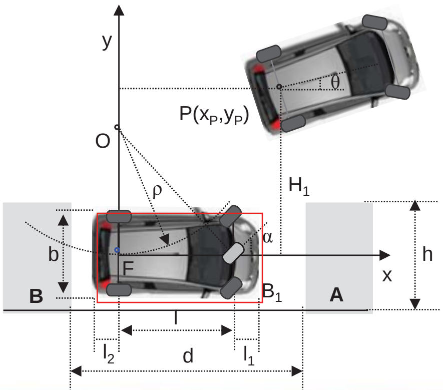
❑ Objective

- Automatic parking by using one or multiple maneuvers, depending on the size of the parking spot and the starting position of the vehicle.

❑ Controller

- Feedback steering (saturated control) and velocity control.

- An approach of using saturated (SAT) control with two different levels of saturation was designed to enlarge the area of starting positions of the vehicle, from which the parking can be achieved in one maneuver.



P. Petrov, F. Nashashibi, "Saturated Feedback Control for an Automated Parallel Parking Assist System", ICARCV 2014, Singapore.

Automated parking

- The problem of stabilizing the vehicle is seen as an extension of the tracking problem.

- The design of the saturated path tracking controller, proposed in this paper is based on "high-gain"-type control design.

- Saturated control (α – steering angle):

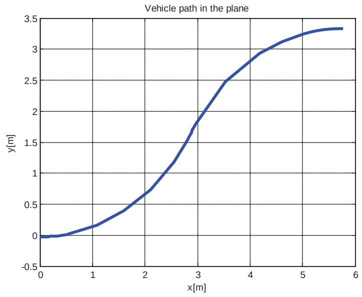
$$u_S = u_m sat(\xi) \qquad u_S = \frac{\tan\alpha}{l}$$

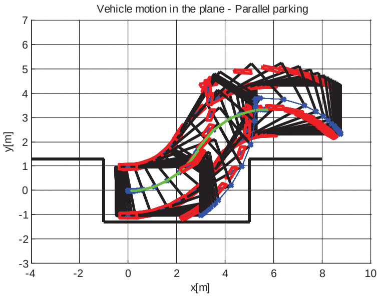
$$sat(\xi) = \begin{cases} -1 & for \quad \xi < -1 \\ \xi & for \quad |\xi| \le 1 \\ 1 & for \quad \xi > 1 \end{cases}$$

$$u_m = abs(u_s) \qquad \xi = \frac{u}{u_m} \qquad u = k\left(e_\theta - k_0 e_y\right)$$

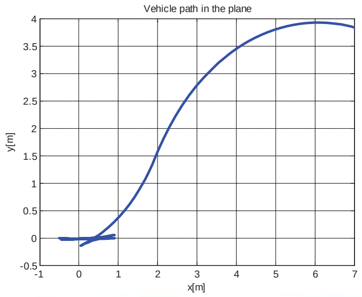
❑ Parallel Parking /Simulation Results/

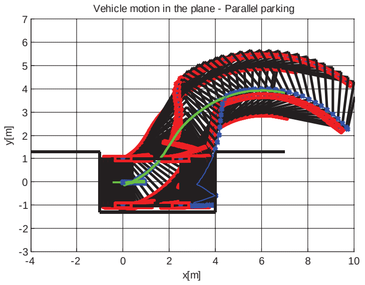
- Parallel parking in one maneuver



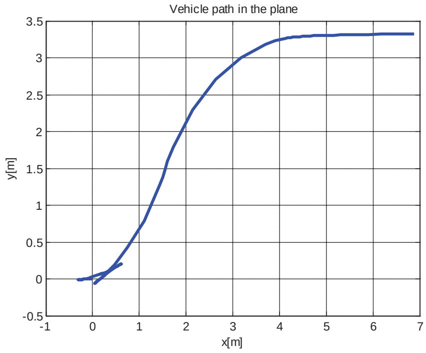


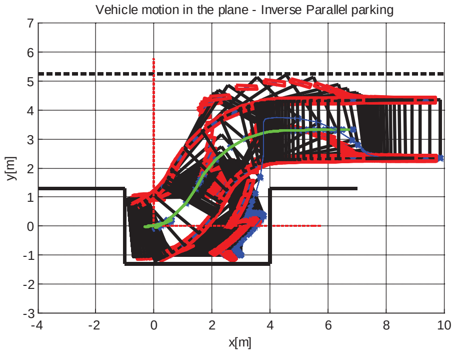
- Parallel parking in multiple maneuvers





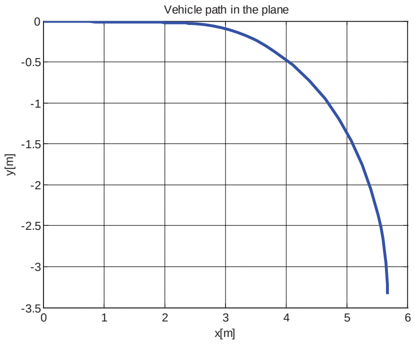
❑ Inverse Parallel Parking /Simulation Results/

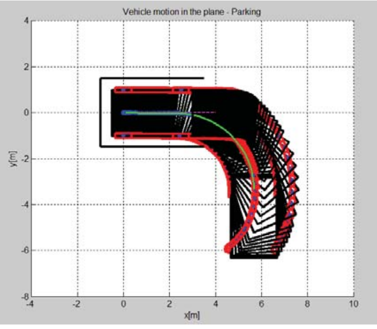




❑ Inverse Parallel Parking /Experiments/

❑ Perpendicular Parking /Simulation Results/





❑ Diagonal Parking /Experiments/

Vehicle Localization

- Needed for the localization of autonomous vehicles in the parking or charging station.

- SLAM based technique for indoor parking
- SLAM+GPS for outdoor parkings

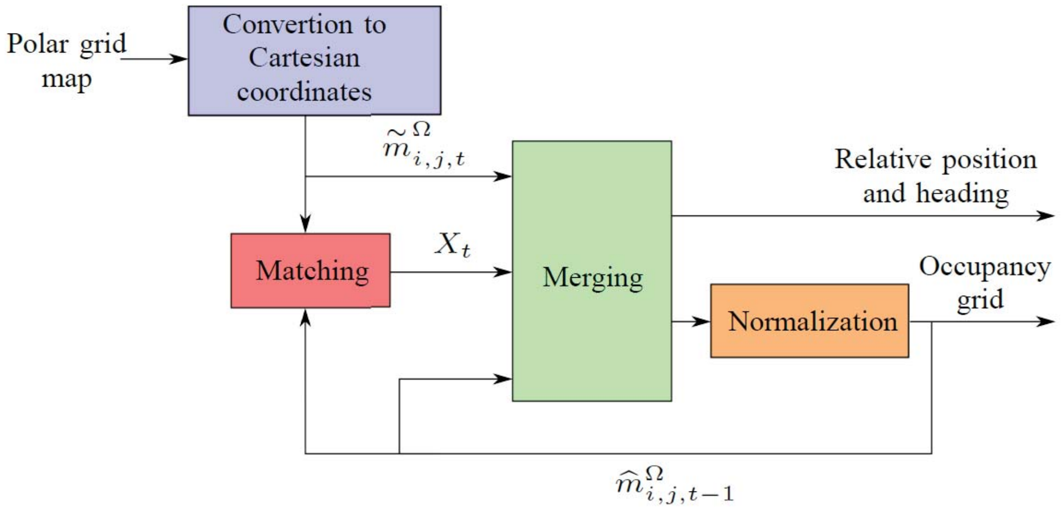
- Two SLAM approaches were designed:

1. ML-SLAM: Maximum Liklihood

J. Xie, F. Nashashibi, M. Parent, and O. Garcia-Favrot, "A real-time robust SLAM for large-scale outdoor environments," ITS World Congress,Busan, Korea, 2010.

2. C-SLAM: Credibilistic

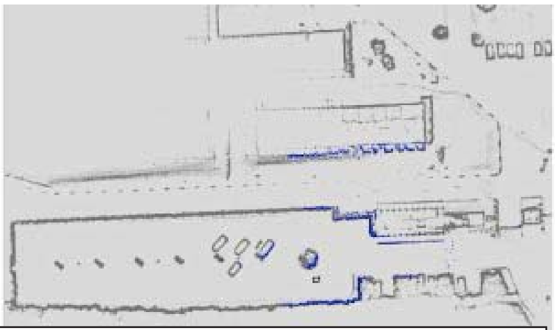
Guillaume Trehard et al., " Credibilist Simultaneous Localization and Mapping with a LIDAR", IROS'2014, Chicago, USA, 2014 !


Polar grid map
Convertion to Cartesian coordinates
$\tilde{m}^{\Omega}_{i,j,t}$
Matching
X_t
Merging
Relative position and heading
Occupancy grid
Normalization
$\widehat{m}^{\Omega}_{i,j,t-1}$

- A Maximum-Likelihood SLAM using occupancy grid to illustrate the limits of probabilities and validate the concept of credibilist SLAM

- A test over 10 sequences of the KITTI database which led to an average of 3.2% in translation drift and of 0.0040 deg/m in rotation drift

B
A



Thank you for your attention

fawzi.nashashibi@inria.fr

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session I

Localization & Mapping

- **Title: An Iterative Graph Optimization Approach for 2D SLAM**
  Authors: He Zhang, Guoliang Liu, and Zifeng Hou

- **Title: Appearance-based Localization across Seasons in a Metric Map**
  Authors: Chris Beall, Frank Dellaert

- **Title: High Precision 6DOF Vehicle Navigation in Urban Environments using a Low-cost Single-frequency GPS Receiver**
  Authors: Sheng Zhao; Yiming Chen, Jay A. Farrell

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

An Iterative Graph Optimization Approach for 2D SLAM

He Zhang, Guoliang Liu, *Member, IEEE,* and Zifeng Hou

Abstract— **The-state-of-the-art graph optimization method can robustly converge into a solution with least square errors for the graph structure. Nevertheless, when a biased edge (erroneous transformation with over-confident information matrix) exists, the optimal solution can produce the large deviation because of error propagation produced by the biased edge.**

In order to solve this problem in graph-based 2D SLAM system, this paper proposed an iterative graph optimization approach. To reduce the errors propagated from the biased edges, we iteratively reconstruct the graph structure by referring to the result of the graph optimization process. Meanwhile, to maintain the information of the other well estimated edges, we strictly update the graph structure by considering the scan-correlation score and the marginal covariance. In addition, we apply a novel key-node mechanism to robustly detect the loop-closure by a linear interpolation algorithm. The experiments show that the proposed method is more robust and accurate than the previous methods when the biased edges exist.

I. INTRODUCTION

The simultaneously localization and mapping (SLAM) problem is a pivotal problem in the robotics community, since it handles the most significant information for the autonomous mobile robot. In 2D SLAM, laser scanner and odometry usually be applied in the-state-of-the-art methods such as Grid Mapping (GMapping) [11] and Graph Optimization (GO) [13]. These methods work well and noisy-resistent, but they will degenerate when motions are estimated erroneously but with high confidence. This often occurs when the vehicle traverses through a corridor-like place and the vehicle slips or applying laser-odometry algorithms [3][7][17]. The previous methods can fail when such edges exist.

The Gmapping implements a rao-blackwellized particle filters(RBPF) to predict its pose distribution under gaussian assumption and update the weights of the particles under the Baysian rule with a likelihood evaluation for each prior pose. Then resampling is carried out to eliminate particles with small weights. However, if motions are estimated with extreme noise such as vehicle slipper, resampling may eliminate particles with true trajectory and could never be recovered.

The graph-based SLAM methods strive to reduce the errors of the odometry constraint and the loop constraint. To reduce the errors of the loop constraint, previous works such as JCBB[16], SCGP[19], RRR[14] involve front-end validation of loop closure to eliminate the false positive loop edges. For the errors of the odometry constraint, they are assumed to be corrected during the graph optimization when true loop edges are added into the graph structure. However, if some odometry constraint contains large errors and over-confident information matrixes, the graph optimization will also result in erroneous solution even when no false loop edges exist.

The key insight of the proposed approach is that we can correct the biased motion estimation in the front-end by referring to the result of the graph optimization in the back-end. When the poses of the nodes are updated, we can use the same scan-matching algorithm to recalculate the edge information with various prior motion guesses. We perceive that in our 2D graph-based SLAM method, biased edges result from poor prior motion guess that can be improved when loops are accurately closed and the graph structure is optimized. Therefore, we propose the iterative graph optimization algorithm to update the graph structure in the front-end by the aid of the optimization process in the back-end.

The minor insight is that, in contrast to detecting loops among nodes in the graph, we construct recoverable key-node to verify loop closures. We perceive that the local map information contained in key-node can be inconsistent and thus we can rebuild the local map for each key-node after the graph optimization. If the local map of a key-node is maintained as a whole in a higher level for the map representation [4][5][23], we can never eliminate the errors in the local map. Therefore, we rebuild the local map of each key-node when the nodes are updated by the graph optimization process. In addition, because of the accumulated motion error, the initial motion guess to align a scan frame with a local map may fall out of the right convergent basin. Thus we provide multiple interpolated initial motion guesses between current pose and the pose of the key-node to robustly detect loops.

In short, the central contributions of this paper are following:

- We apply a new iterative graph optimization approach that reduces the errors propagated by the biased edges in the graph.
- We adopt key-nodes mechanism to robustly detect loop. The corruptly constructed key-nodes could be recovered when good loop edges are inserted into the graph.
- We employ an adaptive linear interpolation algorithm to detect loop and recalculate edge information after graph optimization.

The structure of this paper is as follows. In the fol-

H. Zhang and Z. Hou are with the Institute of Computing Technology, University of Chinese Academy of Sciences and with the Lenovo (Beijing) Ltd. 6th Academy Road, Haidian District Beijing Email: fuyinzh@gmail.com; houzf@lenovo.com

G. Liu is with the Lenovo (Beijing) Ltd. Email: liugl6@lenovo.com

lowing section, the previous related works are discussed. In section III, the process of graph construction will be demonstrated. After that, the iterative graph optimization algorithm is illustrated in detail. Then we carry out two experiments using the real data to prove the superiority of our method in the section V. In the end, the conclusion and future work are presented.

II. Related Work

Kümmerle *et al*. [13][9] illustrate the advantages of the graph-based SLAM methods. But when the graph structure contains false loop edges, these methods will fail catastrophically. To solve this problem, many front-end validation approaches [16][19][14] have been proposed. In addition, other methods strive to alter the graph structure during the back-end optimization process. Sünderhauf and Protzel [20] propose switchable variables to alter the graph structure during the graph optimization process, that can turn off the false loop edges. Olson and Agarwal explicitely [18] model the errors of loop edges and track multiple hypotheses for each edge during the graph optimization process. However, as far as we know, none of these methods make effort to reduce the errors of the biased edges. In fact, the graph optimization could converges into an undesirable solution when the graph contains a biased edge even no false loop edge. Our 2D SLAM method dedicates to reduce the errors of the biased edges by iteratively reconstructing the graph structure with reference to the result of the graph optimization process.

In addition, our 2D SLAM method adopt the submap mechanism to improve the robustness of data association. This idea is not entirely new and has been intensely explored [5][6][4]. The improvements of the proposed method lies in two parts: interpolated initial motion guesses for scan-matching [17] and recoverable mapping. The first part can improve the accuracy of loop edges by aligning a single scan with a previous submap with multiple initial motion guesses. The second part can reduce the errors of the local map by reconstructing the local map after the graph optimization.

III. Graph Construction

The graph-based SLAM algorithm contains two components: the front-end and the back-end. The front-end includes graph construction and the back-end focuses on graph optimization. In this section, we mainly talk about the front-end process, and leave the back-end in the next section. The front-end mainly contains two parts: motion estimation and loop detection. For the motion estimation, we use the particle-type representation to model the uncertainties like the work in [15][10]. We randomly choose a set of hypothesis based on the prior odometry model. And then we employ scan-matcher algorithm [17] to compute the scan correlation score between the current laser frame with the global grid map [21] as the weight of each hypothesis. We choose the hypothesis with maximum weight and add its transformation and information matrix into graph. For the loop detection, we adopt the key-node idea, and construct local grid map in each key-node. We apply scan alignment with interpolated prior guesses to calculate loop constraints.

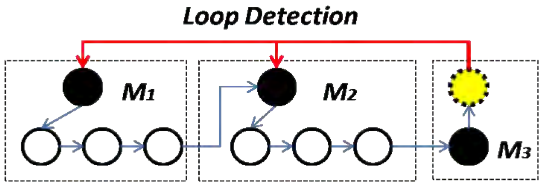


Fig. 1: Graph Structure: black solid circle stands for key-node, dotted rectangle includes the information in each key-node, and the dotted yellow circle means the current node; the blue edges stands for the transformation between the nodes, and the red edges are loop-edges

A. Motion Estimation

In our method, we estimate a set of hypotheses on the relative motion. We provide prior guesses P_{pri} for robot position P_i according to its odometry model with gaussian noise Σ_{odo}.

$$\Sigma_{odo} = \begin{bmatrix} \Sigma_{tt} & \Sigma_{tr} \\ \Sigma_{rt} & \Sigma_{rr} \end{bmatrix} \tag{1}$$

After that, we measure the scan correlation score between current observation and global gridmap M_g with these prior poses P_{pri}. We set P_i as the pose with the maximum scan correlation score. Then we calculate the relative transformation and covariance Σ_{obs} [2] between successive robot pose P_{i-1} and P_i. In the experiment, we found that the Σ_{obs} is often over-confidently estimated. Therefore we take the correlation score into consideration when the covariance is estimated between P_{i-1} and P_i as follows:

$$\Sigma_{i-1,i} = \begin{cases} \Sigma_{obs} & \text{if } Score \geq minScore \\ \Sigma_{obs} + \frac{minScore-Score}{minScore} * \Sigma_{odo} & \text{otherwise} \end{cases} \tag{2}$$

minScore represents the least percentage laser beams for a good estimation in the scan-matcher algorithm. In our experiments we found it works well when it is set as 85% of the total laser beams. Then, we add node n_i and edge $e_{i-1,i}$ into graph. The information matrix in $e_{i-1,i}$ is $\Sigma^{-1}_{i-1,i}$. After that, we construct key-node and detect loops which will be explained as follows.

B. Key-Node Construction

The key-node is the same as other nodes except that it integrates observations into a local map. As shown in the Figure 1, the black solid circle means a key-node. The dotted rectangle shows the nodes and the local gridmap M_i of key-node i. As explained more explicitly in [12][8], the motivation behind key-node is to increase loop detection accuracy by matching with a local submap instead of a single scan. Therefore, when searching loop connections, we only select the key-nodes as potential loop matches. For example in Figure 1, the yellow circle is the current pose of the robot, and it detect potential loops by matching with the key-nodes k_n_1 and k_n_2. In addition, the transition between key-nodes can be triggered under different constrains such as observation area, trajectory length or number of nodes. In our method, we find that using trajectory length can guarantee substantial grid map size but also works as a clue to detect loops. In order to reduce error in the key-node, the maintained grid map M_i of key-node k_n_i will be rebuilt in the iterative optimization process. This will be illustrated in the section IV.

As mentioned previously, we switch to a new key-node when the length of the trajectory contained in current key-node exceeds a predefined distance T_l. T_l is set based on the scale of the environment in which the robot traverses. On the one hand, it must contain enough observed information to make accurate scan alignment when loop occurs. On the other hand, key-nodes should be distributed broadly for the current node to efficiently check whether it reenter the area in one of the previous key-nodes. Once this happen, we verify whether there is loop-closure detected. This process is discussed in the next part.

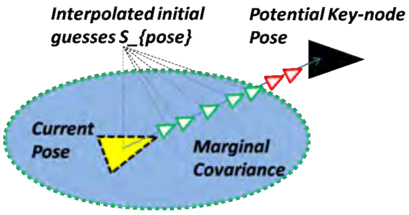


Fig. 2: Interpolated Poses for Initial Guess in Scan-Matcher

C. Loop Closure

When the current pose of robot gets close to a previous key-node, that means the distance between the pose P_i of current node i and the pose P_{k_j} a key-node j is less than T_l, a potential loop may exist. Then we further check if there exist a node n_l in the key-node k_n_j, and the Mahalanobis distance between them satisfy:

$$e_{dis} = P_i \ominus P_l \tag{3}$$

$$\Sigma_{il} = R^t_{li} * \Sigma_l * R_{li} + \Sigma_i \tag{4}$$

$$| e^{-1}_{dis} * \Sigma_{il} * e_{dis} | < \varepsilon \tag{5}$$

Σ_i and Σ_l is the marginal covariance for the node i and l respectively. P_i and P_l are the pose for the node i and l. R_{li} is the rotation part of the transformation from pose P_l to P_i. ε is a predefined threshold. When these conditions are satisfied, we will align the current scan frame and the gridmap M_j in k_n_j to verify whether a consistent transformation can be estimated.

When the robot traverses along a long path, the accumulated motion error becomes large. In this case, the initial guess maybe far from the right convergent basin, therefore using scan-matching algorithms [3][7][17] may fail to precisely estimate the transformation for the loop constraint. Even if the overlap between the current laser frame and the gridmap in a key-node is large, without proper initial guess, the frame alignment algorithm may fall into poor local optima. To solve this problem, we make multiple initial guesses by adaptively interpolating initial motion guesses between P_i and P_{k_j}. As depicted in the Figure 2, we provide multiple initial guesses (small triangles) by linearly interpolating poses between current robot position and key-node position. We select the result from scan-matcher with the biggest scan correlation score. The linear interpolation is a dynamical process, it takes into P_{dis} and marginal covariance Σ_i into consideration. In our experiments, we assume that the proper initial motion values distribute along the direction from P_i and P_{k_j}. Therefore along this direction, we interpolate an initial pose every $5cm$ within P_{dis}.

The Σ_i is the marginal covariance of the node i. When no loop is closed, it accumulates according to:

$$\Sigma_i = R^{-1}_{i-1,i} * \Sigma_{i-1} * R_{i-1,i} + \Sigma_{i-1,i} \tag{6}$$

$R_{i-1,i}$ is the rotation part of the transformation from pose P_{i-1} to P_i. $\Sigma_{i-1,i}$ is the estimated covariance between observations in the node $i-1$ and i by equation 2. If loop closure happens, it is calculated following the rule in [22].

The blue eclipse in the Figure 2 represents the current marginal covariance, and the red small triangles that exceed the eclipse will be discarded. When the loop is verified, a loop edge will be inserted into the graph, and we will perform graph optimization process explained in the next section.

Algorithm 1 Iterative Graph-based Optimization Algorithm

```
function REBUILDGRAPHANDMAP
    $S_l$                              ▷ Loop edges and nodes detected in Frontend
    $M_g \leftarrow empty$             ▷ Global gridmap
    for $n_i$ in graph do
        if $n_i \in S_l$ then
            continue                   ▷ skip loop nodes
        end if
        $S_{pose} = interpolate(P_i, P_{i_{new}})$
        $e_{i-1,i} = maxScanMatcher(n_{i-1}, n_i, S_{pose}, M_g)$
        $P_i = P_{i-1}.oplus(e_{i-1,i})$
        $graph.replace(e_{i-1,i})$     ▷ replace with the new edge
        $graph.update(P_i)$            ▷ reset pose of node
        $M_g.insert(n_i)$              ▷ reconstruct global gridmap
    end for
    for $e_{j,k}$ in $S_l$ do
        $S_{pose} = interpolate(P_k, P_{k_{new}})$
        $e_{j,k} = maxScanMatcher(key\_n_j, n_k, S_{pose}, M_j)$
        $P_k = P_j.oplus(e_{j,k})$
        $graph.replace(e_{j,k})$
        $graph.update(P_k)$
    end for
    $reconstructKeyMap()$              ▷ rebuild gridmap in each key_node
end function
function ITERATIVEGRAPHOPTIMIZATION($iter$)
    $last\_chi2 = optimizeGraph()$
    while $i++ < iter$ do
        $rebuildGraphandMap()$
        $curr\_chi2 = optimizeGraph()$
        if $|curr\_chi2 - last\_chi2| < \varepsilon$ then
            break
        end if
        $last\_chi2 = curr\_chi2$
    end while
end function
```

IV. ITERATIVE GRAPH OPTIMIZATION

The graph optimization dedicates to find a solution with least square errors given the topological graph structure.

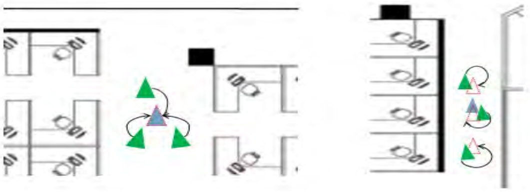


Fig. 3: Scan-Matching using the same laser frame in the Office-like scenario (left) and the Corridor-like scenario (right): blue triangle is the target vehicle pose, green triangles stand for different prior reference vehicle pose, and the red dashed triangle means the convergent vehicle pose calculated by scan-matching

However, even when no false loop edge exists in the graph, some biased edges (erroneous transformation with over-confident information matrix) can result in solution with significant error. To reduce the errors propagated from the biased edges rather than false loop edges, we propose the iterative graph optimization algorithm that will be explained in the following contents.

A. Biased Edges

In our 2D SLAM system, we use scan-matching to estimate the relative motion between sequential vehicle poses. However, the accuracy of scan matching depends highly on the scenarios where the vehicle stays. As shown in Figure 3, in the left office like environment, the scan-matching can result in the same accurate pose (red dashed triangle) even with different priors (green triangles). While in the right corridor like place, its results differ and ranks along the corridor, and thus it's hard to decide where the vehicle stands. With poor prior odometry or laser-odometry, the transformation of the edges in this scenario is often erroneously estimated and the information matrix is over-confidently calculated, that we call biased edges. To improve the accuracy of the biased edges, a better prior motion guess must be provided. We find that when the true loop edges are added, the graph optimization can update the poses of the nodes, which might provide a better prior motion to adjust the biased edges. Therefore, the essence of iteratively optimizing graph is to alter the graph structure in the front-end using the result from the back-end.

B. Iterative Graph Reconstruction

The motivation behind iterative graph reconstruction is to recover the well estimated edges and improve the biased edges. As shown in the Figure 4, the red edge $e(3,4)$ between node 3 and 4 is a biased edge, and the green edge $e(1,5)$ between node 1 and 5 is a validated loop edge. The initial graph structure is depicted in the Figure 4.(1), a loop is closed and a biased edge has been added into the graph. After graph optimization, because of the propagated error from the biased edge, the result of the whole trajectory degenerates, shown in the Figure 4.(2). Then we reconstruct the graph structure as the same process in the front-end, but using the result of the back-end as the prior motion guess. For the good matches, different prior motion guesses can still falls into the same convergent basin as shown in the left part in the Figure 3. While for the biased matches, the relative motion can be updated with a better initial motion guess. This is illustrated in the Figure 4.(3), we see that the relative motion of edges $e(1,2),e(2,3),e(4,5)$ and $e(1,5)$ be recovered, while that $e(3,4)$ be updated. Then we again repeat the same process as shown in the Figure 4.(4)(5), but notice that the red edge $e(3,4)$ in the Figure 4.(3) and the Figure 4.(5) differs, the latter has been improved by iteratively using the result of the back-end optimization process. When the graph structure no longer varies, we again optimize it and obtain the final result as shown in the Figure4.(6).

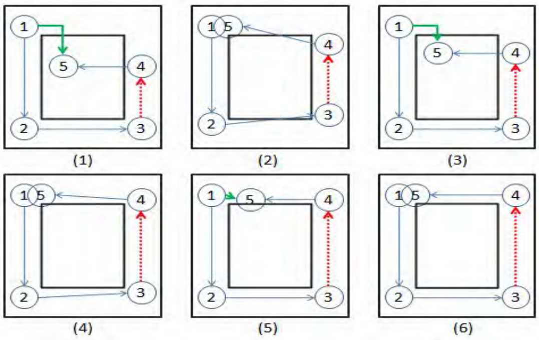


Fig. 4: Iterative Graph Reconstruction: (1) initial graph structure, (2) 1st graph optimization, (3) 1st graph reconstruction, (4) 2nd graph optimization, (5) 2nd graph reconstruction, (6) final graph optimization. Green arrow stands for loop edge, blue for good edge and red dashed for biased edge

C. Algorithm Explantation

The major part of the Algorithm 1 is to rebuild the graph structure and the global gridmap. In line 5 and 6 we skip nodes that has been updated by loop edge for which will be recalculated in the loop set S_l. In line 8 and 16 we interpolate poses between the original position P_i and new position $P_{i_{new}}$ that is updated by graph optimization. We use the same interpolation mechanism as explained in Figure 2 in III-C, and the only difference is that the target position is $P_{i_{new}}$ and current position is P_i. We return the transformation with maximum score, that means most consistent with the rebuilt map M_g. In line 4-14, we rebuild the edges between successive nodes along the robot trajectory. In line 15-21, we recompute the loop edges by aligning laser scan in node n_k with gridmap M_j in the key-node k_n_j. To accurately recover the original estimations once perturbed by graph-based optimization, we strictly reset the pose of each node and the edges between them by considering the marginal covariance and the scan correlation score. After that, in line 22 we reconstruct the local gridmap in each key-node with the newly updated nodes.

The iterative graph optimization part is quite straightforward: we iteratively rebuild graph and optimize it until it converges or the iteration time is more than the threshold *iter*. The *optimizeGraph*() will optimize the graph and return the total square error over the graph.

D. Computational Time Analysis

Suppose the average computational time for graph optimization and scan-matching are $T(o)$ and $T(m)$ respectively. For a graph with E edges, iGO (iterative graph optimization) costs $k(T(o)+E*T(m))+T(o)$ while GO only $T(o)$. k is the average iteration number. However, we can compare the transformation of the edges before and after the graph reconstruction. Therefore, for the edges with no alterations, that means well estimated edges, we will not update in the next loop. For example in Figure, we only recalculate the edges $e(1,2), e(2,3), e(4,5)$ and $e(1,5)$ in the first iteration, and yet update edge $e(3,4)$ in every iteration. Then, the computational time for iGO is $k(T(o)+b*T(m))+E*T(m)+T(o)$. b is the number of biased edges. If no biased edges exist, iGO costs $2*T(o)+E*T(m)$, and the extra time consumed can be seemed as to detect biased edges.

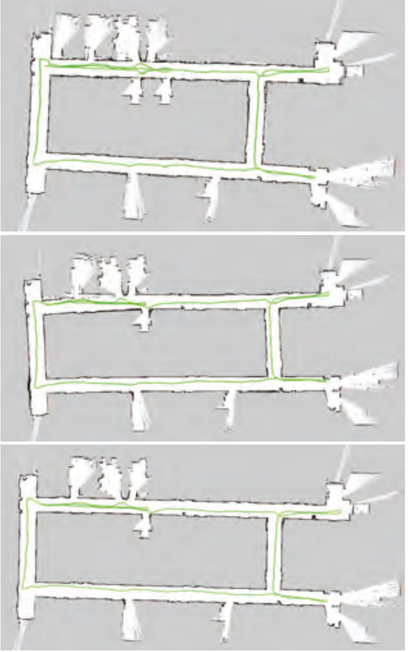


Fig. 5: 2D Gridmap Comparisons Top to Bottom: GMapping, GO and iGO

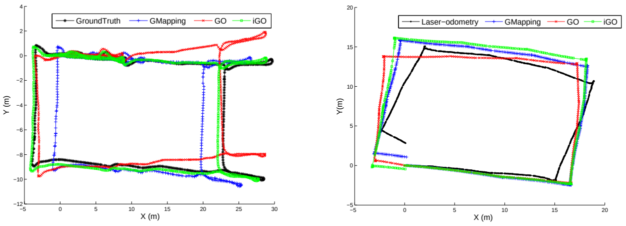


Fig. 6: Trajectory Comparison. Left: GroundTruth, GMapping, GO and iGO; Right: Laser-odometry, GMapping, GO and iGO

V. EXPERIMENT

To demonstrate the proposed method can recover from biased motion estimation, we conducted two experiments using the data from the real environment in two cases:

- When slippage occurs, motion estimation model contains high uncertainty.
- When applying laser-odometry, accuracy of motion estimation varies heavily.

We compare the results via three SLAM methods: GMapping [11], GO [13], and iGO, the proposed method. The only difference between GO and iGO lies in the optimization process in the back-end, while the process in the front-end is the same.

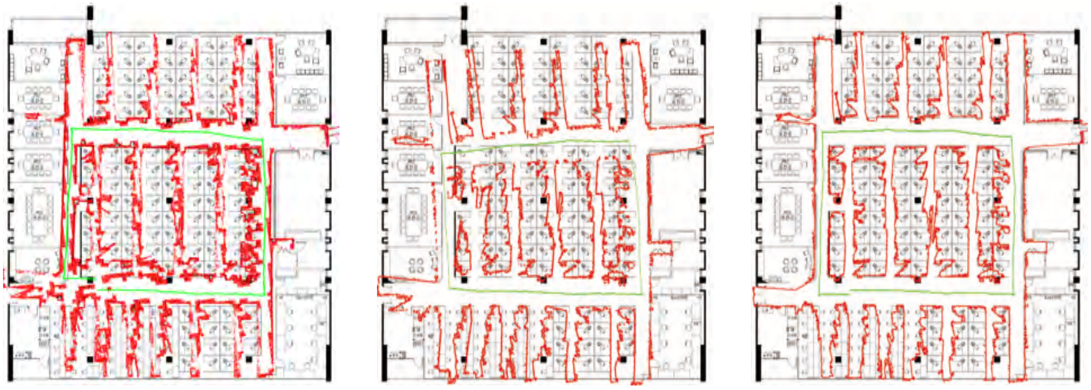
A. experiment I

In the first experiment, we use the uscsal data from the Radish [1]. To simulate vehicle slippage, we increase the motion model covariance Σ_{odo} with $\Sigma_{tt} = 1.6$ and $\Sigma_{rr} = 0.8$. In this case, for GMapping, the weight of particles jumps and resampling occurs which can eliminate some particles with true motion. As shown in the first row of the Figure 5 and the blue cross line in the left part of the Figure 6, the total trajectory shrinks and results in inconsistent 2D gridmap. For GO, some edges along the corridors may contain biased information. Therefore, depicted in the second row of the Figure 5, after closing loops and optimization, the whole graph can converges into a worse solution. Compared to the groundtruth, the result of GO bears large angular error shown in the left part of the Figure 6. In comparison, our approach can maintain information in the well estimated edges and improve the biased edges. As depicted in the third row in the Figure 5 and the green square line in the left part of the Figure 6, the result of our method is more consistent and accurate.

B. experiment II

In the second experiment, we control the vehicle traverse along a circle in the Lenovo B2 office which is about 17m width and 22m length, and the total length of the trajectory is about 80 meters. We use laser-odometry to predict the realtive motion. Therefore the covariance of the motion estimation depends highly on the scenes it traverse. For example, laser-odometry provide less accurate position information in the corridor-like place than other places with rich observation. The trajectory and map comparisons under different methods are shown in the Figure 7 and the right part of the Figure 6.

As shown in the right part of the Figure 6, laser-odometry [17] offers erroneous transformation in the corridor-like places. However, since the laser-scans are matched well along the corridor, it estimates high confidence information matrix. Because of this biased information, GMapping fails to correctly estimate the poses of the particles and result in inconsistent gridmap, Figure 7 (a). For the same reason, even GO can accurately close the loop, the optimization degenerates the whole vehicle trajectory and constructs a worse 2D map Figure 7 (b). On the contrary, our method can



(a) GMapping (b) GO (c) iGO

Fig. 7: Mapping Comparison: the red dots stand for the final 2D grid map, green line is the trajectory and the background is the sketch 2D map for the Lenovo B2 office

not only detect the loop closure precisely, but also reduce the error propagation introduced by the biased edges and improves the transformation in the biased edges. Compared with the sketch map of the work station, the gridmap of the proposed method is more consistent and accurate than those of the other methods, depicted in the Figure 7 (c).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce a new iterative graph optimization method. Our major contributions are two folds: a) iteratively rebuilding and optimizing the graph structure can maintain the well estimated edges, and improve the biased edges; b) a novel key-node mechanism and an interpolation algorithm that can help the loop-closure. The effectiveness of the proposed new ideas have been verified in our experiments.

Although the proposed method has many advantages compared with the previous methods, it can not recover from errors propagated by the false loop edges. A single large misleading loop closure could make the algorithm fail. In the future, we can combine the loop validation algorithms with the proposed method to make the graph-based SLAM method more robust.

REFERENCES

[1] http://cres.usc.edu/radishrepository/view-all.php.

[2] O. Bengtsson and A. Baerveldt. Robot localization based on scan-matchingłestimating the covariance matrix for the idc algorithm. *Robotics and Autonomous Systems*, 44(1):29–40, 2003.

[3] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[4] J. Blanco, J. Fernández-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological slam. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2061–2067. IEEE, 2007.

[5] M. Bosse. *Atlas: a framework for large scale automated mapping and localization*. PhD thesis, Massachusetts Institute of Technology, 2004.

[6] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691, 2008.

[7] A. Censi. An icp variant using a point-to-line metric. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 19–25. IEEE, 2008.

[8] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical slam: Real-time accurate mapping of large environments. *Robotics, IEEE Transactions on*, 21(4):588–596, 2005.

[9] G. Grisetti, R. Ku?mmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43, 2010.

[10] 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.

[11] G. Grisettiyz, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2432–2437. IEEE, 2005.

[12] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999. CIRA'99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325. IEEE, 1999.

[13] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

[14] Y. Latif, C. C. Lerma, and J. Neira. Robust loop closing over time. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[15] J. Min, J. Kim, S. Shin, and I.-S. Kweon. Data-driven mcmc sampling for vision-based 6d slam. *Electronics letters*, 48(12):687–689, 2012.

[16] J. Neira and J. D. Tards. Data association in stochastic mapping using the joint compatibility test. *IEEE T. Robotics and Automation*, 17(6):890–897, 2001.

[17] E. Olson. Real-time correlative scan matching. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4387–4393. IEEE, 2009.

[18] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[19] E. Olson, M. Walter, S. J. Teller, and J. J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In S. Thrun, G. S. Sukhatme, and S. Schaal, editors, *Robotics: Science and Systems*, pages 265–272. The MIT Press, 2005.

[20] N. Snderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *IROS*, pages 1879–1884. IEEE, 2012.

[21] S. Thrun, W. Burgard, and D. Fox. Probability robotics, 2005.

[22] G. D. Tipaldi, G. Grisetti, and W. Burgard. Approximate covariance estimation in graphical approaches to slam. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3460–3465. IEEE, 2007.

[23] H. Zhang, Z. Hou, N. Li, and S. Song. A graph-based hierarchical slam framework for large-scale mapping. *Intelligent Robotics and Applications*, pages 439–448, 2012.

Appearance-based Localization across Seasons in a Metric Map

Chris Beall, Frank Dellaert

Abstract— **In this paper we address the problem of appearance-based long-term outdoor localization across seasons. This is a difficult task due to the changing appearance of visual landmarks across seasons and time of day. Our approach operates based on the premise that combining visual landmarks observed at different times of the year into a single metric map will yield better localization results than a map created from a single sequence alone. We integrate stereo imagery collected at two different times of the year into a unified 3D map, and use this as the basis for localization. A landmark visibility prediction framework is utilized to efficiently retrieve a small subset of landmarks and their feature descriptors from a database of millions of landmarks. The proposed approach is experimentally validated on a challenging sequence collected a year earlier.**

I. Introduction

Vision-based localization systems have received much attention in the past few years. Localization using vision alone is an attractive prospect considering its very low cost compared to other sensor modalities. GPS is useful in many applications, but it is well known that GPS performance is degraded in urban settings due to buildings obstructing the sky. We are particularly interested in the scenario of localizing a moving vehicle, where a coarse localization estimate is available as a prior, either from GPS or from a localization estimate in the immediate past.

While quite a number of visual localization systems have been demonstrated, few have been shown to work robustly in the face of changing scene appearance caused by differences in lighting, seasonal variation, foliage changes, weather, etc. Representing each place as a different experience in a topologically connected map appears to be a particularly promising approach [1], but this sort of technique makes exact localization difficult as the query images are localized in several distinct visual odometry tracks.

In this paper we show that localization across long periods of time (and seasons) within a unified metric map is a feasible approach. We take the view that by combining data from several stereo image sequences into a single map it sufficiently spans the space of possible appearances to enable localization for a wide range of scenarios. This approach clearly presents a number of significant challenges. First, the sequences to be combined into the map must be registered very accurately to ensure the resulting map is geometrically consistent. Since the map contains millions of landmarks, the second challenge is how to decide which landmarks to choose when attempting to localize a query frame.

To the best of our knowledge, this is the first work which explicitly joins data from two sequences into a single metric map as shown in Fig. 1, which is then used for localization. In contrast, previous work which made use of data from different times was topological in nature. The contributions of this paper are:

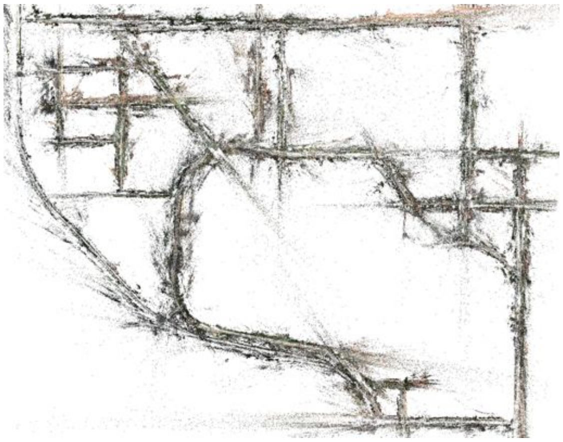


Fig. 1: Color point cloud representing the landmark database of the Georgia Tech campus.

- Vision-only localization in a large-scale metric map created from data collected during different times.
- Landmark visibility prediction in the context of real-time vehicle localization.

The remainder of this paper is organized as follows. We first discuss related work in section II, followed by a detailed discussion of the 3D map building and localization in section III. Section IV has the results.

II. Related Work

In recent years, many vision-based localization algorithms have been proposed. The work most relevant in terms of its application is that of Churchill et. al. [1]. Visual odometry trajectories, termed experiences, are stored each time the vehicle visits a new place and is unable to relocalize itself within already existing experiences. The system keeps collecting new experiences until they become fully adequate for localization. One disadvantage of this work is that these experiences are only topologically linked, and exact metric pose recovery presents a challenge.

Another interesting approach is that of Lategahn et al., who used a pre-computed 3D map, comprising 3D landmarks and their descriptors, to localize a stereo camera without GPS [2]. Given the previous known pose, all landmarks observed by the nearest camera pose used to build the map are used for descriptor matching. Lategahn et al. took a similar approach in [3], with the notable differences being that a monocular camera is used during localization, and the resulting pose is refined in a filter together with IMU measurements.

Milford and Wyeth [4] introduced SeqSLAM. Rather than matching local features between images, sequences of images are compared to establish a loop closure. Image similarity is established using sum of absolute differences. Consequently, no lighting/season invariant descriptors are needed. The method works on sequences with drastically different appearance. The method makes assumptions about relatively constant velocity and direction of travel. A related approach is that of Maddern et al. [5], in a system called CAT-SLAM. Sequential appearance based SLAM is enhanced with metric pose filtering to improve the performance.

Valgren et al. [6] also explored an appearance-based approach across scenes with stark appearance changes, using SIFT/SURF descriptor matching, and tuning the parameters for optimal results.

Deciding which map features to match against is a major challenge, and this is especially true in the case of Structure from Motion (SfM), where unordered datasets with mostly unknown location priors are the norm. Li et al. [7] addressed this difficulty by matching 3D points to image features, rather than the more conventional 2D to 3D matching. Points with higher degree are prioritized. This was further improved upon with bi-directional matching in [8]. A similar approach is taken by Sattler et al. [9], where 2D-3D matching is sped up by indexing all image features into a vocabulary tree that was constructed using the 3D model, and the size of each word cluster is used as a proxy for estimated matching speed. Feature matching is prioritized according to cluster sizes. In [10] this approach is further refined with an active correspondence search in both directions.

Another interesting line of attack is reasoning about descriptor occurrence. One such approach is taken in [11], [12] where robust localization is achieved by computing landmark observation likelihoods based on the number of times a landmark was observed across training runs.

It is standard practice to employ a RANSAC [13] framework to achieve robust matching in the presence of outliers. When inlier ratios become very low RANSAC can take many iterations to find a good model. Chum et al. introduced PROSAC [14] , which progressively increases the sample size. This approach assumes that matches can be prioritized, and in the usual case the descriptor distance is suitable. In [15], [16] feature weighting is integrated into the geometric verification procedure (as opposed to post-processing step).

A different approach to solving the data association problem is taken in [17]. The authors proposed a framework for predicting the visibility of landmarks in the scene. Given a new query image with a pose prior, the landmarks which were previously observed by nearby cameras are probabilistically weighted according to a distance metric which is learned in an offline step. The distance metric takes into account camera rotation and translation. This makes it easy to ignore landmarks which were observed by a camera facing in the opposite direction, even though they are very close to the query camera prior. In this paper we are also interested in localizing a query image given a pose prior, and we adopt this same visibility approach for efficiently retrieving likely visible landmarks from our map.

III. Map Building

In this section we describe how we build a map (3D landmark database) which is used for localization. The main steps consist of applying stereo visual odometry to an image sequence, loop closing within and between data sequences, and large scale bundle adjustment. Each of these will be discussed in detail, but first some notation: We define X^s as the set of camera poses $\{x_i^s\}$ for data sequence s. L^s is the set of landmarks $\{l_j^s\}$ observed in sequence s. $\theta^s \hat{=} \{X^s, L^s\}$ is the set of all variables, which together with a camera-landmark visibility table makes up the map M.

A. Stereo Visual Odometry

We run a conventional stereo visual odometry (VO) algorithm to recover the camera trajectory. For each rectified stereo image pair, SIFT features are extracted and matched across the pair. Matches are only retained if they are mutually optimal according to the ratio test [18], and fall within tight threshold of the epipolar line, which is a horizontal scan-line for rectified images. Points with zero disparity are discarded, and 3D points $(X, Y, Z)^T$are then triangulated. Features are then matched temporally to form a set of putative matches, and a three point algorithm [19] is employed in a RANSAC [13] framework to recover the relative pose.

Features which are successfully tracked for at least two consecutive frames, called feature tracklets, are recorded along with their feature descriptors. As these feature tracklets are geometrically consistent across at least two frames they will be accepted for inclusion in the map. The resulting camera trajectory, together with the accepted landmarks will be optimized later as described in the following sections.

B. Closing the Loop

Loop closures are needed to correct for drift in the VO trajectory, as well as to precisely align multiple passes along the same street. Appearance based loop closure detection as in [20] is a popular approach. However, since the data used to build the map has synchronized GPS, we use this to find loop closure candidates. We are not concerned about real-time performance while constructing the map. In a brute force fashion, we find the nearest neighbor camera poses and attempt feature matching and geometric verification as in Sec. III-A. Loop closure landmark observations are recorded to be incorporated into the map (Sec. III-C).

Loop closure detection is also performed between data sequences to provide constraints to align datasets with respect to each other.

C. Map Optimization

Bundle adjustment, or smoothing and mapping (SAM), has been applied to create highly accurate, city-scale reconstructions from large photo-collections[21], [22]. We apply this technique to optimize several data sequences together into a geometrically consistent map.

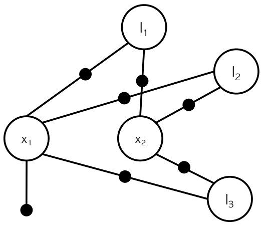


Fig. 2: Factor graph comprising two camera poses, three landmarks, and a GPS prior on camera pose x_1.

The optimization problem at hand is easily represented by a factor graph. A factor graph is a bipartite graph comprising two types of nodes: state variables and factors. Here, the unknown camera poses $X = \{x_i | \, i \in 1...M\}$ and landmarks $L = \{l_j | \, j \in 1...N\}$ make up the set of state variables. The landmark measurements $Z = \{z_k | \, k \in 1...K\}$ as observed by the cameras correspond to factors. An example of a factor graph is shown in Fig. 2.

We minimize the non-linear cost function

$$\sum_{k=1}^{K} \|h_k(x_{i_k}, l_{j_k}) - z_k\|^2_{\Sigma_k} \qquad (1)$$

in a least-squares sense, where $h_k(.)$ is the measurement function of landmark l_j from camera x_i, and the notation $\|.\|^2_{\Sigma}$ represents the squared Mahalanobis distance with covariance Σ. We assume that we have normally distributed Gaussian measurement noise.

For more details on the SAM optimization process, we refer the interested reader to [23].

D. Localization

Given a set of measurements Z_i and the map M, we are interested in efficiently recovering the most likely pose Θ: $P(\Theta|Z_i, M)$. In the case of vehicle localization we also assume that we have a pose prior that comes from the previous pose estimate or GPS. In light of M having many millions of landmarks, it is important to only retrieve landmarks which are likely to be visible in the current stereo frame. We use the visibility prediction framework introduced in [17] to achieve this. The key idea here is that stereo frames which were taken at camera poses X which were nearby the current pose, and also facing in roughly the same direction, are likely to have observed a similar set of landmarks L_v.

The landmark visibility distance metric used in this paper combines Euclidean distance and rotation between the query pose and map poses X. To find the set L_v we compute the distance between the query pose and all poses X, and then collect all of the landmarks observed by the n nearest poses. One important advantage of this approach is that map landmarks observed from a map-building sequence X^s where the vehicle was traveling in the opposite direction along the same road will not be considered visible, which is in accordance with the limits of rotation invariance of the SIFT descriptor.

Date	Frames	VO Fr.	Resolution	Length	Label
Sep 11, 2012	25462	20372	1380 × 480	10.5km	F
Apr 2, 2013	23090	14053	1384 × 680	11.38km	K
Aug 1, 2013	21690	15219	1384 × 680	13.21km	L

TABLE I: Three datasets that were used for the experiments.

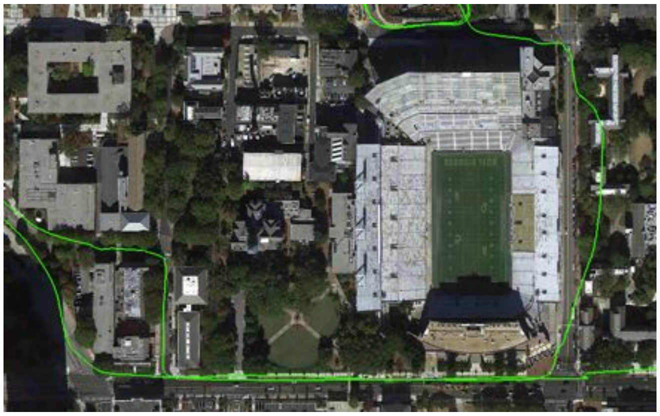


Fig. 3: GPS-INS trajectory superimposed on Google Earth imagery. Severe GPS drift due to multi-path issues can be observed to the east of the stadium.

Given L_v, the standard approach is followed to compute a pose estimate: Detect features in the current stereo pair, match and verify with RANSAC.

In practice, some steps can be taken to further speed up the algorithm described above. Computing the visibility distance metric with respect to all poses X can be costly for large M. Instead, we make use of a quad-tree to pre-prune the set of poses, and only compute the visibility for poses that fall within a bounding box of the query pose.

IV. Experimental Results

To validate our approach we have built a map using two data sequences collected on our campus. One sequence was collected in April, and the other in August of 2013, called sequences K and L. Sequence F is not included in the map, and is used for localization testing only. A listing of all the data sequences used in this paper is shown in table I. Images were collected using two Point Grey Flea 3 GigE cameras, along with a third color camera for visualization purposes. The cameras were triggered through hardware synchronization at 10Hz.

GPS-INS data was collected using a 3DM-GX3-45 GPS-Aided Inertial Navigation System at up to 100Hz. This data was interpolated and synchronized to camera timestamps. The GPS-INS solution occasionally drifts quite noticeably, particularly when driving next to large buildings which hinder a clear view of the sky in all directions. An example is shown in Fig. 3. Visual Odometry is run on each of the sequences, and feature tracklets, as well as their associated descriptors, are saved for the loop closure step.

A. Closing the Loop

As described in Sec. III-B, loop closure detection is performed within each sequence, as well as between the

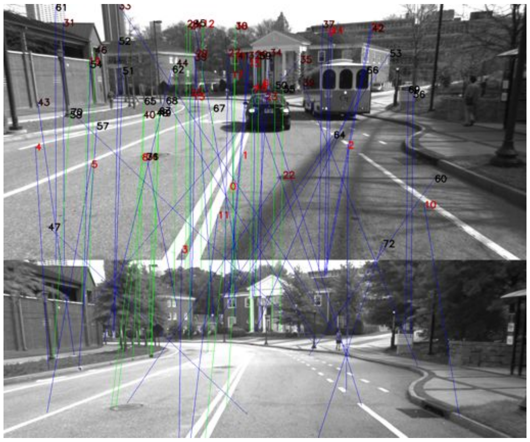
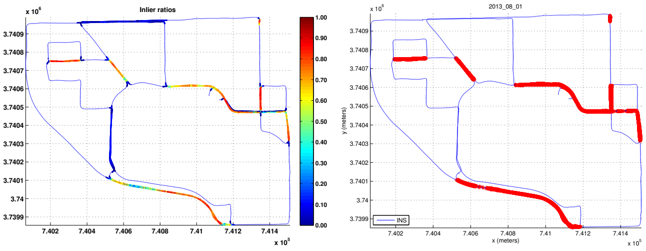


Fig. 4: Successful registration and pose recovery on challenging imagery between frames from sequences K (top) and L (bottom). There are notable differences in lighting, foliage, as well as vehicular occlusions. Putative matches are shown in blue, and accepted inlier matches are shown in green.



(a)

(b)

Fig. 5: Loop closure results for sequence L. a: Inlier ratios b) Accepted loop closures exceeding the inlier threshold and minimum inlier count are shown in red.

two sequences making up the 3D map. Fig. 4 shows a successful loop closure. The goal is to detect as many loop closures as possible as this promises the most accurate map registration possible. Missed loop closures lead to poor map alignment, while false loop closures present difficulties during optimization. Through empirical experimentation we find that a RANSAC inlier ratio of 0.5, and a minimum inlier count of 10 yield satisfactory results. Fig. 5 shows loop closure results for sequence L.

Fig. 6 shows the loop closure result between sequences K and L. As expected, there are no loop closures where the two trajectories do not overlap, but loop closures are also missed in some places, likely due to vastly different appearance, or due to the RANSAC inlier ratio not meeting the required threshold.

B. Map Optimization

Each sequence is optimized individually before all data are combined into a single map. Camera poses X^s are initialized from GPS, and landmarks L^s are initialized from stereo triangulation. We additionally add weak GPS priors to camera poses so the map remains in true alignment with the streets. The Huber cost function is used to achieve robustness against possible outliers. RMS projection errors per camera, before and after optimization, are shown in Fig. 7.

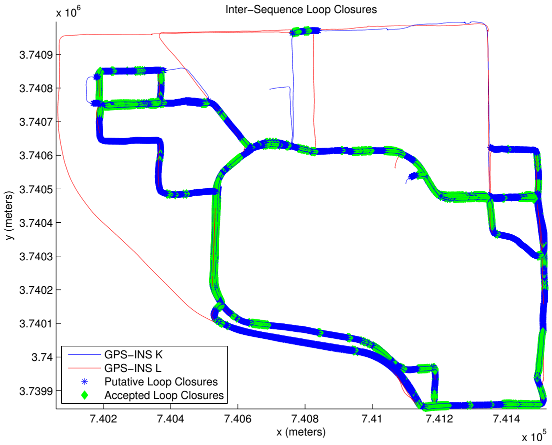
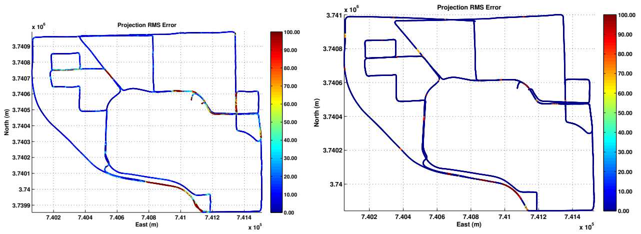


Fig. 6: Loop closures between sequences K & L. Poses where loop closure is possible are shown in blue, and where loop closure was successful is shown in green.



(a)

(b)

Fig. 7: Sequence L. a) Per-camera RMS errors before optimization b) RMS errors after optimization

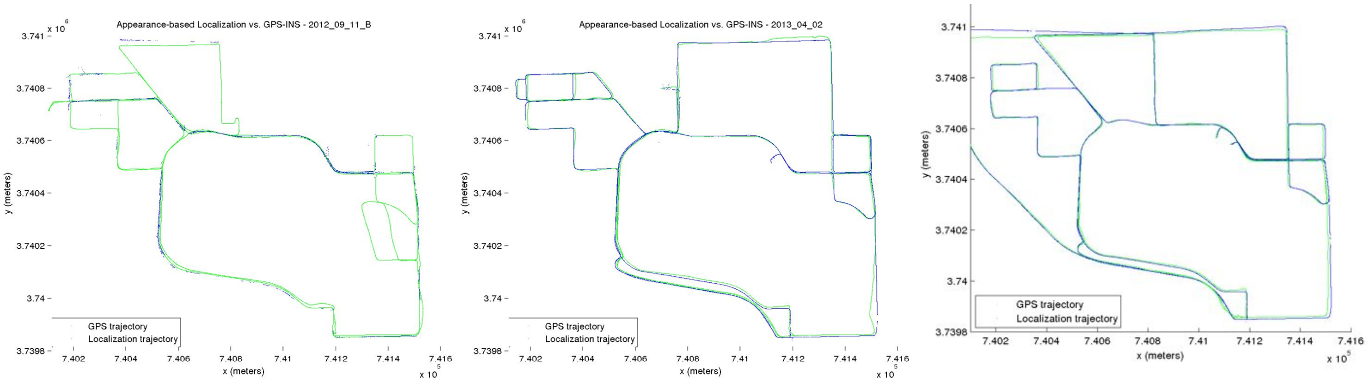
Finally, the two optimized sequences are combined, and landmarks which were observed in both sequences are represented as a single landmark. The final optimized camera trajectories are shown in Fig. 8.

To fully appreciate the structure of the 3D map, Fig. 9 shows a top-down view of all contained landmarks, with landmarks observed in sequences K and L shown in blue and green, respectively. Fig. 1 shows the color point cloud. The complete map, inclusive of feature descriptors has a size of approximately 1.4GB on disk.

C. Localization

We have conducted localization experiments for each of the three sequences, shown in Fig. 10. It is expected that sequences K & L will perform very well, as these contributed to the map. Sequence F, however, is a lot more challenging, since this sequence was taken in the previous year, and scene appearance was drastically different in many places across campus.

Fig. 11 shows a visualization of the smallest visibility distance for each query pose. The smaller the distance, the more likely the camera is to have observed the same



(a) Sequence F (b) Sequence K (c) Sequence L

Fig. 10: Localization results with KL map. Estimated poses are shown blue, GPS-INS priors are shown in green.

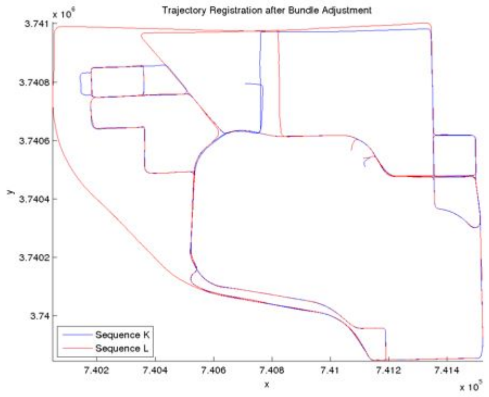


Fig. 8: Optimized camera trajectories after full bundle adjustment of over 12 million factors and over 2.2 million variables.

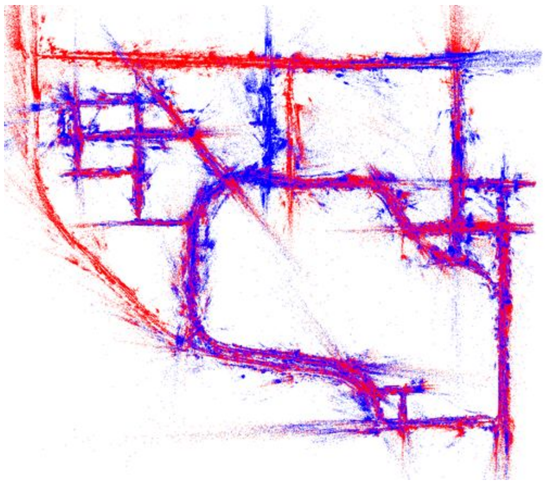


Fig. 9: Point cloud of tracked landmarks. Points shown in blue and red are from sequences K and L, respectively.

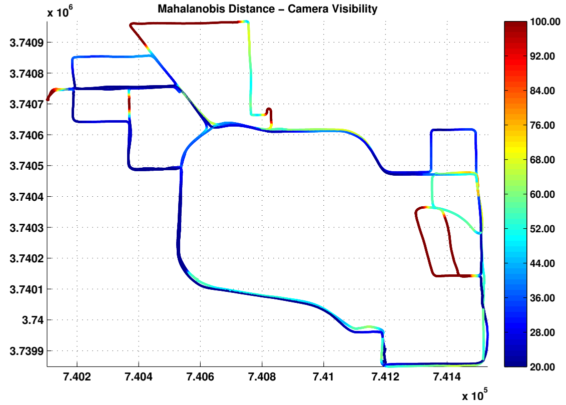
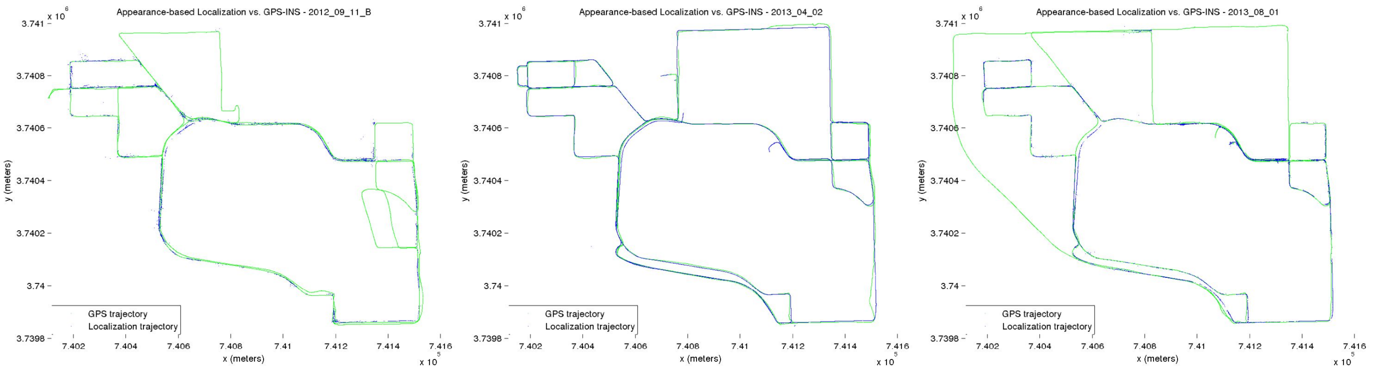


Fig. 11: Camera visibility scores (Mahalanobis distance) per GPS query pose for sequence F with respect to the full database. Lower (blue) is better. Streets which were not covered by the database, or which were traveled in the opposite direction have a large distance (red).

landmarks. For example, note that to the east there is a street block which was not covered in the map, and therefore has a very large visibility distance (deep red).

We have conducted the same experiments using only sequence K as the basis for the landmark map, and these results are shown in Fig. 12. As expected, the results for sequence K are virtually unchanged, and sequence L has gaps in localization where its trajectory does not overlap with K. Sequence F is relatively similar to the previous result, with the notable difference that localization was somewhat worse in areas where the two sequences K & L had poor loop closures. In other words, these were areas where there might exist alignment problems in the map. This underscores the need for very good registration when combining data from multiple sequences into a single metric map, and this is to be addressed in future work. Table II shows the localization performance of the three sequences with respect to a map constructed from sequence K alone vs. a map constructed from K+L.

The visual odometry component of our system runs faster



(a) Sequence F

(b) Sequence K

(c) Sequence L

Fig. 12: Localization results with K map. Estimated poses are shown blue, GPS-INS priors are shown in green.

Map	F	K	L	Total
K	5335	22969	10540	38844
K+L	4417	22819	21245	48481

TABLE II: Number of successfully localized frames of sequences F, K, L against maps created from sequence K alone, and from sequences K and L.

than real-time (10Hz). The performance of the localization module varies greatly, depending on the number of landmarks returned from the map, and depending on the inlier ratio. In the successful case it takes about 5-10ms, depending on the sequence (localizing K or L against the map is faster than F). When localization fails it can take up to hundreds of ms, dependent on RANSAC termination thresholds. However, these results were obtained with unoptimized code, and localization of individual image frames is easily parallelizable.

Conclusion

In this paper we presented a robust localization system based on a unified metric landmark map created from two stereo sequences collected at different times of the year. Efficient vision-based localization was performed by relying on a visibility prediction framework to retrieve a subset of landmarks which are used for descriptor matching. Experiments on real data showed the effectiveness of the approach. In future work we plan to incorporate more datasets into the map, and extending the visibility prediction framework to handle seasonal appearance explicitly.

References

[1] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[2] H. Lategahn and C. Stiller, "City gps using stereo vision," *ICVES*, 2012.

[3] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 719–724.

[4] M. Milford and G. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, may 2012, pp. 1643 –1649.

[5] W. Maddern, M. Milford, and G. Wyeth, "Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 429–451, 2012.

[6] C. Valgren and A. J. Lilienthal, "Sift, surf & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, 2010.

[7] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," pp. 791–804, 2010.

[8] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," pp. 15–29, 2012.

[9] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 667–674.

[10] ——, "Improving image-based localization by active correspondence search," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 752–765.

[11] E. Johns and G.-Z. Yang, "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *Proc. ICRA*, 2013.

[12] ——, "Dynamic scene models for incremental, long-term, appearance-based localisation," in *Proc. ICRA*, 2013.

[13] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[14] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[15] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 500–513.

[16] R. Raguram, J. Tighe, and J.-M. Frahm, "Improved geometric verification for large scale landmark image collections." in *BMVC*, 2012, pp. 1–11.

[17] P. F. Alcantarilla, K. Ni, L. M. Bergasa, and F. Dellaert, "Visibility learning for large-scale urban environment," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011. [Online]. Available: http://frank.dellaert.com/pub/Alcantarilla11icra.pdf

[18] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[20] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *Intl. J. of Robotics Research*, vol. 27, no. 6, pp. 647–665, June 2008.

[21] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *SIGGRAPH*, 2006, pp. 835–846.

[22] 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.

[23] F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.

High Precision 6DOF Vehicle Navigation in Urban Environments using a Low-cost Single-frequency GPS Receiver

Sheng Zhao Yiming Chen Jay A. Farrell

***Abstract*— Many advanced driver assistance systems (ADAS) demand for high precision navigation in urban environments. Traditional high precision dual-frequency RTK GPS receivers are too expensive for the low-cost, massive produced consumer-grade applications. On the other hand, many potential applications will become feasible as the high precision navigation solution becomes affordable using low-cost sensors. Hence, this paper proposed a high precision global navigation system using the low-cost single frequency GPS receiver and MEMS inertial measurement unit (IMU), with the application in GPS-challenged urban environments. By utilizing a sliding-window smoothing estimator, we are able to demonstrate reliable decimeter positioning accuracy in the presence of severe mutlipath errors and intermittent GPS signal receptions. To the best of the authors' knowledge, this is the first literature report of a high performance sliding window smoothing estimator on tightly coupled Differential-GPS/IMU using L1-only measurements in a GPS-challenged urban environment.**

I. Introduction and Related Work

High precision navigation is the core functionality in many advanced driver assistance systems, e.g. self-driving. In these systems, GPS is the primary sensor to obtain the global position of the vehicle. To achieve reliable high precision positioning, differential GPS (DGPS) is a promising approach. As the mobile communication networks (4G or WiFi) becomes ubiquitous, the DGPS technique can be used in most of the urban environments and provides 0.1 to 3 meter positioning accuracy [1].

However, GPS has its own limitations due to many factors. In the urban environments, the GPS signals can be blocked by trees and tall buildings and thus using GPS alone cannot obtain reliable and accurate navigation solutions. As an example of the poor GPS coverage in the urban environment, the satellite availability along the campus testing trajectory is shown in Fig. 1. Moreover, the traditional high-end dual frequency RTK GPS receiver is too expensive for low-cost/consumer-grade applications. In addition, in recent years, the single frequency (L1-only) GPS receivers have become readily available in the market at a much lower price than the dual frequency receiver, and the MEMS IMU is also getting much cheaper. Many potential applications will become feasible as the high precision navigation solution becomes affordable using low-cost sensors. Therefore, this paper proposes a high precision global navigation system using a low-cost single frequency GPS receiver and a MEMS IMU, with the application in GPS-challenged urban environments.

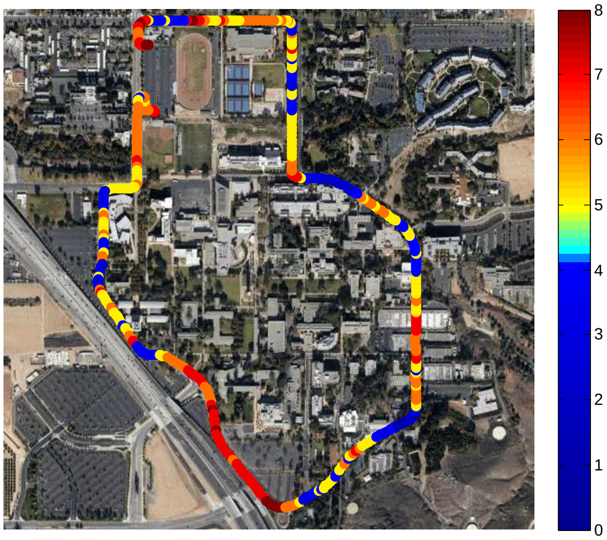


Fig. 1: The testing trajectory marked with colors to represent the number of satellites available to the receiver along the trajectory. Due to the blockage of building and trees, there are many places where less than 4 satellites can be seen (highlighted in blue color). This is a challenging route for the GPS/IMU navigation system. With the proposed method, the reliable decimeter positioning errors is achievable using L1-only GPS measurements.

In the GPS navigation community, RTK GPS positioning is a well-known mature technique. However, for low-cost, single frequency receiver, there still are many challenges, especially in urban environments. Compared to a dual frequency receiver, a single frequency receiver cannot form multi-frequency combinations which would greatly reduce the integer searching effort in realtime. Moreover, the number of measurements from single frequency receiver is half of that from a dual frequency receiver. In urban environments, the intermittent signal reception that caused by the signal blockage also creates problems because everytime the receiver reacquires the satellite signal, the integer in the carrier phase measurement is different. Therefore in practice, it is difficult or takes significantly longer for a single frequency receiver to resolve the integer and obtain the centimeter level accuracy compared to the high-end, dual frequency receivers [2]. As a result, the slow convergence rate of the positioning accuracy using a single frequency receiver is usually intolerable for many ADAS applications.

Without the correctly resolved integer, the phase measurement does not provide absolute range information. In this

Zhao and Chen are Ph.D. students and Farrell is a Professor at the Dept. of Electrical Eng., University of California, Riverside, 92521. {shzhao, yichen, farrell}@ee.ucr.edu.

case, most of the existing approaches use the triple difference technique in an EKF estimator [2], [3], [4]. However, the existing approaches either have slow convergence rate or assume perfect initialization of the estimator. In recent years, the smoothing based estimator have been demonstrated to have better performance than the EKF [5] and have been applied in many vision-aided inertial navigation systems [6], [7]. Hence, in our previous paper [8], the authors utilized a sliding-window smoothing estimator (which is named as the Contemplative RealTime (CRT) estimator) and demonstrated the superior performance to the EKF using pseudorange measurements in open sky environments. To further improve the performance of the CRT estimator in GPS-challenged urban environments, this paper proposes a novel way to utilize the accurate phase measurements in the CRT estimator when the integer cannot be resolved.

The contributions of this paper are:

1) The first literature report of a high performance sliding window smoothing estimator on tightly coupled DGPS/IMU using L1-only measurements in GPS-challenged urban environments.
2) The first literature report of utilizing the phase measurement in the sliding window smoothing estimator to achieve high precision navigation when the correct integer cannot be resolved.

II. Literature Review

For dual-frequency GPS receivers, the integer ambiguity problem is a well researched topic and there are many working solutions available [9], [10]. Once the integer is successfully resolved, an EKF is typically applied to obtain a centimeter accuracy navigation solution.

For a single-frequency GPS receiver, resolving the integer in realtime is much harder. There are many papers working on this topic by using triple difference technique, which differences the phase measurement at two consecutive times to eliminate the unknown integer. In [2], the authors designed an EKF to use the triple-difference phase measurement from a single frequency receiver to achieve submeter accuracy. The reported time needed for the estimator to converge to submeter accuracy is 500 sec and the vehicle needs to stay stationary during this period. In [3], the authors proposed an integer searching and validation method based on the technique developed in [2] with an application on a low-cost mowing robot using single frequency receivers. The time it takes to reliably resolve the integer was not reported. In [4], the authors utilize a modified triple difference technique in an EKF to track the relative positions of the receivers from a perfectly known initial configuration. The global positions are not estimated in their approach. Due to the nature of tracking (perfect initial knowledge), it does not need a long time to converge to a submeter accuracy. The reported accuracy of position tracking is in the decimeter level when the vehicle is driving in high speed.

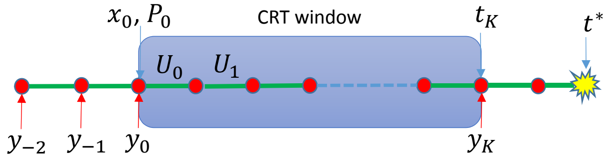


Fig. 2: The illustration of measurement time line and CRT window. The red dots represent the vehicle states at the GPS measurement time. The green lines represent the IMU constraints between two consecutive states. At t_k the optimization problem is formed and the solution of the optimization problem will be available at t^*. At t^*, the vehicle state is propagated from t_k to t^* using IMU measurements.

The method proposed in this paper is related to the triple difference technique. Instead of differencing consecutive measurements to eliminate the integer, we propose a systematic way to eliminate the integer for a time window of measurements and thus creates the constraints between all the vehicle poses in the window. The proposed method is optimal in that it preserves all the information from the measurements and correctly captures the time correlation of the resulted kinematic constraints in contrast to the triple difference technique.

To fully utilize the proposed method, this paper uses a sliding window smoothing estimator. Smoothing related algorithms are getting significant attention in the SLAM community in recent years [11], [12], [6], [7]. However, none of these papers report the performance for tightly coupled DGPS/INS. The most related one is [7]. However, in that paper the GPS is integrated in a loose coupled way. Therefore, they reported similar performance between smoothing and EKF. In a tightly coupled DGPS/INS system, we notice that there is a significant performance improvement compared to the EKF, especially in GPS-challenged urban environments. The navigation system designed in this paper is similar in concept to the one proposed in [6]. However, [6] focus on the visual inertial integration and they have not reported any tightly coupled GPS/IMU results. Based on the best knowledge of the authors, we believe this is the first publication of the high performance capability of smoothing estimator on tightly coupled DGPS/INS.

III. Nonlinear Least Square Problem

This section briefly describes the optimization problem in the CRT estimator. The details of the CRT estimator can be found in [6], [8].

The measurement time line of the CRT estimator is given in Fig. 2. The IMU measurements and GPS measurements are denoted as $\mathbf{U}_k$ and $\mathbf{y}_k$ respectively. At each CRT window (a sliding window of sensor measurements), we solve a nonlinear least square problem:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\{ \sum_{i\in\mathbb{S}} \|\mathbf{e}^i(\mathbf{X})\|_{\mathbf{R}^i} \right\} \qquad (1)$$

where $\mathbf{X}$ is the state vector of the vehicle trajectory defined on the CRT window, the set $\mathbb{S}$ represents all the information within the CRT window which, in our case, consists of the IMU, GPS and prior information, the $\mathbf{e}^i(\mathbf{X})$ is the residual function (IMU: $\mathbf{e}^i_\triangle$, GPS: $\mathbf{e}^i_\rho$ and $\mathbf{e}^i_\phi$, prior: $\mathbf{e}^i_0$), and $\mathbf{R}^i$ is the corresponding covariance matrix. The formulation of IMU and prior residual can be found in [8]. The GPS residuals will be discussed in the next section.

IV. Differential-GPS

DGPS has advantage over stand-alone GPS in that most of the common-mode errors (e.g., ionosphere, troposphere, satellite clock and ephemeris errors) can be removed by differencing measurements between the rover receiver and the GPS base station receiver. For simplicity of notation, it is assumed in this paper that DGPS approach completely removes all common-mode errors. Two types of measurements are provided by the receiver: pseudorange (code) and carrier phase measurements. To avoid the modeling of the receiver clock error which involves the complicated GPS measurement compensation, double differenced GPS measurements are considered in this paper.

A. Pseudorange Measurement

The double differenced pseudorange measurements for the i-th satellite vehicle (SV) can be modeled as

$$\rho^{ic} = \gamma^{ic} + mp^{ic}_\rho + n^{ic}_\rho, \tag{2}$$

where $\gamma^{ic}(t) = \bar{\gamma}^i(t) - \bar{\gamma}^c(t)$, $\rho^{ic} = \bar{\rho}^i - \bar{\rho}^c$, $n^{ic}_\rho = \bar{n}^i_\rho - \bar{n}^c_\rho$, $\bar{\gamma}^i(t) = \|\mathbf{p}_r(t) - \mathbf{p}^i_{sv}(t)\|_2$ is the geometric distance between the vehicle position $\mathbf{p}_r \in \mathbb{R}^3$ and the i-th SV position $\mathbf{p}^i_{sv} \in \mathbb{R}^3$, $\bar{\rho}^i$ is the pseudorange measurement of i-th SV, the superscript c is used to denote the common satellite chosen in the double differencing method, mp^{ic}_ρ is the multi-path error which can be several meters and $\bar{n}^i_\rho$ is the measurement noise whose standard deviation is typically around 2-5 meters.

Thus, the residual function of the double differenced code measurement can be formed from eqn. (2) as:

$$\mathbf{e}^i_\rho(\mathbf{x}(t)) = \rho^{ic} - \gamma^{ic}(t) \tag{3}$$

B. Carrier Phase Measurement

The double differenced phase measurement model for the i-th satellite vehicle (SV) is:

$$\lambda\phi^{ic} = \gamma^{ic} + \lambda N^{ic} + mp^{ic}_\phi + n^{ic}_\phi, \tag{4}$$

where $\phi^{ic} = \bar{\phi}^i - \bar{\phi}^c$ and $n^{ic}_\phi = \bar{n}^i_\phi - \bar{n}^c_\phi$, $\bar{\phi}^i$ is the phase measurement of i-th SV, N^{ic} is an unknown integer of the phase cycle, λ is the wavelength of the signal (L1 signal: 19.05cm), mp^{ic}_ϕ is the multi-path error which is in the centimeter level, and $\bar{n}^i_\phi$ is the measurement noise whose standard deviation is typically in the centimeter level. The integer N^{ic} is constant over time intervals when the receiver has phase lock for SV i and c. The receiver indicates this lock with a flag and lock time counter. Once the unknown integer N^{ic} is resolved, the phase measurement provides the range measurement in a centimeter accuracy.

However, the integer is difficult to resolve reliably in realtime for a single-frequency receiver. Nonetheless, there are at least two reasons why we still want to use the phase measurement when the integer cannot be resolved:

1) Multi-path only introduces a few centimeters of error in the phase measurement while the code measurement can be affected by few meters.
2) Phase measurements over a time window provide the local kinematic constraints of the trajectory at the centimeter accuracy even when the correct integer is not available.

Typically, there are two ways of utilizing the phase measurement when the correct integer cannot be resolved:

1) Use the triple difference technique to create an integer-free measurement. The triple difference measurement $\tilde{\boldsymbol{\phi}}^{ic}_k$ at t_k for i-th SV is defined as:

$$\lambda\tilde{\boldsymbol{\phi}}^{ic}_k = \gamma^{ic}_k - \gamma^{ic}_{k-1} + \tilde{n}^{ic}_\phi \tag{5}$$

where $\tilde{\boldsymbol{\phi}}^{ic}_k = \phi^{ic}(t_k) - \phi^{ic}(t_{k-1})$ and $\tilde{n}^{ic}_\phi = n^{ic}_\phi(t_k) - n^{ic}_\phi(t_{k-1})$. This equation is derived by subtracting eqn. (4) at t_{k-1} and t_k. The benefit of this approach is that the resulted measurement $\tilde{\boldsymbol{\phi}}^{ic}_k$ is independent of the integer and thus it does not violate the integer assumption. However, the triple difference only consider the consecutive measurements and ignore the time correlation of phase measurements over a time window.
2) Estimate the integer as a float number together with other vehicle states in the estimator. This approach correctly accounts for the time correlation of phase measurements. The major drawback of this approach is that the integer constraint is not respected and adding integers into the estimator will increases the computational complexity.

To incorporate the accurate phase measurements in the CRT framework, this paper proposes a new method that takes benefits from both the approaches mentioned above. Furthermore, as an improvement of the triple difference technique, this paper proposes a integer-free phase measurement that is independent of the integer while correctly captures the time correlation of phase measurements. As a nutshell, the CRT estimator uses the proposed integer-free measurements for i-SV when the i-SV is not observed by the first pose in the CRT window. Otherwise the integers are added into the estimator.

V. Integer-free Phase Measurement

To fully utilize the phase measurement without resolving the integer, an integer-free measurement is constructed using methods originally proposed for visual odometry [13]. Given all the measurements of the i-th SV and the common SV c that contributes to the constant integer N^{ic}, which are defined as an integer track Ξ^{ic} in this paper, stacking up eqn. (4) and ignore the multi-path error mp^{ic}_ϕ gives:

$$\lambda\boldsymbol{\phi}^{ic} = \mathbf{h}(\mathbf{X}_{\mathbb{S}^{ic}_p}) + \lambda\mathbf{G}^{ic}N^{ic} + \mathbf{n}^{ic}_\phi \tag{6}$$

where

$$\boldsymbol{\phi}^{ic} = \begin{bmatrix} \phi^{ic}(\bar{t}_1) \\ \phi^{ic}(\bar{t}_2) \\ \vdots \\ \phi^{ic}(\bar{t}_l) \end{bmatrix}, \quad \mathbf{h}(\mathbf{X}_{\mathbb{S}_p^{ic}}) = \begin{bmatrix} \gamma^{ic}(\bar{t}_1) \\ \gamma^{ic}(\bar{t}_2) \\ \vdots \\ \gamma^{ic}(\bar{t}_l) \end{bmatrix}, \mathbf{G}^{ic} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{7}$$

where $\mathbb{S}_p^{ic} = \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_l\}$ denotes a set of consecutive time steps that both the i-th SV and the common SV c are observed, and $\mathbf{X}_{\mathbb{S}_p^{ic}} = \{\mathbf{x}(t) \,|\, t \in \mathbb{S}_p^{ic}\}$. The set $\mathbb{S}_p^{ic}$ always contains consecutive time steps because every time the receiver reacquires the satellite, the integer changes.

Using the methods in [13], the integer can be eliminated from the equation by the following procedure. Define a unitary matrix $\mathbf{A} = [\mathbf{A}_1, \ \mathbf{A}_2]$ such that the columns of $\mathbf{A}_2$ form the basis of the left nullspace of $\mathbf{G}$ ($\mathbf{A}_2^\top \mathbf{G} = \mathbf{0}$). Multiplying $\mathbf{A}_2^\top$ on both sides of (6) gives:

$$\lambda \bar{\boldsymbol{\phi}}^{ic} = \bar{\mathbf{h}}(\mathbf{X}_{\mathbb{S}_p^{ic}}) + \bar{\mathbf{n}}_\phi^{ic} \tag{8}$$

where $\bar{\boldsymbol{\phi}}^{ic} = \mathbf{A}_2^\top \boldsymbol{\phi}^{ic}$, $\bar{\mathbf{h}} = \mathbf{A}_2^\top \mathbf{h}$ and $\bar{\mathbf{n}}_\phi^{ic} = \mathbf{A}_2^\top \mathbf{n}_\phi^{ic}$.

Thus, the integer-free phase measurement induced residual equation $\mathbf{e}_\phi(\mathbf{X})$ can be formed as:

$$\mathbf{e}_\phi^i(\mathbf{X}) = \lambda \bar{\boldsymbol{\phi}}^{ic} - \bar{\mathbf{h}}(\mathbf{X}_{\mathbb{S}_p^{ic}}) \tag{9}$$

Note that the above derived measurement equation is independent of the integer. Equation (9) expresses the relative kinematic constraints between vehicle poses along the trajectory. The constraint is strong because the noise $\mathbf{n}_\phi^{ic}(t)$ has a standard deviation at the centimeter level. Moreover, in the proposed approach, the noise $\bar{\mathbf{n}}_\phi^{ic}$ correctly captures the time correlation of the relative kinematic constraint between all the vehicle poses in the set $\mathbb{S}_p^{ic}$ through its dense covariance matrix $\mathbf{A}_2^\top \mathbf{R}_\phi^{ic} \mathbf{A}_2$, where $\mathbf{R}_\phi^{ic}$ is the covariance matrix of $\mathbf{n}_\phi^{ic}$. In contrast, the triple difference technique only captures the pairwise kinematic constraint between two consecutive vehicle poses, but neglects time correlation between subsequent measurements.

VI. Experimental Results

This section presents analysis of data accumulated during a test drive around the campus of University of California, Riverside, see Fig. 1. Along the test path there are many trees and buildings as is representative of a typical urban environment. In the experiment, the vehicle is equipped with dual-frequency GPS receivers and a MEMS IMU, but no form of compass. For this receiver, the L1 data is more accurate than a typical low-cost single frequency GPS receiver and antenna would produce; however, using identical data is most useful for the present analysis (generating ground truth). GPS measurements are taken at 1 Hz. The GPS provides code (pseudorange) and carrier phase measurements. All GPS measurements are used in a differential mode. The IMU provides measurements at 200Hz. The CRT estimator uses 10s window of data. The vehicle position is initialized by the GPS measurement and the pitch and roll are obtained from the accelerometer assuming the vehicle is stationary. The sensor platform is shown in Fig. 3.

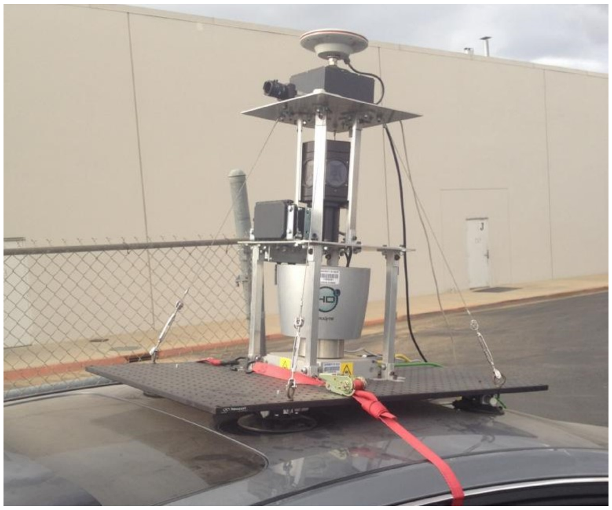


Fig. 3: Navigation and Mapping sensor platform. Equipped with GPS/INS unit, monocular camera, 360 degree camera, 2D LIDAR, RADAR, and Velodyne LIDAR.

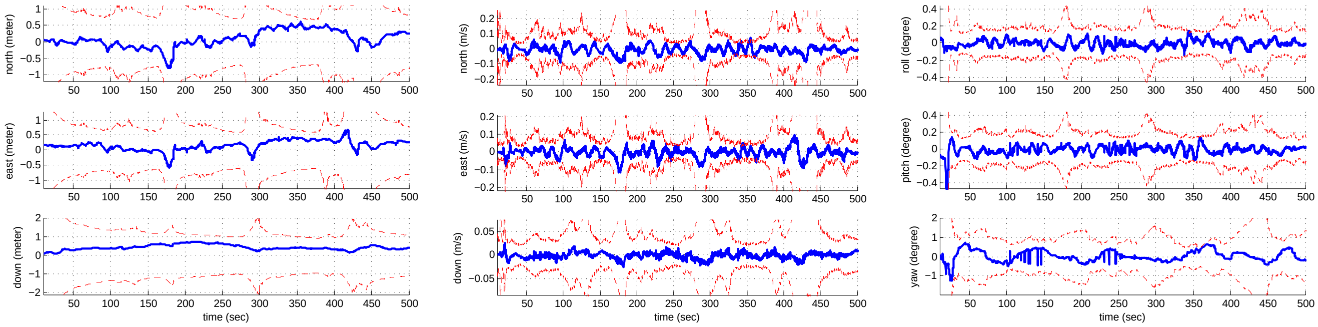
The trajectory estimation error is formulated by subtracting the real-time state estimate at each time from a ground truth state estimate for the same time. The ground truth trajectory is determined by an off-line, post-processed smoother combining the IMU and integer-resolved phase measurements (using L1 and L2 measurements) [14]. The ground truth trajectory is accurate at the centimeter level. All the CRT estimators used in the experiment use L1 only measurements.

Fig. 4 shows the position, velocity and attitude errors of the CRT estimator using code and phase. As we can see from the results, the position error is within $\pm 0.5m$ for most of the time in the horizontal plane (north and east direction), the velocity error is within $\pm 0.1m/s$ and the roll and pitch errors are within $\pm 0.2°$ and the yaw error is within $\pm 1°$ without using a compass.

To demonstrate the benefits of phase measurements, the position errors of the CRT estimator using code only are shown in Fig. 5 with the position errors of the CRT estimator using code and phase. We can see that the phase measurements are able to provide accurate local kinematic constraints that prevent large jumps in the position estimates in the presence of multi-path errors and noisy GPS signal receptions, even when the integer cannot be resolved. Moreover, the estimated trajectory using phase measurement is in general smoother than using the code measurements only.

VII. Conclusions and Future Work

This paper has proposed a novel DGPS/IMU navigation system that significantly improves performance in urban environments. The experimental results demonstrated that the proposed method has the potential to enable the high precision navigation using low-cost, single frequency GPS receivers and MEMS IMU in GPS challenged environments.



(a) Position error.

(b) Velocity error.

(c) Attitude error.

Fig. 4: Navigation system errors. The $\pm 3\sigma$ bound is plotted in red.

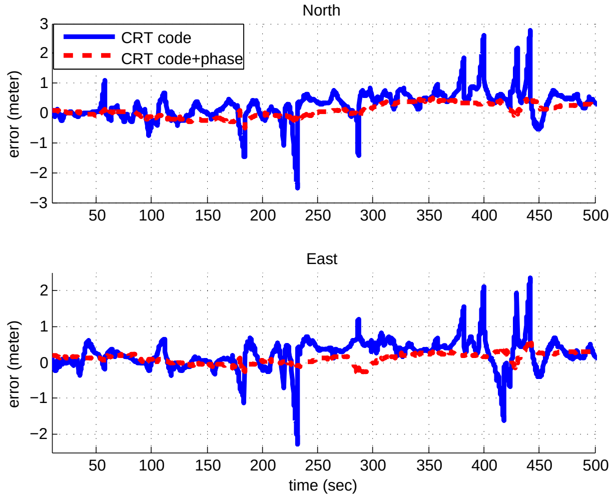


Fig. 5: North and East position errors comparison. Without using the phase measurements, the position errors have huge jumps in the presence of multi-path issues and noisy measurements when under trees.

The new algorithm performs optimization in realtime, for all IMU and GPS measurement within a time window, to provide a state estimate at the current time. The approach leads to improved performance for a few reasons. First, optimization over a time window provides the capability to re-linearize the system kinematic and measurement models around the improved trajectory estimate. This leads to the ability to estimate attitude and biases, especially yaw, accurately without a magnetometer. Second, the large set of measurement data provides sufficient redundancy to allow the effects of noise to be significantly reduced in the optimization. Third, the proposed integer-free phase measurement is able to provide accurate local kinematic constraints, without needing to resolve the integers, which helps to improve the robustness to multi-path errors and GPS noise, which are common in urban environments.

In the future, we plan to integrate the visual odometry into the existing GPS/IMU system to improve the performance. In addition, we are also working on the multi-path error modelling to correctly accounts for it.

VIII. Acknowledgements

This research builds on the software of and technical conversations with Anastasios I. Mourikis and Mingyang Li. These technical collaborations are greatly appreciated.

References

[1] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[2] J. Codol and A. Monin, “Improved triple difference GPS carrier phase for RTK-GPS positioning,” in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*. IEEE, 2011, pp. 61–64.

[3] J.-M. Codol, M. Poncelet, A. Monin, and M. Devy, “Safety robotic lawnmower with precise and low-cost L1-only RTK-GPS positioning,” in *Proceedings of IROS Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment, San Francisco, California, USA*, 2011.

[4] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi, “High-accuracy differential tracking of low-cost GPS receivers,” in *ACM MobiSys*, 2013.

[5] H. Strasdat, J. Montiel, and A. J. Davison, “Real-time monocular SLAM: Why filter?” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.

[6] H.-P. Chiu, S. Williams, F. Dellaert, S. Samarasekera, and R. Kumar, “Robust vision-aided navigation using sliding-window factor graphs,” in *ICRA*, 2013.

[7] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, “Factor graph based incremental smoothing in inertial navigation systems,” in *IEEE Information Fusion (FUSION)*, 2012, pp. 2154–2161.

[8] S. Zhao, Y. Chen, H. Zhang, and J. A. Farrell, “Differential GPS aided inertial navigation: a contemplative realtime approach,” in *IFAC World Congress*, 2014.

[9] T. Takasu and A. Yasuda, “Development of the low-cost RTK-GPS receiver with an open source program package rtklib,” in *International Symposium on GPS/GNSS, International Convention Center Jeju, Korea*, 2009.

[10] X.-W. Chang and T. Zhou, “MILES: MATLAB package for solving mixed integer least squares problems,” *GPS Solutions*, vol. 11, no. 4, pp. 289–294, 2007.

[11] F. Dellaert and M. Kaess, “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing,” *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[12] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental Smoothing and Mapping,” *IEEE Trans. Robotics,*, vol. 24, no. 6, pp. 1365–1378, 2008.

[13] A. Mourikis and S. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation,” in *IEEE ICRA*, 2007, pp. 3565–3572.

[14] A. Vu, J. Farrell, and M. Barth, “Centimeter-accuracy smoothed vehicle trajectory estimation,” *Intelligent Transportation Systems Magazine, IEEE*, vol. 5, no. 4, pp. 121–135, 2013.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session II

Perception

- **Keynote speaker: Chritoph Stiller (KIT, Karlsruhe, Germany)**
  Title: Situation Perception and Prediction for Autonomous Driving
  Co-Authors: Julius Ziegler, Markus Schreiber, Philipp Bender

- **Title: Detection and Tracking of the Vanishing Point on a Horizon for Automotive Applications**
  Authors: Young-Woo Seo, Ragunathan Rajkumar

- **Title: Robot Navigation Using Radio Signal in Wireless Sensor Networks**
  Authors: Ju Wang, Mohammad M Tabanjeh, Tariq Qazi, Brian Bennett, Cesar Flores-Montoya, Eric Glover, Meesha Rashidi

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session II

Keynote speaker: **Chritoph Stiller**
(KIT, Karlsruhe, Germany)

Situation Perception and Prediction for Autonomous Driving

Co-Authors: *Julius Ziegler, Markus Schreiber, Philipp Bender*

Abstract : Vehicle environment perception and to prediction of potential motion of relevant traffic participants is a crucial task for autonomous driving. The contribution of this presentation includes the following aspects. First, we extend the current state of the art in robotic vehicles that require expensive on-roof sensors and GNSS navigation. Instead, we propose signal processing methods that enable us to restrict ourselves to close-to-market sensor configurations. Dominated by vision sensors, we outline vehicle environment perception and scene prediction that enables us to automatically navigate through everyday's traffic. Methods for 3D visual machine perception based mono- and binocular video sensors are presented. The contribution of prior knowledge from digital maps is elaborated as well as its curse in case of erroneous information. Real-time automated decision-making and trajectory planning methods are outlined. Trajectory planning is not only conducted for our ego-trajectory, but we argue that planning of trajectories of other traffic participants is crucial for safe navigation. Extensive results of automated driving are shown in real world scenarios from our AnnieWAY vehicle, the winner of the 2011 Grand Cooperative Driving Challenge, and from the Bertha vehicle that drove autonomously on the Bertha Benz memorial route from Mannheim to Pforzheim through a highly populated area of Germany.

Biography: Christoph Stiller studied Electrical Engineering in Aachen, Germany and Trondheim, Norway, and received the diploma degree and the Dr.-Ing. degree (Ph.D.) from Aachen University of Technology in 1988 and 1994, respectively. He worked with INRS-Telecommunications in Montreal, Canada for a post-doctoral year as Member of the Scientific Staff in 1994/1995. In 1995 he joined the Corporate Research and Advanced Development of Robert Bosch GmbH, Germany. In 2001 he became chaired professor and director of the Institute for Measurement and Control Systems at Karlsruhe Institute of Technology, Germany. Dr. Stiller serves as immediate Past President of the IEEE Intelligent Transportation Systems Society, Associate Editor for the IEEE Transactions on Intelligent Transportation Systems (2004-ongoing), IEEE Transactions on Image Processing (1999-2003) and for the IEEE Intelligent Transportation Systems Magazine (2012-ongoing). He served as Editor-in-Chief of the IEEE Intelligent Transportation Systems Magazine (2009-2011). He has been program chair of the IEEE Intelligent Vehicles Symposium 2004 in Italy and General Chair of the IEEE Intelligent Vehicles Symposium 2011 in Germany. His automated driving team AnnieWAY has been finalist in the Darpa Urban Challenge 2007 and winner of the Grand Cooperative Driving Challenge in 2011.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Karlsruhe Institute of Technology

Situation Perception and Planning for Autonoumous Driving

Christoph Stiller & Julius Ziegler

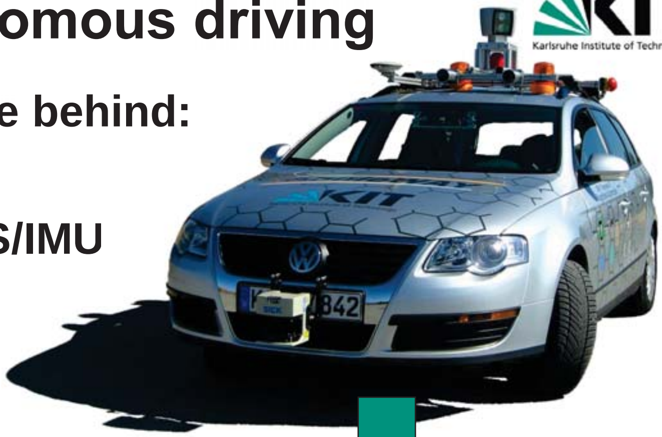
Institut für Mess- und Regelungstechnik

2014 IROS Workshop

Vision-based autonomous driving

Things we wanted to leave behind:

- On-roof sensor suite
- Highly accurate DGNSS/IMU
- High-end lidars



Instead:

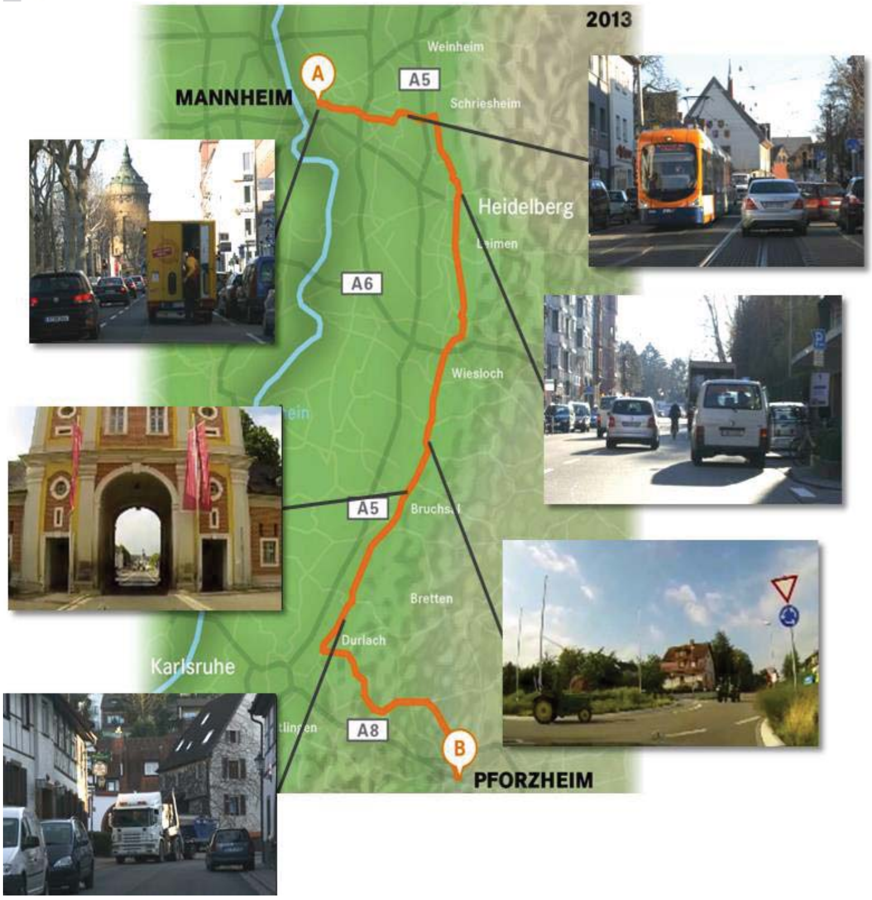
- Normal appearance
- Low cost cameras
- Low cost GPS/IMU
- (Pre-)series sensors
- Map-based

Bertha and Carl Benz ~ 1870

Das Brautpaar Berta Ringer und Karl Benz um 1870
Aus der Sammlung Eugen Benz, Ladenburg

1888 First long distance ride in an automobile by Bertha Benz and her two sons

Bertha Benz Memorial Route



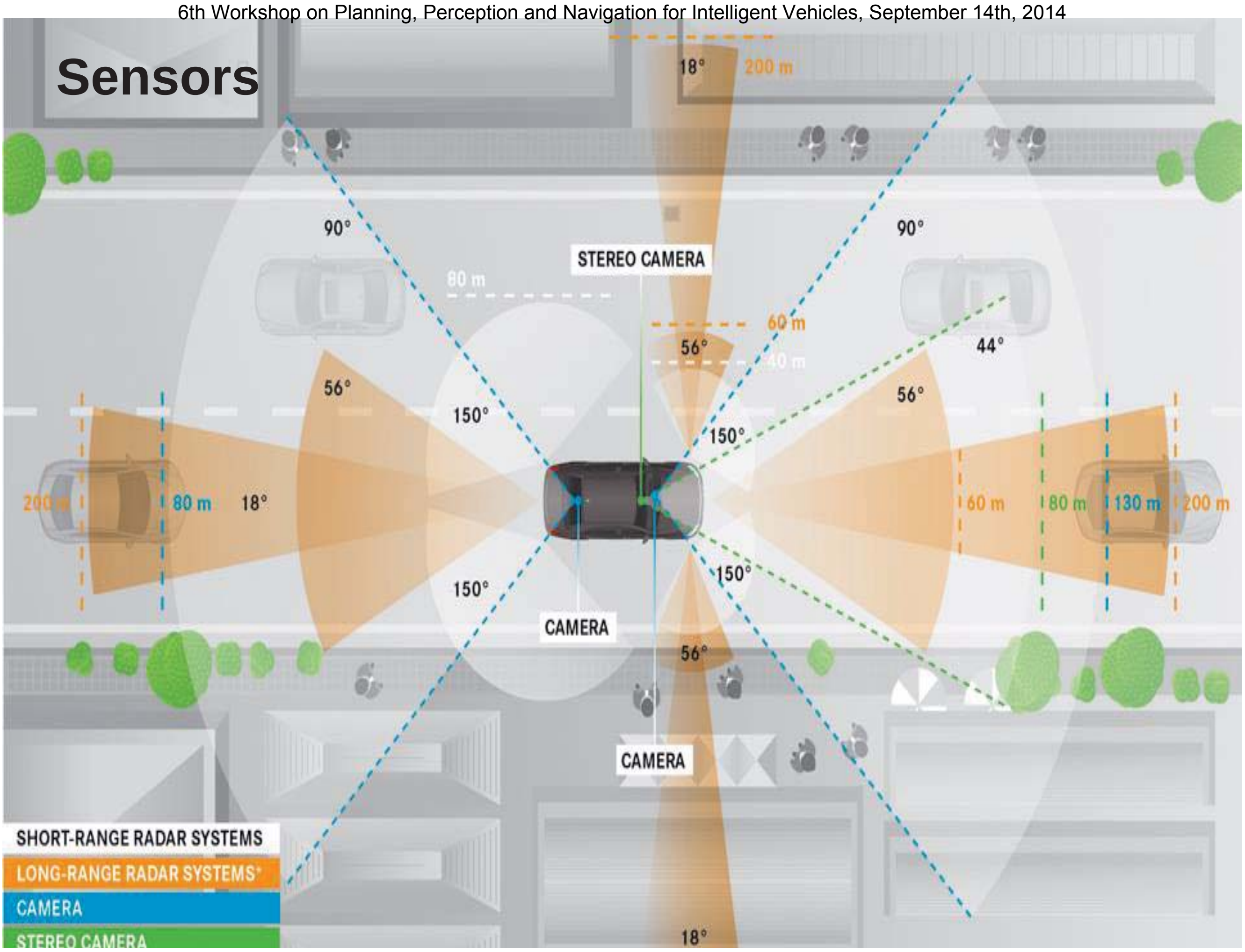
Bertha Benz Memorial Route

- first automotive long distance journey in 1888
- 104 km
- 3 large cities
- 23 smaller towns
- 18 roundabouts
- > 150 traffic lights

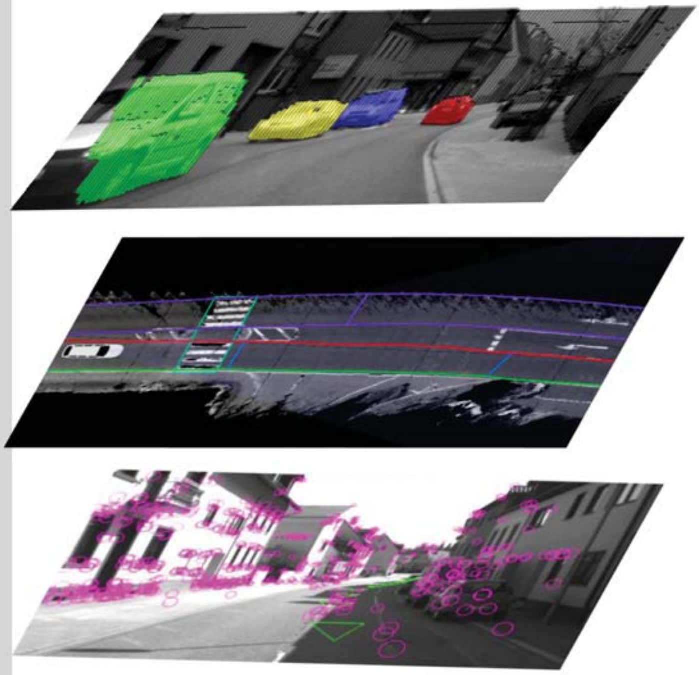
Major KIT/FZI Tasks

- Map generation
- Visual localization (KIT/FZI & Daimler)
- Behaviour decision
- Trajectory planning

Sensors



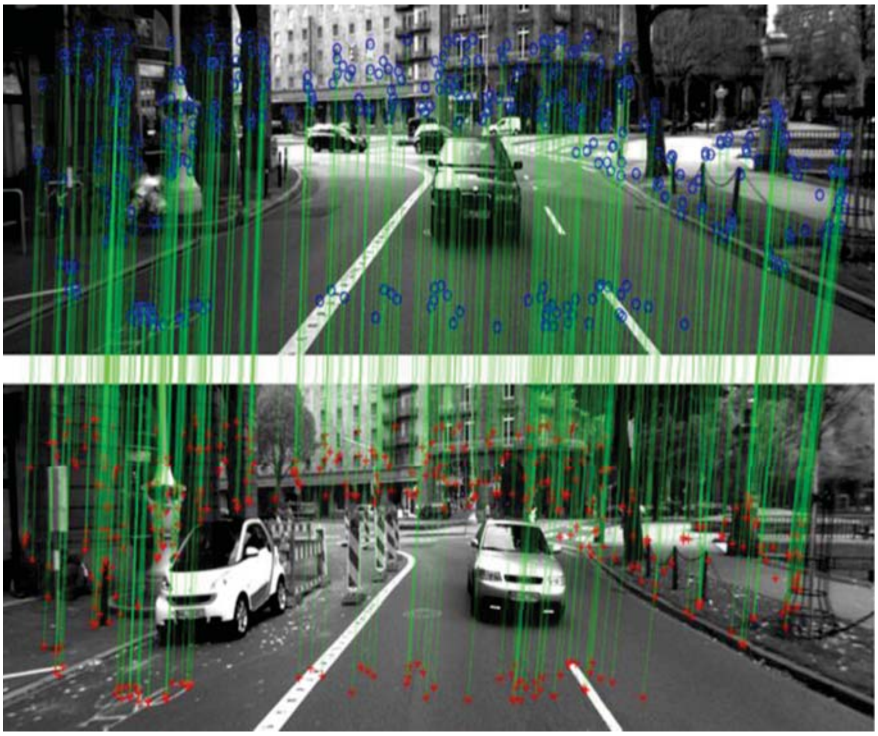
Map Layers



- Dynamic layer
  - dynamic objects
  - new static objects
- Static planning layer
  - 3d geometry, lanelets
  - traffic lights/rules
  - tactical information
- Localization layer
  - 3d landmarks
  - lane markers
  - 6d camera poses

[Lategahn, Bender, Schreiber, Franke et al. 11-14]

Visual Localization from Point Feature Matches



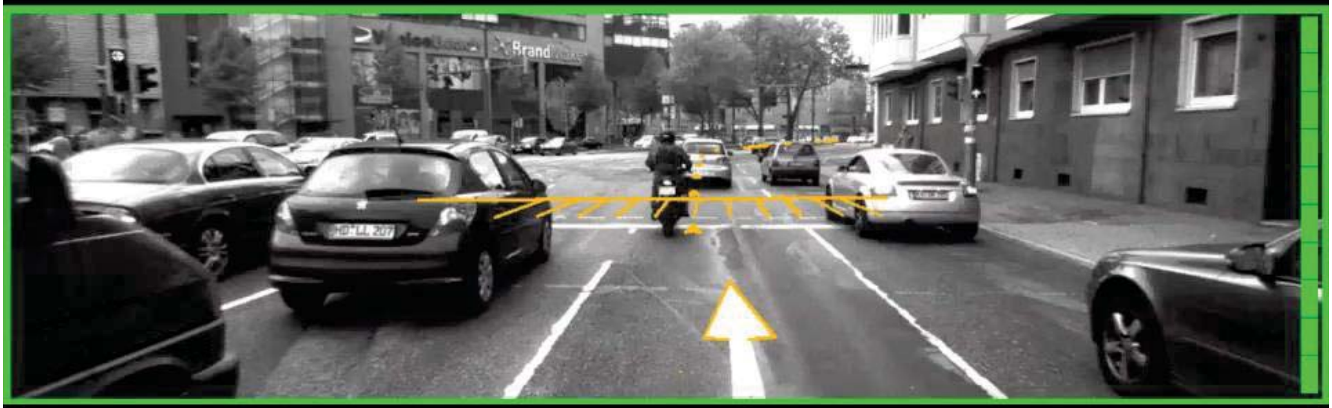
map features

R, t

image features

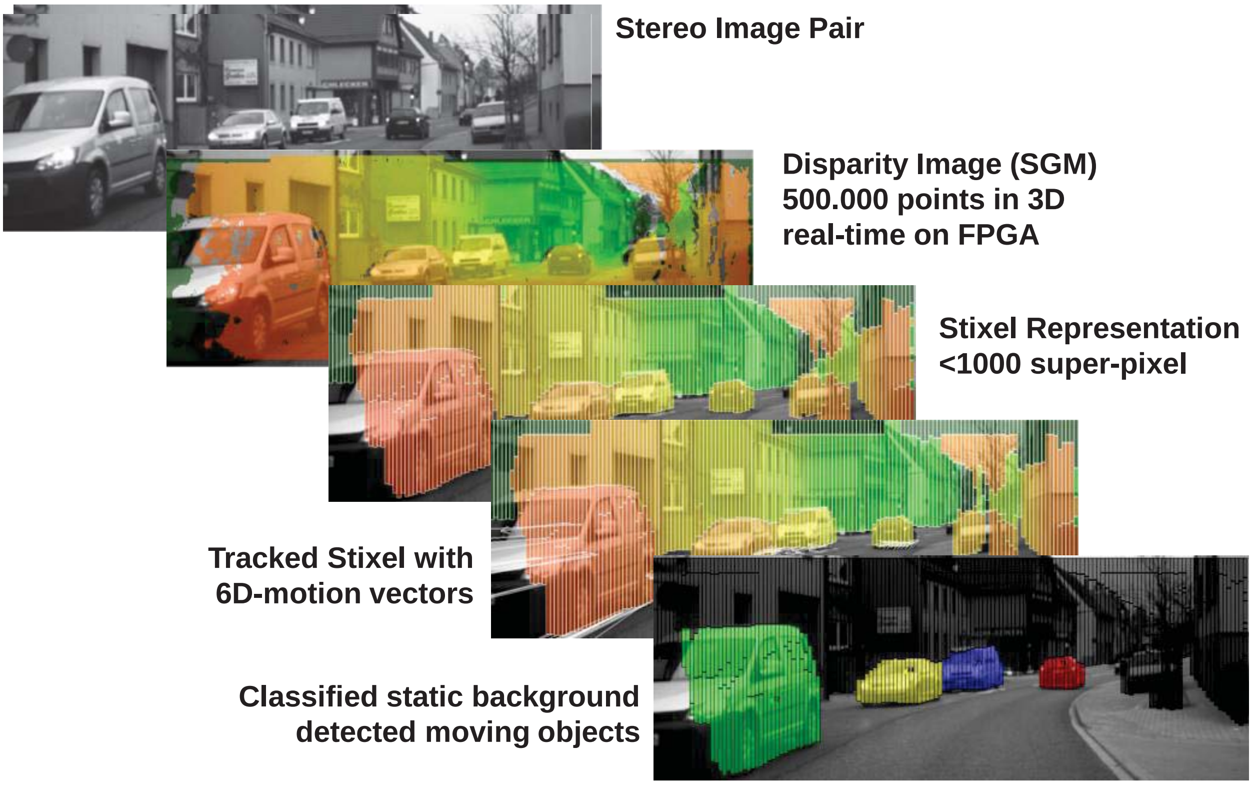
[Lategahn et al. 11-14]
start up company Atlatec UG

Localization



[Lategahn 2009-2013]
Now Atlatec UG

Front Stereo: Stixel Representation



Stereo Image Pair

Disparity Image (SGM)
500.000 points in 3D
real-time on FPGA

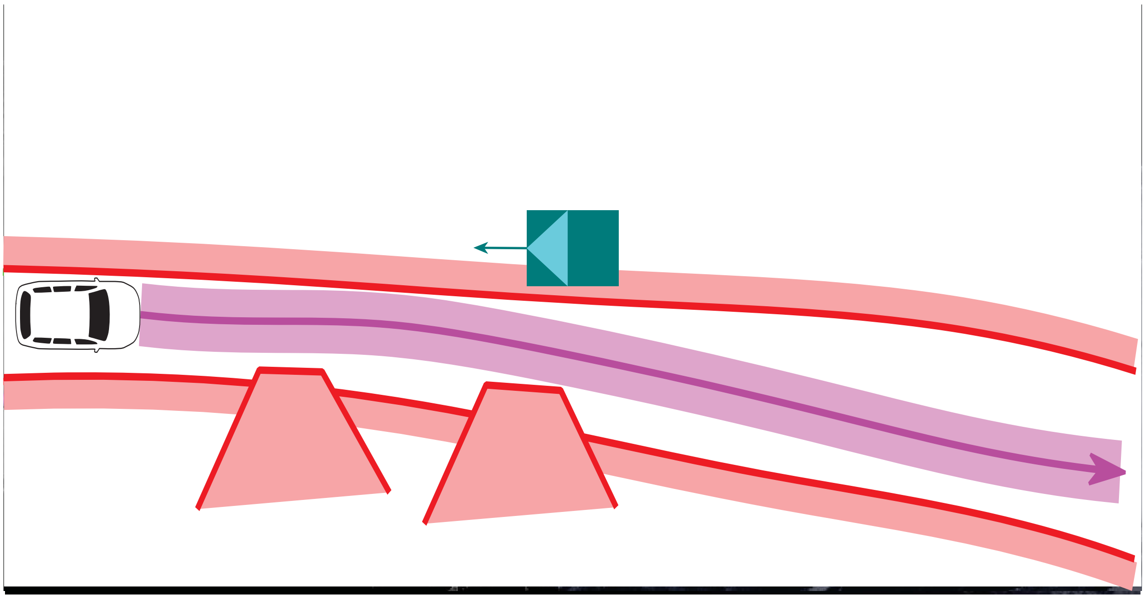
Stixel Representation
<1000 super-pixel

Tracked Stixel with
6D-motion vectors

Classified static background
detected moving objects

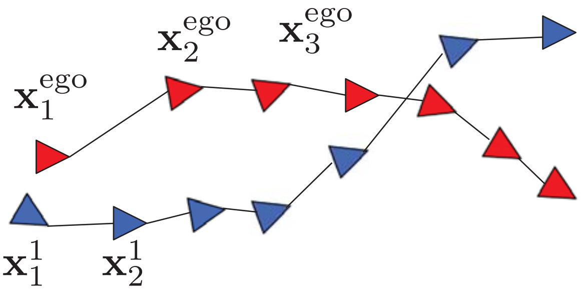
[Franke, Pfeiffer, Rabe, Knoeppel, Enzweiler, Stein, Herrtwich, ICCV Workshop 13]
Courtesy Daimler AG

Bertha's Driving Corridor



[Ziegler et al.2009-2013]

Cooperative Trajectory Planning


$\mathbf{x}^{\text{ego}}_1$
$\mathbf{x}^{\text{ego}}_2$
$\mathbf{x}^{\text{ego}}_3$
$\mathbf{x}^1_1$
$\mathbf{x}^1_2$

pose = (position, orientation)

$$\mathbf{x}_k = (\mathbf{X}_k, \mathbf{R}_k)$$

trajectory

$$\mathbf{x}_{j:l} = (\mathbf{x}_j, \mathbf{x}_{j+1}, \cdots, \mathbf{x}_l)$$

past

$$\mathbf{x}_{j:k} = (\mathbf{x}_j, \mathbf{x}_{j+1}, \cdots, \mathbf{x}_k)$$

future

$$\mathbf{x}_{k+1:l} = (\mathbf{x}_{k+1}, \mathbf{x}_{k+2}, \cdots, \mathbf{x}_l)$$

$$p(\mathbf{x}^{\text{ego}}_{k+1:l} | \mathbf{x}^1_{j:k}, \mathbf{x}^2_{j:k}, \cdots, \mathbf{x}^{\text{ego}}_{j:k}) =$$

$$\int p(\mathbf{x}^{\text{ego}}_{k+1:l} | \mathbf{x}^1_{j:l}, \mathbf{x}^2_{j:l}, \cdots, \mathbf{x}^{\text{ego}}_{j:k}) p(\mathbf{x}^1_{k+1:l}, \mathbf{x}^2_{k+1:l}, \cdots | \mathbf{x}^1_{j:k}, \mathbf{x}^2_{j:k}, \cdots, \mathbf{x}^{\text{ego}}_{j:k}) d\mathbf{x}^i_{k+1:l}$$

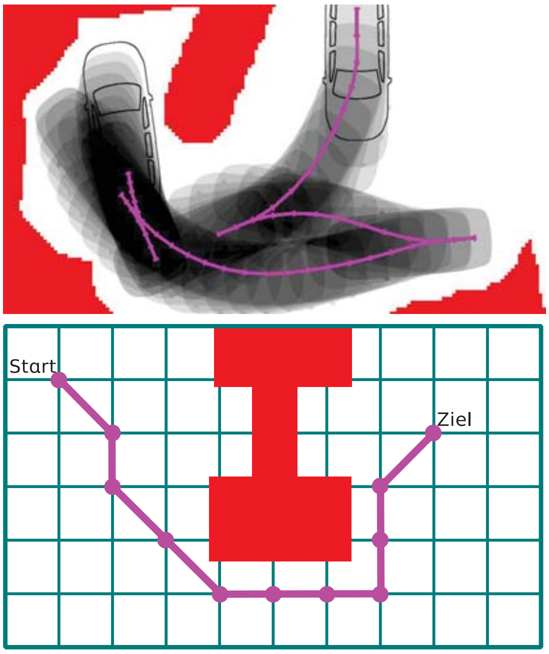
special case „certain prediction“, e.g. through v2v communication

$$= p(\mathbf{x}^{\text{ego}}_{k+1:l} | \mathbf{x}^1_{j:l}, \mathbf{x}^2_{j:l}, \cdots, \mathbf{x}^{\text{ego}}_{j:k})$$

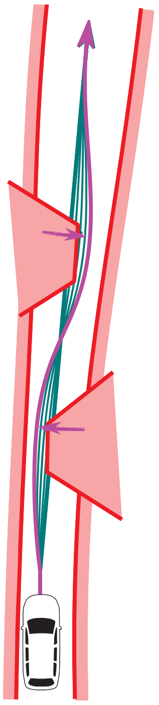
Trajectory Planning Methods

global, discrete, combinatoric

local, continuous, variational

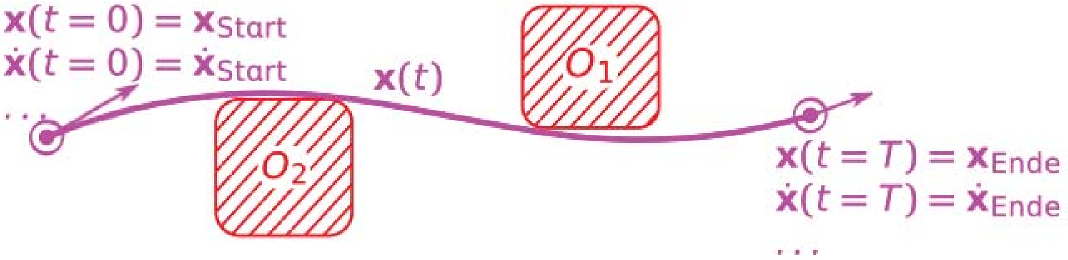


[Ziegler et al.2009–2011]



[Ziegler et al.2011–2014]

Trajectory planning



$\mathbf{x}(t=0) = \mathbf{x}_{\text{Start}}$
$\dot{\mathbf{x}}(t=0) = \dot{\mathbf{x}}_{\text{Start}}$
...
$\mathbf{x}(t)$
O_1
O_2
$\mathbf{x}(t=T) = \mathbf{x}_{\text{Ende}}$
$\dot{\mathbf{x}}(t=T) = \dot{\mathbf{x}}_{\text{Ende}}$
...

optimize cost functional

$$E[\mathbf{x}(t)] = \int_0^T J(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dddot{\mathbf{x}}) \mathrm{d}t$$

$$\begin{aligned} J(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \dddot{\mathbf{x}}) &= w_{\text{lat}} \left| \frac{d_{\text{left}}(\mathbf{x}) - d_{\text{right}}(\mathbf{x})}{2} \right|^2 \\ &+ w_{\text{vel}} \left| (\mathbf{v}_{\text{ref}}(\mathbf{x}) - \dot{\mathbf{x}}) \right|^2 \\ &+ w_{\text{acc}} \left| \ddot{\mathbf{x}} \right|^2 \\ &+ w_{\text{jerk}} \left| \dddot{\mathbf{x}} \right|^2 \\ &+ w_{\text{jawr}} \dot{\theta}^2 \end{aligned}$$

subject to hard inner and outer conditions

inner conditions enforce drivability, e.g.:

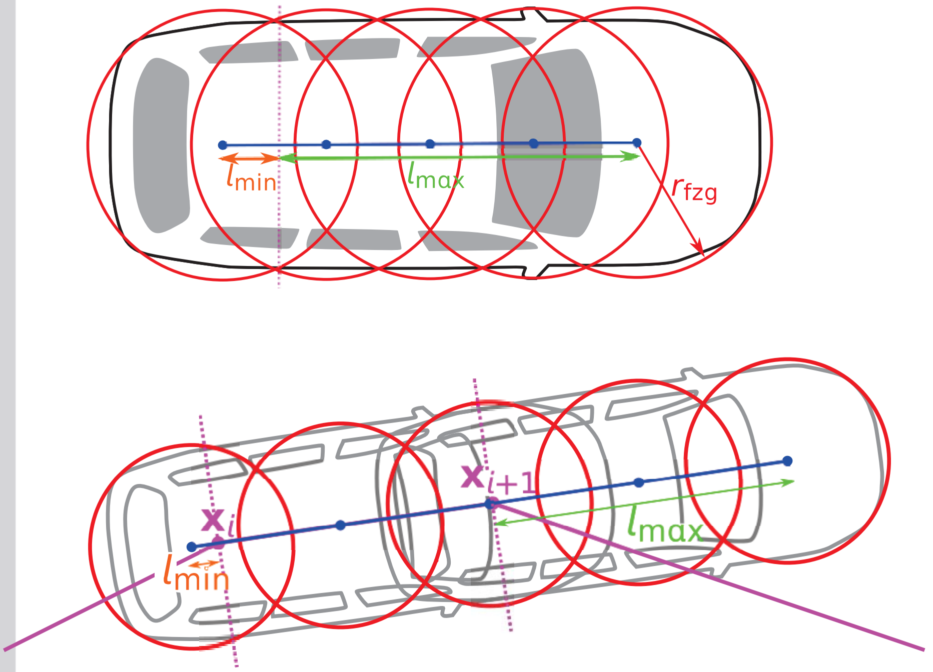
$$\|\ddot{\mathbf{x}}(t)\| < a_{\max}^2$$

outer conditions enforce integrity, e.g.:

$$d\left(\mathbf{x}(t), O_1\right) > 0$$

Fast Collision Checking

Approximation of vehicle shape by a set of circles

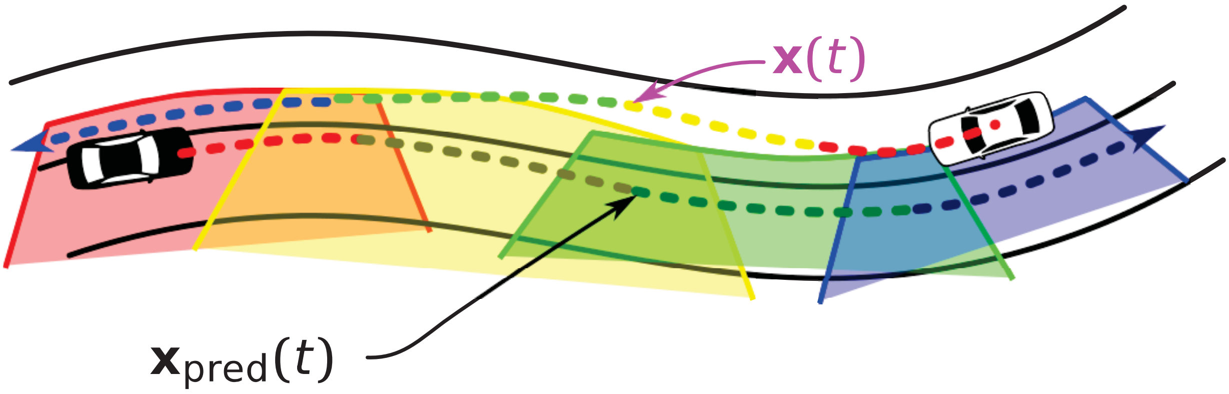


[Ziegler et al. 2011]

Dynamic Objects

We need to plan for ourselves …

… and for others

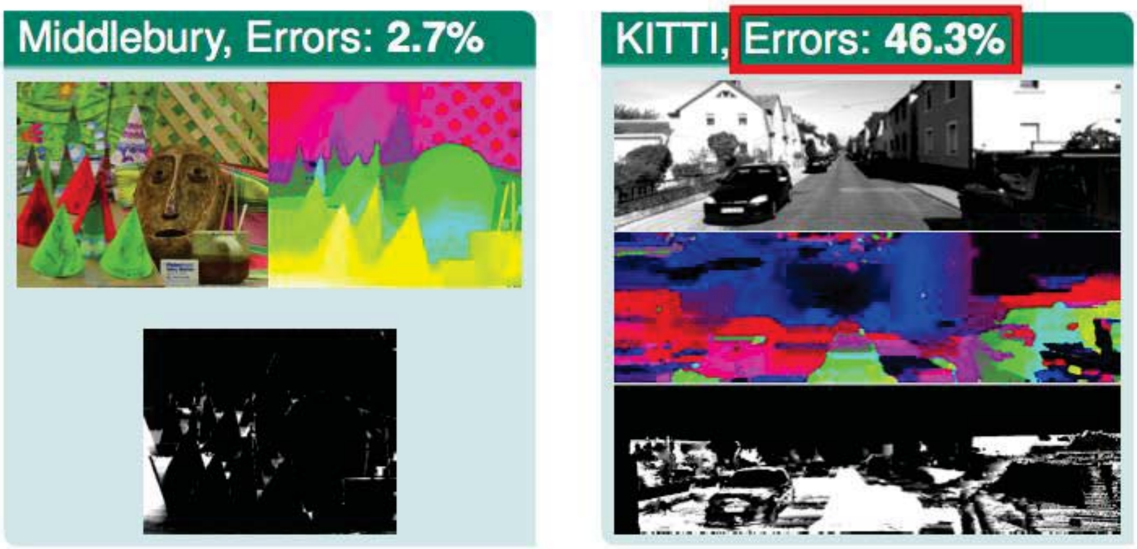


$t \in [t_0, t_1)$ $t \in [t_1, t_2)$ $t \in [t_2, t_3)$ $t \in [t_3, t_4)$

Results

KITTI Vision Benchmark

Fast guided cost-volume filtering (Rhemann et al., CVPR 2011)


Middlebury, Errors: **2.7%**

KITTI, Errors: **46.3%**

- Error threshold: 1 px (Middlebury) / 3 px (KITTI)

[Geiger, et al., International Journal of Robotics Research 32, 2013]

Automotive Vision Bechmark: www.mrt.kit.edu

Grand Cooperative Driving Challenge Holland, May 2011

gcdc Grand Cooperative Driving Challenge

Winner

Summary & Conclusions

- Automated driving using vision, sota sensors and maps is feasible
- Maneuver decisions strongly inferred from map knowledge
- Real-time dynamic trajectory planning
- Automated driving on Bertha Benz Memorial Route
  - In normal traffic and at normal velocities
  - Safety driver still needed
- Many open issues
  - Benchmarks
  - Safety assessment
  - Handling of rare situations
  - Cooperation
- Step-by-step market introduction

[Ziegler et al., IEEE Intelligent Transportation Systems Magazine, 2014]
[Bender et al., IEEE Intelligent Vehicles Symposium 2014]
[Schreiber et al., IEEE Intelligent Vehicles Symposium 2014]
[Geiger, et al., International Journal of Robotics Research 32, 2013]
[Liebner, Klanner, Baumann, Ruhhammer, Stiller, IEEE Intelligent Transportation Systems Magazine, 5 (2), 2013]
[Geiger, et al.IEEE Trans. Intelligent Transportation Systems, 13 (3),2013]
[Kitt, Lategahn, IEEE Intelligent Transportation Systems Conf. 2012]
[Lategahn, et al., IEEE Intelligent Vehicles Symposium 2012-2013]
[Geiger, Ziegler, Stiller, IEEE Intelligent Vehicles Symposium 2011]
[Moosmann, Stiller, IEEE Intelligent Vehicles Symposium 2011]
[Ziegler, Stiller, IEEE Intelligent Vehicles Symposium 2010]
[Ziegler, IROS 2011]
[Stiller, Kammel, Lulcheva, Ziegler, Automatisierungstechnik 2008]
[Özgüner, Stiller, Redmill, IEEE Proceedings 2007]

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session II

Perception

- **Title: Detection and Tracking of the Vanishing Point on a Horizon for Automotive Applications**
  Authors: Young-Woo Seo, Ragunathan Rajkumar

- **Title: Robot Navigation Using Radio Signal in Wireless Sensor Networks**
  Authors: Ju Wang, Mohammad M Tabanjeh, Tariq Qazi, Brian Bennett, Cesar Flores-Montoya, Eric Glover, Meesha Rashidi

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Detection and Tracking of the Vanishing Point on a Horizon for Automotive Applications

Young-Woo Seo and Ragunathan (Raj) Rajkumar

***Abstract*— In advanced driver assistance systems and autonomous driving vehicles, many computer vision applications rely on knowing the location of the vanishing point on a horizon. The horizontal vanishing point's location provides important information about driving environments, such as the instantaneous driving direction of roadway, sampling regions of the drivable regions' image features, and the search direction of moving objects. To detect the vanishing point, many existing methods work frame-by-frame. Their outputs may look desirable in that frame. Over a series of frames, however, the detected locations are inconsistent, yielding unreliable information about roadway structure. This paper presents a novel algorithm that, using the extracted line segments, detects vanishing points in urban scenes and tracks, using Extended Kalman Filter, them over frames to smooth out the trajectory of the horizontal vanishing point. The study demonstrates both the practicality of the detection method and the effectiveness of our tracking method, through experiments carried out using hundreds of urban scene images.**

I. INTRODUCTION

This paper presents a simple, but effective method for detecting and tracking the vanishing point on a horizon appearing in a stream of urban scene images. In urban street scenes, such detecting and tracking would enable the obtaining of geometric cues of 3-dimensional structures. Given the image coordinates of the horizontal vanishing point, one could obtain, in particular, the information about the instantaneous driving direction of a roadway [3], [8], [11], [13], [14], [16], [17], the information about the image regions for sampling the features of the drivable image regions [10], [12], the search direction of moving objects [9], and computational metrology through homography [15]. Advanced driving assistance systems or self-driving cars can exploit such information to detect neighboring moving objects and decide where to drive. Such information about roadway geometry can be obtained using active sensors (e.g., lidars with multi-horizontal planes or 3-dimensional lidar), but, as an alternative, many researchers have studied the use of vision sensors, due to lower costs and flexible usages [1], [3], [9].

A great deal of excellent work has been done in detecting vanishing points on perspective images of man-made environments; their performances are demonstrated on collections of images [2], [7], [18]. Most of these methods, in voting on potential locations of vanishing points, use low-level image features such as spatial filter responses (e.g., Garbor filters) [12], [8], [14], [20] and geometric primitives (e.g., line segments) [7], [15], [17], [18]. To find an optimal vote result, the methods use an iterative algorithm such as Expectation and Maximization (EM).

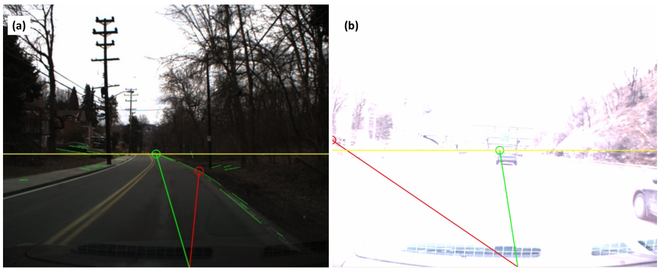


Fig. 1: Sample images show the necessity of a vanishing point tracking for real-world, automotive applications. The red circle represents the vanishing points detected from the input images and the green circle represents the vanishing points tracked over frames. The yellow line represents the estimated horizon line. The existing, frame-by-frame vanishing point detection methods would fail when (a) the extracted image features overfit and (b) relevant image features are not present at the input images.

However, these frame-by-frame vanishing point detection methods may be impractical for real-time, automotive applications primarily because 1) they require intensive computation per frame and 2) they expect a presence of low-level image features. In particular, it may take longer than a second simply to apply spatial filters to large parts of or the entire input image. Meanwhile, a vehicle drives a number of meters with no information about road geometry. Furthermore, these frame-by-frame methods would fail to detect the vanishing point appearing on over- and under-exposed images. Such images are acquired when a host-vehicle is emerging from tunnels or overpasses. Figure 1 (b) shows a sample image acquired when our vehicle emerges from a tunnel. When this happens, these methods would fail to continuously provide information about the vanishing point's location. Because of such a practical issue, some researchers developed Bayes filters to track the vanishing point's trajectory [12], [17]. In addition to the two aforementioned concerns, we have one of our own. In an earlier work [15], we demonstrated the ability to acquire, using a monocular camera sensor, the information of a vehicle's lateral locations as well as metrological information of the ground plane. To correctly compute metric information such as lateral distances of a vehicle to both boundaries of the host road-lane, it is critical to accurately estimate the angle between the road plane and the camera plane. To do this, we detect the vanishing point on the horizon to estimate the angle (i.e., pitch) between two planes. But, because, image features relevant to detecting the vanishing point are missing in certain frames, our vanishing point detection fails to correctly locate the vanishing point, resulting in incorrect angle measurements and distance computations.

Young-Woo Seo is with the Robotics Institute and Ragunathan (Raj) 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

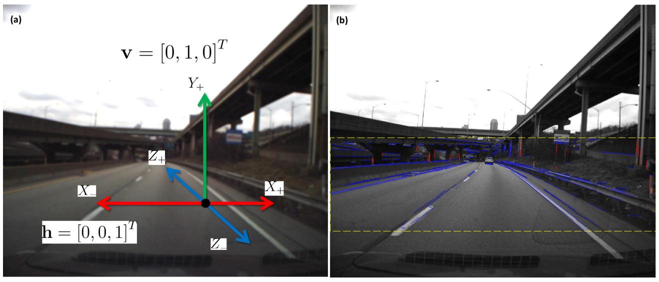


Fig. 2: (a) A prior for the line classification. (b) An example of line detection and classification. The red (blue) lines are categorized into the vertical (horizontal) line group. The yellow, dashed rectangle represents a ROI for line extraction.

To address these practical concerns, we have developed a novel method of detecting and tracking the vanishing point on the horizon. In what follows, Section II-A details how we extract line segments from an input image and how we detect, using extracted line segments, the vanishing point on the horizon. Section II-B describes our implementation of Extended Kalman Filter (EKF) for tracking the detected vanishing point. Section III explains experiments conducted to demonstrate the effectiveness of the proposed algorithms and discusses the findings. Finally Section IV lays out our conclusions and future work.

The contributions of this paper include 1) a method, based on line segments, for fast detection of vanishing points, 2) a novel vanishing point tracking algorithm based on a Bayes filter, and 3) empirical validations of the proposed work.

II. A Bayes Filter for Tracking a Vanishing Point on the Horizon

This section details our approach to the problem of detecting and tracking a vanishing point on a horizon appearing on perspective images of urban streets. A vanishing point on a perspective image is the intersection point of two parallel lines. In urban street scenes, as long as the image is under normal exposure, one can obtain plenty of parallel line segment pairs, pairs such as longitudinal lane-markings and building contour lines. Section II-A describes how we extract line segments and, with them, detect vanishing points. The image coordinates of the vanishing points detected from individual frames may be temporally inconsistent because line segments relevant to and important for vanishing point detection may not have been extracted. To smooth out the location of vanishing points over time, we develop an extended Kalman filter to track vanishing points. Section II-B details the procedure and measurement model of our EKF implementation.

A. Vanishing Point Detection

Our algorithm detects, by using line segments, vanishing points appearing on a perspective image. In an urban scene image, one can extract numerous line segments from urban structures, like man-made structures (e.g., buildings, bridges, overpasses, etc.) and traffic devices (e.g., Jersey barriers, lane-markings, curbs, etc.) To obtain these line segments, we tried three line-extraction methods: Kahn's [6], [7], the probabilistic and the standard Hough transform [4]. We found Kahn's method to work best in terms of the number of resulting line segments and their geometric properties, such as lengths or representation fidelity to the patterns of low-level features. To implement Kahn's method, we first obtain Canny edges and run the connected component-grouping algorithm to produce a list of pixel blobs. For each pixel blob, we compute the eigenvalues and eigenvectors of the pixel coordinates' dispersion matrix. The eigenvector, $\mathbf{e}_1$, associated with the largest eigenvalue is used to represent the orientation of a line segment and its length, $\mathbf{l}_j = (\theta_j, \rho_j) = (atan2(\mathbf{e}_{1,2}, \mathbf{e}_{1,1}), \bar{x}\cos\theta + \bar{y}\sin\theta)$, where $\bar{x} = \frac{1}{n}\Sigma_k x_k, \bar{y} = \frac{1}{n}\Sigma_k y_k$. The two parameters, θ_j and ρ_j, are used to determine two end points, $\mathbf{p}^1_j = \left[x^1_j, y^1_j\right]$ and $\mathbf{p}^2_j = \left[x^2_j, y^2_j\right]$, of the line segment $\mathbf{l}_j$. Figure 2 (b) shows an example of line detection result.

Given a set of the extracted lines, $L = \{\mathbf{l}_j\}_{j=1,\ldots,|L|}$, we first categorize them into one of two groups: vertical L_V or horizontal L_H, $L = L_V \cup L_H$. We do this to use only a relevant subset of the extracted lines for detecting a particular (vertical or horizontal) vanishing point. For example, if vertical lines are used to find a horizontal vanishing point, the coordinates of the resulting vanishing point would be far from optimal. To set the criteria for this line categorization, we define two planes: $\mathbf{h} = [0,0,1]^T$ for a horizontal plane and $\mathbf{v} = [0,1,0]^T$ for a vertical plane in the camera coordinate. We do this because we assume that the horizontal (or vertical) vanishing points lie at a horizontal (or vertical) plane at the front of our vehicle. Figure 2 (a) illustrates our assumption about these priors. We transform, the coordinates of the extracted line segments' two points into those of the camera coordinates, $\mathbf{p}_{cam} = \mathbf{K}^{-1}\mathbf{p}_{im}$, where $\mathbf{p}_{cam}$ is a point in the camera coordinates, $\mathbf{K}$ is the camera calibration matrix of intrinsic parameters, and $\mathbf{p}_{im}$ is a point in the image coordinates. We then compute the distance of a line segment, $\mathbf{l}_j = [a_j, b_j, c_j]^T$,[1] to the horizontal, $\mathbf{h}$, and the vertical plane, $\mathbf{v}$. We assign a line to either of two line groups based on the following:

$$\begin{array}{ll} L_V \leftarrow \mathbf{l}_j, & \text{if } \mathbf{l}_j^T \cdot \mathbf{v} \leq \mathbf{l}_j^T \cdot \mathbf{h}, \\ L_H \leftarrow \mathbf{l}_j, & \quad\text{Otherwise} \end{array} \tag{1}$$

where $\mathbf{l}_j^T \cdot \mathbf{v} = \frac{[a_j, b_j, c_j]^T [0,1,0]}{\sqrt{a_j^2 + b_j^2 + c_j^2}}$. Figure 2 (b) shows an example of line classification result; vertical lines are depicted in red, horizontal lines in blue. Such line categorization results help us use a subgroup of the extracted lines relevant to computing the vertical or horizontal vanishing point. Our approach of using line segments to detect vanishing point is similar to some found in earlier work [7], [17], [18]. All uses line segments (or edges) to detect vanishing points. Our distinguishes itself in terms of line classification. Suttorp and Bucher's method relied on a heuristic, to cluster lines into left or right sets for vanishing point detection [17]; Tardif [18] used a J-linkage algorithm to group edges into the same clusters. In contrast, our method distinguishes horizontal line segments from vertical ones by computing the similarity of line segments to the priors about the ideal locations of vanishing points.

Given two sets of line groups (vertical and horizontal), we run RANSAC [4] to find the best estimation of a vanishing point. For each line pair randomly selected from

[1] Using two end-points of a line segment, we can represent a line segment in an implicit line equation, where, $a_j = y^1_j - y^2_j, b_j = x^2_j - x^1_j, c = x^1_j y^2_j - x^2_j y^1_j$.

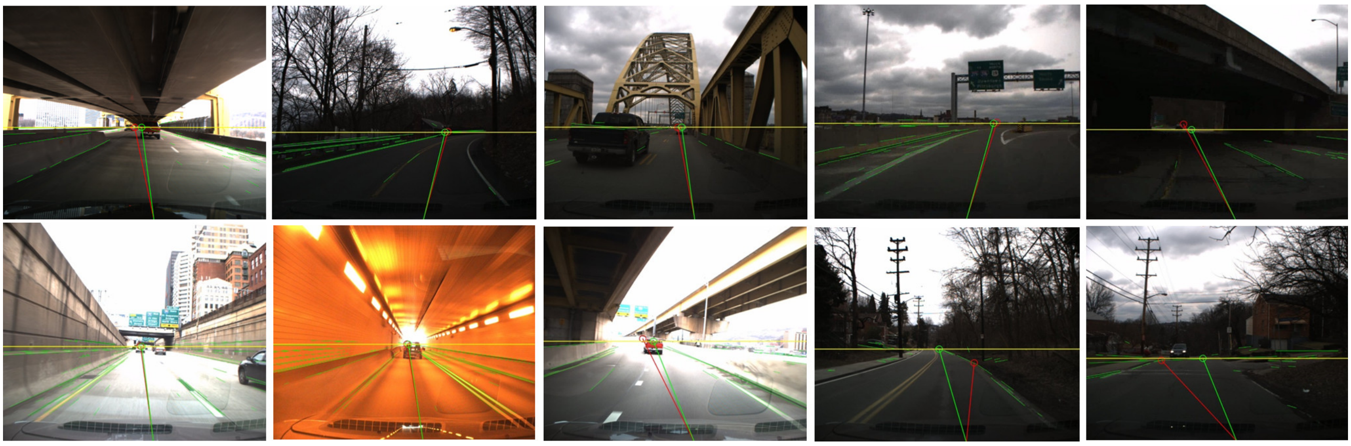


Fig. 3: Some examples of vanishing point detection results. For most of testing images, our vanishing point detection worked well as good as the tracking method. But it often failed to correctly identify the location of the horizontal vanishing point. For the last two images, our detection method found the locally optimal vanishing points (red circles) based on the lines extracted from those images. By contrast, our tracking method were able to find the globally optimal locations (green circles) of the vanishing points.

the horizontal and vertical line groups, we first compute the cross-product of two lines, $\mathbf{vp}_{ij} = \mathbf{l}_i \times \mathbf{l}_j$, to find an intersection point. The intersection point found thus is used as a vanishing point candidate. We then claim the vanishing point candidate with the smallest number of 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. Figure 3 shows sample results of vanishing point detection.

B. Vanishing Point Tracking

The previous section detailed how we detect vanishing points using line segments extracted from urban structures. Such frame-by-frame detection may result in, however, inconsistent locations of the same vanishing point over frames. This is because some image features (i.e., line segments) relevant to detecting vanishing points on the previous frame may not be available in the current frame. When this happens, any frame-by-frame, vanishing point detection algorithm, including ours, fails to find an optimal location of the horizontal vanishing point. This results in incorrect information about roadway geometry [15].

To address such potential inconsistency, we develop a tracker to smooth out the trajectory of the vanishing point of interest. Our idea for tracking the vanishing point is to use some of the extracted line segments as measurements, thus enabling us to trace the trajectory of the vanishing point. To implement our idea, we developed an Extended Kalman Filter (EKF). Algorithm 1 describes the procedure of our vanishing point tracking method.

For our EKF model, we define the state as, $\mathbf{x}_k = [x_k, y_k]^T$, where x_k and y_k is the k step's camera coordinates of the vanishing point on the horizon. We initialize the state, $\mathbf{x}$ and its covariance matrix, $\mathbf{P}$ as:

$$\mathbf{x}_0 \quad = [\mathrm{IM_{width}}/2/\mathrm{f_x}, \mathrm{IM_{height}}/2/\mathrm{f_y}]^T ,$$

Algorithm 1 EKF for tracking the vanishing point.

```
Input: IM, an input image and $L$, a set of line segments
    extracted from the input image, $\{\mathbf{l}_j\}_{j=1,\dots,|L|} \in L$
Output: $\hat{\mathbf{x}}_k = [x_k, y_k]^T$, an estimate of the image coordinates
    of the vanishing point on the horizon
Detect a vanishing point, $vp^h$ = Detect(IM, $L$)
Run EKF iff $vp^h_x \leq \mathrm{IM_{width}}$ and $vp^h_y \leq \mathrm{IM_{height}}$.
    Otherwise exit.
EKF: Prediction
$\hat{\mathbf{x}}^-_k = f(\hat{\mathbf{x}}_{k-1}) + \mathbf{w}_{k-1}$
$\mathbf{P}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}^T_{k-1} + \mathbf{Q}_{k-1}$
EKF: State Estimation
for all $\mathbf{l}_j \in L$ do
    $\tilde{y}_j = z_j - h(\hat{\mathbf{x}}^-_k)$
    $\mathbf{S}_j = \mathbf{H}_j\mathbf{P}_j\mathbf{H}^T_j + \mathbf{R}_j$
    $\mathbf{K}_j = \mathbf{P}_j\mathbf{H}^T_j\mathbf{S}^{-1}_j$
    Update the state estimate if $\tilde{y}_j \leq \tau$
    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^-_k + \mathbf{K}_j\tilde{y}_j$
    $\mathbf{P}_j = (\mathbf{I}_2 - \mathbf{K}_j\mathbf{H}_j)\mathbf{P}_j$
end for
```

$$\mathbf{P}_0 \quad = \begin{bmatrix} \left(\frac{x_{im}}{f_x}\right)^2 & 0 \\ 0 & \left(\frac{y_{im}}{f_y}\right)^2 \end{bmatrix}$$

where x_{im} and y_{im} are our initial guesses about the uncertainty of the state in pixels, along the x- and y-axises, and f_x and f_y are focal lengths of the vision sensor we use. The initial values need to be scaled by focal lengths because the state is represented in the normalized camera coordinates.

Given an input image, our algorithm predicts the location of the vanishing points, $\hat{\mathbf{x}}^-_k = \mathbf{I}_2\hat{\mathbf{x}}_{k-1} + \mathbf{w}_{k-1}$, where $\mathbf{I}_2$ is 2×2 identity matrix and $\mathbf{w}_{k-1}$ is a 2×1 vector of process model's noise, normally distributed, $\mathbf{w}_k \sim N(0, \mathbf{Q})$.[2] While doing so, we neither define a motion model (i.e., $f(\hat{\mathbf{x}}_k)$)

[2] The $\hat{\mathbf{x}}$ represents an estimate and the superscript, $\hat{\mathbf{x}}^-$, indicates that it is a predicted value.

nor incorporate any information about ego-motion. We set the process noise as a constant, $\mathbf{Q}_{2\times 2} = \text{diag}(\sigma_Q^2)$, where $\sigma = \frac{x_{im}}{f_x}$.

For the measurement update, we first change the representation of an extracted line segment, $\mathbf{l}_j$, as a pair of image coordinates of its mid-point and orientation, $\mathbf{l}_j = [\mathbf{m}_j, \theta_j]^T$, where $\mathbf{m}_j = [m_{j,x}, m_{j,y}]^T$, $\theta_j \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Note that the line segments we use as measurements for EKF are the same ones used for detecting the vanishing point. Our approach is similar to Suttorp and Bucher's method [17], but both employ different measurement models. We then compute the residual, $\tilde{y}_j$, the difference between our expectation on an observation, $h(\hat{\mathbf{x}}_k^-)$ and an actual observation, $z_j = \theta_j$.

We presume that if a selected line segment, $\mathbf{l}_j$, is aligning with the vanishing point of interest, $\hat{\mathbf{x}}_k = [x_k, y_k]^T$, the angle between the vanishing point and the orientation of the line should be zero (or very close to zero). Figure 5 illustrates the underlying idea of our measurement model that investigates the geometric relation between an extracted line and a vanishing point of interest. Based on this idea, we design a model of what we expect to observe, our observation model, as

$$h(\mathbf{x}_k) \quad = \quad \tan^{-1}\left(\frac{y_k - m_{j,y}}{x_k - m_{j,x}}\right) \tag{2}$$

To linearize this non-linear observation model, we take the first-order, partial derivative of $h(\mathbf{x}_k)$, with respect to the state, $\mathbf{x}_k$, to derive the Jacobian of the measurement model, $\mathbf{H}$.

$$\frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}} = \left[\frac{-(y_k - m_{j,y})}{d^2}, \frac{(x_x - m_{j,x})}{d^2}\right] = \mathbf{H} \tag{3}$$

where $d^2 = \sqrt{(x_k - m_{j,x})^2 + (y_k - m_{j,y})^2}$. We set the measurement noise, $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$ and $\mathbf{R}_{1\times 1} = \sigma_R^2$, where $\sigma = 0.1$ radian. We then compute the innovation $\mathbf{S}_j$ and the Kalman gain $\mathbf{K}_j$ for the measurement update.

Before actually updating the state using these measurements, we treat individual line segments differently based on their lengths. This is because the shorter the length the higher the chance of the line being a noise measurement.[3] To implement this idea, we compute a weight of the line based on its length and heading difference, to update the measurement noise.

$$\mathbf{R} = R_{max} + \left(\frac{R_{min} - R_{max}}{l_{max} - l_{min}}\right) |\mathbf{l}_j| \tag{4}$$

where R_{max} (e.g., 10 degrees) and R_{min} (e.g., 1 degree) define the maximum and the minimum of heading difference in degree, and l_{max} (e.g., 500) and l_{min} (e.g., 20) define the maximum and minimum of observable line lengths in pixels, $|\mathbf{l}_j|$ is the length of the line. This equation ensures that we treats the longer line more importantly when updating the state and we only use lines of which heading differences are smaller than the threshold, τ.

In summary, the task of our EKF is to analyze the extracted line segments to estimate the location of the vanishing point on the horizon. Figure 4 shows some example results that one can see the difference of the locations between the detected and the tracked vanishing points.[4]

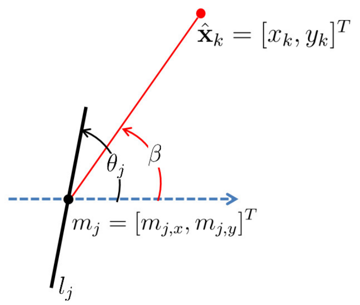

$\hat{\mathbf{x}}_k = [x_k, y_k]^T$
β
θ_j
$m_j = [m_{j,x}, m_{j,y}]^T$
l_j

Fig. 5: The line measurement model. The red circle represents the vanishing point, $\hat{\mathbf{x}}_k$, tracked until kth step. θ_j is the orientation of the jth line, l_j, and β is the orientation between the line's mid-point and the vanishing point. The orientation difference is the residual of our EKF model.

We use the tracked vanishing point to compute the (pitch) angle between the camera plane and the ground plane. The underlying assumption is that, if the road plane is flat and perpendicular to an image plane, the vanishing point along the horizon line is exactly mapped to the camera center. Based on this assumption, we derive the location of the vanishing point on the horizon line as [5]:

$$\mathbf{vp}_h^*(\phi, \theta, \psi) = \left[\frac{c\phi s\psi - s\phi s\theta c\psi}{c\theta c\psi}, \frac{-s\phi s\psi - c\phi s\theta c\psi}{c\theta c\psi}\right]^T \tag{5}$$

where ϕ, θ, ψ are yaw, pitch, and roll angle of the camera plane with respect to the ground plane and c and s for cos and sin. Since we are interested in estimating the pitch angle, let us suppose that there is no vertical tilt and rolling (i.e., the yaw and the roll angles are zero). Then the above equation yields:

$$\mathbf{vp}_h^*(\phi = 0, \theta, \psi = 0) = \left[\frac{0}{c\theta}, -\frac{s\theta}{c\theta}\right] \tag{6}$$

Because we assume that there is neither yaw nor roll, we can compute the pitch angle by computing the difference between the y-coordinate of the vanishing point and that of the principal point of the camera as

$$\theta = \tan^{-1}\left(|p_y - vp_y|\right) \tag{7}$$

where p_y is the y coordinate of the principal point. Figure 6 shows our setup to verify the accuracy of our pitch angle estimation. Because no precise angle measurement exists between the two planes, we instead measure the distances between the camera and markers on the ground to evaluate the accuracy of the pitch angle computation. We found that the distance measurements have, on average, a sub-meter accuracy (i.e., less than 30cm).

III. Experiments

To evaluate the performance of our vanishing point detection and tracking algorithm, we drove our robotic car [19] on a route of inter-city highways, to collect some image data

[3] Recall that we extract line segments from Canny's edge image where short edges may originate from artificial patterns, not from actual objects' contours.

[4] Some of the vanishing point tracking videos are available from, `http://www.cs.cmu.edu/~youngwoo/research.html`

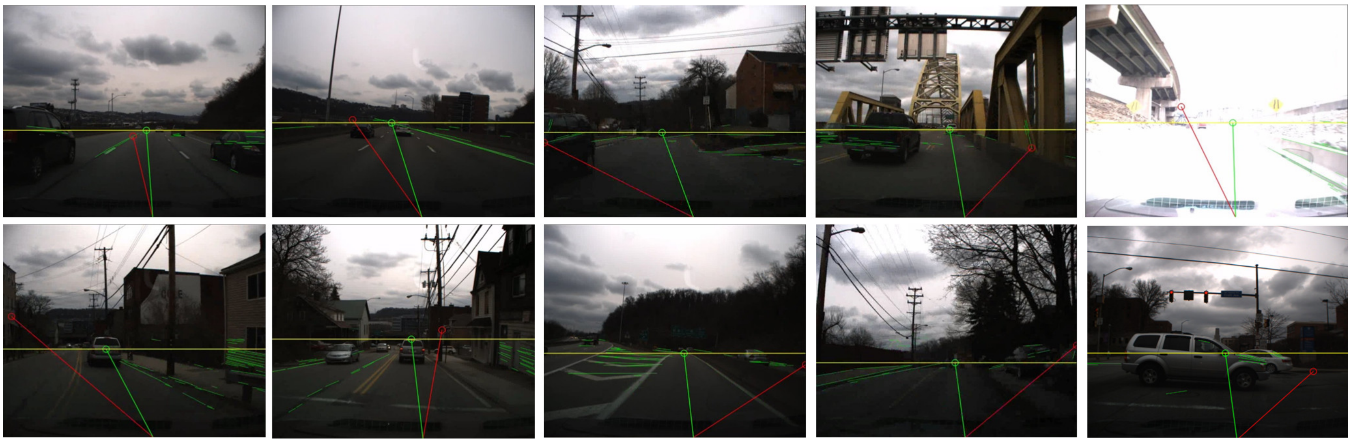


Fig. 4: A comparison of vanishing point locations by the frame-by-frame detection and by the EKF tracking.

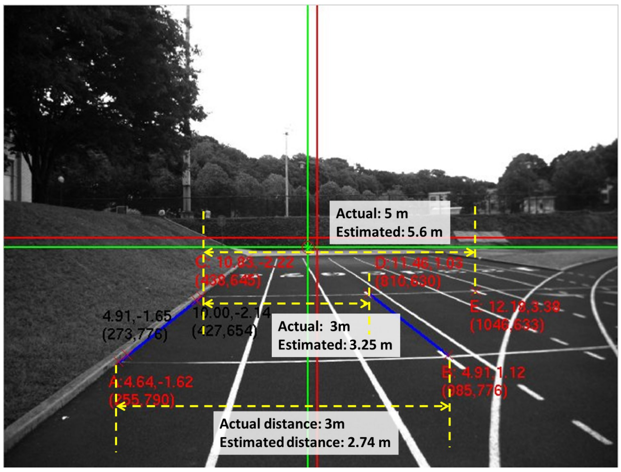


Fig. 6: A setup for verifying the accuracy of our world-coordinate computation model. The intersection point of the two red lines represents the camera center and the intersection point of the two green lines represents a vanishing point computed from the two blue lines appearing on the ground.

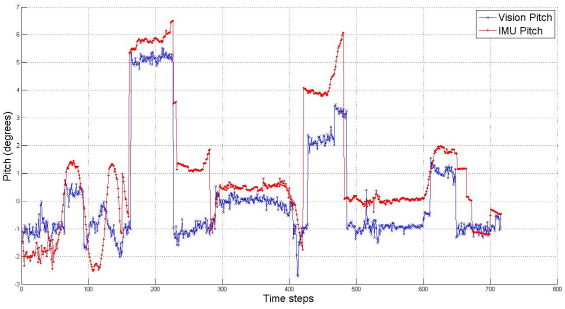


Fig. 7: A comparison of the estimated pitch angles by an IMU and by the proposed method.

and the vehicle's motion data. Our vehicle is equipped with a military-grade IMU which, in root-mean-square sense, the error of pitch angle estimation is 0.02 with GPS signals (with RTK corrections) or 0.06 degree with GPS outage, when driving more than one kilometer or for longer than one minute. The vision sensor installed on our vehicle is PointGrey's Flea3 Gigabit camera, which can acquire an image frame of 2,448×2,048, maximum resolution at 8Hz. While driving the route, we ran the proposed algorithms as well as the data (i.e., image and vehicle states) collector. We implemented the proposed methods in C++ and OpenCV that runs about 20Hz. The data collector automatically syncs the high-rate, ego-motion data (i.e., 100Hz) with the low-rate, image data (i.e., 8Hz). To estimate the camera's intrinsic parameters, we used a publicly, available toolbox for camera calibration[5] and define a rectangle for the line extraction ROI, $x_1 = 0$, $x_2 = \mathbf{I}_{width-1}$, $y_1 = 1300$ and $y_2 = 1800$. For the line segment weighting, we empirically found that $R_{max} = 10$, $R_{min} = 1$, $l_{max} = 500$, and $l_{min} = 20$ worked best.

We evaluated quantitatively and qualitatively the performance of the presented vanishing point tracking method.

For the quantitative evaluation, we analyzed the accuracy of the pitch angles estimated from the vanishing point tracking. Figure 7 shows the comparison of the pitch angles measured by the IMU and estimated by a monocular vision sensor. Although the pitch angles estimated from our algorithm have some periods underestimate (or overestimate) the true pitch angles, the two graphs have, at a macro-level, similar shapes where the blue curve follows the ups-and-downs of the red curve. The mean-square error is 2.0847 degrees.

For the qualitative evaluation, we examined how useful the output of the tracked vanishing point is in approximating the driving direction of a road way. Figure 8 shows some example results that, within a certain range, the driving directions of roads can be linearly (or instantaneously) approximated by linking the locations of the tracking vanishing point to the center of the image bottom (i.e., the image coordinates our camera is projected on).

IV. Conclusions and Future Work

This paper has presented novel methods of detecting vanishing points and of tracking a vanishing point on the horizon. To detect vanishing points, we extracted line segments and applied RANSAC to the locally optimal vanishing point

[5] http://www.vision.caltech.edu/bouguetj/calib_doc/

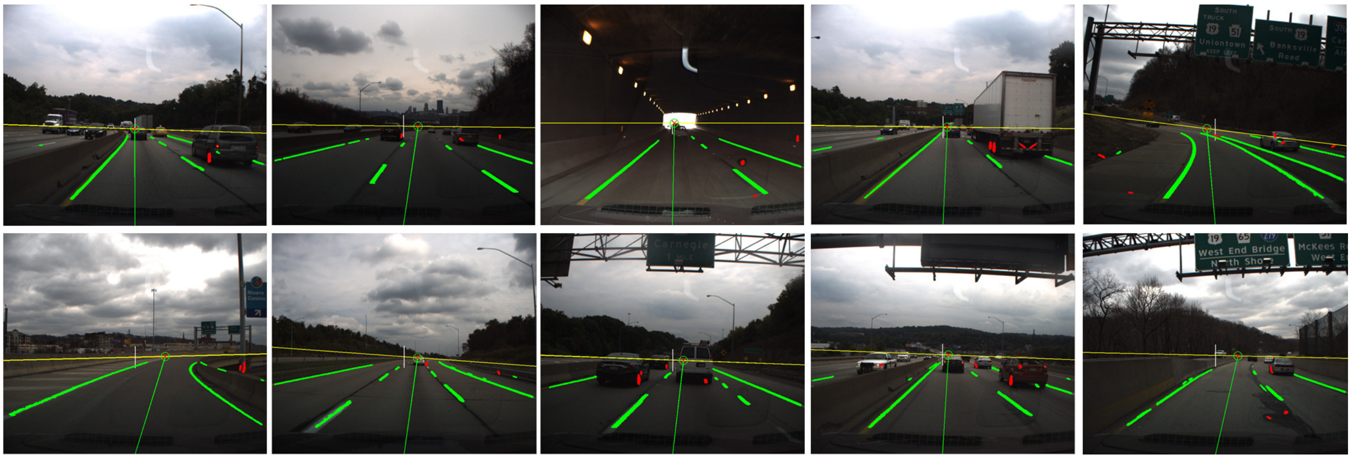


Fig. 8: This figure shows the idea of using the results of vanishing point tracking to approximate the driving direction of a roadway. We used such approximated driving directions to remove false-positive lane-marking detections [16]. The green blobs are the final outputs of lane-marking detection and the red blobs are the false-positive lane-marking detections that are removed from the final results. Refer to [16] for more detail.

from a given input image. Occasionally, however, our method failed to detect the vanishing point because relevant image features were unavailable. Our previous computer vision application for autonomous driving required metric computation to accurately measure the vehicle's lateral position. To obtain this measurement, we need an accurate measurement of the angle between the camera and the ground planes. To compute this angle, we used the detected vanishing point. Thus, when the vanishing point location was inaccurately estimated, it led to an imprecise measurement of the vehicle's lateral motions. To tackle such inconsistent positions of the vanishing point over frames, we developed an EKF and addressed this jumpy trajectory of the vanishing point.

As future work, we would like to determine the limits of our algorithms and so continue testing it against various driving environments. In addition, we would like to study the relation of ego-vehicle's motion between in the world coordinates and image coordinates and develop a motion model to enhance the performance of our tracking method.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Myung Hwangbo for the fruitful discussion on 3D geometry and to the members of GM-CMU Autonomous Driving Collaborative Research Lab for their efforts and dedications.

REFERENCES

[1] 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.

[2] James M. Coughlan and A.L. Yuille, Manhattan world: compass direction from a single image by bayesian inference, In *Proceedings of IEEE International Conference on Computer Vision* (ICCV-99), pp. 941-947, 1999.

[3] Ernst D. Dickmanns, *Dynamic Vision for Perception and Control of Motion*, Springer, 2007.

[4] David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2002.

[5] 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.

[6] 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.

[7] Jana Kosecka and Wei Zhang, Video compass, In *Proceedings of European Conference on Computer Vision* (ECCV-02), pp. 476-490, 2002.

[8] Hui Kong, Jean-Yves Audibert, and Jean Ponce, Vanishing point detection for road detection, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR-09), pp. 96-103, 2009.

[9] 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.

[10] Ondrej Miksik, Petr Petyovsky, Ludek Zalud and Pavel Jura, Robust detection of shady and highlighted roads for monocular camera based navigation of UGV, In *Proceedings of IEEE International Conference on Intelligent Robots and Systems* (IROS-11), pp. 64-71, 2011.

[11] Ondrej Miksik, Rapid vanishing point estimation for general road detection, In *Proceedings of IEEE International Conference on Robotics and Automation* (ICRA-12), pp. 4844-4849, 2012.

[12] Peyman Moghadam and Jun Feng Dong, Road direction detection based on vanishing-point tracking, In *Proceedings of IEEE International Conference on Intelligent Robots and Systems* (IROS-12), pp. 1553-1560, 2012.

[13] Marcos Nieto, Jon Arrospide Laborda, and Luis Salgado, Road environment modeling using robust perspective analysis and recursive Bayesian segmentation, *Machine Vision and Applications*, 22:927-945, 2011.

[14] Christopher Rasmussen, Grouping dominant orientations for ill-structured road following, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR-04), pp. 470-477, 2004.

[15] Young-Woo Seo and Raj Rajkumar, Use of a monocular camera to analyze a ground vehicle's lateral movements for reliable autonomous city driving, In *Proceedings of IEEE IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles* (PPNIV-2013), pp. 197-203, 2013.

[16] Young-Woo Seo and Raj Rajkumar, Utilizing instantaneous driving direction for enhancing lane-marking detection, In *Proceedings of the IEEE Intelligent Vehicles Symposium* (IV-2014), pp. 170-175, 2014.

[17] Thorsten Suttorp and Thomas Bucher, Robust vanishing point estimation for driver assistance, In *Proceedings of IEEE Intelligent Transportation Systems Conference* (ITSC-06), pp. 1550-1555, 2006.

[18] Jean-Philippe Tardif, Non-iterative approach for fast and accurate vanishing point detection, In *Proceedings of IEEE International Conference on Computer Vision* (ICCV-09), pp. 1250-1257, 2009.

[19] Junqing Wei, Jarrod Snider, Junsung Kim, John Dolan, Raj Rajkumar, and Bakhtiar Litkouhi, Towards a viable autonomous driving research platform, In *Proceedings of IEEE Intelligent Vehicles Symposium* (IV-13), pp. 763-770, 2013.

[20] Qi Wu, Wende Zhang, and B.V.K Vijaya Kumar, Example-based clear path detection assisted by vanishing point estimation, In *Proceedings of IEEE International Conference on Robotics and Automation* (ICRA-11), pp. 1615-1620, 2011.

Robot Navigation Using Radio Signal in Wireless Sensor Networks

Ju Wang, Mohammad M Tabanjeh, Tariq Qazi, Brian Bennett, Cesar Flores-Montoya, Eric Glover, Meesha Rashidi

Abstract— **We investigate the navigation problem of a land robot in a Wireless Sensor Network (WSN) to perform network maintenance works. Our proposed method employs a distributed RF sensing technique with the aid of a directional antenna. Our method only requires partial and coarse RF profiling of the field network area. Another advantage is that it does not require knowledge of the locations of the beacon nodes. In 2D navigation, the directional RSS measurements allow us to achieve location accuracy beyond the grid resolution of the RF profile. The robot is able to navigate within 2 feet from the target sensor in indoor environments and within 5 feet in outdoor tests.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are used in many research, industry and agriculture applications to provide valuable field data, however the maintenance of such networks pose a significant challenge. Typically deployed in outdoor environment, the network will gradually lose its functionality due to depleted battery and/or physical wear in sensor node. A single mal-functioning node is often enough to cause network segmentation and render some wireless sensors unreachable (Figure 1). A practical solution is to use Unmanned Ground Vehicle to perform maintenance workload such as recycling dead nodes and deploying new sensor nodes in the field. The problem then becomes the ability of the robot to locate and navigate to a wireless sensor device based on RF transmission. The challenge come from two aspects: (1) could RF sensing provide sufficient location accuracy? and (2) could we track the location of the wheeled robot precisely in the off-road environment [1]. Our focus here will be on the first challenge.

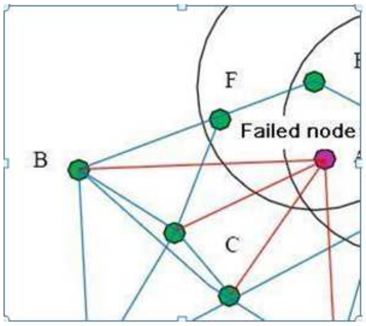


Fig. 1. Sensor localization in WSN

In the simplest case, radio sensing based navigation involves measuring radio signal strength (RSS) of a target transmitter to estimate the distance between the transmitter and the receiver. The method attracted more attention recently [4], [6], [8], [10], [11] when the application scenarios disallowed conventional navigation method such as GPS. Another application scenario would be for search and rescue of missing personnel in a remote wild land. Using RF sensing technology, robots can search a large area in a coordinated fashion. Bahl etc [4] reported that in a complex indoor environment, RSS based empirical method has higher accuracy up to a few meters than model-fitting based method. However simple RSS localization has been proven difficult to use due to the high complex and nonlinear radio channel model in real deployments. In most cases comprehensive RF profiling of the target area must be conducted. There are several follow-up techniques to improve the accuracy of RF localization [5], [12], some employ comprehensive RF profiling; some use multiple fixed-location beacon nodes. However RF sensing remains to be a low accuracy positioning tool even when RF profiling is used.

The specific research question is: could RF sensing technology provide sufficient accuracy to navigate a robot to a wireless sensor device with only coarse RF profiling?

The answers to the problem have tremendous impact on WSNs in network lifespan and reliability. In terms of localization and navigation, such a robot must possess two capabilities: a middle range locating capability that does not require visual data, and a terminal locating capability to locate sensor nodes in close range, which most likely is vision aided.

We present a locating and navigation method based on distributed RF sensing. The objective is to accurately locate/track a target wireless sensor and the robot itself using only wireless beacon measurements and a coarse RF profile established in prior. Due to high non-linearity of the RF signal strength model, our localization method adopt a modified particle filtering [17]. Our method also takes into account that the observation data is a vector function consisting of measurements from independent beacon nodes. This is different from the scalar observation function assumed in conventional sensor. Such would mandate how the robot location estimation will be updated.

To estimate the target sensor location, we calculate the cumulative Maximum Likelihood (ML) probability of the received observation vector. This requires us to determine which beacon nodes should be included in the active set for location estimation. Two beacon set selection strategies are

*This work was supported by NSF under Award Award No.1040254

[1] Mathematics and Computer Science, Virginia State University, 1 Hayden Rd, Petersburg, VA USA jwang@vsu.edu

studied and the results show that the cumulative beacon approach outperforms a single beacon by a significant margin. The proposed ML probability model is used to update the posterior location probability of the robot.

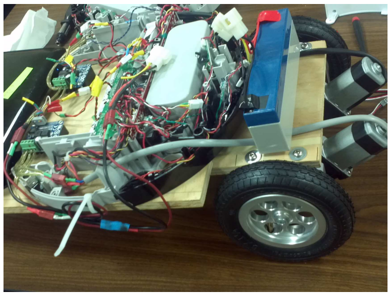


Fig. 2. Prototype of UGV equiped with RF sensor and onboard processing unit

Our navigation method adopts a novel way to determine the optimum motion angle using directional RSS measurements. We use a ’indirect matching’ technique to calculate the motion direction corresponding to the *shortest RSS path*. Our experimental data shows that such a method results in a better motion direction than the shortest path direction in Euclidean space.

In the rest of the paper, Section 3 gives an overview of the problem. Section 4 presents our navigation algorithm in an indoor setting. Section 5 discusses the estimation of optimum motion angles in an outdoor environment.

II. Background and Literature Review

Despite many drawbacks, using RF sensing as localization and navigation technology has several advantages in WSN. RF sensing is more reliable and less terrain-limited compared to the GPS technology. Another benefit is that no additional hardware is required since RF sensing is integrated in most wireless communication chip. In a large WSN network a hop-by-hop navigation plan [6], [9], [12] is usually required.

There exists a wealth collection of robot location estimation methods in a global or locally defined coordinate system. When infrastructure is not available, the location knowledge is typically obtained by analyzing local measurements from a range sensor, a visual sensor (landmarks, or special tags [11]), sound echoes (as in sonar), or a radar sensor, combined with local odometer or GPS data when available. Typically a Bayesian Network based computational method, such as EKF or particle filter [14], [17], [3], will be used to produce location estimation. This is accomplished by processing two streams of input data: one from **Process or Motion Model** such as odometer and IMU sensor, and the other is **Observation Model** such as ranger sensor, GPS sensor, and in our case, radio sensor.

In an infrastructure environment such as mobile cellular network [15], the time-of-flight (TOF), angle-of-arrival (AoA), or signal strength of beacon radio signals could be used.

The feasibility of radio sensing as a localization technique for WSN are discussed in [2], [4], [10], [12]. In the simplest scenario, a single RSS measurement offers rough estimation of relative distance between a transmitter and a receiver. Bahl etc [4] reported that in a complex indoor environment, RSS based empirical method has higher accuracy than model-fitting based method. Whitehouse etc., [6] demonstrates that RSS with low power radios can be used for direct distance estimation in an ideal open, outdoor environment. However simple RSS localization has limited applicability due to the complex radio propagation in real scenario. Some of the main reasons that contribute to the difficulty in RSS based localization are: (1) radio channel could be very unpredictable, (2) reflections of the signal against walls, floor, and (3) severe multi-path interference at the receiving side.

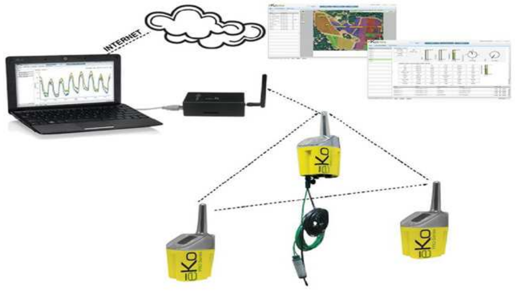


Fig. 3. Wireless Sensor Network in VSU farm

Many schemes have been proposed to improve the RSS localization technique. In [5], a distributed sensing algorithm is used to cancel RF measurement errors for an indoor environment. To reduce the localization noise, a stream of RSS measurement are observed and processed by Kalman filter [14] and particle filter [12].

III. Proposed Method Overview

Most existing RSS based localization methods only consider RF sources in a stationary environment. In our investigation, the localization and navigation problems are considered from a mobile robot equipped with a directional antenna. This gives us some unique advantages that are not considered in previous WSN location studies: the mobile robot can perform RSS measurements of multiple beacon sensors from different spatial locations.

The high level navigation algorithm consists of a three-step loop:

1) resample the candidate location of the robot
2) estimate the location of the robot and the target sensor with new RSS data.
3) determine a motion distance and angle θ.

We assume a set of N beacon nodes $\{s_1, s_2, ...s_N\}$ and a mobile wheeled robot m. The general 2-D robot state (position) is given by $S(k) = (x_r, y_r, \theta)$. We assume a simple Motion (or Process) Model where the robot position is update by odometry difference between consective time

$$S(k+1) = S(k) + \Delta(x, y, \theta) + w_0 \quad (1)$$

where k is the discretized time, (x, y, θ) is the grid location of the robot and its orientation, and w_0 is the odometry noise,

To estimate $S(\hat{k})$ we use sensor observations $SS_i(S(k))$, which is the RSS measurement of the i-th beacon node at the current robot location. SS_i follows observation model:

$$\hat{SS}_i(S(k)) = SS_i(S(k)) + w_i \tag{2}$$

Here we assume a Gaussian random noise term w_i.

We use a familiar particle filtering framework: we maintain a set of particle (robot location candidates); at the arrival of new RSS measurement, the posterior probability of each location candidate are re-evaluated; a new population is re-sampled based on posterior probability. The multiple observation sources allow continuous refinement of the location estimation [14] [10]. Secondly, the utilization of a directional antenna in our localization process proves to be significant. Such an option has not been exploited in the past, partially because directional antenna requires much more space than a single dipole antenna.

The project's original goal was to maintain the high-tunnel greenhouse WSN at VSUs Randolph Farm. The network was deployed in 2009 to collect environmental data in the greenhouse environment (Figure 2). From time to time, the network would not be accessible due to a failed relay node or poor signal quality. An all-terrain field robot called PatchBot has been developed as a mobile service node that could retrieve a troubled node and fill the hole in the network. The first robot prototype is built upon a 6-wheel platform. Each wheel is driven by a gear motor and independently controlled. The low level driver is implemented with a Cyclone II FPGA to provide a rich set of driving signal. The second prototype (See Figure 2) utilized a modified iRobot Create chassis with a major upgrade in motor driver (up to 43 Amp) and gear motor. High level navigation, as well as communication and task dispatch, is implemented on top of the Tekkotsu framework [16] and Robotic Operating System (ROS).

IV. Indoor hallway Localization

The indoor hallway assumption lead to a 1-D world and the robot state variable is degenerated to a single offset d.

The sensor input (observations) is RSS of beacon packets measured by m or other nodes of interest. In our indoor hallway problem setting, the RSS is described by a nonlinear function $SS_i(d)$, where d is the location of the mobile robot in the reference frame and the subscription i denotes the $i{-}th$ beacon node. The objective of the navigation algorithm is to estimate $\hat{d}$ given observations $\hat{SS}_i$. We further assume the following observation model:

$$\hat{SS}_i(d) = SS_i(d) + w_i \tag{3}$$

The main difficulty is that the actual radio model is often non-linear and dependant to the particular physical environment. Hence we opt to use an empirical model based on RF profiling.

One immediately notice that the observation data is a vector function, instead of the conventional scalar function. Such would mandate how the robot location estimation will be updated. In the following, we first discuss our experiments to establish necessary the indoor observation model— RF profile for all beacon nodes. We will then present the effectiveness of Maximum-Likelihood estimation algorithm combined with different beacon selection strategies.

A. Indoor radio profiling

We recorded radio signal strength (RSS) as a function of the sensor location in the hallway setup. RSS is reported in units of dBm. The RSS measurement is conducted in the hallway next to the Robotic Lab. Three wireless sensor nodes are programmed to broadcast beacon packets at channel 0x0B (2405 MHz) of ZigBee frequency band. The channel is selected to minimize the interference from local WIFI activity. Each beacon packet contains a local sequence number and a sensor node ID to distinguish it from other beacon packets (when there are multiple beacon nodes). The beacon nodes are positioned along the wall of the hall about 2 feet above the ground. The respective locations of the beacon/sensors are shown in Figure 3. Beacon node 6 is placed at the left side of the hallway, node 5 in the middle, and node 3 at the right side of the hallway. The three wireless sensors are programmed to transmit BEACON packets every 200 ms. CSCA is used at the node MAC layer to avoid packet collision.

To receive the beacon packets, a crossbow MIB510 wireless base station is connected to the robot's onboard computer (an ASUS netbook). The MIB510 is programmed to listen to the wireless channel and report all received beacon packets. The physical layer packet from MIB510 base station includes a received signal strength indicator (RSSI) and the link-quality-indicator (LQI) fields. RSSI is reported in scale from 0 (-91 dBm) to a maximum observation of 84 (-7 dBm). In a test run, the PatchBot will move along the hallway from the left side while collecting BEACON packets.

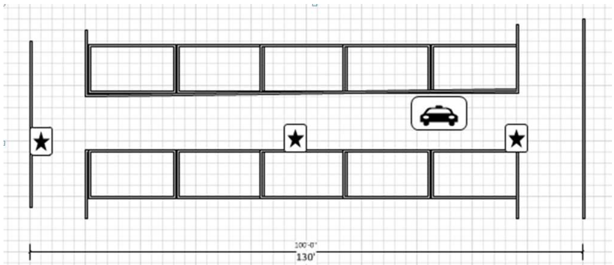


Fig. 4. Hall way RSSI measurement

The measurement result is shown below in Figure 5. As the robot moves from one end of the hallway to the other end, the signal strength observed changes accordingly, but in a non-trivial manner. The measurement of a particular beacon reaches the strongest point when the robot is close to or at the corresponding sensor node. The signal strength decreases as the robot moves away from the beacon node. The strongest signal strength is in the range of 40 (-51 dBm). The lowest RSSI observed is 12 (-79 dbm) before packet becomes completely undecipherable.

Our observations confirmed many finds on indoor RSS measurement reported in literature. We find that the standard deviation of slow fading is normally small (within 2 dBm). The RSS measurement result also shows a strong multi-path effect as the RSS is not monotonic of distance.

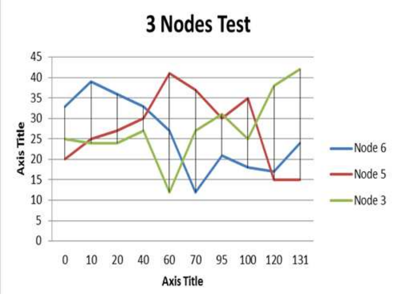


Fig. 5. RSSI Measurement from three beacons

B. Maximum Likelihood (ML) Navigation

The Maximum-likelihood estimation maximizes the posterio probability of the robot location based on current RSS observations and a set of candidate locations P. As in typical particle filtering implementation, P is re-sampled from the prior location estimation using the Motion model and odometer record.

The general form of Maximum-Likelihood estimation of the robot location d is as the following:

$$\hat{d} = argmax_{d\in P}\{\sum_{i\in R} Pr_i(S\hat{S}_i|d)\} \qquad (4)$$

This model is called **Cumulative-Maximum-Likelyhood**, since we sum the conditional probability Pr_i for all observable beacon nodes. The active beacon set is defined by $R = \{i|SS_i > \xi\}$, where ξ is the tolerance margin of radio sensor. The computation of Pr_i is based on the RF profile obtained above and the Gaussian model in equation (1). The selection of the active beacon set R could have significant impact on the results. Several other possible strategies are:

- **Fixed one**: one particular beacon node is fixed and not changed during the navigation course.
- **Strongest RSS first**: R only contain the beacon node with the highest RSS reading. The rational of this strategy is that high RSS reading means less noise interference.
- **Closest to Target first**: here R will contain the beacon node that is closest to the target sensor.
- **Highest gradient first**: here R will contain the beacon node whose RSS at the estimated robot position is the steepest.

The additional requirement is that all beacon nodes will measure the RSSI reading of the target sensor node and report the result to the robot. The ML procedure will be used to estimate the location of the target sensor node.

Using the above estimations, the probability $P_(r,t)$ of the robot observing $SS_i(r,t)$ can be calculated. If $P_(r,t)$ is greater than a pre-defined threshold, the result is accepted and the robot will drive towards the estimated target node.

C. Localization Results

Figure 5 shows the cumulative distribution function (CDF) of the location error based on the **Fixed One** strategy. The location error is calculated through 5 test runs where the robot is manually driven through the hallway. We re-randomize the particle set every 1 foot to emulate a robot kidnapping event. In general, we observed that the location accuracy provided by the three test beacon nodes is very location dependant. For example, using beacon node #6, the location error in the middle section of the hallway is mostly below 2 feet. However the error increase to 10 feet in the two end sections of the hallway. Such is the direct result of the non-linear RF profile.

Figure 6 shows the localization error using **Additive-Running-Set** strategy with three beacon nodes. The reduction in localization error is significant compared to the simple **Fixed one** approach. In most position the location error is below 1.5 feet, and the highest location error is 2.7 feet, a very acceptable result.

We hence conclude that the cumulative ML method with 3 beacon nodes is able to provide sufficient accuracy for hallway localization even with sever RF multipath.

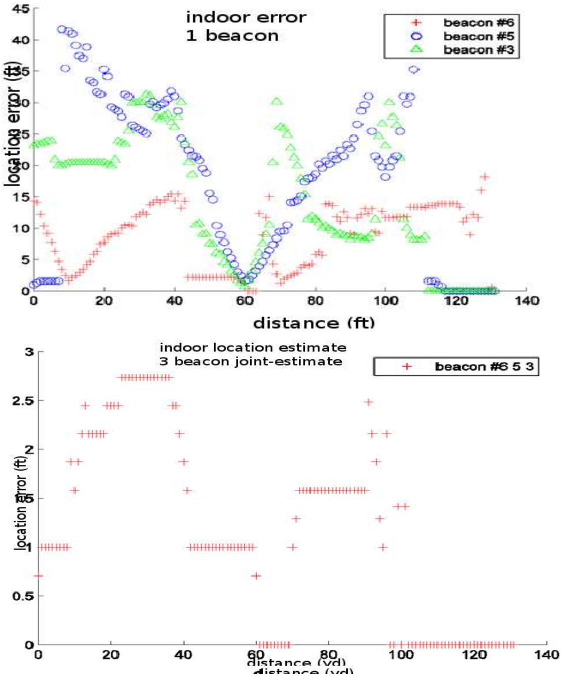


Fig. 6. Hallway Test

V. Outdoor 2D Navigation

In an outdoor environment, the locations of the robot and the target sensor node are estimated in a similar manner as described above, except that the RF profile is established in a 2-D grid. In typical 2D navigation, the estimated grid locations for the target sensor and the robot are sufficient to decide the movement direction θ, which is the direction of the shortest path in Euclidean space. However this usually demands high accuracy in rotational tracking [1]. We derive the motion angle differently by examining the shortest path in the *RSS space*. This is accomplished by comparing local beacon measurements at the robot and that at the target sensor node. These measurements are then matched against the RSS profile to determine θ.

The directional antenna plays a critical role in the navigation algorithm. Depending on whether the robot can hear the target sensor node or not, two different modes of navigation are possible: (1) Mode 1 (direct matching): If the RF of the target node is functional and the robot is close enough to hear beacon packets from the target node, the robot will scan the directional antenna and record beacon packets from the target node. The antenna-bodyframe angle with the strongest RSS reading is selected as θ. The robot will rescan after it moves forward by 1 foot. (2) Mode 2 (indirect matching): If the target node can't be heard, the robot is guided by matching the RSS reading of the beacon nodes between the robot and the target node. We refer to this navigation mode as indirect matching since we are comparing the RSS of beacon packets instead of packets directly from the target node. More detail will be discussed.

The rationale behind the *indirection matching* is because of the 2-D nature of outdoor field and (rotational) odometer drifting in some terrain [1], which makes navigation more complicated than the indoor case. While the odometer-based orientation tracking has a cumulative error, the relative orientation between the robot's directional antenna and the beacon nodes show a rather static error. The use of a directional antenna alleviates both problems and is the main source of improved locating accuracy.

A. Navigation base on Indirect Matching

The indirect matching method allows the target sensor node to be several hops away from the robot, assuming that the beacon measurement $\vec{SS}_t$ from the target node is successfully relayed to the robot via WSN. If the target node fails to provide current beacon measurement, the multi-hop navigation algorithm can still utilize WSN network topology to determine a closest neighbor node as an alternate navigation goal. This topic however is beyond the scope of this paper.

With N beacon nodes, we assume a partial RF profile field function

$$\vec{\mathbf{SS}}(\mathbf{x},\mathbf{y}) =< \mathbf{ss_1}(\mathbf{x},\mathbf{y}), \mathbf{ss_2}(\mathbf{x},\mathbf{y}), ...\mathbf{ss_N}(\mathbf{x},\mathbf{y}) > \qquad (5)$$

where $ss_i(x,y)$ is the RSS profile of the $i-th$ beacon signal at (x,y). If point (x,y) is one of the preselected measurement points, referred to as grid points, the actual measurement value is used. For a non-grid point, $ss()$ is interpreted linearly from its neighboring grid points.

The navigation loop can be described as follow:

1) Denote $\vec{SS}^t = (ss_1^t, ss_2^t, ...ss_n^t)$ as the beacon measurement at the target node, the location of the target wireless sensor node $(\hat{x_t}, \hat{y_t})$ is estimated by
$$(\hat{x_t}, \hat{y_t}) = argmax_{(x,y)}\{\sum Pr_i(ss_i^t|(x,y)\} \qquad (6)$$
This is obtained with the Maximum-Likelyhook estimation algorithm discussed in section 3.
2) The RSS measurement at the robot is denoted by $\vec{SS}^r(\theta) = (ss_1^r(\theta), ss_2^r(\theta), ...ss_n^r(\theta)$. The angle of the directional antenna is taken from a range $\theta \in [0^o \; 270^o]$.
3) Resample the particle set for the robot location P, the location of the robot itself $(\hat{x_r}, \hat{y_r})$ is estimated using a processed version $\vec{SS}^r = (max(ss_1^r(\theta)), max(ss_2^r(\theta)), ...max(ss_n^r(\theta)))$.
$$(\hat{x_r}, \hat{y_r}) = argmax_{(x,y)\in P}\{\sum Pr_i(ss_i^t|(x,y)\} \qquad (7)$$
4) calculate $\delta_{SS} = |\vec{SS}_t - \vec{SS}_r|$, if $\delta_{SS} < \zeta$, stop.
5) Else, an optimum movement direction θ^* is calculated such that the RSS discrepency δ_{SS} will be reduced the most.
6) The robot will travel along θ^* for D meters.

The optimun θ^* is selected by a local greedy procedure:

- Descretize θ_i in the scanning range $[0 \; 270^o]$ at a step size of 0.06^o.
- calculate a potential location
$$P_i = (\hat{x_r} + D.\sin\theta_i, \hat{y_r} + D.\cos\theta_i) \qquad (8)$$
- calculate the projected RSS at location P_i based on RSS profile:
$$\vec{SS}(P_i)$$
- calculate the potential gain: $\delta_{SS_i} = |\vec{SS}_t - \vec{SS}(P_i)|$
-
$$\theta^* = argmin_{\theta_i}\delta_{SS_i} \qquad (9)$$

B. Directional Antenna Setup

A motorized plat directional antenna assembly is fabricated to provide additional navigation accuracy at the close range. The antenna is mounted in front of the robot. A Kondo digital servo controls the antenna to scan a specific area or points to a particular angle. The maximum scan range of the servo is 270 degrees and the resolution is 270/4096 degree/step.

The RSSI reading from the directional antenna is strongest if the radio source is in the middle of the antenna's field of view. The readings fall off as the target move to the side of the field view. The further away the radio source is to the robot, the better angular resolution we can get. If the radio source is too close to the robot, the direction becomes indistinguishable due to RF signal saturation. At close range, the directional antenna is effective with the radio source placed one foot from the robot. The figure below shows the measured RSS of a source node three feet away from the robot.

C. Outdoor experimental results

Our outdoor experiment is conducted on VSU football field. Three Beacon nodes are deployed at the end of the field. To evaluate the performance of the navigation algorithm, we place a test wireless sensor node in the field as the target. The navigation algorithm will stop if it believes itself within

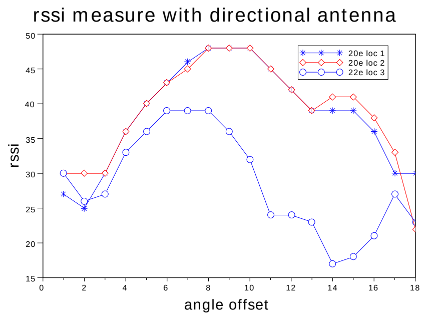


Fig. 7. RSS vs Angular offset measured by Directional antenna in room HM22E, target-robot distance: 3 feet.

1 foot from the target. The actual distance to the target is measured manually as the location error.

The measurement points are located from the goal line up to the 110-yd line every 10 yards; the total length of the field is about 330 feet. On each measured yard line, 5 data points are selected from sideline to sideline, making a grid size of 30 x 50 feet.

The test area is about 160 feet in length. This is determined by a minimum RSS reading of 5. Although the Beacon packets can still be received after 110 yard line, there are many noticeable packet drops and moving robot further line will not generate any location estimation. Figure 8 (top) shows the 2-D RF profile we collected in the testing field from one of the beacon node. The bottom subfigure shows the navigation error when we place the test node at different locations along the middle of the field. The results show that the navigation error is closely related to the relative distance between the target sensor and the beacon nodes. The starting location of the robot has little effect on the final result. The location error is less than 5 feet as long as the target sensor is within 120 feet from the beacon.

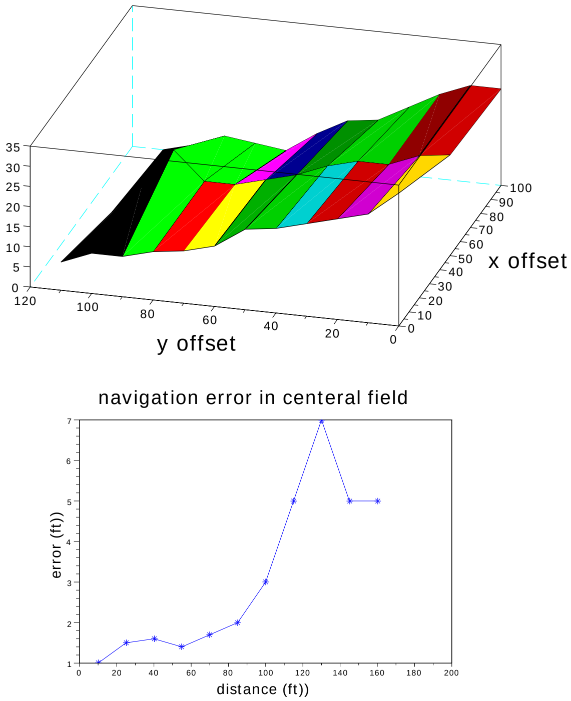


Fig. 8. (top) Outdoor RF profile, (bottom) location error

VI. Conclusions

We present a radio sensing based location and navigation method for outdoor WSN applications. The method utilizes a vector of RSS measurements to estimate the location of the target sensor and optimum navigation direction. Our result shows that RSS based navigation can achieve reasonable accuracy in a coarsely profiled field.

References

[1] O. Hach, R. Lenain, B. Thuilot, P. Martinet, Avoiding steering actuator saturation in off-road mobile robot path tracking via predictive velocity control, IROS11, San Francisco, USA, september 26th - 29th, 2011.

[2] Sebastian Thrun, Dieter Fox, Wolfram Burgard, Frank Dellaert, 2001, Robust Monte Carlo localization for mobile robots, Artificial Intelligence: 128,

[3] Nando de Freitas, et al., “Diagnosis by a Waiter and a Mars Explorer”, Proc Of The IEEE, Special Issue on Sequential State Estimation, April 2003.

[4] P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In INFOCOMM, 2000.

[5] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In Proceedings of ACM Sensys-04, Nov 2004.

[6] Kamin Whitehouse, Chris Karlof, and David Culler, A practical evaluation of radio signal strength for ranging-based localization, SIGMOBILE Mob. Comput. Commun. Rev. 11, 1 (January 2007), 41-52.

[7] Sangjin Han , Sungjin Lee , Sanghoon Lee , Jongjun Park , Sangjoon Park, Node distribution-based localization for large-scale wireless sensor networks, Wireless Networks, v.16 n.5, p.1389-1406, July 2010

[8] T. Oka, M. Inaba and H. Inoue, 1997, Describing a Modular Motion System based on a Real Time Process Network Model, in Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[9] J. Bachrach and C. Taylor, 2005, Localization in Sensor Networks, in Handbook of Sensor Networks: Algorithms and Architectures.

[10] Cesare Alippi, Giovanni Vanini, Wireless Sensor Networks and Radio Localization: a Metrological Analysis of the MICA2 received signal strength indicator, 29th Annual IEEE International Conference on Local Computer Networks (LCN’04).

[11] Edwin Olson, Johannes Strom, Rob Goeddel, Ryan Morton, Pradeep Ranganathan, and Andrew Richardson, Exploration and mapping with autonomous robot teams, ACM Commun, 56, 3 (March 2013), 62-70.

[12] S. Lanzisera, D.T. Lin, and K.S.J. Pister. Rf time of ight ranging for wireless sensor network localization. pages 1-12, June 2006.

[13] David Culler, Deborah Estrin, and Mani Srivastava. Guest editors introduction: Overview of sensor networks. Computer, 37(8):41-49, 2004.

[14] F. Gustafsson et al., ”Particle filters for positioning, navigation, and tracking,” IEEE Transactions on Signal Processing, 2002.

[15] Tong Liu, Paramvir Bahl, Imrich Chlamtac, Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks, IEEE Journal on Selected Areas in Communications, vol. 16, no. 6, 1998

[16] David S. Touretzky and Ethan J. Tira-Thompson. 2005. Tekkotsu: a framework for AIBO cognitive robotics. In Proceedings of the 20th national conference on Artificial intelligence - Volume 4 (AAAI’05), Anthony Cohn (Ed.), Vol. 4. AAAI Press 1741-1742.

[17] Jayesh H. Kotecha, et al., “Gaussian Particle Filtering”, IEEE Trans on Signal Processing, Vol.51, No.10, 2003, pp 2592 - 2601.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session III

Interactive session

- **Title: Task and motion plannig for Selective Weed Conrol using a Team of Autonmous Vehicles**
  Authors: I.A. Hameed, A. la Cour-Harbo, K.D. Hansen

- **Title: Remaining Range Indicator System for Electric Vehicle**
  Authors: R. Potarusov, I. Kobersy, J.-P. Lebacque

- **Title: Landmark Discovery for Single-View Cross-Season Localization**
  Authors: Ando Masatoshi, Chokushi Yuuto, Tanaka Kanji

- **Title: Obstacle Detection and Avoidance from an Automated Guided Vehicle**
  Authors: Roger Bostelman, Will Shackleford, Geraldine Cheok

- **Title: Frontier Based Exploration with Task Cancellation**
  Authors: P.G.C.N. Senarathne, Danwei Wang

- **Title: A Pareto Front-Based Multiobjective Path Planning Algorithm**
  Authors: Alexander Lavin

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Task and motion plannig for Selective Weed Conrol using a Team of Autonomous Vehicles

I A Hameed*; A la Cour-Harbo; K D Hansen

Aalborg University, Faculty of Engineering and Science, Dept. of Electronic Systems, Section of Automation and Control,
Fredrik Bajers Vej 7C, 9220 Aalborg, Denmark
* Corresponding author: ih@es.aau.dk

***Abstract*— Conventional agricultural fields are sprayed uniformly to control weeds, insects, and diseases. To reduce cultivation expenses, to produce healthier food and to create more environmentally friendly farms, chemicals should only be applied to the right place at the right time and exactly with the right amount. In this article, a task and motion planning for a team of autonomous vehicles to reduce chemicals in farming is presented. Field data are collected by small unmanned helicopters equipped with a range of sensors, including multispectral and thermal cameras. Data collected are transmitted to a ground station to be analyzed and triggers aerial and ground-based vehicles to start close inspection and/or plant/weed treatment in specified areas. A complete trajectory is generated to enable ground-based vehicle to visit infested areas and start chemical/mechanical weed treatment.**

Keywords—path planning; turning trajectory; motion control; ASETA;

I. Introduction

Adaptive Surveying and Early treatment of crops with a Team of Autonomous vehicles (ASETA) is a research projected funded by the Danish Council for Strategic Research and aims to explore the efficient and safe task execution and cooperation between a number of ground-based and airborne vehicles and its use in the early detection and treatment of weeds in row crops e.g. sugar beets [1]. In traditional automated weed control systems, unmanned ground vehicle (UGV) scans the surface of the entire field area and identify weed species and treats it chemically or mechanically directly. The main disadvantage of these approaches is that the robot has to scan or comb the whole field looking for weed spices which is a time consuming and costly process and in addition, more crop plants are most likely to be subjected to a potential full or partial damage. In this project, surveillance is carried out based on small unmanned helicopters equipped with a range of sensors, including multispectral and thermal cameras. A path-planning algorithm for efficient unmanned aerial vehicles (UAVs) guidance based on a predefined set of waypoints and dubbin curves is used [2]. The helicopters scan large field areas, data collected are transmitted to a ground station, which analyses the data and triggers aerial and ground-based vehicles to start close inspection and plant treatment in specified areas. Small-scaled helicopters are used to provide the system with multi-spectral aerial images. Using data from the helicopters, the system identifies infestations and intensive weed spots in the field and then dispatches autonomous ground vehicles to the infestations to exactly identify and localize the weeds [1-3].

UGVs require high-precision control, continuous operation, increased efficiency, and the removal of a human operator from an unsafe environment. Although autonomous vehicles have been for long the subject of research, only recently have sensor and computer technology made autonomous vehicles practical [4]. The advent of Global Positioning Systems (GPS) sensors which has offered engineers the high precision necessary for accurate vehicle control and the relatively inexpensive computers which are capable of running complex control and estimation algorithms make it practical for real-time control. With all the tools necessary for economical real-time land-vehicle control, specific commercial applications are stimulating research into effective vehicle control systems. Agriculture has emerged as one of the first potential applications of real-time vehicle control. Certain types of repetitive farming tasks such as seeding, spraying, fertilizing, weed control, and harvesting could benefit from high-precision control, control which is available in all visibility conditions [4].

In this paper, a motion planning and control approach to give the UGV the capabilities to visit weed spots in the optimized manner is developed. This includes; 1) the generation of a path which follows the crop rows to minimize crop damage, 2) the generation of an optimized turning path at the end of crop rows to join rows and to enable a UGV to drive smoothly between these rows in a manner which reduces soil compaction, operational time and total travelled distance in the field, and 3) a trajectory following controller, namely the Helmsman Controller (HC), to follow the desired trajectory is proposed.

The entire work is organized as follows; we first introduce the mechanical specification of the UGV platform used in ASETA project (see Fig. 2) in Section 2.2. The dynamic model of the used UGV is presented in Section 2.3. In section 2.4, two controllers are presented. A Field coverage approach is introduced in Section 3.1. In Section 3.2, an approach for assigning rows which are the closest to the weed spots are presented. Different types of turning trajectory generation for a blind turning are presented in Section 3.3. Finally, the developed approach is tested and validated through a number of simulation experiments are performed to evaluate the tracking performance of the robot under different field and operational conditions. Finally, a brief concluding remarks and future work in Section 5 are presented.

II. Unmanned Ground Vehicle (UGV) Platform

A. RobuRoc Robot

The provided Unmanned Ground Vehicle (UGV) is a commercial four wheel skid steering vehicle called RobuROC-4 from Robosoft®, four-wheel drive (4WD) vehicle with custom on board computer, sensor suite, and camera setup as it is shown in Fig. 1. This robot is built for outdoor use and has 4 independent wheels mounted. They are powered by 4 brushless DC motors working together two on the left and two on the right, making the robot able to turn on the spot. The top speed is rated to 2 m/s and its weight is approximately 140 kg and up to 100 kg extra payload can be added [1]. The robot has an embedded controller, called RobuBOX, running Windows XPe. The robot is equipped with odometer, proxymeter, and a bumper sensor. The robot has a wireless emergency stop and can be driven in two modes, by a Xbox360 wireless gamepad or robuBOX commands. The commands can be sent by a UDP connection to the robot via Ethernet, and can be used for setting angular and linear velocities, and requesting data from sensors in the RobuROC-4. The sensor data is returned from the robot at a specified rate (e.g., 10 Hz).

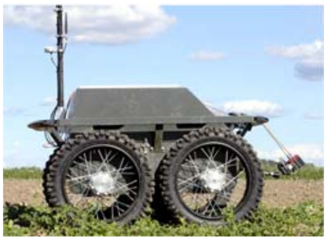


Fig. 1. UGV platform.

B. Robot Controller

The main goal of this section is to develop a controller to enable the robot to follow a predefined trajectory under normal field conditions. Two controllers are developed and used simultaneously; one for speed control (i.e., open-loop trajectory following controller) and the second one is for position control (i.e., Helmsman controller). Block diagram of the robotics feedback control system is shown in Fig. 2.

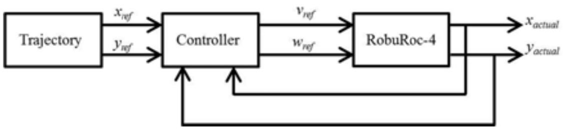


Fig. 2. Robot control system where v_{ref} and w_{ref} are the desired linear and angular velocities.

A trajectory following controller (TFC) is developed in two ways; (1) Open loop control form where the desired heading angle and consequently the desired angular and linear velocities are obtained from the reference trajectory through differentiation, and (2) Closed loop control form based on Helmsman controller principal, shown in Fig. 3. In the lower level, two PI controllers are used to regulate the linear and angular speeds of the vehicle motors.

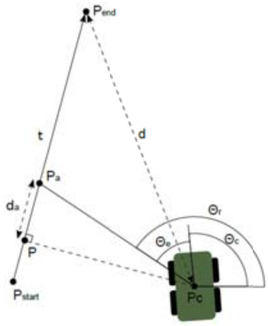


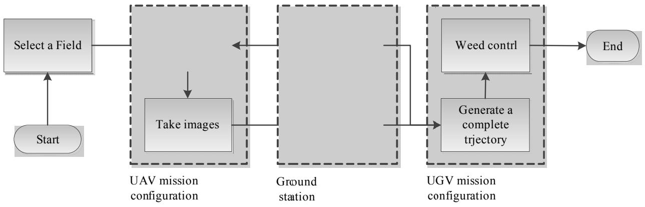
Fig. 3. Helmsman controller principall [1].

In Helmsman controller, two standard PI controllers are used for adjusting the vehicle"s linear and angular speed in a manner which reduces the distance error, *d* (m), and heading angle error, θ_e, respectively, where P_{start} is the start waypoint of the current segment, P_{end} is the target waypoint, P_a is the aiming point on the current line segment *t*, P_c is the vehicle's current position, and *P* is the projection point of P_c on line segment *t*. d is the distance from the vehicle's current position to the target position (m), d_a is the distance from aiming point to the projection point (m), θ_r is the reference angle, θ_c is the current heading angle of the vehicle.

III. Task and Motion Planning Approach

A. Mission planner

The system consists of a UGV and an unmanned airborne vehicle (UAV). Each vehicle has allocated a great deal of onboard autonomy and provided with a high level controller that is capable of performing its own behaviors path planning, mission execution, and obstacle avoidance. However, the coordination between these vehicles was handled by a centralized component known as the mission planner. The primary purpose of the mission planner is to provide the system with a mechanism of which allows it to carry out tasks with a high degree of autonomy. The process starts by selecting a field, scanning it using UAV, collecting data and sending it to a Ground Station (GS) where data analysis takes place, sending coordinates of weed spots back to the UAV and UGV for further inspection and then task execution, as it is shown in Fig. 4.

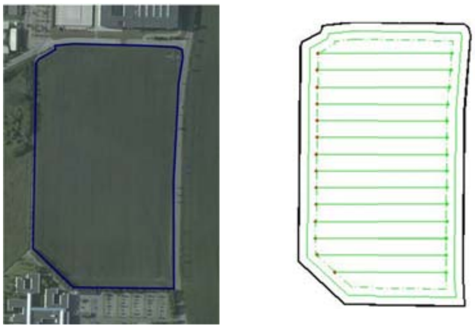


Select a Field
Start
Take images
UAV mission configuration
Ground station
Weed contrl
Generate a complete trjectory
UGV mission configuration
End

Fig. 4. System block diagram.

B. *Path planning*

Field coverage is the process in which field tracks and headland paths are generated in a manner to cover the whole field area. In recent years, many field coverage approaches have been developed to generate an optimized coverage path regardless of the complexity of the field shape and the presence of obstacles [4-7]. Different techniques have been developed to optimize field operations in a manner which minimizes operational time, cost and maundering over field area [8-10] through optimizing is first generated in terms of the driving angle and operating width of the vehicle. First the field is geometrically represented; a tool developed by Hameed et al. is used [7]. The 2D geometrical representation of a field involves the generation of a geometrical map which is made up by discrete geometric primitives, such as points, lines, and polygons, providing a concise representation of the environmental data that can be readily used for operational planning. The input consists of the set of coordinates of the points on the field boundary, the operating width of the implement, the number of headland paths, and the tested driving direction. The tool generates the set of the parallel field-work tracks (i.e., rows) for the complete field area coverage and gives as an output the coordinates of the points representing the starting and the ending point of each track or row, and of the points representing the headland paths, as it is shown in Fig. 5.



(a) (b)

Fig. 5. Example implementation of the 2D geometrical field representation: (a) Satellite image of a field of an area of around 4.5 ha located at (57° 0' 51.56" N, 9° 59' 32.97" E) with the outer field boundary in blue (b) The geometrical representation of the field for an operating width of 20m, a driving angle of 0°, and a single headland path

C. *Trajectory Generation*

In this section, a complete trajectory to enable the vehicle to visit the intensive weed spots is generated. First the closest field rows or tracks from the weed spots are selected and ordered in a manner which reduces the total driving distance and the maneuvering over headland area. Second, turnings between the selected and ordered field rows are generated and combined with the rows to create a complete trajectory to guide the UGV through the execution of the weed control operation. Complete trajectory generation is carried out through the following steps:

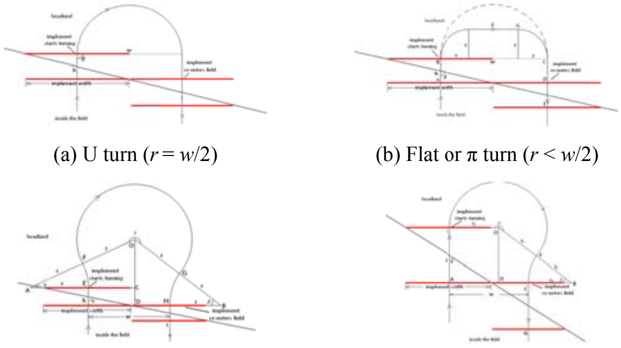
1) *Rows assignment*

A set of coordinates representing the locations of the intensive weed spots is obtained from the helicopter. Each weed spot is assigned to a track where the vehicle can use to reach that spot. The perpendicular distances between crop row, *i*, and weed spot, *j*, are calculated using the formula given by Eq. (1). The spot with less perpendicular distance, d_{ij}, to that crop row is assigned to that particular row.

$$d_{ij} = \left\| \left(t_{x_i}, t_{y_i}\right) - \left(w_{x_j}, w_{y_j}\right) \right\|_2 \qquad (1)$$

2) *Connecting selected crop rows*

At the end of each track the vehicle has to turn to enter to the next track. Relying on vision system to carry out a successful and safe turning is not reliable under some weather conditions. Also, sharp turnings should be avoided even if the robot is of the omnidirectional type which can turn around its center point because this type of turning can cause severe damage to soil structure and plants. In addition, a sharp turning may not be possible due to the mechanical restrictions of the vehicle and therefore a soft turning type which reduces maneuvering over headland area is required. In this paper, some typical turning types, shown in Fig. 6, will be used to join selected crop rows in a manner which reduces total travel distance and operational time [11].



(a) U turn ($r = w/2$)

(b) Flat or π turn ($r < w/2$)

(c) Bulb or Ω turn ($r > w/2$)

(d) Hook turn ($r > w/2$)

Fig. 6. Turning types where *r* is the minimum turning width of the vehicle and w is the distance between crop rows [11].

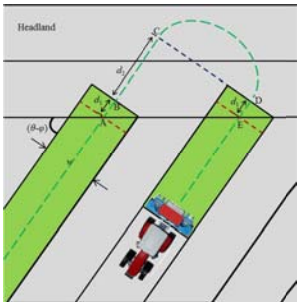


Fig. 7. Alignment of crop rows before the vehicle starts to turn; *w* is the implement width, *θ* is the driving or tracks angle and φ is the headland or edge direction [9].

3) *Rows alignment*

Before a turning trajectory is generated, rows must be aligned in order to carry out a successful turning. When the centre point of the implement, shown in Fig. 7, reaches point A, a part of the implement started to exit the interior of the field. However, in order to completely finish the coverage of the current swath or track, the vehicle needed to keep moving straight ahead until point B was reached. The vehicle cannot make a turn until both tracks are aligned by keep moving until point C is reached. The vehicle then made a turn from C to D, and starts to re-enter the field from D, until the entire width of the implement was inside the field at point E. Distances d_1 and d_2 are computed using the formulas given by Eq. (2).

$$d_1 = w/\left(2\times\tan\left(\theta-\varphi\right)\right)$$
$$d_1 + d_2 = norm(\mathrm{C}-\mathrm{D})/\tan\left(\theta-\varphi\right) \qquad (2)$$

IV. Results

The field shown in Fig. 5 is used to demonstrate the functionality of the developed approach. A simulated set of UTM coordinates representing the locations of the weed spots which are supposed to be received from a helicopter is shown in Fig. 8(a). For a minimum turning radius, r = 10m, a trajectory with simple turning types is generated as it is shown in Fig. 8(b) where only U and flat turning types are employed. For r = 15m a trajectory with more complex turning types using U and Ω turning types is generated as it is shown in Fig. 8(c). One drawback of the current approach is that the turns may be carried out outside the headland area, as it is shown in Fig. 8(c). Enlarging the headland area or increasing the number of headland paths could be used to maneuver the UGV without going outside the headland.

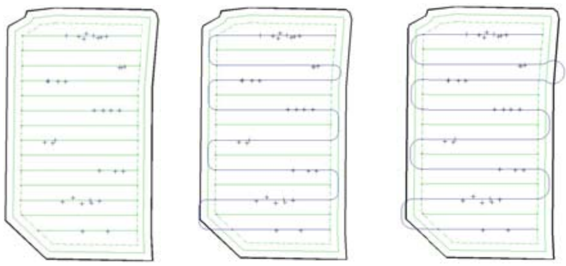

(a) (b) (c)

Fig. 8. (a) Coordinates of intensive weed spots received from a helicopter scanning the field, (b) a complete trajectory with U and flat turnings for r = 10 m, and (c) a complete trajectory with U and Ω turnings for r = 15 m.

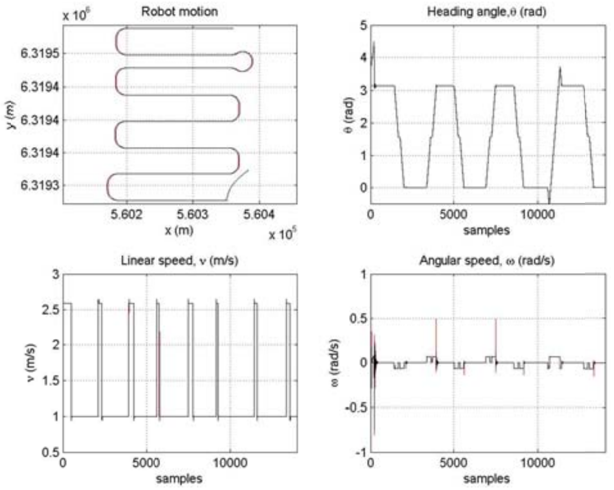
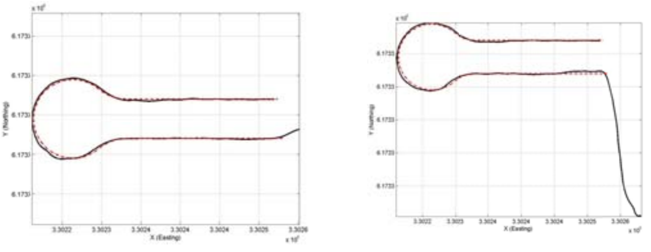


Fig. 9. Robot motion in case of using other turning types such as U, π and Ω turning curves (the robot initial position is at +57° 0' 48.42", +9° 59' 39.72" with a heading angle of 108°) where reference values are in red and actual values are in black.

The trajectory following controller is tested for a trajectory obtained for r = 15m where the controlled helped the robot model to follow complex turning shapes of U and Ω turning types as it is shown in Fig. 9. The actual and reference positions of the robot, heading angles, linear and angular speeds are shown in Fig. 9. The robot has also been tested in the real field trial starting from two different initial positions as it is shown in Fig. 10 a & b. The minimum turning distance of the robot was chosen to be 5 m and therefore a bulb- or Ω-turning type was obtained. The Helmsman controller was able to guide the robot from its initial position to the desired track and move in straight line to the other end of the track (i.e., crop row). At the end of the current track, a buld- or Ω-turning was generated and used to guide the robot through a smooth turning over the headland area. At the end of the turning trajectory, the robot entered the next track (i.e., next crop row) and the controller showed a good ability to drive the robot it in a straight line to the other end of the track.



(a) (b)

Fig. 10. Testing the robot's ability to follow a trajectory from two different initial positions: desired trajectory is in dotted red line and actual robot trajectory is in solid black line.

V. Conclusions

In this paper, a novel approach for generating a complete operational trajectory to enable unmanned ground vehicle to visit specific infested weed spots identified from aerial images for a crop field is presented. A complete trajectory consists of the selected crop rows and the turning path connecting these rows is generated to enable an unmanned ground vehicle to visit these weed spots in an optimized manner. Four types of turning types; namely, U-, π-, Ω-, and hook- turning types are generated and used to join selected crop rows. To enable a robot to follow a predefined trajectory, a trajectory following controller based on Helmsman controller principal is proposed. Simulation results showed that the controller is capable of accurately steering the robot to follow complex trajectories and reducing the tracking error under different operational conditions. An experimental field test showed that the developed controller is capable of following a predefined trajectory with high accuracy. In future work, the trajectory will be extended to 3D dimensional space in order to improve the controller performance for different field terrains.

References

[1] A. la Cour-Harbo, "ASETA Project – Adaptive Surveying and Early treatment of crops with a Team of Autonomous vehicles," retrived from URL http://www.aseta.dk, 2010.

[2] K.D. Hansen, F. Garcia-Ruiz, W. Kazmi, M. Bisgaard, A. la Cour-Harbo, J. Rasmussen, and H.J. Andersen, "An Autonomous Robotic System for Mapping Weeds in Fields", 8th IFAC Symposium on Intelligent Autonomous Vehicles, Austrial, 2013, 217-224.

[3] W. Kazmi, M. Bisgaard, K.D. Hansen, F. Garcia-Ruiz, and A. la Cour-Harbo, "Adaptive Surveying and Early Treatment of Crops with a Team of Autonomous Vehicles," In: *Proceedings of the 5th European Conference on Mobile Robotics*, 2011, 253–258.

[4] T. Bell, "Precision robotic control of agricultural vehicles on realistic farm trajectories," Doctoral dissertation, Stanford University, Stanford, CA, 1999.

[5] T. Oksanen, and A. Visala, "Coverage path planning algorithms for agricultural field machines", *Journal of Field Robotics*, 2009, 26(8), 651-668.

[6] S. Choudhury, and S. Singh, *"A Coverage Planning Algorithm for Agricultural Robots*", Technical Report, IIT Kharagpur, India, 2009.

[7] I.A. Hameed, D.D. Bochtis, C.G. Sørensen, and M. Nøremark, "Automated generation of guidance lines for operational field planning," *Biosystems Engineering*, 2010, 107(4), 294–306.

[8] I.A. Hameed, D.D. Bochtis, and C.G. Sørensen, "Driving angle and track sequence optimization for operational path planning using genetic algorithms," *Applied Engineering in Agriculture*, 2011, 27(6), 1077-1086.

[9] J. Jin, and L. Tang, "Optimal coverage path planning for arable farming on 2D surfaces", *Transactions of the ASABE*, 2010, 53(1), 283-295.

[10] A.A. Bakhtiari, H. Navid, J. Mehri, and D.D. Bochti, "Optimal route planning of agricultural field operations using ant colony optimization", *Agricultural Engineering International: CIGR Journal*, 2011, 13(4), 1-16.

[11] J. Jin, "*Optimal field coverage path planning on 2D and 3D surfaces*," Graduate Theses and Dissertations, Iowa State University, 2009, paper 11054.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Remaining Range Indicator System for Electric Vehicle

R. Potarusov*, I. Kobersy**, J.-P. Lebacque***

*IFSTTAR/COSYS/GRETTIA,
France-77447 Marne-La-Vallée
potarusov.roman@gmail.com
**Taganrog Polytechnic Institute of Don State Technical University
Russia-347900 Taganrog,
iskobersi@gmail.com
***IFSTTAR/COSYS/GRETTIA,
France-77447 Marne-La-Vallée
jean-patrick.lebacque@ifsttar.fr

Abstract— **This paper presents a Breadth-First Search-based Indicator System for remaining range estimation and representation in battery electric vehicle driving range indicators. The representation enables detailed illustration of electric vehicle's "distance to empty". To build up a remaining range graph the Breadth-First Search (BFS) algorithm is coupled with a simple electric energy consumption model taking into account the driver-desired speed, road data in a near horizon (route network topology, legal speed, grade and grip), ambient temperature, headwind speed and state of charge of electric battery. The presented study focuses on investigation of the effect of the ambient temperature variations on the vehicle remaining range. Simulation results clearly show an increased energy requirement at low temperature resulting in a reduction of the vehicle range.**

Keywords: advanced driver assistance systems, navigation, remaining range estimation, electric vehicle, energy consumption, breadth-first search.

I. INTRODUCTION

Since its creation in 2010 the Laboratory of Engineering of Surface Transportation networks and Advanced Computing (GRETTIA) of the French Institute of Science and Technology for Transport, Development and Networks (IFSTTAR) contributes to transport networks and systems development taking into consideration integration, intermodality, reliability and system analysis issues. The areas of research include road sector, collective, and particularly guided transport. In this framework, GRETTIA together with the Taganrog Polytechnic Institute of Don State Technical University participate in development of new approaches for remaining range estimation and representation in battery electric vehicle (BEV) driving range indicators.

Range estimation remains complicated by the fact that:

- Future driving behavior is often unknown [1,2].
- Road data as well as weather, traffic conditions (often uncertain) have to be taken into account [1], [3,4,5], [8].
- Batteries are subject to external influences and aging [4].

Several studies have been performed for conventional and electric vehicles to estimate/predict the remaining range or extend it by providing (for a given road segment) the optimal speed profile aiming at reducing the electric energy consumption. Rodgers et al. [1] investigated conventional and novel methods for estimating an electric vehicle's "Distance to Empty" (D_{TE}), the actual distance the vehicle can be driven before recharging is required. They proposed a Novel Regression-based D_{TE} algorithm that reduces the error in D_{TE} estimation if the future changes in driving conditions are detected beforehand by obtaining route information from the user.

Yuhe et al. [3] introduced an estimation method, in which to save time and computing resources, range estimation is classified into rough range estimation and precise range estimation according to remaining battery energy.

Besselink et al. [4] analyzed the energy usage and range of a battery electric ECE VW Golf, using over 20000 km of real life data. The study showed the impact of ambient temperature: in cold weather conditions additional energy was needed to heat the interior resulting in a higher specific energy usage.

To the authors' knowledge, modern indicators use two different ways to represent the remaining range estimation results on the road map (Fig. 1):

- By simply drawing a circle containing all the achievable road network nodes (in red).
- Detailed graph-based representation (in green).

To plan a ride we use two different on-board devices:

- A range indicator computing the vehicle's DTE.
- A GPS-navigator providing us with the optimal path/distance to the destination point.

Having that information in mind we then take a decision on whether to schedule a stop for refueling/recharging or not. **And why not to couple those two devices?**

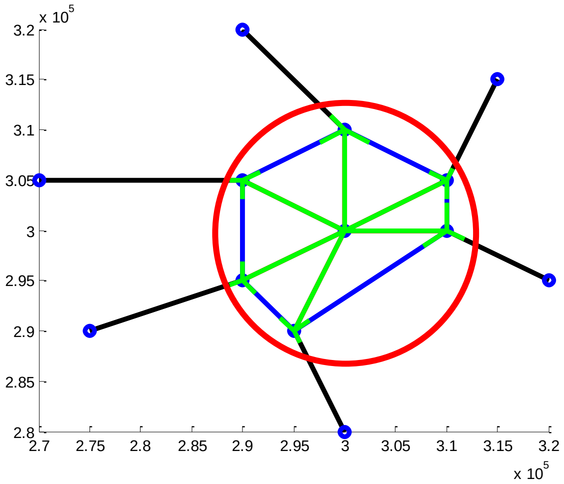


Figure 1. Circle- and graph-based representations.

For the purposes of the BFS-based Indicator System we have chosen the detailed representation of the output graph of road segments all starting from the initial EV position and ending at the farthest achievable road nodes (Fig. 2). The chosen graph-based representation allows defining more accurately the achievable road network zones. This makes it possible to easily couple the BFSIS with modern on-board GPS navigators.

This paper extends the study presented in [5] by investigating the effect of the ambient temperature variations on the EV remaining range. The remaining of this paper is organized as follows. Section 2 introduces the architecture of the BFS-based Indicator System and the consumption model used. Section 3 shows and discusses some simulation results. The conclusion of the paper and future work are given in section 4.

II. BFS-based Indicator System

The BFS-based Indicator System (BFSIS) uses the extension of the BFS [6] coupled with a simple electric energy consumption model to build up the BEV remaining range graph (in green, Fig. 2) taking into account the driver-desired speed, road data in a near horizon (route network topology, legal speed, grade, grip), ambient temperature and state of charge (SoC) of the battery.

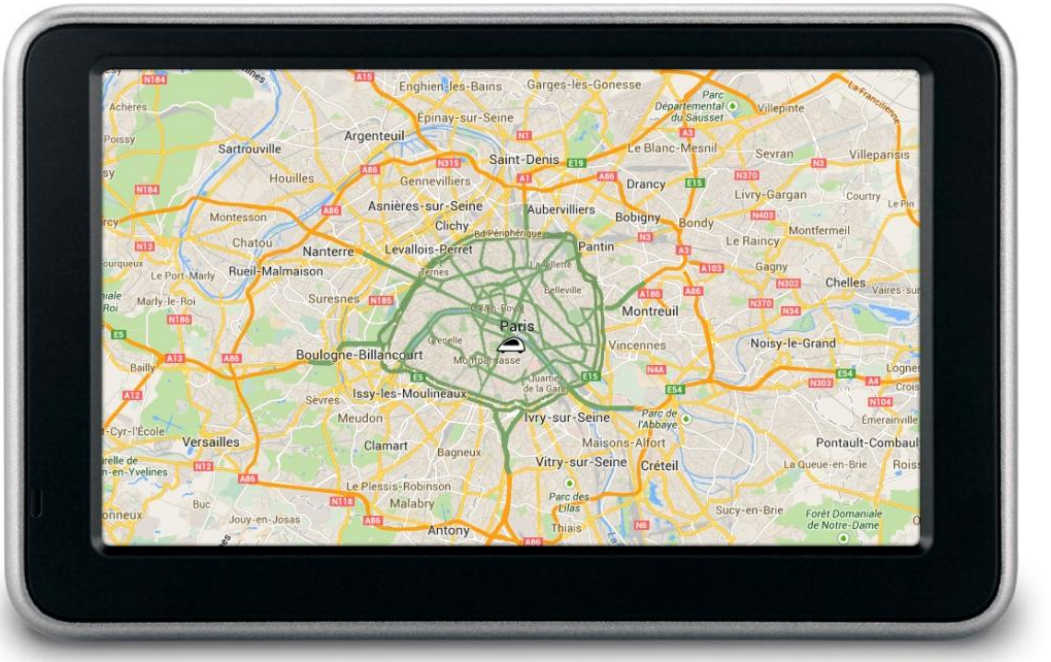


Figure 2. BFSIS illustration: remaining range graph (in green) on Google Maps.

TABLE I. Nomenclature

Variable	Description	Units
V	Average vehicle speed (within a road segment)	m/s
V_{driver}	Driver-desired speed	m/s
eHorizon	Data structure of predefined size containing the road data and weather forecast	-
Route graph	Road network graph (has to be known ahead of time)	-
Av. Vleg	Average road segment legal speed	m/s
Av. Grade	Average road segment grade (slope)	-
Av. Grip	Average road segment grip (road slipperiness or road friction)	-
Amb. Temperature	Outdoor air temperature	°C

Av. Vhw	Average headwind speed	m/s
SoC	Battery state of charge	-
Rem. Range Graph	EV remaining range graph	-
E_k	Kinetic energy	J
η_g	Transmission efficiency	-
T_e	Engine torque	N*m
w_e	Engine speed	rad/s
E_{elec}	Electric energy	J
E_{aux}	Energy requirement for heating/cooling	J
η_b	Battery efficiency	-
η	Electric motor efficiency	-
r	Wheel radius	m
M	Vehicle mass	kg
rho	Air volumetric mass	kg/m^3
SCx	Vehicle drag area	m^2
g	Gravitational acceleration	m/s^2
Crr	Rolling resistance coefficient	-

Low-level description

Fig. 3 below presents the low-level description of the BFSIS. The eH block converts the data of eHorizon into an appropriate format that can be read by the BFSIS. eH provides the following data for each road segment (input road graph arc):

- Road network topology.
- Average road grade.
- Average road grip.
- Average legal speed.
- Average outdoor air temperature.
- Average headwind speed.

We assume the driver always prefers to keep the maximum speed allowed within a given road segment:

$$V = \min(Av.Vleg, Vdriver).$$

Based on the provided data the output remaining range graph is constructed by the BFSIS as described below.

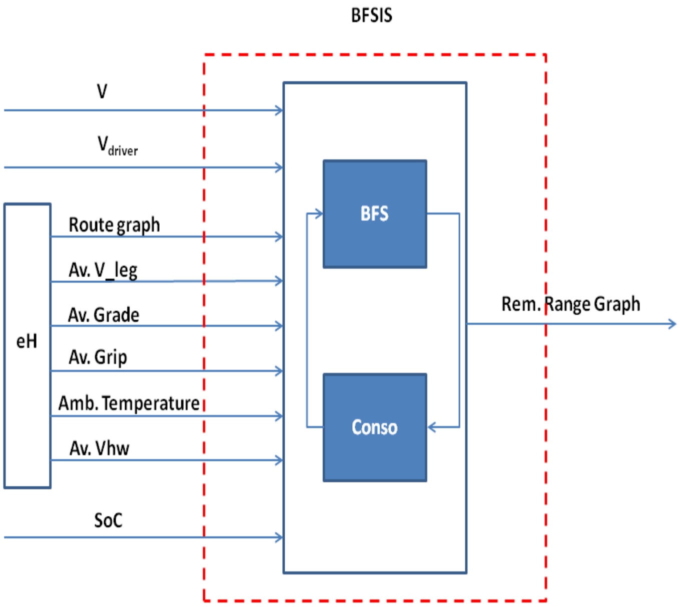


Figure 3. Block diagram of the BFS-based Indicator System.

Consumption model

The kinetic and the electric energies (consumed during a time slice *t*) are evaluated based on the following equations:

$$E_k = \frac{1}{\eta_g} T_e w_e.$$

$$E_{elec} = \frac{\eta_b}{\eta(T_e, w_e)} E_k + E_{aux}(Amb.Temperature).$$

The engine speed computation is performed based on the absence of gearbox:

$$w_e = V/r.$$

The torque T_e is computed as follows:

$$T_e = r * (0.5 * rho * SCx * (V + Vhw)^2 +$$

$$+ M * g * Crr + M * a)/2.$$

BFS approach for remaining range estimation and representation

The proposed extension of the BFS approach for remaining range representation is depicted below. It consists of two phases. The first phase calculates the path cost for each road segment by means of the consumption model described in the previous sub-section and returns the *RoadSegmentEnergyCost* array. Then the phase 2 performs the BFS algorithm which uses a queue data structure to store intermediate results as it traverses the graph, as follows:

Input: *NNodes* – number of nodes of the road network graph, *DCs* – set of direct children of a node, *Eelec* – *NNodes***NNodes* matrix of electric energy needs to traverse a road segment, *RemRange* – 1**NNodes* array of nodes' remaining range values, *Examined* – 1**NNodes* Boolean array of examined nodes.

Initialize: *RemRange*(initial vehicle location) = *SoC*, *Examined* [1..*NNodes*] = 0.

```
Enqueue the root node (initial vehicle location).
While the queue is not empty
    Dequeue a node, v = node, IndInVertices = index of v in Vertices set, v is labeled as examined, Examined(v) = 1
    for each $i \in DCs(v)$
        RayToDraw = min(Eelec(i,IndInVertices), RemRange(IndInVertices));
        Plot the remaining range ray
        if !Examined(i)
            Enqueue i
            RemRange(i) = RemRange(IndInVertices) – RayToDraw, i is labeled as examined, Examined(i) = 1
        endif
    endfor
endwhile
```

III. Simulation Results

The system is implemented in Matlab/Simulink. Table 2 and Table 3 summarize the vehicle and simulation parameters based on which the simulation has been performed. The efficiency profile of YASA-400 Advanced Axial Flux Electric motor [7] has been used for the simulations.

TABLE II. VEHICLE PARAMETERS

Variable	Value/Range
r	0.30
M	1700
SCx	0.7
η_g	0.9
η	[0.01, 0.95]
η_b	0.7

TABLE III. SIMULATION PARAMETERS

Variable	Value
Crr	0.012
V (initial)	16
Vdriver	30
SoC	From 0.35 to 0.2
eHorizon	100 km

Test scenarios

To simulate D_{TE} we have generated a route network graph depicted in Fig. 4. The graph contains 20 nodes, including the root node ($X = Y = 3.00$) where the EV is initially located.

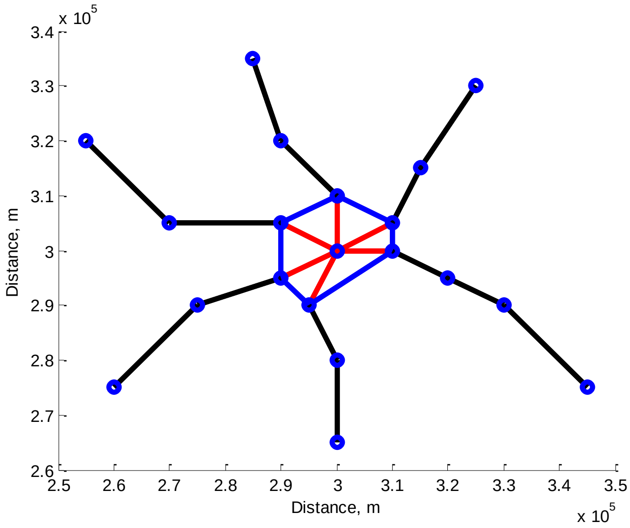


Figure 4. Road network topology.

Table 4 summarizes the *Av. Vleg* values used for all four test scenarios described below.

TABLE IV. AVERAGE ROAD SEGMENT LEGAL SPEED VALUES

Road segment color in Fig. 4	Value
Red	16
Blue	22
Black	30

Scenarios 1 to 4

All four scenarios are temperature-oriented. They investigate the effect of the ambient temperature variations on the EV remaining range. Table 5 lists the simulation parameters for each of them. All the parameters are applied to the entire road network. The scenarios differ from each other by the air density *rho* and the *Amb. Temperature* values.

TABLE V. SCENARIO 1 TO 4 SIMULATION PARAMETERS

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Av. Grade	0	0	0	0
Crr	0.012	0.012	0.012	0.012
Vhw	0	0	0	0
Amb. Temperature	-15	0	20	30
rho	1.37	1.29	1.20	1.16

The auxiliary power (heating/cooling) measurements (executed on a distance of 100 km) for different values of the ambient temperature have been performed by Tober [8] and presented in Fig. 5 below.

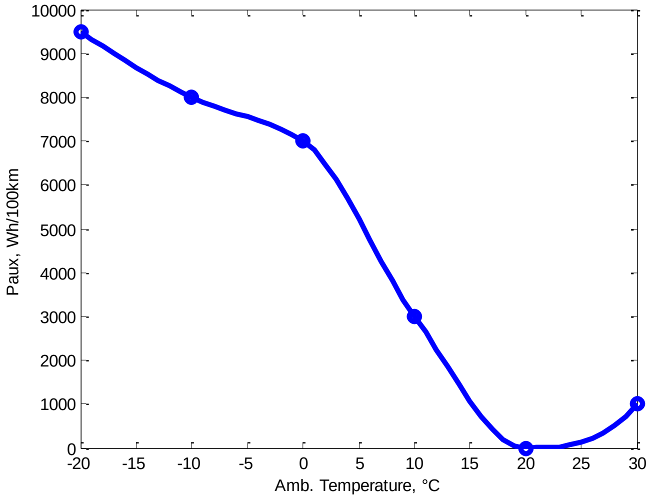
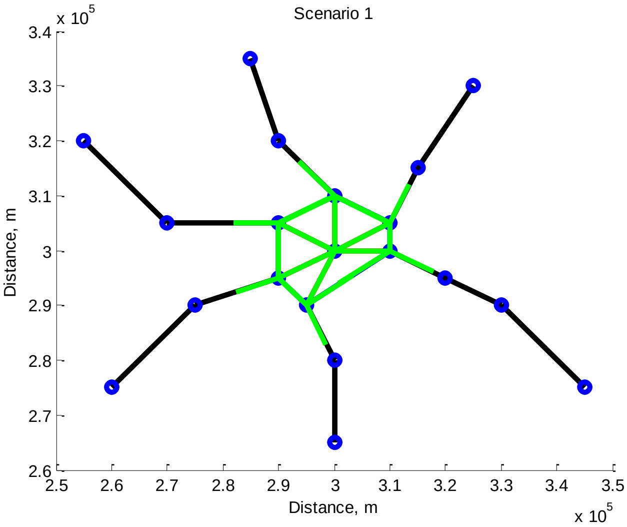
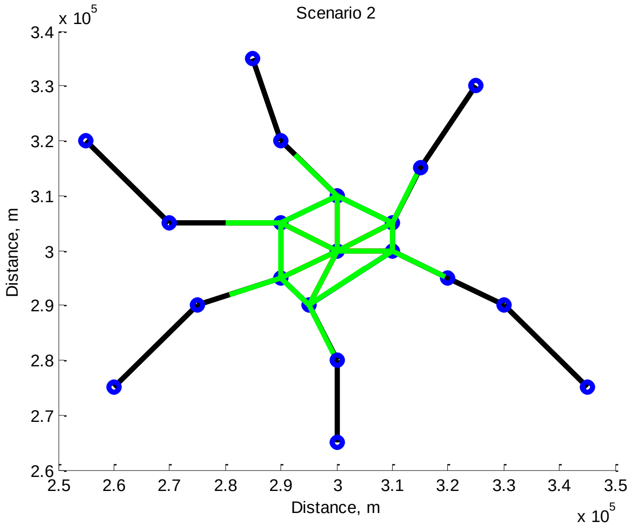


Figure 5. Energy requirement for heating/cooling as a function of the ambient temperature for Nissan Leaf electric car [8].

In Fig. 6 and 7 the proposed BFS Indicator System is used to output the remaining range graph for four different scenarios defined in Table 5.

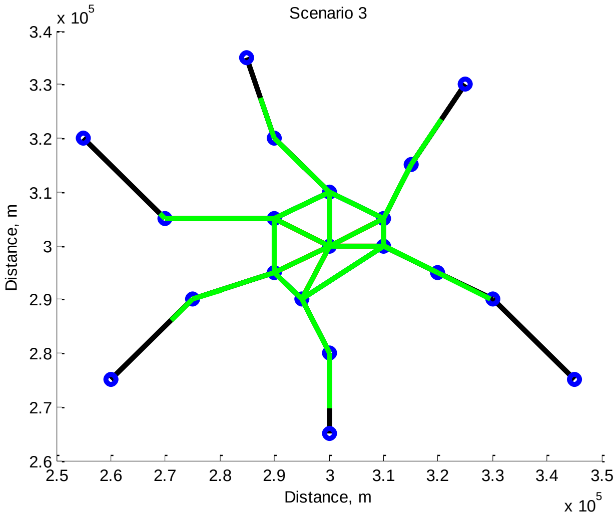


a) DTE ((3.0, 3.0) to (2.9, 3.2) via (3.0, 3.1)) = 18.8 km

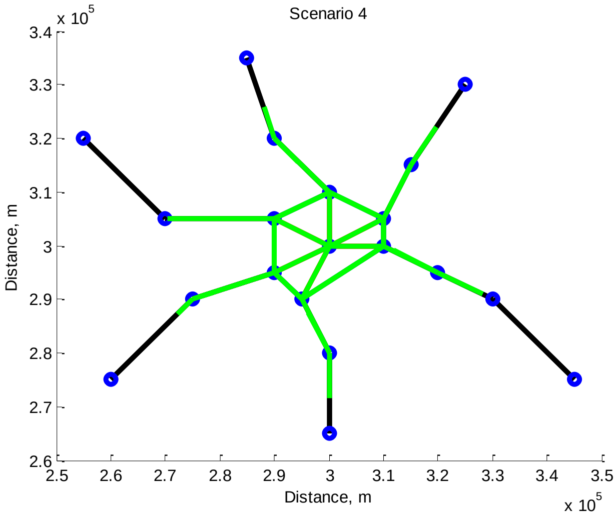


b) DTE ((3.0, 3.0) to (2.9, 3.2) via (3.0, 3.1)) = 21.5 km

Figure 6. Scenario 1 (a) and 2 (b) simulation results.



a) DTE ((3.0, 3.0) to (2.85, 3.35) via (3.0, 3.1) and (2.9, 3.2)) = 32 km



b) DTE ((3.0, 3.0) to (2.85, 3.35) via (3.0, 3.1) and (2.9, 3.2)) = 30.3 km

Figure 7. Scenario 3 (a) and 4 (b).

Results discussion

The Breadth-First Search-based Indicator System has been tested on the four previously described scenarios. The simulation results show that the simplified consumption model used for the purpose of the system reacts properly to the ambient temperature changes. The tests report an increased energy usage at low temperatures, resulting in a major reduction of the vehicle range. Scenario 1 is the 'coldest' scenario with its critical *Amb. Temperature* = -15°C and D_{TE} = 18.8 km. In the Scenario 2 the vehicle range slightly increases until D_{TE} = 21.5 km at *Amb. Temperature* = 0°C. In the Sceario 3 the weather conditions are optimal and there is no need to heat/cool the interior of the vehicle, thus resulting with the maximum D_{TE} = 32 km. Then in the last Scenario 4 the size of the remaining range graph (the number of green edges and their length) is affected by the energy requirements for cooling at 30°C, D_{TE} = 30.3 km.

To extend the vehicle range we propose to:

1) "Eco-drive", keeping, for example, the constant speed of 80 km/h (22 m/s) all over the road network.

2) Couple the BFS Indicator System with the Smart and Green ACC driver assistance system developed in [9].

IV. Conclusions and future work

In this paper, the Breadth-First Search-based Indicator System for electric vehicle has been tested on several temperature-oriented scenarios. The tests have reported an increased energy usage at low temperatures, resulting in a major reduction of the vehicle range. The BFS-based remaining range representation we have chosen for the purposes of the BFSIS allows detailed illustration of electric vehicle's "distance to empty".

Future developments will be oriented at using of a map-based service (Google Maps, OpenStreetMap or others) to show the output road graph overprinted in real time. Different traffic conditions will be taken into account when calculating the average vehicle speed and road segments' energy costs.

To validate the proposed system we build a moke-up, which will represent a road network drawn at a rectangular piece(s) of wood. The road segments will be built in such a way to represent uphill, downhill and slippery roads. Different average legal speed as well as different traffic density values will be assigned to all the segments of the road network.

The cars will be car-like robots moving from their initial locations to their destinations. All of them will have a small LCD direct user interaction and information display visualizing the road network and the RR graph overprinted in real time. The spectators will be invited to choose the initial location of a robot(s) and its destination point. Then the BFSIS will compute the time-optimal (or the most efficient in terms of fuel/energy consumption) path taking into account the road data in a near horizon (route network topology, legal speed, grade and grip), traffic conditions as well as ambient temperature. If the destination point is not achievable, the BFSIS will propose the optimal path via the nearest refueling/recharging station.

V. References

[1] Rodgers, L., Wilhelm, E., Frey, D.: Conventional and Novel Methods for Estimating and Electric Vehicle's 'Distance to Empty'. In: ASME IDETC/ATV, Portland, Oregon, USA (2013).

[2] Smart, J., Powell, W., and Schey, S.: Extended Range Electric Vehicle Driving and Charging Behavior Observed Early in the EV Project. SAE Technical Paper 2013-01-1441, 2013, doi:10.4271/2013-01-1441.

[3] Yuhe, Z., Wenjia, W., Kobayashi, Y., Shirai, K.: Remaining driving range estimation of electric vehicle. In: IEEE Electric Vehicle Conference, Greenville (2012), doi:10.1109/IEVC.2012.6183172.

[4] I.J.M. Besselink, J.A.J. Hereijgers, P.F. van Oorschot, H. Nijmeijer.: Evaluation of 20000 km driven with a battery electric vehicle. In: European Electric Vehicle Congress, Brussels (2011).

[5] Potarusov, R. Lebacque, J.-P.: Breadth-First Search-Based Remaining Range Estimation for Electric Vehicle. SAE Technical Paper 2014-01-0273, 2014, doi: 10.4271/2014-01-0273.

[6] Knuth, Donald E. (1997), "The Art Of Computer Programming Vol 1. 3rd ed.", Boston: Addison-Wesley, ISBN 0-201-89683-4.

[7] YASA-400 Advanced Axial Flux Electric Motor, http://www.yasamotors.com/products/yasa-400/

[8] W. Tober.: The energy requirement of battery electric vehicles under different conditions. In: 4th EVE Meeting, Geneva (2013).

[9] Potarusov, R., Nouveliere, L., Orfila, O., Glaser, S.: Smart and Green Adaptive Cruise Control for an Electric Vehicle: First Results. In: 7th IFAC Conference on Manufacturing Modelling, Management, and Control, Saint-Petersburg (2013), doi:10.3182/20130619-3-RU-3018.00190.

Landmark Discovery for Single-View Cross-Season Localization

Ando Masatoshi Chokushi Yuuto Tanaka Kanji

***Abstract*— We tackle a challenging task of single-view cross-season localization. The main problem we face is how to obtain discriminative and compact visual landmarks, which are necessary to cope with changes in appearance in an environment. We address this issue by proposing the use of raw image matching, which is contrastive to popular bag-of-words methods which rely on vector quantized visual features. A direct implementation of raw image matching can be time / space intractable due to the high dimensionality of raw image data. We propose to exploit raw image matching, not for the direct matching between query and database images, but for mining an available visual experience to find discriminative visual landmarks. The result is a bounding box -based scene descriptor that crops the mined landmark objects with respect to the visual experience. We develop a practical localization system, by employing both efficient and reliable subsystems for raw image matching, including RANSAC geometric verification, common pattern discovery, and approximate near neighbor search. Experimental results show that our proposed framework tends to produce stable localization results despite the fact that our scene descriptor is significantly space / time efficient.**

***Index Terms*— single-view localization, cross-season localization, landmark discovery, raw image matching**

I. Introduction

Cross-season robot localization has gained much attention in autonomous robotics today. Most of the work so far has concentrated on coping with changes in appearance of an environment [1]–[8]. A single place can look differently depending on the geometric conditions (e.g., viewpoint trajectories, object configuration) and the photometric conditions (e.g., illumination). Such changes in appearance lead to difficulties in scene matching, and thereby raise the need for highly discriminative, compact scene descriptor (Fig.1).

Existing approaches are broadly divided into those which describe a variety of visual appearances of scenes in a sole map, or those in which multiple independent maps are employed to describe different visual experiences. [1] developed a state-of-the-art robust localization framework, called SeqSLAM, in cross season navigation tasks separated by months or years and in opposite seasons. However, SeqSLAM algorithm explicitly assumes that image sequence measurements (i.e., multiple views) are available for robot localization and relies on an image sequence -based scene descriptor. [2] proposed a robust approach that can capture the typical time varying appearance of an environment in multiple different maps and the number of experiences required tends to a constant. [3] showed that by quantizing local features in both feature and image space, discriminative statistics can be learned on the co-occurrences of features at different times of the day. However, it is not clear whether those approaches can make the scene description compact as they directly memorize a variety of multiple visual experiences. A notable exception is [4], where the issue of compactness is addressed with a question: “how little and what quality of visual information is needed to localize along a familiar route? ”. Although impressive results have been demonstrated, it also relies on the assumption of image sequence measurements.

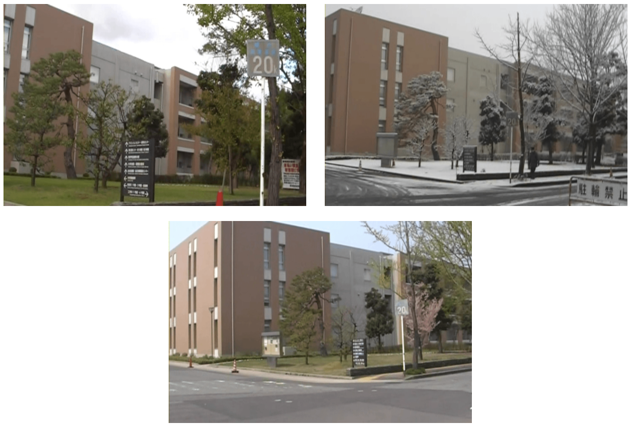


Fig. 1. Single-view cross-season localization. A single place can look quite differently depending on the geometric conditions (e.g., viewpoint trajectories, object configuration) and the photometric conditions (e.g., illumination). Such changes in appearance lead to difficulties in scene matching, and thereby raise the need for highly discriminative, compact scene descriptor.

In this paper, we tackle a challenging task of single-view cross-season localization. Single-view localization is a family of localization tasks with important applications, where the robot’s view sequence only sparsely overlaps with pre-mapped views. The main problem we face is how to obtain discriminative visual landmarks, which are necessary to cope with changes in appearance in an environment. We address this issue by proposing the use of raw image matching, in contrast to popular bag-of-words methods which rely on vector quantized visual features. A direct implementation of raw image matching can be time / space intractable due to the high dimensionality of raw image data. We propose to exploit raw image matching, not for the direct matching between query and database images, but for mining an available visual experience to find discriminative visual landmarks. The result is a bounding box -based scene descriptor that crops landmark objects with respect to the visual experience. We develop a practical localization system by employing

This work was partially supported by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) 23700229 and Grant-in-Aid for Scientific Research (C) 26330297 (“The realization of next-generation, discriminative and succinct SLAM technique: PartSLAM”), by KURATA grants, and by the Telecommunications Advancement Foundation in Japan.

M. Ando, Y. Chokushi, and K. Tanaka are with Faculty of Engineering, University of Fukui, Japan. tnkknj@u-fukui.ac.jp

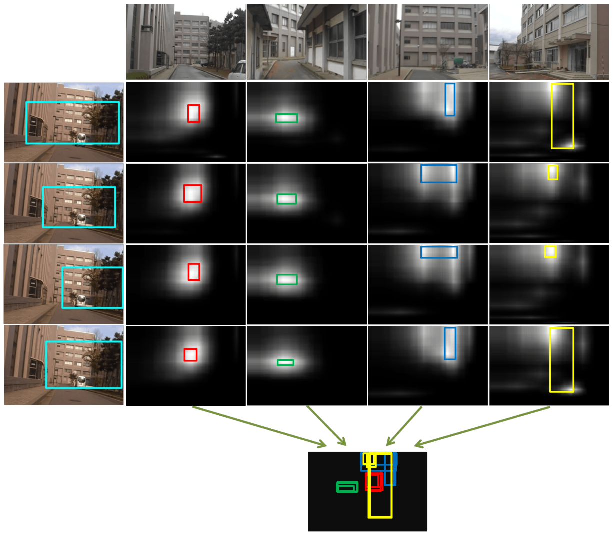


Fig. 2. The landmark discovery process. Left-most column: $I = 4$ subimages with different views (cyan rectangular boxes) are sampled from an input query / database image. Top-most row: $J = 4$ reference images that are most similar to the input image, are searched by an efficient SIFT-based approximate near neighbor search, followed by a RANSAC step to ensure the geometric consistency between the tentative SIFT correspondences. 4x4 panels in the right bottom: A raw image matching between each input subimage and each reference image is performed to find common visual patterns, i.e., discriminative landmarks. Bottom-most: the result is a discriminative and compact scene descriptor that consists of $IJ = 16$ bounding boxes.

efficient matching techniques, including RANSAC geometric verification, common pattern discovery, and approximate near neighbor search. We evaluate the proposed framework through challenging single-view cross-season localization tasks.

II. Localization System

The localization task consists of two main steps: (1) scene description, and (2) database search. The first step interprets a query scene image into a scene descriptor. Descriptors are also computed for all images in the map database. The second step searches over the map database to find images with similar descriptors to the query descriptor. Then, image with highest similarity score is viewed as the localized image. The both subtasks are respectively detailed in the following subsections.

A. Scene Description

To interpret a given scene image into a scene descriptor, we assume a dictionary or a library of random L reference view images to be given (Fig.2). Those reference images are not required to be associated with spatial information (e.g., viewpoint, location ID, geotag) such that the viewpoint and orientation are known. Such images are cheaper than those mapped images with spatial information required by the database, and can be more easily available. For example, they could be a visual experience obtained by the robot-self in a previous navigation, or shared via information sharing networks by other colleague robots [5], and potentially, they could be resource publicly available image data on the web, such as Google StreetView.

A small subset of J appropriate reference images that are most similar to a given input image are selected and used to interpret the image. Our experimental results suggest that high localization performance tends to be associated with coverage of the robot's route by these reference images. To select most similar J reference images $\{\mathscr{R}_j\}_{j=1}^{J}$ to a given input image, the pairwise similarity between the input and each reference image is evaluated as the number of similar SIFT matches between the image pair. Approximate near neighbor search (ANN) [9] is iterated for efficient search of similar SIFTs to each query SIFT, followed by a RANSAC step to ensure the geometric consistency between the tentative SIFT correspondences, and those J similar images that are supported by highest number of SIFT matches are considered as the relevant reference images. Note that although this is a naive implementation of our approach, it does not rely on quantized SIFT vectors as the popular bag-of-words image model do. Instead, our image model is based on the precise, raw SIFT features. Although it is computationally more demanding, the search process is fast as we only require a small library of reference images.

Then, common pattern discovery (CPD) between an input and the reference images is performed to mine a set of rectangular image patches, termed visual phrases (VPs), that effectively explain an input image. Any CPD algorithm could be adopted, but for our purposes, we selected the algorithm of randomized visual phrase (RVP) [10], as it provides fast and stable detection of common visual patterns and can generally handle scale variations among objects without relying on any image segmentation or region detection.

The original RVP algorithm employs a bag-of-words image representation. We do not rely on vector quantized visual features, but do raw SIFT matching between the input and each reference images. We assign unique ID to each reference SIFT feature, and do the ANN over the reference to find similar SIFT features to each input SIFT feature, followed by a verification step to ensure the normalized L1-distance between the SIFT descriptor pair is smaller than 0.4. We then assign IDs of the similar SIFTs as visual words to the input SIFT feature, i.e., multiple visual word per feature. Note that our bag-of-words image representation is free from vector quantization errors.

Given the bag-of-words image representations of the reference and the query images, the RVP algorithm efficiently evaluates the likelihood of the query object being located at each pixel on the reference image. The result is a voting image whose pixel indicates the evaluated likelihood. We binarize the voting image to obtain the bounding box. This binarization process employs a threshold that is determined on-the-fly, so that the size of a bounding box should be sufficiently small that it can be localized well, and should not exceed 10% of the area of the reference image. We also take into account the fact that the views of input and reference images are usually different depending on their viewpoints. To address the difference of viewpoint, a collection of random I subimages are sampled from each query image, and used as the view images with slightly different viewpoints. Then, CPD is iterated for each pairing of input subimage and the reference image. We finally obtain I bounding boxes for each of the J reference images.

B. Database Search

The scene description process outputs a compact scene descriptor, in a form of bag-of-bounding-boxes (BoBB), which consists of J pairings of

- a reference image ID (an integer),
- I visual phrases (BBs on the reference image).

Because a BB is a far lower dimensional representation than many existing feature descriptors such as 128 dimensional SIFT vectors, the search of similar BBs to a query BB can be done quite quickly. Note that a BB carries appearance information of a VP as it indicates size and location of common visual patterns with respect to the visual experience. Suppose a function $Overlap(\mathscr{B}_{i,j}, \mathscr{B}'_{i',j'})$ returns the area of overlap between a given BB pair $\mathscr{B}_{i,j}$, $\mathscr{B}'_{i',j'}$ when they belong to the same reference image or 0 otherwise. Note that our current implementation ensures that each bounding box is well localized, i.e., smaller than 10% of the image area, and we found there is no need to penalize the size of bounding boxes. A large value of $Overlap(\mathscr{B}_{i,j}, \mathscr{B}'_{i',j'})$ indicates that the VPs cropped by the BBs are similar between the image pair, and vice versa. By aggregating the VP-level similarity, we obtain the image-level similarity:

$$f_{VP}(\mathscr{I}, \mathscr{I}') = \frac{1}{IJ} \sum_{j=1}^{J} \sum_{i=1}^{I} \max_{i',j'} Overlap(\mathscr{B}_{i,j}, \mathscr{B}'_{i',j'}). \quad (1)$$

The bag-of-bounding-boxes (BoBB) descriptor does not produce meaningful results when there is no common visual pattern between the input and the reference scenes. We propose to use the traditional bag-of-words (BoW) descriptor complementary with the proposed BoBB descriptor, and a modified image-level similarity:

$$f(\mathscr{I}, \mathscr{I}') = C_{VP} \cdot f_{VP}(\mathscr{I}, \mathscr{I}') + C_{VW} \cdot f_{VW}(\mathscr{I}, \mathscr{I}'), \quad (2)$$

where C_{VP}, C_{VW} are weighting coefficients and $C_{VP} \gg C_{VW}$. Currently, we use the FAB-MAP as the bag-of-words method and view its output likelihood value as the similarity f_{VW}.

III. Experimental Results

We evaluate the performance over several data sets that are collected in different seasons and paths. The dataset used in these experiments consists of collections of view images taken around a university campus, using a hand-held camera as a monocular vision sensor.

Fig.3 shows a bird's eye view of our experimental environments, viewpoint paths, and examples of the dataset. We consider a typical scenario that deals with view images that are taken relatively far (1m-5m) from each other [2], greatly reducing the memory required to describe a given path. Occlusion is severe in the entire scenes, and people and vehicles are dynamic entities occupying the scene. We went each path three times, collected three independent collections of images and use each for query, library and database image collection.

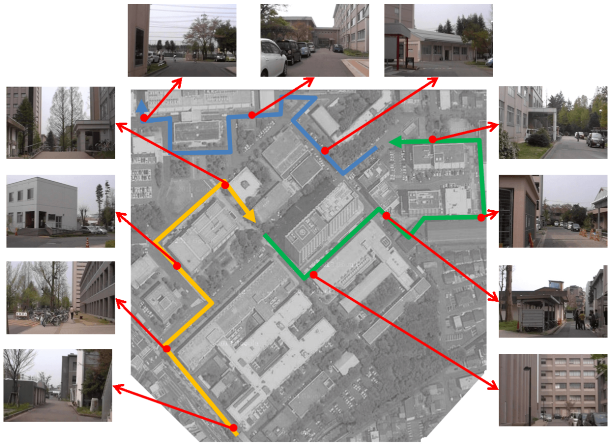


Fig. 3. Experimental environments. A collection of 448-592 images are collected for three different seasons, autumn (2013/10/21), winter (2013/12/21) and spring (2014/4/17) and for three different paths #1, #2 and #3, respectively indicated by orange, green and blue curves on the figure. Shown in the bottom three rows are examples of single-view cross-season dataset collected in spring, autumn, and winter.

Fig.4 shows example results of landmark discovery. For common pattern discovery (CPD), our method selects a set of $J = 4$ reference images out of the size $L = 100$ library and learns $I = 4$ VPs for each reference image, based on ANN and CPD as described in the previous section. One can see that discriminative landmarks are successfully found for all the image pairs shown here. However, the reasons for each landmark found are various depending on the content of the input and reference images. In the first case, an image of a house that commonly appear in the input query or database image is mined via the visual experience mining, and a window of the house is found as a discriminative landmark via the landmark discovery. In the second case, there are mainly three dominant objects, (from left to right) trees, white house, and large building, and among them, only the house is selected as the discriminative landmark, as the appearance of the house does not change among the scenes. In the third case, large portion of the scene is occupied by dynamic objects (e.g., cars), and despite the difficulty, a part of the house is successfully selected as the discriminative landmark. In the fourth case, a part of one of dominant building object is selected as the landmark.

Fig.5 shows peformance results. We evaluated the proposed VP method ("VP") in terms of the retrieval accuracy and compare it with FAB-MAP 2.0 ("FAB-MAP") [11]. For FAB-MAP, we used the same code as the authors. Series of independent 200x3 retrievals are conducted for each of 200 random query images for each of the 3 different paths. The retrieval performance was measured in terms of averaged normalized rank (ANR) in %, which is a ranking-based retrieval performance measure, where the smaller value is better. To evaluate ANR, the rank assigned to the ground-truth relevant image was evaluated for each of the 200 independent retrievals, and then the rank was normalized by the database size and averaged over the 200 retrievals. From Fig.5, one can see that our approach outperformed FAB-

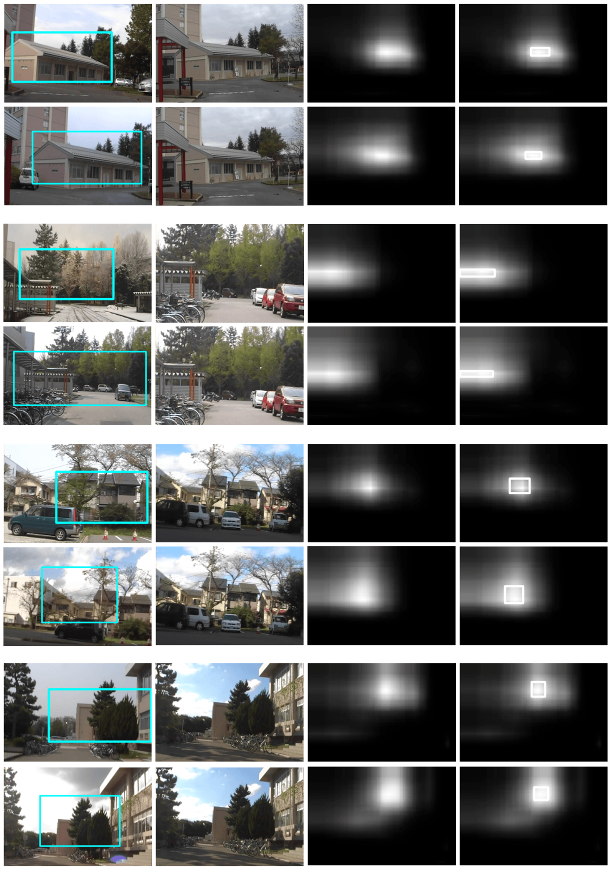


Fig. 4. Example results of CPD.

MAP in most of the retrievals considered here.

Fig.6 shows the input query image, the ground truth image, and the database images topranked by FAB-MAP and by the proposed method. One can see that both the FAB-MAP and the proposed method tend to find database images that are similar to the query image in some sense. However, FAB-MAP tends to fail when there are confusing images in the database whose appearance is partially similar with the query image but with different structure. Shown in the 3rd row in the figure is a typical exmaple where FAB-MAP could not distinguish two different buildings with partically similar appearance and quite different structure, i.e., flip left and right. On the other hand, the proposed method successfully distinguish such a confusing image, due to the discriminative power of raw image matching used as the matching criteria.

IV. Conclusions

The main contribution of this paper is to tackle the challenging tasks of single-view cross-season localization and to propose a novel discriminative and compact scene descriptor. In contrast to popular bag-of-words scene descriptor that relies on vector quantized feature vectors, our criteria for scene matching is based on raw image matching. Instead of direct raw image matching between query and database images that is space time intractable, we proposed to do raw image matching between a query or database image and known reference images, e.g., visual experience, information shared by other colleague robots, or publicly available image data on the web. We developed a practical localization system, by employing both efficient and reliable subsystems for raw image matching, including RANSAC geometric verification, common pattern discovery, and approximate near neighbor search. Experimental results showed that our proposed framework tends to produce stable localization results despite the fact that our scene descriptor is significantly space / time efficient.

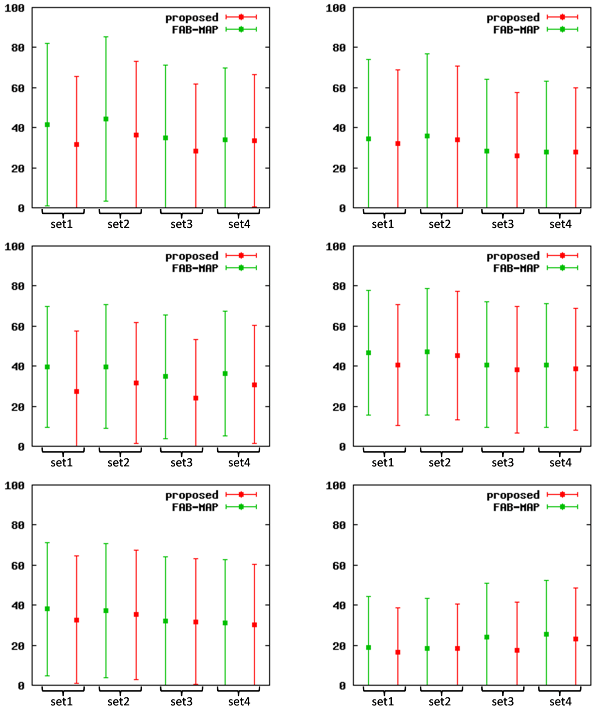


Fig. 5. Performance results. We conducted scene retrieval experiments for 6 different combinations of query - database / library images: autumn -winter, autumn - spring, winter - autumn, winter - spring, spring - autumn, and spring - winter. For each experiment, we constructed and used 4 different sets of query (Q), library (L), and databsse (D) images: from left to right, set#1={ Q:path1+path2, L:path1+path2, D:path1+path2 }, set#2={ Q:path1+path2, L:path2+path3, D:path1+path2 }, set#3={ Q:path2+path3, L:path2+path3, D:path2+path3 }, set#4={ Q:path2+path3, L:path1+path2, D:path2+path3 }

References

[1] Michael Milford and Gordon Fraser Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *ICRA*, pages 1643–1649, 2012.

[2] W. Churchill and P. Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *ICRA*, pages 4525–4532, 2012.

[3] E. Johns and Guang-Zhong Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *ICRA*, pages 3212–3218, 2013.

[4] Michael Milford. Vision-based place recognition: how low can you go? *I. J. Robotic Res.*, 32(7):766–789, 2013.

[5] Alexander Cunningham, Kai M. Wurm, Wolfram Burgard, and Frank Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *ICRA*, pages 1093–1100, 2012.

[6] Albert S. Huang, Matthew E. Antone, Edwin Olson, Luke Fletcher, David Moore, Seth J. Teller, and John J. Leonard. A high-rate, heterogeneous data set from the darpa urban challenge. *I. J. Robotic Res.*, 29(13):1595–1601, 2010.

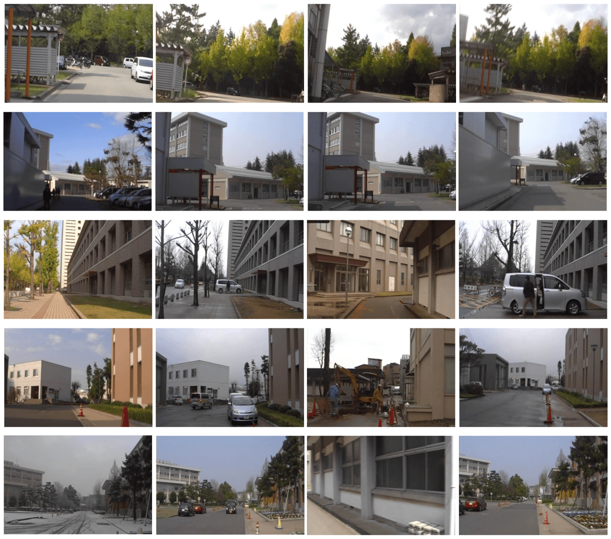


Fig. 6. Examples of scene retrieval by using our scene descriptor. From left to right, each panel shows a query image, the ground truth image, the database image topranked by FAB-MAP and by the proposed method.

[7] Nicholas Carlevaris-Bianco and Ryan M. Eustice. Long-term simultaneous localization and mapping with generic linear constraint node removal. In *IROS*, pages 1034–1041, 2013.

[8] John McDonald, Michael Kaess, Cesar Dario Cadena Lerma, José Neira, and John J. Leonard. Real-time 6-dof multi-session visual slam over large-scale environments. *Robot Auton Systems*, 61(10):1144–1158, 2013.

[9] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Int. Conf. Computer Vision Theory and Application*, pages 331–340. INSTICC Press, 2009.

[10] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Randomized visual phrases for object search. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3100–3107, 2012.

[11] Mark Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics: Science and Systems*, 2009.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Obstacle Detection and Avoidance from an Automated Guided Vehicle

Roger Bostelman, Will Shackleford, Geraldine Cheok

Abstract— **Current automated guided vehicle (AGV) technology typically provides material handling flow along single or dual opposing-flow lanes in manufacturing and distribution facilities. An AGV stops for most any obstacle that may be in its path which then halts other AGVs behind it until the obstacle is removed. An alternative to serial AGV flow is to provide parallel flow in particular areas, such as buffer zones and appropriate lanes where a stopped AGV can be passed by other AGVs. This paper describes two obstacle detection and avoidance (ODA) methods developed and tested. These methods will allow current off-the-shelf AGVs to advance towards unstructured environment navigation.**

I. Introduction

Automated guided vehicle (AGV) technology has been used since 1953 [1] for material handling in manufacturing and distribution facilities. Tug-, unit-load-, and forklift-style AGVs are readily available with typical onboard low-level control of drive, steer, position sensing, guidance sensing, obstacle detection, emergency stop and automatic restart, and start/stop controls among other capabilities. Today, occurrence handling, for example when an obstacle is detected in the vehicle path, is mostly handled locally by the onboard safety sensors – two dimensional (2D), laser detection and ranging (LADAR) sensors – that directly control the vehicle to slow and/or stop via direct electrical connection to the drive amplifiers. Non-contact and/or contact (bumpers) safety sensors are mandatory onboard AGVs, according to the American National Standards Institute/Industrial Truck Standards Development Foundation (ANSI/ITSDF) B56.5 [2] AGV safety standard, where sensors must provide low-level stop capability prior to the AGV structure contacting an obstacle.

Typically, centralized [3] off-board higher-level controllers command AGVs through wireless communication that provides waypoint positions, segment information between waypoints, navigation method and handling, traffic management (e.g., admittance into or decline movement into a particular facility zone), etc. Many AGVs navigate by triangulating laser-based detection of reflectors mounted on walls, resulting in centimeter or smaller repeatability. The AGV movement along segments is programmed into the controller with knowledge of speed, steer method (e.g., Ackerman or quad-steer), onboard equipment adjustment, etc., from one waypoint to the next. Segment information is sent to the onboard AGV controller, typically as the AGV approaches upcoming segments. This ensures that the AGV does not have the entire facility navigation plan that may be uninterruptable and that forces the AGV to follow without updates.

AGVs transporting material usually travel along single or dual, opposing-direction lanes. Therefore, when an AGV halts for an obstacle in its path, it serially stops the flow of other AGVs behind it until the obstacle is removed on its own (e.g., a person walking) or by an AGV supervisor (e.g., a piece of broken pallet dropped in the lane). Workers can anticipate AGV flow in known directions and lanes and at known rates. This method of material handling flow provides intuitive movement for nearby workers and operations. However, this method potentially slows production rates and may require more or faster vehicles to achieve the continuous material flow rates desired by the facility owner. Obstacle detection and avoidance using mobile robot systems is well known in the literature, as a simple internet search illustrates. However, this is not so for AGVs, where only two instances in our search provided examples. One AGV company stated "autonomous navigation provides increased responsiveness, operational flexibility, and improved material flow." [4] Another company demonstrated a floor cleaning robot that navigates around an obstacle in an open area. [5]

The National Institute of Standards and Technology's (NIST) Smart Manufacturing Program has been researching AGV control for developing safety and performance test methods for several years [6, 7, 8]. NIST mobile autonomous vehicle research has investigated performance of obstacle detection algorithms and sensors with respect to standard test pieces, human forms, and overhanging obstacles to foster more intelligently controlled AGVs. Past AGV controls research was enabled through open-source controls and algorithms developed by NIST.

Recently, NIST procured an industrial AGV with stock controls for developing performance metrics and test methods for mobile robots within smart manufacturing facilities. These newer manufacturing settings may have minimal infrastructure, with humans working in close proximity to robots, and may require AGVs to carry advanced onboard equipment such as robotic arms. Using existing technology to conduct this research dramatically reduces the risk to current AGV users and manufacturers. The 2025 Material Handling and Logistics Roadmap suggests that "as confidence in algorithms increases, many routine and even complex decisions will be turned over and automated" and "real-time optimization algorithms for dynamic control of logistics systems should be developed and widely used." [9] Detecting and avoiding obstacles may be considered complex for some in the industry, although it directs them towards future unstructured environment navigation even with their current systems.

Roger Bostelman, Will Shackleford, and Geraldine Cheok are with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (phone: 301-975-3426; fax: 301-990-9688; e-mail: roger.bostelman, will.shackleford, geraldine.cheok@nist.gov).

Currently, AGVs typically have closed-source proprietary controls. NIST is investigating whether current, commercial off-the-shelf (COTS) AGVs could be controlled to perform with greater flexibility, such as adapting to changing environments, and if so, how well they perform. Should the AGV adapt appropriately, overall factory performance would benefit, for example, through enabling parallel material handling AGV flow, where a stopped AGV or obstacle could be passed by other AGVs in buffer zones or adjacent lanes. Providing a centralized traffic manager with knowledge of AGV intent to pass obstacles or other AGVs is an obvious issue that was only briefly addressed in recent NIST research and left for future research.

This paper describes two obstacle detection and avoidance methods developed and tested at NIST using an industrial AGV and controller. One method used alternative pre-planned paths drawn on a layout tool offline. The other method planned paths around obstacles after the positions of obstacles were detected during run-time. Multiple AGV control is briefly discussed, followed by performance measurements compared to ground truth.

II. AGV Obstacle detection and Avoidance Algorithms

A. Using Predetermined Paths

An obstacle detection and avoidance algorithm, called obst_avoid, was written and implemented using an NDC8 transport structure [10]. This algorithm was executed when the onboard safety sensor slow field (detection area causing the AGV to slow speed) detected an obstacle in its main segment or path (in this case straight ahead). Paths are defined in a layout application designer, similar to a computer aided design (CAD) software system, that defines the AGV paths and stop points. Upon obstacle detection, the modified controller uses this information to redirect the AGV to a new, predetermined path positioned to drive around the obstacle. The obstacle avoidance path can be an adjacent AGV lane or a buffer area used for this option. The following strawman control algorithm includes obstacle detection, avoidance, and high level AGV controller alert when more than one AGV is being used in the same facility. The following outline describes how the algorithm works when the AGV detects an obstacle in its path, where the waypoint numbers used in the algorithm are shown in Figure 1 (a). Figure 1 (a) also shows the AGV path, points, and obstacle.

1. Drive along typical path from point 1 to 6 - predetermined points.
2. Safety sensor detects an obstacle (red dot marks detect location) in the path within the slow field detection area when the AGV is between point 1 and point 2. Slow the AGV.
3. Send obstacle detect alert to the high level AGV control system (HL-AGVS) (if there are other vehicles in the same facility/zone) of an obstacle in the path.
4. Vehicle control receives a reply from HL-AGVS, of no approaching AGVs – clear adjacent lane or obstacle-passing area, to move from segment 1-2 to point 6 via point 4.
5. Future option: Continuously monitor with onboard safety sensors that the lane/area is clear. Method: Use raw sensor data to determine if obstacles are in the path and to plan well in advance of the AGV slow field detection area.
6. Left/right crab or steer into approaching lane or clear area, from segment 1-2 to point 5 via segment 3-4.
7. Drive past obstacle. Future option: detect using side and rear sensing that the obstacle has been passed and that the AGV lane is clear to reenter.
8. Right/left steer/crab into original AGV lane towards point 6 via segment 4-5.
9. Continue on to previously commanded goal (point 6)
10. End.

Note: If the obstacle was detected when the AGV was between points 2 and 3, the vehicle would still move along segment 3-4 because the AGV has not reached the next segment choice (3-5 or 3-4) However, if the obstacle was detected after point 3 or beyond, the vehicle would stop using the safety sensor stop field detection and control.

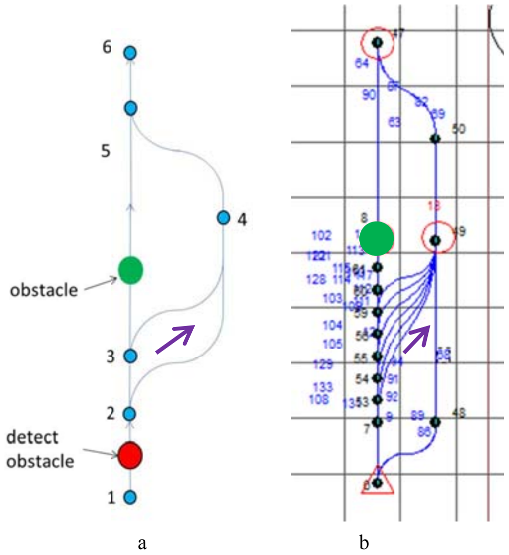


Figure 1 – (a) AGV path, points and obstacle for strawman algorithm development. (b) Layout application plan showing the predetermined AGV paths (blue). The straight vertical path is the commanded path. The red triangle is the start location; the green spot is the obstacle; violet arrows show the obstacle avoidance path; the grid size is 1 m.

The experimental results from several trials showed that the obst_avoid control using a layout application (see Figure 1 (b)) and algorithm performed well: the AGV was able to avoid an obstacle detected using its safety slow field sensor by following a newly-commanded path to drive around the obstacle. Figure 1 (b) shows the paths directly through the obstacle and around the obstacle. No emergency stop was required or automatically occurred and the AGV reached the point 6.

The algorithm did not cause the vehicle to stop at the beginning of each obstacle avoidance start point, therefore traffic flow continued. However, when no obstacle was in the path, the AGV hesitated at each start point. This is because the vehicle was programmed to stop at each point, wait for data on whether there was an obstacle ahead, time for data to reach the traffic manager, and then time for the traffic manager to respond with the ‘go-ahead’ to the next point. The vehicle could have been programmed to go through multiple points without stopping, but then obstacles would not have been avoided. It is possible that this does not occur with all AGVs, although if so, this control method is disruptive to AGV movement. Other problems with this method include:

1. Only two alternative paths are considered. It may be that both paths are blocked by obstacles and a third path might allow the AGV to successfully avoid the obstacle.

2. Communications delays and the limited range at which an AGV could reliably detect an obstacle limit the speed at which the AGV could be traveling when it detects the obstacle in order to switch to the alternative path before reaching the point that the two paths diverge.

3. More work is required of the AGV installer to calculate and verify the large number of possible alternative paths ahead of time. A smaller number of paths could be calculated, but that would reduce the number of places an obstacle could be detected and successfully avoided.

As such, a different type of obstacle detection and avoidance algorithm was researched using B-Splines detailed in the following section.

B. Using B-Spline Paths

The vendor of the controller for our AGV sells an option to allow the AGV to accept external paths during run-time at the start of a segment. Paths are drawn off-line using the same tool as shown in Figure 1 except some segments are marked as “external”. The entire layout, which could include both normal and external segments, is downloaded to the vehicle. A Transmission Control Protocol (TCP) server on the vehicle allows third-party applications, such as the one developed at NIST for our experiments, to monitor the state of the vehicle over a wireless network to know when the vehicle is approaching a segment marked as external. The third party application can then send a list of up to 20 control points (X, Y locations) over that same TCP connection to be used to generate a B-spline that the vehicle will then follow.

There are two significant restrictions:

1. As with most commanded segments, the spline must also start at the original layout start position for that segment and end at the original layout end position. The algorithm can replan up until the start of the blocked segment as the facility sensors provide continuous obstacle position information. However, if the end point of that segment is blocked by an obstacle but some further point in the layout would have been reachable, no replanning is performed since the end-point cannot be bypassed. In other words, the algorithm cannot replan to the final goal because the end point is blocked by an obstacle, nor can it replan to some segment-start closer to the goal.

2. The spline cannot be changed after the vehicle begins following it. This means that new sensor data on the size, shape, and/or position of the obstacles obtained as the spline is being or has already been executed cannot be used to redirect the vehicle around the obstacles. However, should the obstacle move into the new spline AGV path, the onboard AGV safety sensor will still function normally and slow/stop the vehicle prior to contact with the obstacle.

The NIST Spline Generator consists of two parts: obstacle detection and path planning. Obstacle detection reads raw data from two external (i.e., not on the AGV) laser line scanners. A single laser line scanner can provide an estimate of the position of the front of obstacles but since the obstacles are only seen from one side, the size and shape of the back side would be unknown. Therefore, we used two line scanners to measure the size and shape of the obstacle. Figure 2 shows photos of the facility layout, AGV, obstacle, and one of the line scanners. More line scanners could be used to better cover a larger area. The scanners were mounted, with variable height, to posts set on the floor and used to scan in a horizontal plane a few inches above the floor. Since the scanners detect obstacles only in a single plane a few inches from the floor, they can detect things such as the legs of a table or ladder. Other line scanners could be mounted on a motorized tilt or a 3D sensor could be used to detect higher or overhanging obstacles.

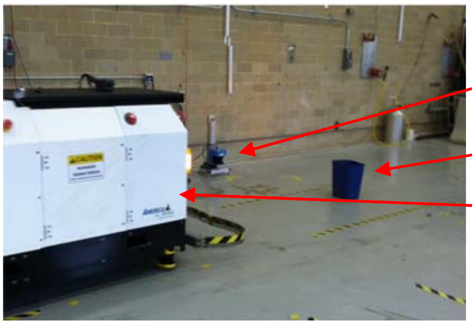


Figure 2 shows photos of the facility layout, AGV, obstacle and line scanners.

Using external sensors rather than sensors mounted on the AGV has several advantages because external sensors:

1. Can be used to provide information for multiple AGVs.
2. Can be placed so that obstacles can be detected from more than one direction.
3. Can be changed without modifying the AGV, adding any additional weight, drawing power, and/or changing the footprint of the AGV (i.e., sensor on an AGV could protrude from the vehicle causing a potential hazard).
4. Uncertainty in the position of the vehicle does not add to uncertainty in the position of the obstacles. Unfortunately, although the vehicle would seem to have an accurate position estimate of the AGV for use internally, this does not appear to be available for third party applications through the TCP connection previously discussed. Even if it were available, unexpected delays in receiving the position could result in errors in the position associated with the obstacle.

The primary disadvantage of mounting the line scanners throughout the facility is that if there is a large area to cover and only a few AGVs, then mounting the sensors to the room will require more sensors than using a sensor mounted to the vehicle. Ideally, safety line scanners that are typically used for detecting and slowing/stopping the vehicle when obstacles are in the vehicle path would also provide relative position information of obstacles beyond the slow and stop zones. Today, there are such sensors available in the market. To use this feature would require simultaneous AGV position and relative obstacle position to the AGV from the onboard line scanners and a filter to only detect obstacles in the AGV path.

Figure 3 shows a graphical display of the obstacle detection program. The obstacle detection filters out background points, through use of a simpler set of fixed boundaries, corresponding to the walls and fixed equipment that the motion planner will avoid. Filtering out these points reduces the computational work-load for the planner.

The range points from the line scanners are converted to 2D points in the room using the calibrated position of the sensor and then grouped so that the planner only needs to deal with a rather small set of obstacles. It also filters out dynamic obstacles or obstacles that have not remained at the same position for the last five seconds. It generally makes more sense to wait for dynamic obstacles, such as people or other vehicles, to move out of the way than to plan around them. It is important to understand that the AGV still has the original safety system running unaffected by the obstacle avoidance system. That safety system, and not the obstacle detection system, is responsible for stopping the vehicle to prevent injury to people or equipment.

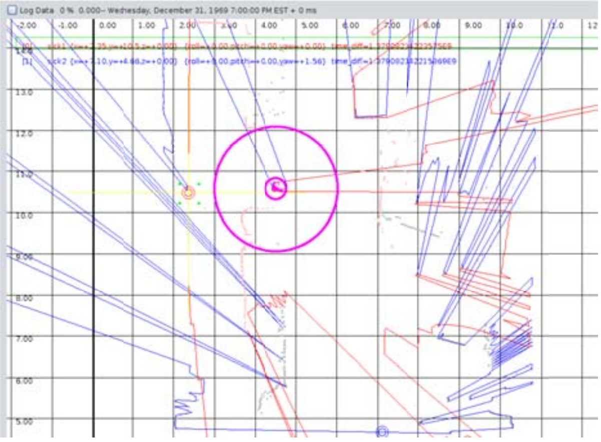

Log Data 0 % 0.000-- Wednesday, December 31, 1969 7:00:00 PM EST + 0 ms
-2.00
-1.00
0.00
1.00
2.00
4.00
5.00
9.00
13.0
12.0
10.0
8.00
7.00
6.00
5.00

Figure 3: Graphical Display of Obstacle Detection program: red lines show range data from one line scanner, blue lines show range data from the other line scanner, the detected obstacle is shown with two concentric purple circles

Path planning takes the following as input:

1. List of obstacles detected from the current sensor data. Each obstacle is associated with an X, Y center point and a radius.
2. List of boundaries drawn off-line by hand using the room's CAD drawing. Each boundary is associated with two X, Y locations for the two end points of the line segment.
3. The start position and orientation for the next external segment.
4. The end position and orientation for the next external segment.
5. Whether the segment will be performed in crab mode or not.
6. The dimensions of the vehicle's footprint. This includes the distance from the commanded point to the farthest point in front of the vehicle that an obstacle could trigger an emergency stop, as well as the distances to the farthest points to the rear of the vehicle and to each side.

The planning algorithm is based on the common A* search where the nodes in the graph represent candidate X,Y control point locations and the edges are weight costs based on the distance between the points. Edges are only added if the spline between the two points could be traveled without any part of the vehicle's foot print entering an obstacle radius or crossing a boundary.

The algorithm uses two lists of points. One list contains unopened planner points whose reachability from the start location is unknown. The other list contains opened points that are reachable from the start location and have at least one untested potential next point. The unopened list initially contains the goal point and points strategically placed around each obstacle or boundary. The opened list initially contains only the start position. Each point also has an associated list of untested next points. For the start point, this is initialized with a copy of the unopened points list.

A loop is repeated until either a path to the goal is found (see Figure 4) or there are no more points on the opened list. Within the loop, the best point on the opened list is determined as the point with the lowest value of the heuristic function. The heuristic function is the distance of the shortest obstacle-free path found so far to that point plus the minimum over all untested next points of the distance to that point plus the distance to the goal from that point. The path from the best point from the open list to the best untested next point from the list for that point is checked. The check determines if an obstacle or boundary would be encountered by the area swept out by the AGV while moving between the two points.

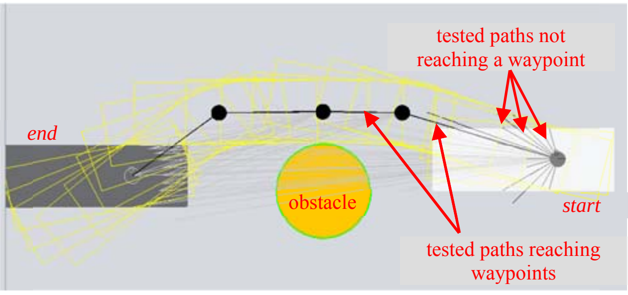

tested paths not reaching a waypoint
end
obstacle
start
tested paths reaching waypoints

Figure 4: Plot showing the planned AGV path (black line) and tested paths (gray lines to no point) around an obstacle. The white and dark gray filled boxes are the AGV at the start and end points and the yellow boxes are the sweep of the AGV as it moves along the planned path.

If no obstacles or boundaries would be encountered, the point last tested is added to the list of opened points and the points on the unopened list copied to the list of untested next points for the point being added to the opened list. In addition, if the test succeeds and the last point was the goal, the loop can be ended with success. Regardless of the result of the check, the point is removed from the untested next points for the first point taken from the open list and this will result in a change in its associated heuristic value. If the untested next points are now empty, the first point is removed from the opened list and if that makes the opened list itself empty, the loop is ended in failure.

Psuedo code:

```
unopened_list ← [ goal + points around each obstacle and boundary ]
start_point.untested_next_points ← copy(unopened_list)
opened_list ← [ start point ]
while opened_list is not empty :
  best_opened ← find_best_point(opened_list)
  best_next ← find_best_point(best_opened.untested_next_points)
  if test(best_opened,best_next) is ok :
     best_next.prev ← best_opened
     best_next.untested_next_points ← copy(unopened_list)
     opened_list ← [ opened_list + best_next ]
     if best_next is goal :
        end the loop , we found the path to the goal
  best_opened.untested_next_points.remove(best_next)
  unopened_list.remove(best_next)
  if best_opened.untested_next_points is empty :
     opened_list.remove(best_opened)
     if opened_list is empty :
        end the loop, failure no path to the goal
```

The test involves generating a series of polygons for a move and then testing each boundary and obstacle to determine if a boundary has either end point in the polygon or intersects any edge of the polygon or an obstacle is within any polygon or closer to any edge than its radius. Figure 6 shows a screen snapshot of the resulting algorithm planning a path around obstacles in the AGV path detected by facility line scanners.

The algorithm makes several assumptions that may cause the path found to be less than optimal or to fail to find a path even when one was available.

1. It only considers the finite list of points added to the initial unopened point list.
2. It will not check whether a point could be reached if the starting orientation were different. There may be more than one way to reach the same point and therefore more than one orientation the vehicle would have at that point, however the tests to next points will assume the orientation that would result from the first path found to that point.

The path shown in Figure 5 was computed in 54 ms (on an 8 core x 2.6 Ghz laptop). Figure 6 shows a series of snapshots of the AGV performing obstacle avoidance using the spline method.

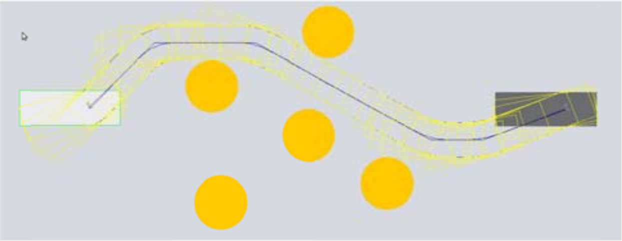


Figure 5: Graphical output of path planner, starting footprint of the AGV is in white, the goal position is dark grey rectangle. Yellow rectangles show the area swept out as the AGV would travel, blue curve shows the resulting spline, orange circles represent obstacles

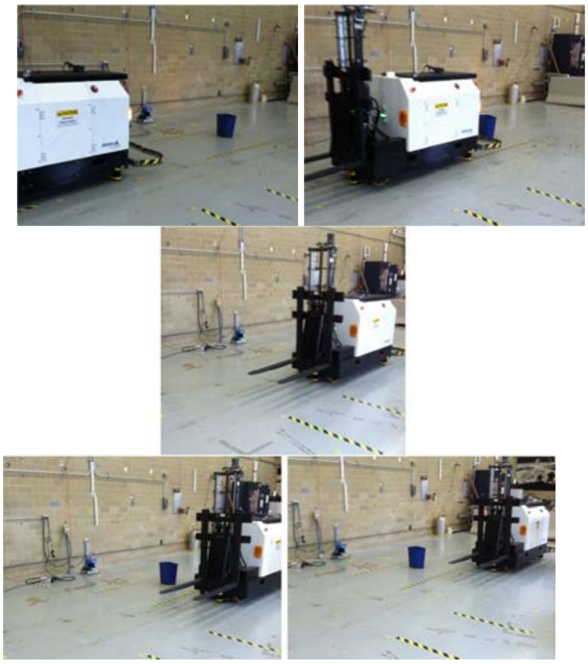
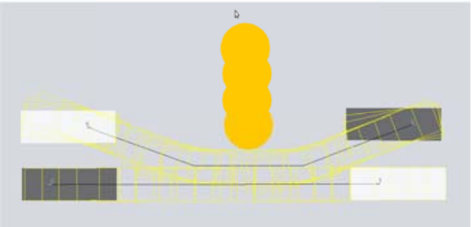


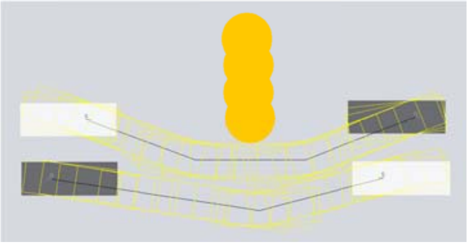
Figure 6: (top-left to bottom-right) Series of snapshots showing the AGV performing obstacle detection and avoidance using the spline method.

III. Multiple Vehicles

Although we only have one test AGV, the effect of one AGV's obstacle avoidance on other AGV is known to be an important issue. A feature called “Exclusive Tracks” was added to prevent AGVs from approaching each other from different directions that likely result in both AGVs sensing the other and going into E-Stop mode. With this option, one AGV is given priority. The path that it sweeps out adds boundaries that will be considered in planning the path of the other AGV. Figure 7 shows the effect of both with and without the “Exclusive Tracks” feature enabled.



a



b

Figure 7: (a) One AGV plan to the right over another AGVs plan to the left shows conflicting areas of the path since the "Exclusive Tracks" option was not selected. (b) The same setup as in (a) with the "Exclusive Tracks" option enabled causing the AGV moving to the left to avoid the conflicting area and reduce the chances of an E-STOP.

IV. Algorithm Performance Compared to Ground Truth

Upon development of the algorithm, the performance of the algorithm was evaluated in an experiment where ground truth measurements were obtained from a laser tracker. It was intended here to mainly compare the best fit path to ground truth, not the interpolated b-spline curve at turns. The laser tracker had an uncertainty of 18 µm at 12 m to track AGV motion (see Figure 8). An SMR (spherical mounted retroreflector) was placed on top of the AGV above the AGV center. The AGV center was defined as the centroid of the four AGV wheels. The planar motions of the AGV, with an obstacle (small or large) in its path, were compared to ground truth measurements.

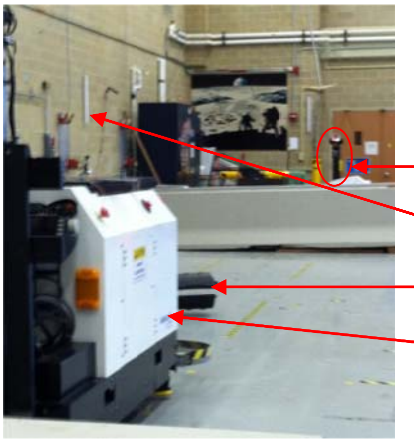


Figure 8: Photo of ground truth measurement system (laser tracker) setup.

A set of six runs were performed with a small obstacle and a set of six runs were performed with a large obstacle. For the set of six runs, three runs were in the 'forward' direction and three runs were in the 'reverse' direction. That is, one run was from waypoint A to B (forward direction) and the next run was from waypoint B to A (reverse direction). Figure 9a shows a plot of the ground truth measurements for one of the six runs with the small obstacle (orange markers), and Figure 9b shows a plot with the large obstacle (purple markers). As seen in the figure, the largest deviations of the AGV from the commanded waypoints occur at the turns (indicated by the red circles in Figure 9).

Figure 10 shows the comparison of the average deviations of the six runs for the small and large obstacle. The error bars indicated one standard deviation of the average. As seen in Figure 10, the magnitude and variability of the deviations are greater for the large obstacle. Excluding the turns, the average deviation for the small obstacle is less than 10 mm and 40 mm for the large obstacle. The maximum average deviations on the turns are 65 mm and 170 mm for the small and large obstacle, respectively.

The commanded points are control points for the b-spline. Control points are typically the exact stop or cross points for AGVs using current AGV path planners. However, control points for a b-spline determine the shape of the curve but are not necessarily expected to lie on the curve. These deviations indicate neither a flaw in the algorithm that generated the control-points nor in the lower level software/hardware responsible for following the curve. This b-spline effect is not related to obstacle avoidance, but only to the tightness of the curve and the inherent characteristics of b-splines. The performance measurement method proved useful to understand the effect and the distance from the b-spline waypoints to the actual path followed by the AGV (i.e., the interpolated b-spline). Exact (interpolated) points to be crossed by the AGV could be calculated when using a b-spline where the performance measurement demonstrated may prove useful and results better fit the interpolated path, especially for AGV navigation in highly confined, unstructured areas.

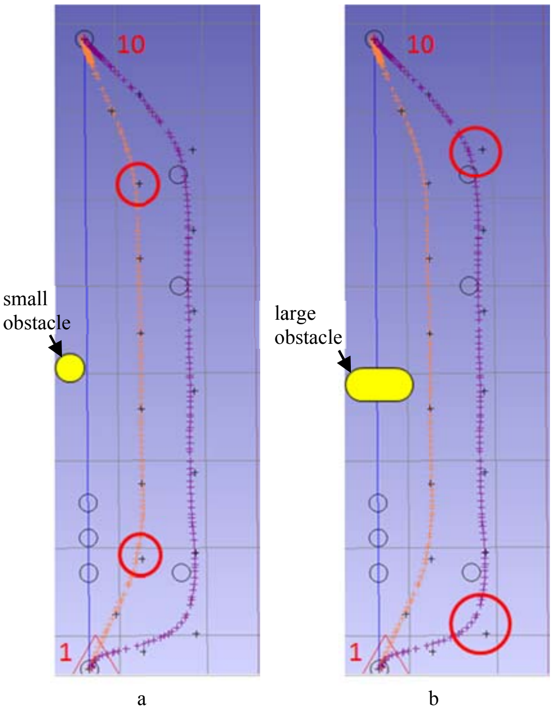


Figure 9: Measurement results as compared to ground truth for obstacle detection and avoidance using a spline algorithm for runs with a (a) small obstacle and (b) large obstacle. Black '+' symbols are commanded waypoints. Orange and purple '+' symbols indicate ground truth AGV positions for the small and large obstacle runs, respectively.

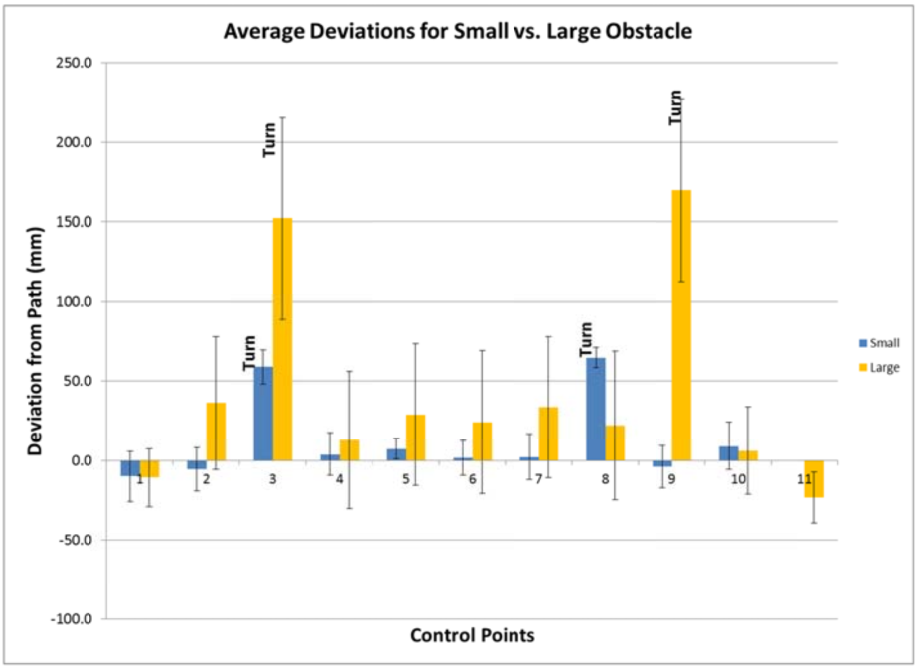


Figure 10. Comparison of the average AGV deviations from the commanded waypoints for a small obstacle vs. a large obstacle.

V. Conclusion

Based on internet searches and conversations with AGV industry representatives, at least two companies have performed obstacle detection and avoidance tests using AGVs. The research also found that other AGV manufacturers described traffic management issues as a low priority issue, although they expected that obstacle detection and avoidance was feasible using current sensor technology. NIST concludes from past research with mobile robots and the 2025 Material Handling and Logistics Roadmap that future benefits to the industry for performing obstacle detection and avoidance would allow: increased productivity for users, perhaps require fewer yet faster AGVs, and facilitate more capable AGVs with minimal modifications to current controllers that would allow navigation through less structured environments.

Two obstacle avoidance control algorithms were designed and tested on the NIST AGV. One includes using many pre-computed paths for the AGV to select from to avoid an obstacle depending upon when the obstacle was detected. The second algorithm uses a b-spline function and higher-level program with facility sensor information about the obstacle position. The second algorithm provides much more control flexibility to avoid more than one obstacle while planning complex paths. Both algorithms could allow adjacent traffic information to minimize risk of AGV to AGV collisions or bottlenecks in AGV traffic. Performance measurement of the B-spline controlled AGV indicated that the largest deviations of the AGV from the commanded waypoints occurred at the turns. Waypoint access tolerance at the turns could, however, be set at a minimum for improved navigation performance between narrow facility passages.

Acknowledgment

The authors would like to thank the AGV manufacturers and users for their input to this research, especially Tommy Hessler from America In Motion.

References

[1] Savant Automation, http://www.agvsystems.com/history-agvs, 2012.

[2] American National Standards Institute/Industrial Truck Standards Develop Foundation (ANSI/ITSDF) B56.5:2012 Safety Standard for Driverless, Automatic Guided Industrial Vehicles and Automated Functions of Manned Industrial Vehicles, 2012.

[3] Y. Uny Cao, Alex S. Fukunaga, Andrew B. Kahng, “Cooperative Mobile Robotics: Antecedents and Directions,” Autonomous Robots 4, 7–27 (1997).

[4] Video of obstacle avoidance, http://youtube.com/JVigBUcvA-c, consulted October 2013.

[5] Video of autonomous floor cleaning robot avoids obstacles, http://www.youtube.com/watch?v=4GJ00EBbBfQ, consulted October 2013.

[6] Roger Bostelman, Will Shackleford, Geraldine Cheok, Richard Norcross, “Standard Test Procedures and Metrics Development for Automated Guided Vehicle Safety Standards,” Performance Metrics for Intelligent Systems (PerMIS’12) Workshop, College Park, MD, March 2012.

[7] Roger Bostelman, Will Shackleford, Geraldine Cheok, Kamel Saidi, “Safe Control of Manufacturing Vehicles Research Towards Standard Test Methods, International Material Handling Research Colloquium (IMHRC) 2012, Gardanne, France, June 2012.

[8] Roger Bostelman, Richard Norcrossa, Joe Falco, Jeremy Marvel, “Development of Standard Test Methods for Unmanned and Manned Industrial Vehicles Used Near Humans, SPIE Defense, Security, and Sensing 2013, Baltimore, MD, May 2013.

[9] Material Handling and Logistics U.S. Roadmap, http://www.mhlroadmap.org/,2014.

[10] NDC8 controller, www.ndc8.com.

Frontier Based Exploration with Task Cancellation

P.G.C.N. Senarathne and Danwei Wang

Abstract— **Classical frontier based exploration strategies operate by iteratively selecting the next best sensing location myopically and moving to the specified location, until the entire environment is explored. And it does not consider the new information added to the map through continuous observations by the robot along the way to a selected location. This can sometimes lead to redundant traversal by the robot, such as traveling towards a dead-end when the nearby area is already mapped. In this work, we augment the classical frontier based exploration strategy to include a probabilistic decision step that decides whether further motion on the planned path is desirable or not. If the motion is not desirable, it is interrupted and a new sensing location is selected as the next sensing task. Experiments were conducted using a Pioneer 3AT robot to explore an indoor environment and is demonstrated that the proposed method on average is capable of exploring environments more efficiently.**

I. INTRODUCTION

Autonomous exploration of unknown environments is one of the most important tasks carried out during robotic missions such as environmental mapping [1], search and rescue/find [2], [3]. For a robot to autonomously explore an unknown environment, it must sequentially sense the environment at new sensing locations until complete sensor coverage of the environment is achieved. At a given instance during an exploration mission, there are multiple candidate locations where the robot could perceive the environment to expand the existing mapped area. Once the robot senses the environment through one of these caldidate locations, a new set of candidate locations appear in the map. This explosion of the candidate sensing space with each sensing action, coupled with the high uncertainties in the information gain predictions for each of these sensing locations, makes finding an optimal sequence of sensing locations for complete exploration of the unknown environment intractable. Therefore, all the exploration strategies are reduced to finding the next best sensing location as an intermediate target and moving to that target as the next task and repeating this process until complete mapping of the environment is achieved. The most popular heuristic used to generate the set of candidate locations, to find the next best sensing location (i.e. intermediate target), is to extract cells in the boundary between the mapped free and unmapped cells from an Occupancy Grid Map representation for the environment [4]. These boundary cells are called *frontier* cells and the exploration strategies based on this heuristic are called Frontier Based Exploration.

In almost all the frontier based exploration strategies, the robot is left to travel to the selected target (i.e. completion of current sensing task), before the next decision is made. While this method is simple and reduces the exploration problem to a one finding the best intermediate target to expand the map in discrete steps, it does not consider the changes to the frequently evolving frontiers, which could invalidate the usually over-estimated information gain for the intermediate target. This happens when robots select frontier cells, that are very close to obstacles which are yet to be revealed in the map, as their next intermediate targets. As a robot moves towards such a target, continuous sensing reveals the obstacle, which encloses the explored free space, fully or partially, and stops the expansion of most of the frontier cells in the neighborhood, an extreme situation of which is depicted in fig. 1. This disappearance of frontiers in the current target's neighborhood results in the further movement of the robot towards the target, largely redundant as large portion of the map does not get expanded.

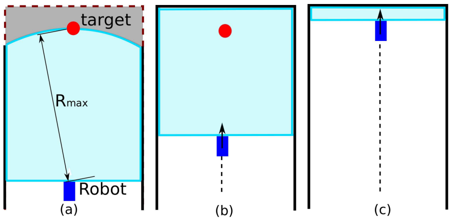


Fig. 1. (a) Illustrates a situation where a frontier cell that is close to a yet to be revealed obstacle boundary, marked in dashed lines, is selected as the next intermediate target. The white area is the already explored free space while solid black lines are the already discovered obstacle boundaries. Gray represents unmapped area. R_{max} is robot's maximum sensing range and light-blue area illustrate robot's current sensing field of view. (b) As the robot moves towards the target (as shown by the arrow), the entire area becomes fully explored (c) The robot continues to move towards the target in traditional exploration strategies, redundantly

Therefore it is important to check if the map continues to get expanded in the direction of the robot's final desired heading. This expansion is directly reflected by the distribution of the frontier cells near the current target as the robot moves towards the target. Existence of frontiers in the direction of the robot's desired heading indicate that the map gets expanded in that direction. Frontiers in other directions indicate the options available for the robot for exploration in those directions. Depending on the distribution of the desirability of these two types of frontiers, decisions can be made to either continue the current motion or to cancel it and generate a new exploratory task in another direction. The work presented in this paper augments the classical frontier based exploration strategy to include this decision step in order to conduct more efficient exploration missions.

Both authors are with EXQUISITUS, Center for E-City, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore senarathne@ntu.edu.sg, edwwang@ntu.edu.sg

The article begins by providing a summary of various works on autonomous exploration strategies and a description on the baseline frontier based exploration strategy. The next section describes the approach used to arrive at the task cancellation decision. It includes a probabilistic formulation for the decision event and how the parameters decide the aggressiveness or the conservativeness of the exploration process. It also details the generation of the frontier selection probability distribution used for the decision event. Details of the experiments, the results and analysis are provided in the subsequent sections.

II. Related Work

Robot exploration strategies are mainly driven by the concept of iteratively selecting the next best sensing location in the environment as the intermediate target in order to expand the map. Frontier based exploration [4] selects the next best sensing location from a collection of candidate target locations generated from the boundary between the mapped-free and unknown grid cells (i.e. frontiers) in the occupancy grid map [5]. Different criteria used for the selection of the next intermediate target has resulted in various extensions of the basic frontier based exploration strategy. Selecting the closest frontier cell, balancing the information gain of frontier cells with travel cost [6], [7], [8], use of *hysteresis value* to restrict robot from frequently switching exploration tasks [9] are some of them. While the popular heuristic is to use frontier cells as candidate targets for exploration, random generation of candidate target locations have also been proposed. The works in [10] discuss the generation of candidate target locations randomly while [11] proposes the biasing of the random target generation towards the frontier boundaries.

Several exploration algorithms based on topological map representations are also proposed in the literature. These include the works of Kuipers et. al, [12], Choset et. al, [13] and Ge et. al, [14]. However, frontier based exploration strategies have become prevalent due to the ease of generation and management of occupancy grid maps compared to topological maps hence are the focus of this article. The baseline strategy used for comparison of the proposed method is a frontier based exploration strategy. The utility $U(\lambda)$ of the frontier cell λ is generated as $U(\lambda) = \alpha\mathcal{I}(\lambda) - \beta\mathcal{C}(\lambda)$ where $\mathcal{I}(\lambda)$ is the estimation of the information gain and $\mathcal{C}(\lambda)$ is the estimated travel cost to cell λ. α and β are two parameters that can be varied to decide the relative importance of the information gain and cost components. The frontier cell with the highest utility is selected as the next intermediate target. Calculation of the information gain can be done using entropy/mutual information based methods [8], [7] or by counting nearby unexplored cells, generated by thresholding the occupancy probabilities [9]. Since the second method is simpler to implement and is not considered inferior [15], it is used to calculate the information gain. The travel cost is estimated as the distance of the planned path to the frontier cell.

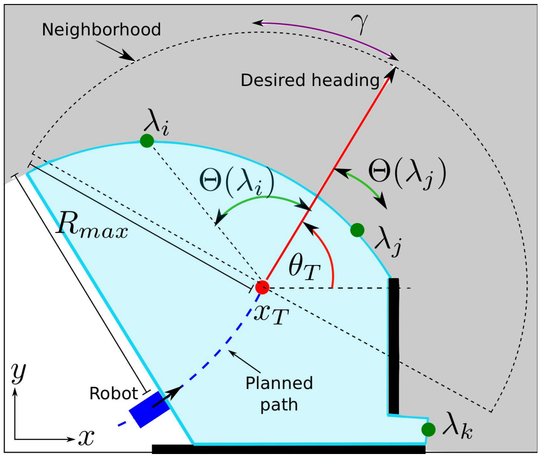


Fig. 2. x_T is the current intermediate target, θ_T is the desired heading angle. Both λ_i, λ_j are example depictions of two frontier cells neighboring x_T. Note that frontier cell λ_k is outside the neighborhood of x_T, so not considered for the cancellation decision. The neighborhood is depicted by the dashed semi-circle with a radius of R_{max}. The light-blue shaded area represents robot's current sensor field of view.

Improving the efficiency of frontier based exploration using updated frontier information has previously been mentioned by Keidar et. al, [16]. However, their work focuses only on efficiently generating frontiers in high frequency to support such improved strategies. Holz et. al, [17] discusses the drawback of continuing to move towards the intermediate target without considering the continuously updated frontier information. A repetitive rechecking approach is proposed where, the assigned intermediate target is checked for being a valid frontier cell in order to reduce redundant exploratory motion. Our work extends this approach to a more general formulation that considers the utility around the target and to decide when to cancel the current motion.

III. Cancelling Exploratory Motion

Let us define the current intermediate target of the robot as $x_T \in \mathcal{R}^2$ and the final heading of the robot's planned motion at x_T as θ_T. Consider that the map is updated at discrete steps during the planned motion to x_T. Let $\Theta : \mathcal{R}^2 \to [0, \frac{\pi}{2}]$ be the random variable describing the absolute difference in angle between robot's desired heading θ_T and $\overrightarrow{x_T\lambda}$ where λ is any neighboring frontier cell of x_T as illustrated in fig. 2. The set of neighboring frontier cells of target x_T is denoted by $\mathcal{F}(x_T)$. At each map update step k, a probability density function $f^k_\Theta : [0, \frac{\pi}{2}] \to [0, 1]$ can be defined. This distribution describes the probability of selecting a frontier cell from $\mathcal{F}(x_T)$ in a specified absolute angle difference with θ_T, as the next intermediate target. Suppose γ to be the angle tolerance used to consider frontier cells as belonging to robot's final heading direction. Then, the robot's motion towards its current target x_T is cancelled if $P(\Theta \le \gamma) < p_T$. The cumulative probability on the left hand side of the inequality provides a measure of desirability of the frontier cells in the 'direction' (based on angle tolerance γ) of the robot's final heading for the robot. If this desirability is less than a certain probability threshold p_T, the robot's motion

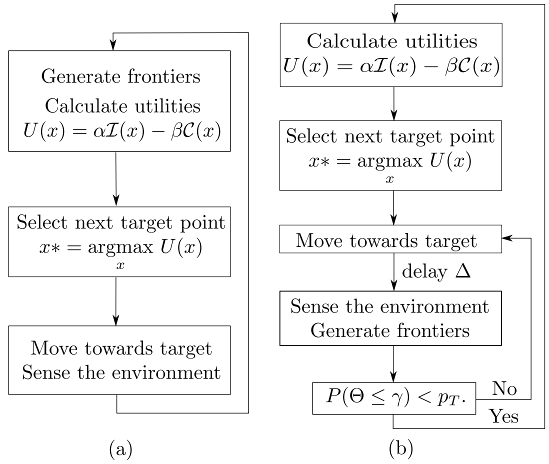


Fig. 3. (a) Major steps in the base-line exploration strategy (b) Major steps in the proposed exploration strategy. Frontiers are generated at a certain frequency and the motion is preempted to check for redundant motion. And the motion continues if the motion towards the target is still desirable. If not desirable, a new target is selected and the process continues.

towards the current target is cancelled as illustrated by the steps in fig. 3. The value of p_T can be changed according to the level of cancellation desired. A more aggressive form of exploration can be activated by setting p_T to a higher value and a more conservative form of exploration can be activated by a lower p_T value. Setting p_T to zero makes the inequality false for all scenarios and will not cancel the robot motion, hence will make the robot behave identically to the baseline strategy. $P(\Theta \leq \gamma)$ is calculated as follows.

$$\begin{aligned} P(\Theta \leq \gamma) &= \int_0^\gamma f_\Theta^k(\theta)\,\mathrm{d}\theta \\ &= \int_0^\gamma \sum_{\lambda \in \mathcal{X}_\theta^T} f_X^k(\lambda)\,\mathrm{d}\theta \end{aligned} \tag{1}$$

Here, the set $\mathcal{X}_\theta^T$ is defined as $\{\lambda \in \mathcal{F}(x_T) \text{ s.t. } \Theta(\lambda) = \theta\}$ and contains all the neighboring frontier cells of x_T with an absolute angle difference to θ_T equal to θ. $f_X^k : \mathcal{R}^2 \rightarrow [0, 1]$ is the probability mass function (p.m.f.) that provides a measure of desirability of selecting neighboring frontier cell $\lambda \in \mathcal{F}(x_T)$ as an alternative target.

IV. ESTIMATING NEIGHBORHOOD FRONTIER CELL SELECTION PROBABILITY

The probability of selecting a neighboring frontier cell (i.e. desirability of frontier cell as an alternative target) is calculated based on their individual utilities. Hence the p.m.f. can be defined as,

$$f_X^k(\lambda) = \frac{U(\lambda)}{\sum_{\lambda_i \in \mathcal{F}(x_T)} U(\lambda_i)} \tag{2}$$

However, it is reasonable to assume that the effect of the travel cost is negligible for the set of neighboring frontiers of x_T, $\mathcal{F}(x_T)$, as the variation among these travel costs is minimal. Hence Eq. 2 can be approximated using the information gain values $\mathcal{I}(\lambda)$ as follows.

$$f_X^k(\lambda) = \frac{\mathcal{I}(\lambda)}{\sum_{\lambda_i \in \mathcal{F}^T(x_T)} \mathcal{I}(\lambda_i)} \tag{3}$$

While crude calculation of information gain for each frontier cell, $\mathcal{I}(\lambda)$, can provide the necessary p.m.f values, it is desirable to generate these p.m.f. values with less computational burden as they need to be computed at each frontier update step. Hence the apparent correlation of information gain values among nearby frontier cells is used to quickly arrive at estimates of actual information gain values. Frontier cells generally belong to clusters of cells representing map boundary contours that share a common unknown area that is used to estimate the information gain. Corners of these contours are adjacent to already mapped area restricting the information gain, while the mid points of these contours are generally the farthest points away from mapped area allowing a higher information gain. This heuristic knowledge of information gain progression along a frontier cell contour is used to define a function that approximates the information gain for each frontier cell.

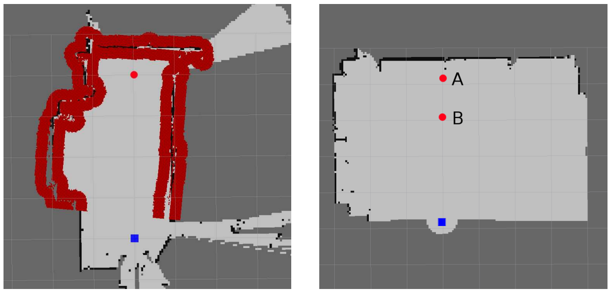
Consider a frontier contour c, then the information gain of cell $\lambda \in c$ is approximated as $\mathcal{I}(\lambda) = \mathcal{I}_c\psi_c(\|\lambda - \lambda_{\mu,c}\|)$ where $\mathcal{I}_c$ corresponds to the contour wide common information gain term. The function $\psi_c(d)$ approximates the fraction of $\mathcal{I}_c$ the robot gains by visiting a frontier cell d distance away from it's associated contour's center $\lambda_{\mu,c}$. R_c is the maximum distance to a corner of the frontier contour from $\lambda_{\mu,c}$.

$$\psi_c(d) = \begin{cases} 1 & \text{if } d + R_{max} \leq R_c \\ \frac{1}{\sqrt{2\pi}R_{max}} e^{\frac{-(d+R_{max}-R_c)^2}{R_{max}^2}} & \text{else} \end{cases} \tag{4}$$

The function value is kept at 1, indicating the highest possible information gain, when the frontier cell is more than R_{max} distance away from the frontier contour's corners, towards the center $\lambda_{\mu,c}$, thus not restricting the maximum sensor range. When the frontier cell's position makes the sensor range goes beyond the contour's corners, it restricts the sensing of the robot and the fraction of the information gained by the robot is approximated to decline according to a normal function.

V. EXPERIMENTAL RESULTS

Experiments are conducted using a Pioneer 3 AT robot equipped with a Hokuyo LRF for sensing. The exploration strategies were implemented using the navigation software layer provided by ROS [18]. The occupancy grid map update/access frequency is set to 1Hz. In all experiments, α and β parameter values are set to 0.8 and 0.3 respectively. Changes to these two values do not affect the performance of the task cancellation decision as they are used only for selection of the intermediate target, hence are kept constant throughout the experiments. Two types of experiments were conducted to compare the performance of the proposed strategy. In the first type, effect of cancellation on a single sensing task is evaluated while on the second type of experiments,



(a)

(b)

Fig. 4. (a) Map of the narrow passage. The blue box is the starting position of the robot, red circle is the target near the dead end. The red band circling the mapped space is the inflated obstacle grid cells. (b) Map of the room type environment. Two targets A and B are used

the effect of task cancellation was evaluated for complete exploration missions. Travel distance of the robot is used for comparisons between the two strategies.

A. Analyzing the effect of parameters on different environments

In the first type of experiments, the cancellation of exploration tasks is compared with traditional exploration tasks to better understand the effects of the two parameters γ and p_T. The two environments used for comparing parameter effects is shown in fig. 4. The robot's maximum sensor range, R_{max}, is set to 4m. The first environment considers the situation of a robot approaching a narrow passage with a dead end, which are often found in indoor environments. The end of the passage is 4.5m from robot's starting point. Hence, there exists unmapped area towards the dead end and robot selects a target which is 3.5m away and close to the dead-end.

Fig. 5 depicts the travel distance of the robot towards the dead-end with varying γ and p_T values. It can be seen that irrespective of the γ value, when the threshold probability is set to zero, it behaves identically to the classical exploration strategy and continues to move towards the target which is close to the dead end even though the entire passage gets fully explored by a small forward motion. In all other situations, the robot's motion gets cancelled within 0.5-1.0 m forward motion and behaves identically. This is because as the robot moves forward about 0.5m-1.0m, due to the narrowness of the passage, the entire environment is fully explored and there are no more frontiers to be selected as intermediate targets. Hence the two parameters do not affect the cancellation decision differently.

In the second environment type, two targets are used. Target A and B are placed about 0.5m and 1.5m away from the room boundary respectively as shown in fig. 4(b). In both cases, the target is at the frontier with the robot's sensor facing towards the target. In each experiment, the robot is sent to the specified intermediate target in an exploratory motion following a straight trajectory. The travel distance at the time the robot stops/gets interrupted is recorded for each run. For each target A and B, the area the robot could explore is measured by hand. This value is used to generate the expected information gain percentage function for the robot's currently travelled distance. It can be noted that, the remaining information gain exhibits a diminishing returns property with respect to the travelled distance. This is because as the robot moves towards the targets, most of the area in front of the robot gets mapped. Remaining unmapped area resides to the two sides of the robot's motion. Hence, the straight motion of the robot does not map the unknown area with high efficiency. This results in a lower net-gain with increasing travelled distance. This observation justifies cancelling the motion before the robot reaching the assigned target and reassigning a new target, when the robot's motion towards the previous target is less 'desirable'.

The two graphs in fig. 6 and fig. 7 summarizes the results from these experiments and illustrates the effect of γ and p_T in deciding the desirability of the robot's motion. For both targets, setting p_T to zero, irrespective of the γ value, makes the robot behaves exactly similar to the classical exploration approach and moves the robot to the target which is 3.5m away. And also as $\gamma = 90°$ includes the entire neighborhood as the robot's 'heading', the robot moves forward until there are no frontiers in the neighborhood, hence the long travel distances. For target A, it can be seen that, travel distances between 1.0m-1.5m explores between 70%-85% of the neighboring area approximately. Cancelling the sensing task during this travel distance interval can be considered more efficient for the exploration mission as the robot could cover the remaining unknown areas more efficiently by employing a different motion from the current one. For targets that lie very close to obstacle boundaries in a room type environment, such as A, selecting $\gamma = 75°$ and $p_T = [0.1 - 0.5]$ or $\gamma = 60°$ and $p_T = [0.1 - 0.3]$ is observed to generate motions that are neither too agressive nor too conservative in exploring according to fig. 6. For target B, the travel distance interval 2.0m-2.75m explores between 77%-90% of the neighboring area approximately. Hence, similar to target A, it can be considered that selecting $\gamma = 60°$ and $p_T = [0.3 - 0.5]$ or $\gamma = 45°$ and $p_T = [0.1 - 0.3]$ generate exploratory motions that exhibit the correct balance of agressiveness and conservativeness for targets that are not too close or far away from obstacle boundaries relative to the sensor range. It can also be observed that for $\gamma = 75°$, the generated motions are either too conservative, for $p_T = [0.1 - 0.7]$, or too agressive, for $p_T = [0.8 - 1.0]$ hence not suitable for exploration missions.

B. The effect of task cancellation on exploration missions

The previous section analyzed the effect on exploration with cancellation of a single sensing task. An exploration mission is a sequence of such sensing tasks. Experiments were conducted to measure the effect of the cancellation strategy on complete exploration missions of an indoor environment, depicted in fig. 8. The travelled distances to complete the exploration missions were recorded. Based on the results from the previous section, $\gamma = 45°$, $p_T = [0.1 - 0.3]$ and $\gamma = 60°$, $p_T = [0.1 - 0.5]$ are selected as the parameter space for full exploration missions. For each γ, P_T

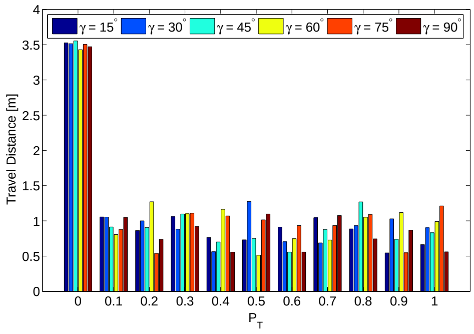


Fig. 5. Travel distances of robot during exploratory motion in the narrow passage

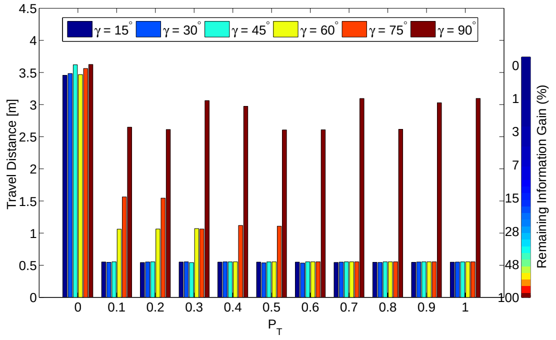


Fig. 6. Travel distance of robot and remaining information gain percentage during exploratory motion to target A

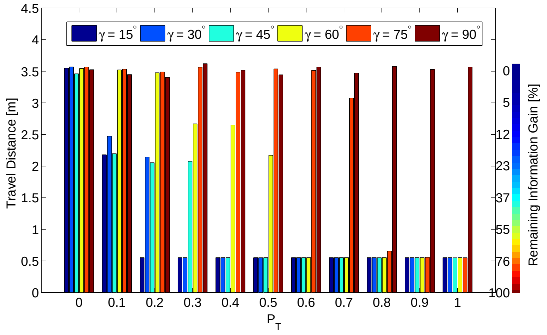


Fig. 7. Travel distance of robot and remaining information gain percentage during exploratory motion to target B

parameter pair used for experiments, 10 runs were conducted with different starting pisitions to negate any bias arising from the starting point. The numbered points in the figure illustrate these starting points. The greedy target selection strategy sometimes makes the robot move longer distances for sensing tasks during the final stages of the exploration mission. Inorder to remove any biases of the results due to this scenario, the travel distances when the environment is 95% explored are also reported. The maximum range of the sensor is set to 3.0m.

Table I summarizes the results for the various exploration missions conducted. The first row corresponds to the results from the classical exploration mission, with $p_T = 0$.

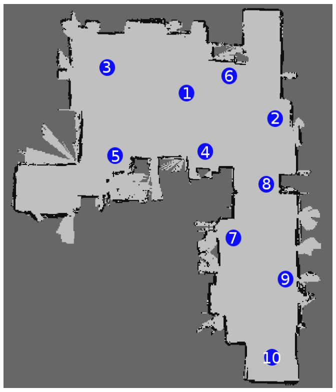


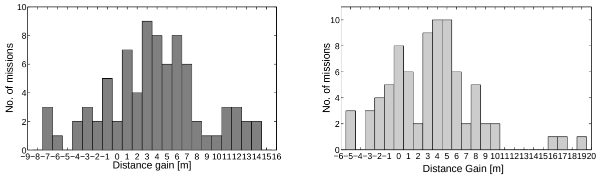
Fig. 8. The map of the indoor area used for experiments. The numbered points are the different starting points used for each (γ, P_T) parameter pair

Overall, augmenting the classical exploration process with cancellation of sensing tasks results in lower travel distances, on average for both the 100% and 95% explored scenarios. However, the standard deviation indicates that, the perceived average performance improvements are not statistically significant. Gain in travel distance during exploration occurs when the classical method selects frontier cells that are near obstacle boundaries as targets and the proposed method avoid reaching such targets. However, selection of such frontier cells by the classical method does not occur in all target selection steps during a mission. In some steps, the selected target may not be closer to any obstacle boundary and both strategies would perform identically on average. Hence, the improvements on the travelled distance by the proposed method gets averaged out over long travel distances and multiple experiments, and reduces the statistical significance of the data. However, it is observed that the number of times the task cancellation based missions having a positive gain for each parameter pair is much higher. Of the total 80 missions executed, 62 have provided a positive travel gain considering 100% exploration and 58 have provided positive travel gains considering 95% exploration. Therefore, inorder to have a better understanding on the effect of task cancellation on missions, the distribution of the travel gains should also be considered. Travel gain distributions for both 100% and 95% explored scenarios, illustrated in fig. 9, indicate that exploration with task cancellation approach generates more efficient motion with high probability. It is also observed that with high p_T values, the number of motion cancellations increase leading to more aggressive exploration missions as predicted. However, the results are inconclusive about the effect of the agressiveness of task cancellation on the efficiency in this environment. Fig. 10 qualitatively compares the efficiency of the proposed method with the classical approach. While the robot ventures in to two narrow passages out of three and moves very close to obstacles as depicted in 10(a) and taking sharp turns (top left of path) in the classical approach travelling 39.91m, the proposed approach avoids entering all the narrow passages and completes the exploration mission in 32.99m and conducts a more efficient exploration mission as expected.

TABLE I

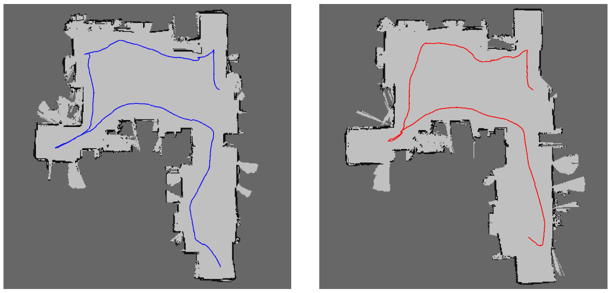
PERFORMANCE OF EXPLORATION MISSIONS

(γ, P_T)	Avg. Travel Dist.		Std-dev.		# Avg. Cancels	# +ve Gains (out of 10 experiments)	
	100%	95%	100%	95%		100%	95%
(*,0)	44.48	39.94	6.53	4.99	-	-	
(45,0.1)	39.36	35.71	5.53	4.84	8	9	8
(45,0.2)	40.29	35.32	4.66	5.28	14	8	8
(45,0.3)	39.23	35.90	3.95	4.24	22	9	8
(60,0.1)	42.31	36.98	3.04	5.28	7	6	6
(60,0.2)	41.44	37.72	3.76	3.82	9	8	7
(60,0.3)	41.20	37.44	4.52	4.10	15	8	6
(60,0.4)	41.91	37.94	4.33	4.55	19	7	7
(60,0.5)	40.13	35.67	4.14	4.80	23	7	8



(a) (b)

Fig. 9. (a) Histogram of travelled distance gain for 100% exploration (b) Histogram of travelled distance gain for 95% exploration



(a) (b)

Fig. 10. Qualitative comparison of classical exploration approach and the proposed approach. Robot start from starting point No. 2. (a) The path taken by robot during the classical exploration mission. Travel distance 39.91m (b) The path taken by robot with interruption. Travel distance 32.99m

VI. CONCLUSIONS AND FUTURE WORK

This work augmented the classical frontier based exploration strategy to include a decision step that cancels sensing tasks if they are no longer desirable. The check for desirability was formulated as a probabilitic decision step and the classical approach is shown to be a special case of this augmented strategy. The experiments revealed that the augmented strategy is capable of conducting efficient exploration missions than the classical approach with high probability. However, the experiments on a single environment type was not sufficient to evaluate the effect of two parameters and variances in efficiency on complete exploration missions though their effect on a single sensing task were evaluated. In future works, we expect to conduct more experiments in different environments to evaluate and find the best range of parameter values for the proposed approach.

REFERENCES

[1] M. A. Batalin and G. S. Sukhatme, “Spreading out: A local approach to multi-robot coverage,” in *in Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems*, 2002, pp. 373–382.

[2] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, “Human-robot teaming for search and rescue,” *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 72–79, jan.-march 2005.

[3] D. Apostolopoulos, L. Pedersen, B. Shamah, K. Shillcutt, M. Wagner, and W. Whittaker, “Robotic antarctic meteorite search: outcomes,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 4174–4179 vol.4.

[4] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings., 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jul. 1997, pp. 146–151.

[5] H. P. Moravec, “Sensor fusion in certainty grids for mobile robots,” *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[6] H. H. Gonzalez-Banos and J.-C. Latombe, “Navigation strategies for exploring indoor environments,” *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, Oct. 2002.

[7] F. Amigoni, V. Caglioti, and U. Galtarossa, “A mobile robot mapping system with an information-based exploration strategy,” in *International Conference on Informatics in Control, Automation and Robotics*, 2004, pp. 71–78.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.

[9] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, and et al., “Coordination for multi-robot exploration and mapping,” in *Proceedings of the Seventeenth National Conference on Artificial Intelligence, AAAI.*, 2000.

[10] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, “The srt method: randomized strategies for exploration,” in *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on*, vol. 5, april-1 may 2004, pp. 4688–4694 Vol.5.

[11] L. Freda and G. Oriolo, “Frontier-based probabilistic strategies for sensor-based exploration,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, april 2005, pp. 3881–3887.

[12] B. Kuipers and Y.-T. Byun, “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations,” *JOURNAL OF ROBOTICS AND AUTONOMOUS SYSTEMS*, pp. 47–63, 1991.

[13] H. Choset and J. Burdick, “Sensor-based exploration: The hierarchical generalized voronoi graph,” *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, Feb. 2000.

[14] S. S. Ge, Q. Zhang, A. T. Abraham, and B. Rebsamen, “Simultaneous path planning and topological mapping (sp2atm) for environment exploration and goal oriented navigation,” *Robotics and Autonomous Systems*, vol. 59, no. 3-4, pp. 228–242, 2011.

[15] F. Amigoni, “Experimental evaluation of some exploration strategies for mobile robots,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, may 2008, pp. 2818–2823.

[16] M. Keidar and G. A. Kaminka, “Efficient frontier detection for robot exploration,” *The International Journal of Robotics Research*, p. 0278364913494911, 2013.

[17] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “Evaluating the efficiency of frontier-based exploration strategies,” in *Proceedings of Joint 41st International Symposium on Robotics and 6th German Conference on Robotics*, Jun. 2010.

[18] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

A Pareto Front-Based Multiobjective Path Planning Algorithm

Alexander Lavin
Computational Engineering & Robotics Laboratory
Carnegie Mellon University
Pittsburgh, PA
lavin@cmu.edu

***Abstract*—Path planning is one of the most vital elements of mobile robotics. With *a priori* knowledge of the environment, global path planning provides a collision-free route through the workspace. The global path plan can be calculated with a variety of informed search algorithms, most notably the A* search method, guaranteed to deliver a complete and optimal solution that minimizes the path cost. Path planning optimization typically looks to minimize the distance traversed from start to goal, but many mobile robot applications call for additional path planning objectives, presenting a multiobjective optimization (MOO) problem. Past studies have applied genetic algorithms to MOO path planning problems, but these may have the disadvantages of computational complexity and suboptimal solutions. Alternatively, the algorithm in this paper approaches MOO path planning with the use of Pareto fronts, or finding non-dominated solutions. The algorithm presented incorporates Pareto optimality into every step of A* search, thus it is named A*-PO. Results of simulations show A*-PO outperformed several variations of the standard A* algorithm for MOO path planning. A planetary exploration rover case study was added to demonstrate the viability of A*-PO in a real-world application.**

Keywords—multiobjective optimization; path plan; search algorithm; A*; Pareto; mobile robot; Mars rover

I. Introduction

A crucial task for mobile robots is to navigate intelligently through their environment. It can be argued that path planning is one of the most important issues in the navigation process [1], and subsequently much research in field robotics is concerned with path planning [2], [3]. To complete the navigation task, methods will read the map of the environment and search algorithms will attempt to find free paths for the robot to traverse. The robot's path is generated by defining a sequence of waypoints between the initial and end positions, while avoiding objects and obstacles. Path planning methods find a path connecting the defined start and goal positions, while environmental parameters play the role as algorithm inputs, and the output is an optimized path from the start to goal [4]. The important issue in mobile robot navigation is optimizing path efficiency according to some parameters such as cost, distance, energy, and time. Of these criteria, time and distance are typically the most important for researchers [5], and methods typically optimize the path efficiency for only one criterion [6]. Yet, many mobile robot operations call for a path plan that optimizes for several parameters. Path optimization over several parameters – e.g. distance and energy – is a multiobjective optimization (MOO) problem. The best path is not necessarily the shortest path, nor the path calling for the least amount of energy expenditure.

Combining the optimization criteria into a single objective function is a common approach, often with tools such as thresholds and penalty functions, and weights for linear combinations of attribute values. But these methods are problematic as the final solution is typically very sensitive to small adjustments in the penalty function coefficients and weighting factors [7]. Evolutionary algorithms, particularly genetic algorithms, have been used widely for MOO problems, including success in path planning [6], [8]. The merging of path segments can result in offspring solutions with high scores across several fitness criteria. The *non-dominated* paths are favored in the population, and this increases generation over generation [10]. Non-dominated solutions are those in which there exist no other solutions superior in all attributes. In attribute space, the set of non-dominated solutions lie on a surface known as the *Pareto front*. Fig. 1 illustrates the two-dimensional case, where there is a tradeoff between minimizing both f_1 and f_2. The goal of a Pareto evolutionary algorithm is to find a set of solutions along the Pareto front, optimal for a combination of criteria [9]. Some state-of-the-art algorithms for multi-objective evolutionary computation include NSGA-II and SPEA2 [11], [12].

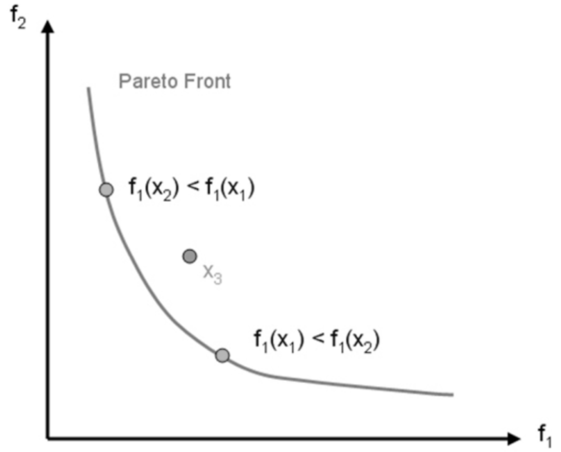


Fig. 1. Two-dimensional Pareto space, where points x_1 and x_2 lie on the Pareto front [13].

This study looks to use multiobjective optimization for mobile robot path planning, but with a Pareto front cost function. Other studies have applied Pareto optimality to evolutionary planning for synchronous optimization of several objectives [8], and domination metrics are used in some evolutionary algorithms for path planning, including NSGA-II and SPEA2 [11], [12]. Yet these algorithms compare complete paths for domination. In order to sort a population according to the level of non-domination, each path must be compared with every other path in the population to find if it is dominated [9]. The algorithm presented in this study, however, checks for non-domination at each search step, resulting in a single, optimal path. The path planning algorithm is novel because each step is Pareto optimal.

The next section further discusses Pareto optimality and the application to mobile robot path planning. Section III discusses the technical approach used in this study, and Section IV presents the results. Included in Section IV is a Mars rover case study as an example application of the new A*-PO algorithm. Other applications for mobile robots with global path planning include agricultural harvesting and information gathering (i.e. drones), disaster relief, DARPA challenges, factory and residential robot workers, and exploration rovers. Section V concludes the paper with discussion and future work.

II. Materials and Methods

A. *Mobile Robot Path Planning*

The aim of mobile robot path planning is to provide an efficient path from start to goal that avoids objects and obstacles. An efficient path is one which minimizes path costs, where the cost is typically the travel distance or time.

Path planning methods can be categorized as either *static* or *dynamic*, according to the environmental conditions. The positions of all obstacles and objects in the static environment are fixed and known. The dynamic environment, on the other hand, may have obstacles and objects which vary positions with time. Similarly, an unknown environment calls for dynamic path planning because more is learned as the mobile robot progresses through the environment. The algorithms for path planning are also in two categories: *local* and *global*. Local algorithms function as the robot moves through the environment, revising the path based on environmental changes. Global algorithms use *a priori* knowledge of the environment to plan the path, and are thus applicable to planning in static environments. Each method has its own pros and cons depending on the environment and application type [8].

The control architecture in mobile robotics is typically a combination of local and global planners, organized as shown in Fig 2. The reactive layer handles local information, with real-time constraints. The deliberative, or global, layer considers the entire world, likely requiring computation time proportional to the problem size [15]. The algorithm presented in this paper is a global path planner.

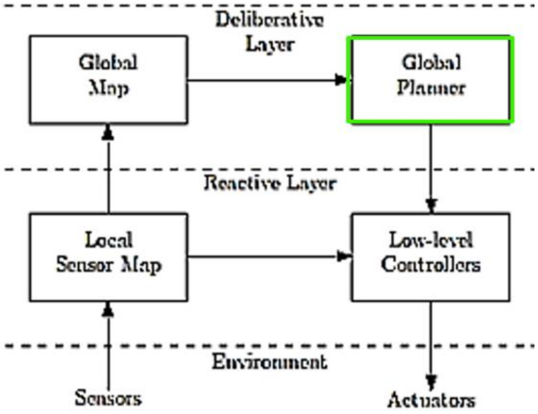


Fig. 2. High-level block diagram of the standard hybrid control system architecture for mobile robots [14]. The focus here is global path planning.

There are two main components of global path planning. First is the robot representation of the world in the *configuration space*: data structures that show the position and orientations of objects and robots in the workspace area, including both the free and obstructed regions. The configuration spaces of path planning algorithms are usually represented by either an occupancy grid, a vertex graph, a Voronoi diagram, generalized cones, or a quad-tree [1].

The methods discussed in this study use an *occupancy grid*, where the environment is represented by a two-dimensional layout of square cells. The values of these cells are binary states, where 0s and 1s represent free and occupied spaces, respectively. The robot occupies a cell, with or without orientation. For a given cell currently occupied by the robot, there are eight feasible cells in the path that can be successors. This is shown in Fig. 3, where the robot in the green position is capable of moving into a neighboring yellow position, but not the occupied gray cells. Feasible solution paths never collide the robot with an obstacle.

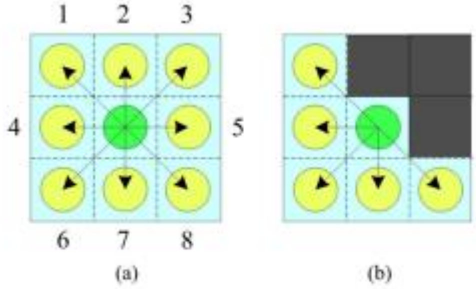


Fig. 3. (a) The robot (green cell) has at most eight possible path steps. (b) The set of feasible successor cells is narrowed because of the three occupied cells (gray) [9].

The second main component of global path planning is implementing an algorithm to find an optimal path from start to goal states. That is, for two arbitrary points in the area – the start and the goal – the algorithm finds a drivable path between them that minimizes distance, energy, or some other criteria. The algorithm employed for the problem must coordinate with the configuration space representation [1]. Potential solution paths connect the start cell to the goal cell via free cells. Searching for the optimum path is an optimization problem, where the optimum path is defined as that which minimizes the path cost, or the *objective function*.

A candidate path can be denoted by

$$P = \{p_1, p_2, \dots, p_n\} \tag{1}$$

where p_i is the ith waypoint of the path P. The MOO problem is then framed as determining a path

$$P^* \in P \tag{2}$$

that satisfies

$$F(P^*) = \min\{F_1(P), F_2(P), \dots, F_t(P)\} \tag{3}$$

where $F_i$denotes the ith cost function of the path planning problem. The study here considers three cost functions, or $t = 3$. They are defined in (4) and (5) below, and (6) later in the Mars rover case study.

Equation (4) gives the total length of the path:

$$F_1(P) = \sum_{i=1}^{n-1} |p_i, p_{i+1}| \tag{4}$$

where $|p_i, p_{i+1}|$ is the Euclidean distance between subsequent cells in the path. Minimizing F_1 finds the path of shortest length from start to goal.

Equation (5) gives the average elevation of the path:

$$F_2(P) = \sum_{i=1}^{n} e_i/n \tag{5}$$

where e_i is the elevation at waypoint $i \dots n$. With the fixed start and goal states at constant elevation, the minimization of F_2 gives the path which climbs up the least amount of incline (or alternatively moves the robot down the most decline).

Search algorithms are employed for finding the minimal cost paths through the configuration space. *Uninformed search* methods are used when no information about the states are known beyond the problem definition [14]. The global path planning problem discussed here has *a priori* knowledge – a map of the exploration area. Thus uninformed search methods, like Dijkstra's breadth-first algorithm, can be ignored in favor of *informed search* methods. The general approach of these methods is *best-first*, which traverses a graph or grid using a priority queue to find the shortest, collision-free path [4]. The decision of the next node expanded, the *successor*, is based on an *evaluation function*, $f(n)$: estimated cost of the cheapest solution through node n. The choice of $f(n)$ determines the search strategy. A bonus of informed search is including a *heuristic function* $h(n)$: the estimated cost of the cheapest path from a node n to the goal state. Greedy best-first search is built solely on this heuristic, where $f(n) = h(n)$, expanding the node closest to the goal at each search step. The incorporation of the heuristic into the path cost makes the search algorithm more efficient.

Algorithm 1 A* Search

```
Initialize open and closed lists
Put the starting node in the open list
Define f, the cost function
While the open list is not empty
    q ← node on open list with smallest f
    Remove q from open list
    Generate q's 8 successors, set their parents to q
    For each successor
        If successor is a goal, then stop search
        successor.g ← q.g + distance between successor and q
        successor.h ← distance from successor to goal
        successor.f ← successor.g + successor.h
        If a node with same position as successor is in the open list & has a lower f than successor, then skip this successor
        If a node with same position as successor is in the closed list & has a lower f than successor, then skip this successor
        Else, add the node to the open list
    End For
    Push q to the closed list
End While
```

The A* algorithm is perhaps the most popular best-first search method, adding to the heuristic the cost to reach the node, $g(n)$. That is, $f(n) = h(n) + g(n)$. The search algorithm, looking for the cheapest path, tries (expands) the node with the lowest $f(n)$ [15], [16]. To determine the optimal sequence of waypoints, the A* algorithm is a favorite for route search problems [17], [18]. For graph search, as opposed to tree search, a *consistency condition* is required to guarantee optimality. A heuristic is consistent if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n':

$$h(n) \le c(n, a, n') + h(n') \tag{6}$$

Norvig and Russel [15] explain how the A* heuristic satisfies the consistency condition, and also that A* is *optimally efficient*: no other optimal algorithm is guaranteed to expand fewer nodes than A*. As long as a better-informed heuristic is not used, A* will find the least-cost path solution at least as fast as any other method.

For real-time planning, where computational speed is a priority, previous studies [19], [20] have modified A* for fast planning. The D* algorithm is a dynamic version of A*, built to be capable of fast rerouting when the robot encounters new obstacles in the environment [4]. The speed of these searching algorithms is increased dramatically, but at the cost of sub-optimal solution paths [14].

B. *Pareto Optimality*

The MOO problem presents multiple cost criteria, where a solution stronger for one criterion may be weaker for another.

There are two general approaches to optimizing for multiple objectives: (i) combine the individual objectives into one composite function, and (ii) determine a *Pareto optimal* solution set. The first can be accomplished with weighted sums or utility functions, but selection of parameters is difficult because small perturbations in the weights can lead to very different solutions. The second option finds the Pareto optimal set of the population, which is a set of solutions that are non-dominated with respect to each other. That is, moving between Pareto solutions, there is always sacrifice in one objective to achieve gain in another objective [21]. It is advantageous to incorporate Pareto fronts in evolutionary algorithm fitness functions when tackling MOO problems. Simply summing over the fitness criteria presents difficulties. Yet in search methods it is common the cost function sums over the cost criteria at each step; the A* algorithm sums $h(n)$ and $g(n)$.

For minimization of objective function f, a point n^* is said to be a Pareto optimal point if there is no n such that $f_i(n) \leq f_i(n^*)$ for all $i = 1 \dots t$, where there are t optimization objectives.

Point $n^* \in C$ is a non-inferior solution if for some neighborhood of n^* there does not exist a Δn such that $(n^* + \Delta n) \in C$,

$$f_i(n^* + \Delta n) \leq f_i(n^*), i = 1, \dots, m, \text{ and}$$

$$f_j(n^* + \Delta n) \leq f_j(n^*) \text{ for at least one } j.$$

Multiobjective optimization is, therefore, concerned with the generation and selection of non-inferior solution points – those on the Pareto front. Pareto optimality is a crucial concept for finding solutions to MOO problems because identifying a single solution that simultaneously optimizes across several objectives is an impossible task [22].

It is worth noting that summing over the costs to calculate a composite f presents another possible issue in search algorithms: depending on the current development of the path, some cost criteria may be favored over others, and this changes as the path development continues. For instance, the A* heuristic – the estimated cost of the cheapest path from the current cell to the goal cell – will contribute more to the cost function close to the start than it will close to the goal. That is, near the start state $h(n)$ will have greater influence on f than will $g(n)$, and vise-versa for the goal state. Thus, as the path develops from start to goal, the heuristic value will contribute less and less. Using a Pareto front solves this issue because each cost criterion is valued as its own dimension in the Pareto space, not summed together.

III. Technical Approach

A. *Costmap*

To calculate cost functions at each step the search algorithm uses a *costmap*. This representation of the configuration space is built off of the aforementioned occupancy grid, but now a cost value is assigned to each cell. Traversing a free space adds a unit cost to the path total, and the obstacles are represented by infinite cost; thus, they are not traversable. If traversing straight across a cell carries a unit distance cost, the cost for traversing a cell at a diagonal (a 45° angle) carries a cost of $\sqrt{2}$.

Yet this costmap only reflects the distance of taking a given path through the configuration space. For a MOO problem, the path cost needs to consider the other cost criteria, for which we use additional *layers*. Each additional cost layer adds a dimension to the Pareto space, from which the Pareto front is calculated. The first costmap layer is the distance cost, $g(n)$. The second layer is the heuristic, $h(n)$. These two suffice for traditional A* search, but we're also interested in optimizing the robot's path for elevation – i.e. minimize (5). A third layer, $e(n)$, is then added to the costmap. With three layers, the Pareto space is three-dimensional. That is, points on the Pareto front are optimal across the three dimensions, one for each cost – distance, the heuristic, and elevation.

B. *A*-PO Search Algorithm*

The algorithm presented in this study, A*-PO, is essentially the standard A* search algorithm but for a key modification: rather than computing the cost function f by summing cost criteria, A*-PO calculates the Pareto front of the cost criteria. Lines 8-16 in the A* pseudocode of Fig. 3 are replaced by the pseudocode shown below.

With a set of at most eight possible directions for the robot to continue at each step, it is very possible the Pareto front will contain multiple successor nodes. That is, q may contain multiple Pareto points. For this scenario where multiple nodes makeup the Pareto front, one is chosen from the set of Pareto points via the normalized A* cost calculation. For instance, of eight directions on a given step of the path, perhaps three fall on the Pareto front. The algorithm will first normalize the three nodes for each cost criteria such that the range for each criterion is [0:1] for the set of nodes on the open list. Then the A* cost metric is used to decide between these Pareto front nodes. Thus, the A*-PO search algorithm still maintains the quality that every step is Pareto optimal.

Algorithm 2 A*-PO Search (replaces line 8+ of Alg.1)

```
8   For each successor
9       If successor is a goal, then stop search
10      successor.g ← q.g + distance between successor and q
11      successor.h ← distance from successor to goal
12      successor.e ← elevation of succesor
13      scoreMatrix(successor) ← [successor.g, … successor.h, successor.e]
14  End For
15  q ← Calculate Pareto front of scoreMatrix
16  If multiple points on Pareto front
17      Normalize scoreMatrix
18      q ← run std. A* cost function on Pareto front nodes
19  Push q to the closed list
20 End While
```

IV. Results

The MOO path planning algorithms are tested in simulated mobile robot environments. The computer simulation environment includes a Lenovo notebook computer with Intel Core i5 vPro CPU and 4 GB memory, running on Windows 8.1. The software is written in MATLAB R2013a.

A. *Algorithm Comparison*

The A* search algorithm, and subsequently the A*-PO algorithm presented here, are guaranteed complete and optimal, but not necessarily for MOO path problems. We evaluated A*-PO by comparing it with A* for a set of 40 simulated environments.

The workspaces were setup as a 20x18 cell grids of randomly assigned free spaces and obstacles, the obstacles accounting for 20% of the workspace. The start and goal locations were fixed at the upper left (0,0) and lower right (20,18), respectively. The elevations for the goal and start states were at 0 in each in configuration, where the terrain ranged [0:1]. Eight unique terrains were used in the simulations.

The optimization objectives, as presented above in (3) and (4), were to minimize the total path distance and elevation. Fig. 4 shows an example of the resulting paths for each search method in pink, where the gray squares represent the path steps; the red and green marked squares represent the start and goal states, respectively. The left side diagrams of Fig. 4 are the final solution paths over a grid of obstacles (black) and free spaces (white), representing the occupancy grid layer of the costmap. The right side diagrams show the same paths over a contour map, representing the elevation layer of the costmap.

For one of the 40 simulation runs, Fig. 4a shows the final solution path of the standard A* algorithm for the distance travelled, the heuristic, and the elevation cost criteria. At each search step the costs for each criteria were normalized [0:1] over the nodes in the open list. It was necessary to normalize the path costs at each search step because the elevation values are small relative to the distance values; without normalization the elevation metric would be insignificant. This normalization is unnecessary for the A*-PO algorithm because each cost value is relative to the cost metric's dimension in Pareto space. Fig. 4b shows the solution path for the A*-PO algorithm of the same simulation environment as A* in Fig. 4a.

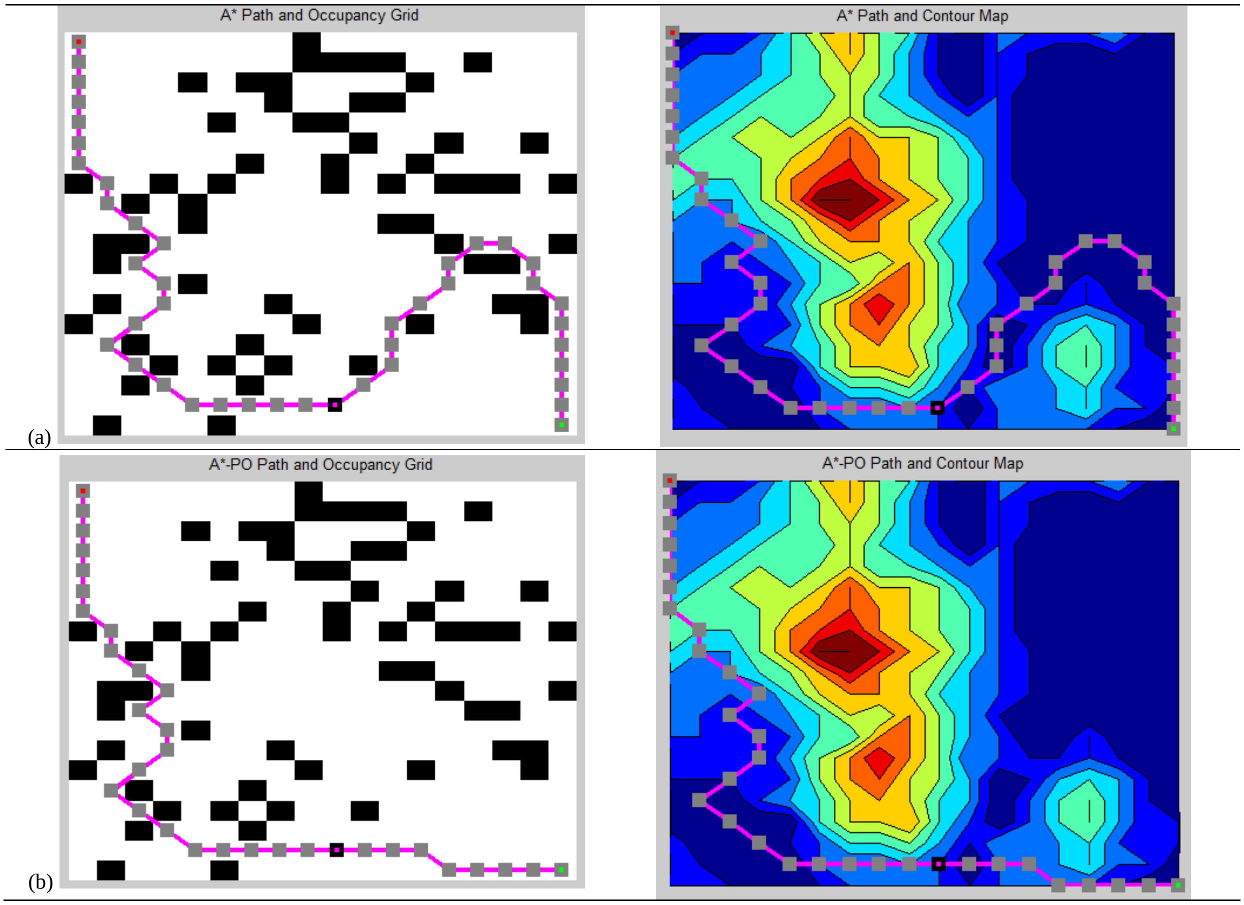


Fig. 4. Solution paths for (a) A* and (b) A*-PO from one of the 40 simulation runs., where the path cost at each step includes distance travelled, the heuristic, and elevation. The divergence in the two paths is plotted with a black square.

For the sample workspace and terrain in this example, it is clear to see the benefits of calculating the Pareto front at each search step. The data over the set of 40 simulations echo these results, as shown in Table 1. The A*-PO algorithm outperforms the other A* variations for the optimization objective functions $F_1(P)$ and $F_2(P)$, the path length (steps) and average elevation (normalized), respectively. The search time results show paths with Pareto optimal steps can be obtained efficiently with the A*-PO algorithm, with only a slight increase in computation time over the standard A* search algorithm. All algorithms gave complete solution paths.

The average elevation of each solution path is used as a metric to compare the robot's net incline from start to goal. A path of a given average elevation implies the robot traversed up less slope (or down more slope) as compared to a path of higher average elevation.

B. *Case Study*

A case study is presented to demonstrate the application of the A*-PO algorithm in a real environment. An example Mars terrain was sourced from HiRISE, the High Resolution Imaging Science Experiment conducted by the University of Arizona, NASA, JPL, and USGS [23]. Fig. 5 shows a digital terrain model of a Mars landscape, from which a section (red square) was extracted for use in the case study.

The extracted section was converted to a terrain map with elevation values [0:1], as shown in Fig. 6. The overlaid occupancy grid was generated randomly, with obstacles accounting for 30% of the workspace. The dimensions are 100x100, where each cell represents a 1m^2 area.

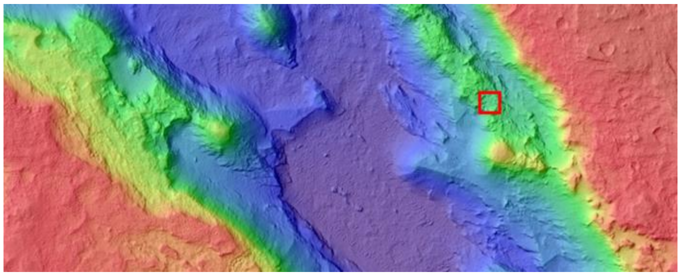


Fig. 5. Terrain map showing exposures of layered bedrock northwest of the Hellas Region of Mars. The selection within the red square was used for simulation.

In addition to the path planning objectives used above, the case study included an additional aim of maximizing the solar incidence on the rear of the rover. That is, the MOO problem included an additional optimization objective to minimize the total angular deflection of sunlight from the solar panel. This was computed by minimizing the dot product of the rover vector $\vec{r}$ and the solar ray vector $\vec{s}$:

$$F_3(P) = \vec{r}\cdot\vec{s} = |r||s|cos\theta \qquad (7)$$

The solar incidence cost criteria was incorporated as an additional layer to the costmap. However, this layer was dependent on the robot's orientation in the configuration space, and was thus dynamic. That is, the costmap changed at each step in the path, depending on the two-dimensional rover vector. For this case study, the solar incidence angle was held constant and two-dimensional. Thus, there were eight possible variations of the solar costmap, or one for each angle between the solar and rover vectors.

TABLE I. RESULTS – SIMULATION AND CASE STUDY

Algorithm	**Environment**	**Comparison Metrics (mean values)**			
		F_1: Path Length (steps)	***F_2: Average Elevation [0:1]***	***F_3: Solar Incidence [0:1]***	***Search Time (seconds)***
A*	Simulation	31.1	0.72	n/a	0.19
A*-PO		28.3	0.65	n/a	0.28
A*	Mars	161.4	0.45	0.53	1.21
A*-PO		151.3	0.39	0.69	1.42

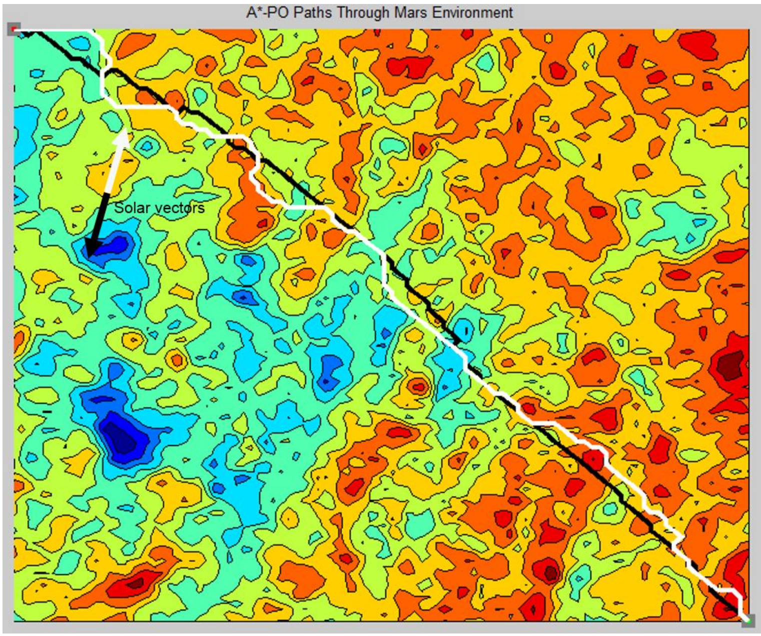


Fig. 6. The A*-PO solution paths through the Mars workspace (total elevation variation is 34.4m). The two paths both minimized the criteria for distance travelled, the heuristic, elevation, and solar angle deflection. The white path reflects solar incidence at angle 70°, while the black path is for 250°.

The result paths shown in Fig. 6 are Pareto optimal at each step across three independent cost functions, F_{1-3} of (4), (5), (7). The three functions cover the four cost criteria because both the distance travelled and the heuristic contribute to F_1. The case study shows the A*-PO algorithm provides the least-cost global path according to several independent preferences for a mobile robot in practice.

Further studies may aim to more accurately include the solar incidence as a cost metric. This can be done by varying the angle of sunlight with time, as the rover progresses along its path. Or calculating the solar incidence in three-dimensional space. Additionally, one may account for more elaborate thermal constraints, such as heating of sensitive components by direct sunlight.

V. Conclusion

In this study, global path planning for mobile robots is investigated. The optimal path is generated according to several cost criteria, solving the multiobjective optimization problem with the presented A*-PO algorithm. As demonstrated in the previous section, A*-PO is capable of providing paths where each step is Pareto optimal, and computes these solutions efficiently. In comparison to the traditional A* algorithm, it can be concluded the incorporation of Pareto fronts in A*-PO offers a better MOO search algorithm.

In future work, Pareto optimality may be incorporated into other algorithms of the mobile robot control system architecture (Fig. 1). The mobile robot community has put an increased emphasis on suboptimal path planning methods which meet the time-critical constraints over slow, optimal algorithms [14]. Local and dynamic path planners, such as D*, may improve with Pareto cost functions.

Acknowledgment

The author would like to thank Professors Matthew Eicholtz and David Wettergreen of Carnegie Mellon University for their continued support and mentorship.

References

[1] Sariff, N.; Buniyamin, N., "An Overview of Autonomous Mobile Robot Path Planning Algorithms," *Research and Development, 2006. SCOReD 2006. 4th Student Conference on* , vol., no., pp.183,188, 27-28 June 2006

[2] P. M. Bhushan Mahajan, “Literature review on path planning in dynamic environment,” International Journal of Computer Science and Network, vol. 2, no. 1, pp. 115–118, 2013.

[3] J.-A. Meyer and D. Filliat, “Map-based navigation in mobile robots:: Ii. a review of map-learning and path-planning strategies,” Cognitive Systems Research, vol. 4, no. 4, pp. 283 – 317, 2003

[4] Stentz, Anthony. "Optimal and efficient path planning for partially-known environments." Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on. IEEE, 1994.

[5] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” Robotics, IEEE Transactions on, vol. 21, no. 3, pp. 354–363, 2005.

[6] Hu Jun; Zhu Qingbao, "Multi-objective Mobile Robot Path Planning Based on Improved Genetic Algorithm," *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on* , vol.2, no., pp.752,756, 11-12 May 2010

[7] Jon T. Richardson, Mark R. Palmer, Gunar E. Liepins, and Mike Hilliard. 1989. Some guidelines for genetic algorithms with penalty functions. In Proceedings of the third international conference on Genetic algorithms, J. David Schaffer (Ed.). Morgan Kaufmann, 191-197.

[8] Samadi, M.; Othman, M.F., "Global Path Planning for Autonomous Mobile Robot Using Genetic Algorithm," Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on , vol., no., pp.726,730, 2-5 Dec. 2013

[9] Jin Yuan; Tao Yu; Wang, Kesheng; Xuemei Liu, "Step-spreading map knowledge based multi-objective genetic algorithm for robot-path

planning," *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on* , vol., no., pp.3402,3407, 7-10 Oct. 2007

[10] Horn, J.; Nafpliotis, N.; Goldberg, D.E., "A niched Pareto genetic algorithm for multiobjective optimization," *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* , vol., no., pp.82,87 vol.1, 27-29 Jun 1994

[11] Deb, K.; Pratap, A; Agarwal, S.; Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on* , vol.6, no.2, pp.182,197, Apr 2002

[12] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," in Parallel Problem Solving from Nature - PPSN VIII, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervs, J. A. Bullinaria, J. Rowe, P. Tino, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3242, pp. 832–842.

[13] Bonino, Simone. "Multi-objective Optimization - Design of Control Arm Using HyperWorks - Altair HyperWorks Insider." *Altair HyperWorks Insider*. N.p., 7 Oct. 2010. Web. 15 July 2014.

[14] Wooden, David T. "Graph-based Path Planning for Mobile Robots." Thesis. Georgia Institute of Technology, 2006.

[15] Norvig, Peter, and Stuart Russell. Artificial Intelligence: A Modern Approach. NJ: Pearson Education Limited, 2013.

[16] Wu, Peng, and Hehua Ju. "Mission-Integrated Path Planning for Planetary Rover Exploration." Journal of Software 8.10 (2013).

[17] Howard, A.; Seraji, H.; Werger, B., "Fuzzy terrain-based path planning for planetary rovers," Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, vol.1, pp.316-320, 2002.

[18] Johnson, Aaron, Jeffrey Hoffman, Dava Newman, Erwan Mazarico, and Maria Zuber. "An Integrated Traverse Planner and Analysis Tool for Planetary Exploration." AIAA SPACE 2010 Conference & Exposition, Anaheim, California, 2010. 1-28 p. American Institute of Aeronautics and Astronautics.

[19] Kuo-Chin Fan and Po-Chang Lui, "Solving the find-path problem in mapped environments using modified A* search algorithm," in IEEE Transactions on Systems, Man, and Cybernetics, vol.24, no.9, September 1994, pp 1390-1396.

[20] Charles W. Warren, "Fast path planning method using modified A* method," in . IEEE International Conference on Robotics and Automation, pp 662-667.

[21] Konak, Abdullah, David Coit, and Alice Smith. Multi-Objective Optimization Using Genetic Algorithms. Department of Industrial and Systems Engineering, Rutgers University. www.ise.rutgers.edu.

[22] Xiao-Bing Hu; Ming Wang; Di Paolo, E., "Calculating Complete and Exact Pareto Front for Multiobjective Optimization: A New Deterministic Approach for Discrete Problems," Cybernetics, IEEE Transactions on , vol.43, no.3, pp.1088,1101, June 2013

[23] "HiRISE." *High Resolution Imaging Science Experiment*. Web. 15 July 2014.<http://www.uahirise.org/>.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session IV

Navigation, Control, Planning

- **Keynote speaker: Alonzo Kelly (Robotics Institute, CMU, Pittsburgh, Pennsylvania, USA)**
  Title: Formulation, Calibration, and Uses of Constrained Kinematic Models for WMRs
  Co-Authors: Neal Seegmiller

- **Title: Modified flatbed tow truck model for stable and safe platooning in presences of lags, communication and sensing delays**
  Authors: A. Ali, G. Garcia, P. Martinet

- **Title: Global Robot Ego-localization Combining Image Retrieval and HMM-based Filtering**
  Authors: Cedric Le Barz, Nicolas Thome, Matthieu Cord, Stephane Herbin, Martial Sanfourche

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session IV

Keynote speaker: **Alonzo Kelly (Robotics Institute, CMU, Pittsburgh, Pennsylvania, USA)**

Formulation, Calibration, and Uses of Constrained Kinematic Models for WMRs

Co-Authors: Neal Seegmiller

Abstract : Recent work on wheeled mobile robot modeling enables a purely kinematic representation of such effects as suspension, wheel slip, and terrain following contact. Such models are relevant to high speed on-road or rough terrain off-road situations and they present compelling advantages in terms of higher estimation and prediction accuracies for less computation, or both. This talk will show results for how such models can be calibrated on-line in real time and how they can be used to improve solutions to state estimation, adaptive control, predictive control, trajectory generation, and motion planning.

Biography: Alonzo Kelly is a professor at the Robotics Institute of Carnegie Mellon University. He received his B. A. Sc. in Aerospace engineering from University of Toronto in 1984, his B. Sc. in computer science from York University in 1990, and his Masters and Ph.D. degrees in robotics from Carnegie Mellon University in 1994 and 1996 respectively. His research interests include perception, planning, control, simulation, and operator interfaces for indoor and outdoor mobile robots. He has worked in the aerospace industry for a decade on various robotics projects for NASA and JPL and on numerous robotics projects in the defense, nuclear, manufacturing and material handling applications. He is the author of Mobile Robots: Mathematics, Models and Methods.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Modeling Mobile Robots

… for High Speed and Off Road
… autonomous and supervisory control

Alonzo Kelly
Professor
Robotics Institute
Carnegie Mellon University

Outline

- Autonomy
- WMR Models
- Applications
- Conclusion

THE ROBOTICS INSTITUTE

Outline

Carnegie Mellon

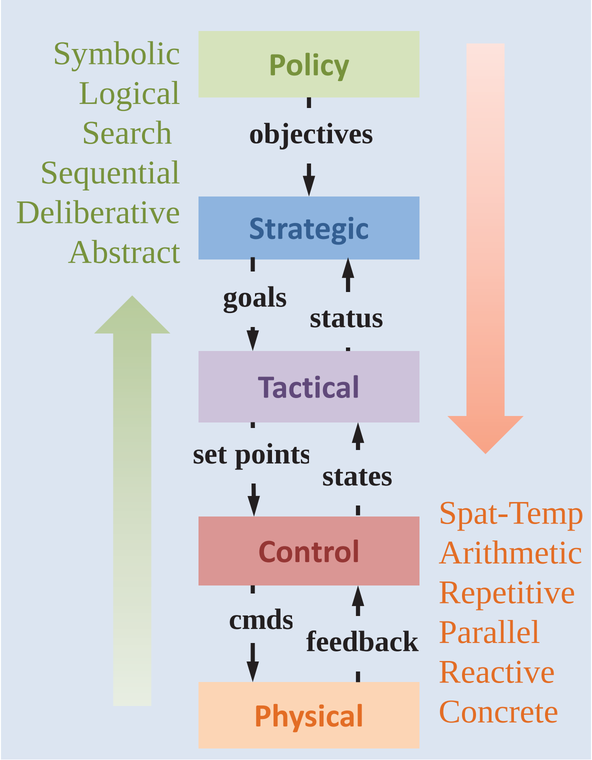
- Autonomy
  - Computation
  - Architecture
- WMR Models
- Applications
- Conclusion

THE ROBOTICS INSTITUTE

Computations

Carnegie Mellon

- Upper levels:
  - Symbols
  - Graphs
  - Propositions
  - Concepts
- Lower levels:
  - Signals
  - Fields
  - Vectors



Symbolic
Logical
Search
Sequential
Deliberative
Abstract

Policy
objectives
Strategic
goals
status
Tactical
set points
states
Control
cmds
feedback
Physical

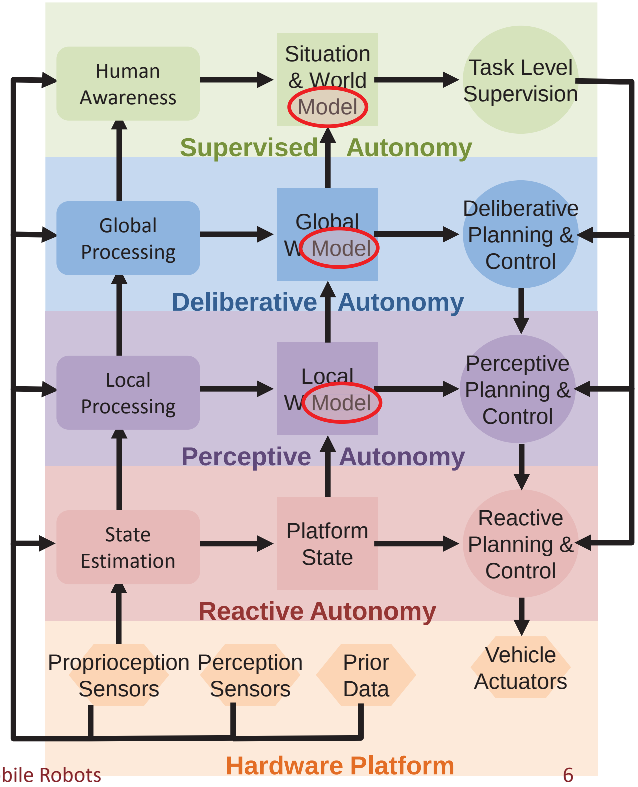
Spat-Temp
Arithmetic
Repetitive
Parallel
Reactive
Concrete

Outline

- Autonomy
  - Computation
  - Architecture
- WMR Models
- Applications
- Conclusion

Autonomy in 5 Layers

- Nested control loops.
  - Commands, state, and models at all levels.
- Processing Levels
  - Supervise = …
  - Deliberate = decide
  - Perceive = see
  - React = …

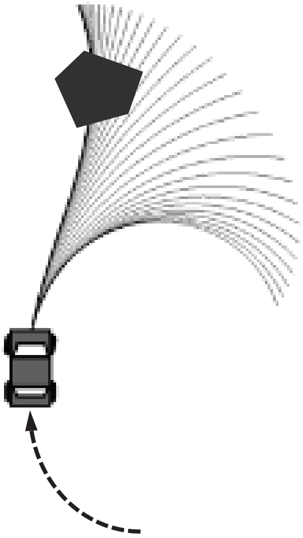


Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
  - Calibration
- Applications
- Conclusion

Need Fast, Accurate Models

- Need correct predictions for:
  - Estimation
  - Control
  - Planning
  - Human interfaces
- Need 1000X faster than real time (with 1% CPU).
  - 10 X a second
  - simulate 10 seconds motion
  - for 10 trajectories.



Trying to avoid the obstacle
On left side at high speed
will cause a collision

Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
  - Calibration
- Applications
- Conclusion

WMR Models : Nature

- Differential:

$$\dot{x} = f(x, u, t)$$

- Underactuated:

$$x \in \Re^n \qquad u \in \Re^m$$

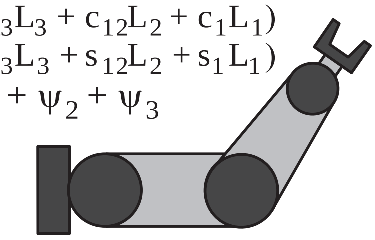
- OverConstrained:

$$v_w \cdot \hat{y} = 0$$

$$z_w = z_{terrain}(x,y)$$

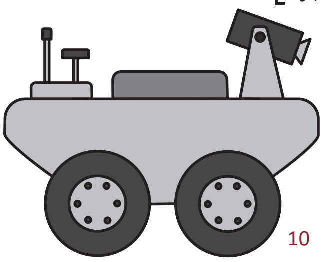
Manipulator

$$\begin{aligned} x &= (c_{123}L_3 + c_{12}L_2 + c_1L_1) \\ y &= (s_{123}L_3 + s_{12}L_2 + s_1L_1) \\ \psi &= \psi_1 + \psi_2 + \psi_3 \end{aligned}$$



WMR

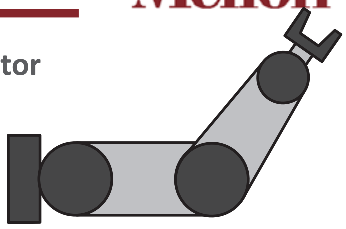
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \zeta(t) \end{bmatrix} = \begin{bmatrix} \cos\zeta(t) & -\sin\zeta(t) & 0 \\ \sin\zeta(t) & \cos\zeta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x(t) \\ V_y(t) \\ \dot{\zeta}(t) \end{bmatrix}$$



Implications

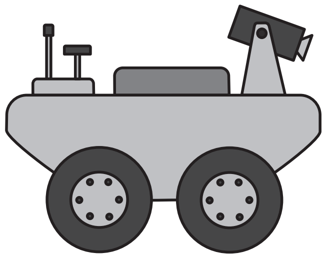
- IK does not exist in closed form.
  - Best case is Fresnel integrals.
  - Requires a numerical approach
- Solution does not exist at all for arbitrary trajectories.
  - Only some motions are feasible.

Manipulator



$$\psi_2 = \operatorname{acos}\left[\frac{(k_1^2 + k_2^2) - (L_2^2 + L_1^2)}{2L_2L_1}\right]$$

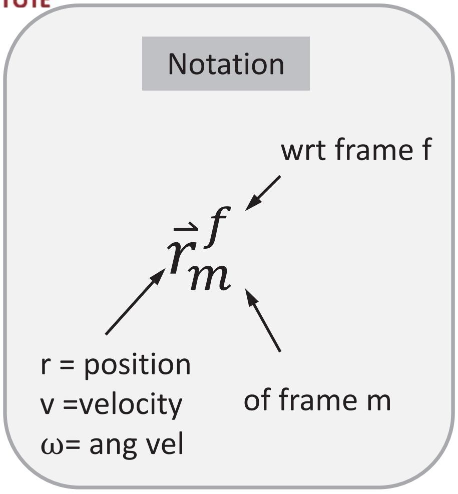
WMR



Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
    - Kinematics
    - DAEs
    - Constraints
  - Calibration
- Applications
- Conclusion

Enabling Kinematics - Transport Theorem

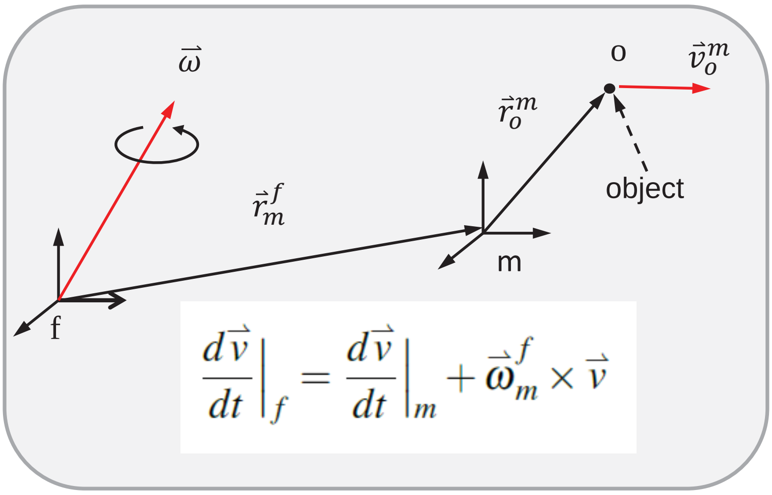

Notation

wrt frame f

$\vec{r}_m^f$

r = position
v =velocity
ω= ang vel

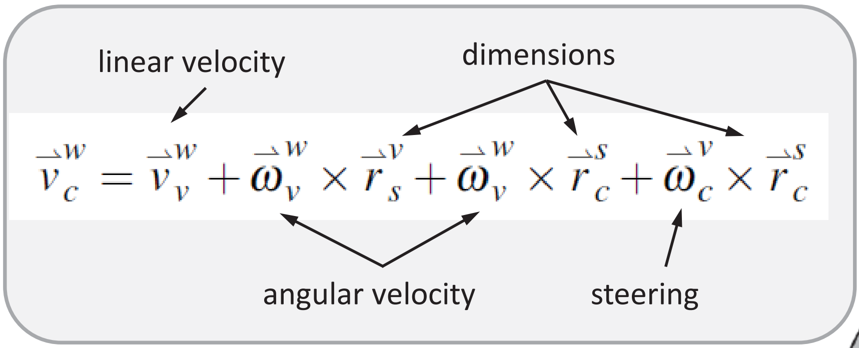
of frame m



$$\left.\frac{d\vec{v}}{dt}\right|_f = \left.\frac{d\vec{v}}{dt}\right|_m + \vec{\omega}_m^f \times \vec{v}$$

- Basic mechanism to convert measurements from moving (robot) frame to fixed (world) frame.

Wheel Equation

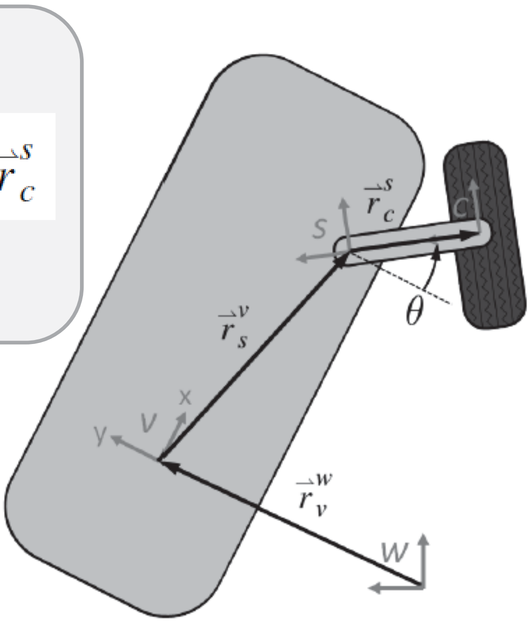

linear velocity

dimensions

$$\vec{v}_c^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_s^v + \vec{\omega}_v^w \times \vec{r}_c^s + \vec{\omega}_c^v \times \vec{r}_c^s$$

angular velocity

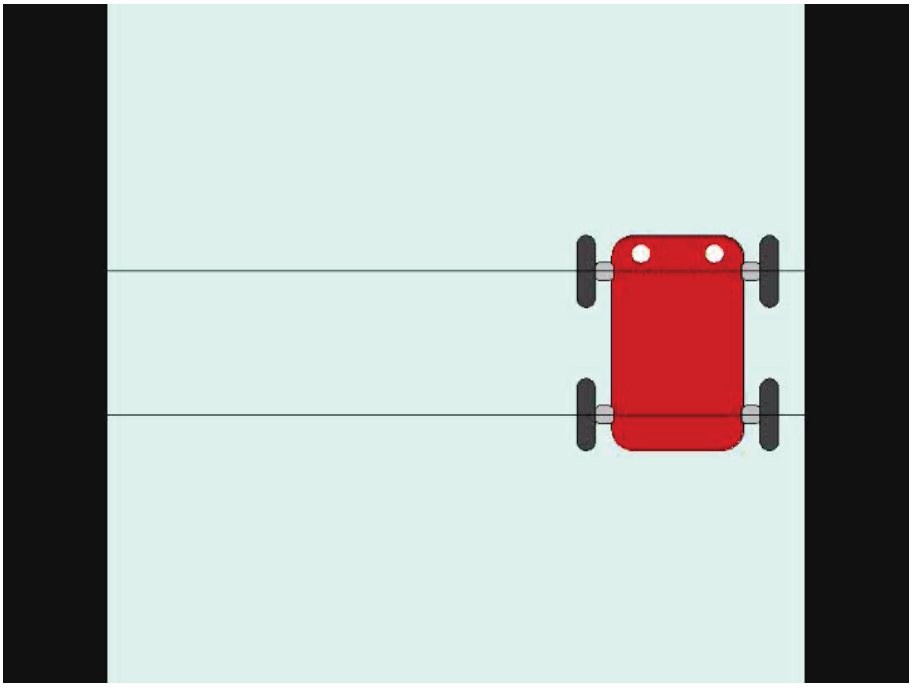
steering



- Vector formulation that relates wheel rotation rates to body linear and angular velocities.

Kelly & Seegmiller, Recursive Kinematic Propagation, to appear IJRR.

Example: 4 Wheel Steer

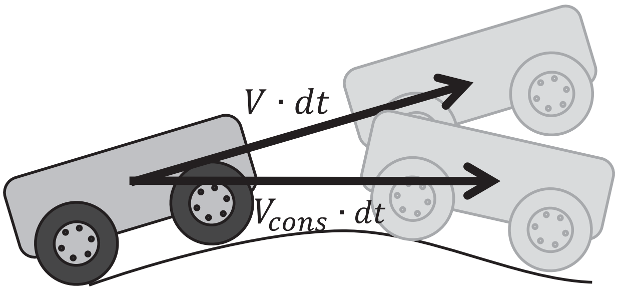


Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
    - Kinematics
    - DAEs
    - Constraints
  - Calibration
- Applications
- Conclusion

Why DAEs

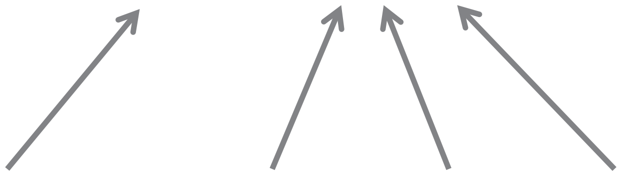
- Solves for unknown constraint forces/velocities automatically.
- Provides constrained derivatives needed for fast, accurate ODE solvers.



$$\|V \cdot dt\| \neq \|V_{cons} \cdot dt\|$$

Notation

$$\dot{x} = f(x, u, t)$$



"state derivative" (velocity etc.)

"state" (x,y,θ)

"inputs" (speed, steer)

time (omitted)

$$x = \int_0^t f(x, u, t)\,dt$$

DAEs

- We formulate velocity kinematics for wheeled vehicles as a constrained, first order, differential equation:

$$\boxed{\begin{aligned} \underline{\dot{x}} &= \underline{f}\left(\underline{x},\underline{u}\right) \\ \underline{c}\left(\underline{x},\underline{u}\right) &= \underline{0} \end{aligned}}$$

1st order ODE

Constraints

$$\begin{bmatrix} H_x^\top H_x & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} \underline{\dot{x}} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} H_x^\top \underline{v}_x \\ \underline{v}_d \end{bmatrix}$$

- Compare that with Lagrange dynamics:

$$\boxed{\begin{aligned} \underline{\ddot{x}} &= \underline{f}\left(\underline{\dot{x}},\underline{x},\underline{u}\right) \\ \underline{c}\left(\underline{x},\underline{\dot{x}},\underline{u}\right) &= \underline{0} \end{aligned}}$$

2nd order ODE

Constraints

$$\begin{bmatrix} M & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \underline{\ddot{x}} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \underline{F}^{ext} \\ \underline{F}_d \end{bmatrix}$$

9/16/2014

Modeling Mobile Robots

DAE Formulation

- Constrained ODE:

$$\underline{\dot{x}} = \underline{f}(\underline{x},\underline{u}) \longleftarrow$$

system dynamics

$$\underline{c}(\underline{x}) = \underline{0} \longleftarrow$$

terrain following

$$\underline{d}(\underline{x})^T \underline{\dot{x}} = \underline{0} \longleftarrow$$

wheel no-slip

- Solve for Lagrange Multipliers (or do nullspace projection) at each iteration:

$$\begin{bmatrix} I & \underline{c}_x^T \\ \underline{c}_x & 0 \end{bmatrix} \begin{bmatrix} \underline{\dot{x}} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \underline{f}\left(\underline{x},\underline{u}\right) \\ \underline{0} \end{bmatrix}$$

- Then integrate w.r.t. time.

Kelly & Seegmiller, WMR modelling with DAEs, submitted IJRR.

Example of DAE Models

Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
    - Kinematics
    - DAEs
    - Constraints
  - Calibration
- Applications
- Conclusion

Wheel Slip Constraint

- Write the wheel equation in contact point coordinates.

$$\vec{v}_c^{w} = H_V V + H_{\dot{\theta}}\dot{\theta}$$

- Set lateral component to zero.

$$\hat{x}\cdot\vec{v}_c^{w} = 0$$

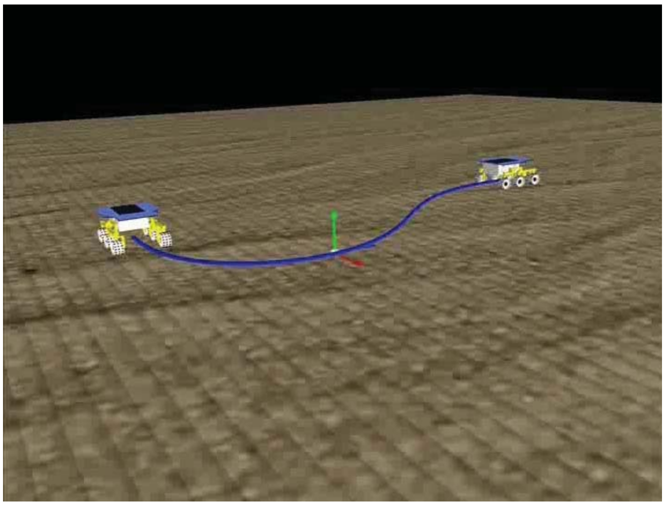
- This is a constraint on $\underline{V}$.

Terrain Following Constraint

- Disallow wheel motion along terrain normal.

$$\hat{n}\cdot\vec{v}_c^{\,w} = 0.$$

- Compute the gradient of this by dot product with system Jacobian.



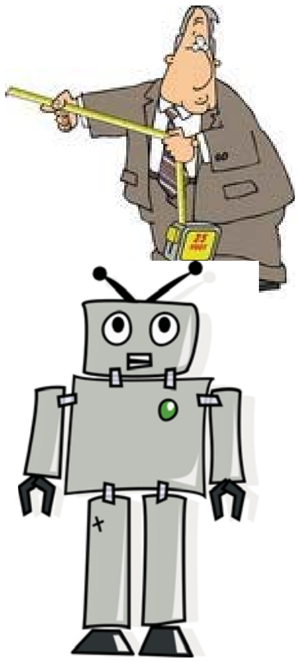
$$\underline{d}(\underline{x})^T = {}^v\hat{n}^T H_v(\underline{\theta})$$

Outline

- Autonomy
- WMR Models
  - Motivation
  - Nature
  - Formulation
  - Calibration
- Applications
- Conclusion

System Identification - Slip

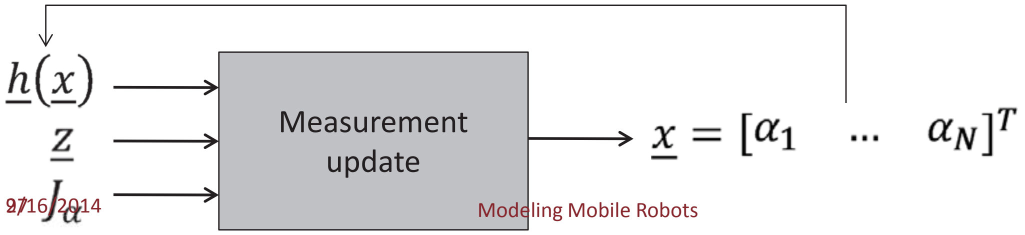
- Model prediction error as an unknown variation (perturbation).
- Form prediction residuals and solve for parameters iteratively in real time.



observation

state

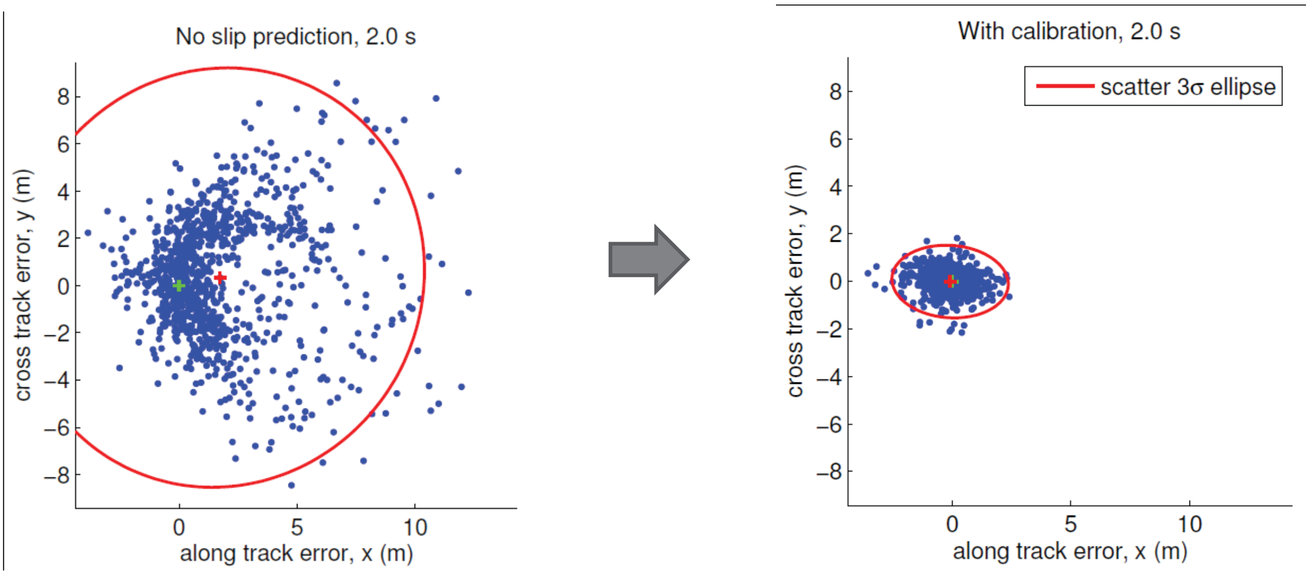
$$\delta \underline{\rho}(\underline{\alpha},t)=\Phi(t,t_0)\delta\underline{\rho}(t_0)+\int_{t_0}^{t}\Gamma(t,\tau)\delta\underline{u}(\underline{\alpha},\tau)d\tau$$



$\underline{h}(\underline{x})$
$\underline{z}$
J_α
Measurement update
$\underline{x}=[\alpha_1 \;\dots\; \alpha_N]^T$

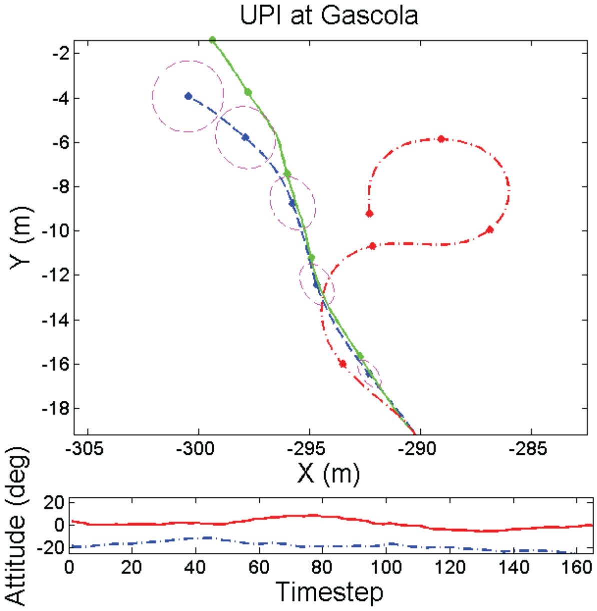
Real Time Slip Model Identification

Results at Extremes

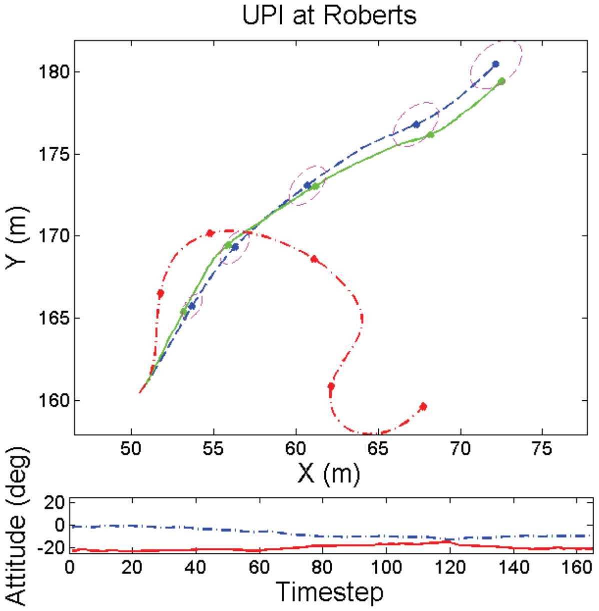


Kelly & Seegmiller, Integrated Prediction Error Minimization, IJRR.

Results at Extremes



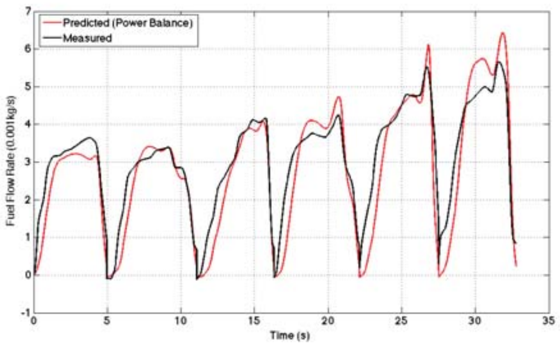
20° Roll Angle

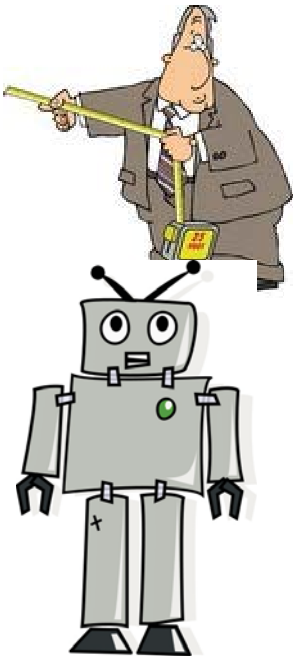


20° Pitch Angle

System Identification - Mass

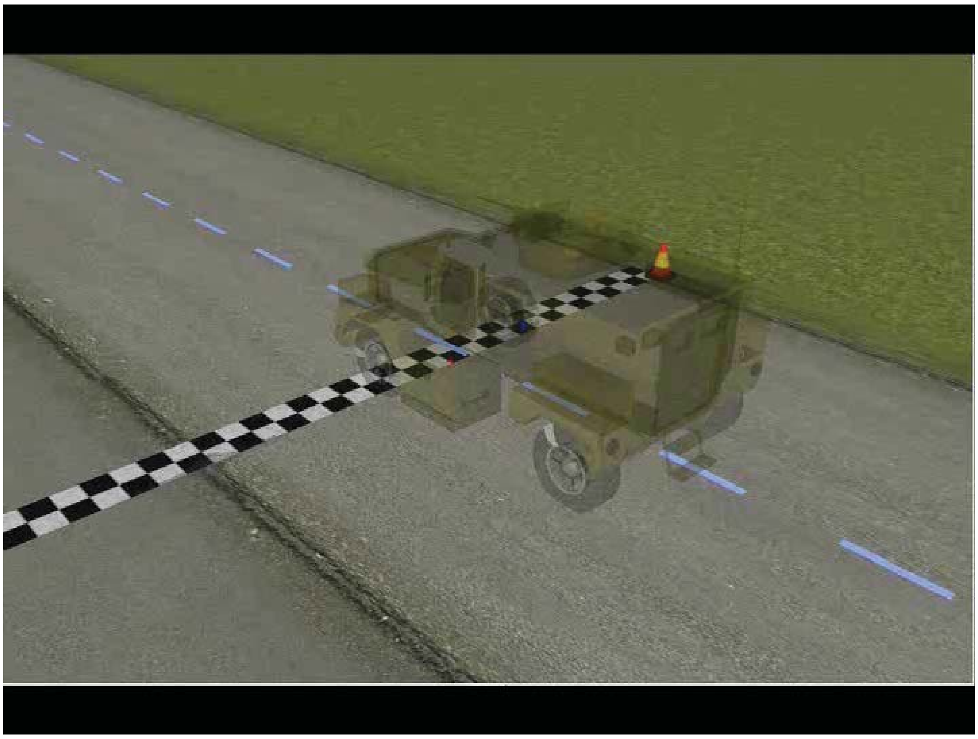
- Calibrate parameters (c.g., stiffness) in motion and structural dynamics.





- Use results for adaptive stability control.

Cg Calibration and Adaptive Stability Control



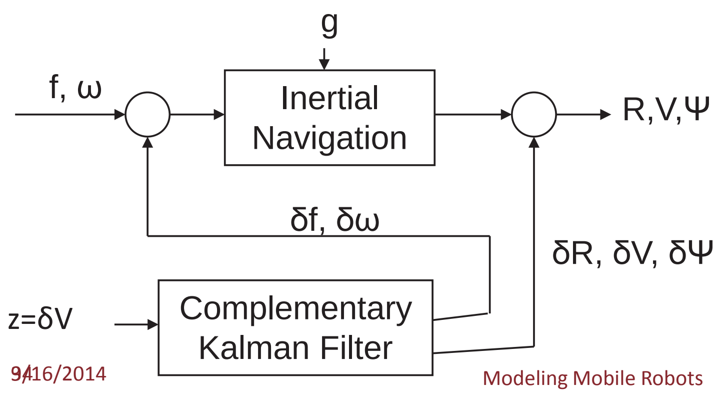
Diaz-Calderon & Kelly, Online Stability Margin …, IJRR

Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Trajectory Generation
  - Motion Planning in Traffic
  - Motion Planning in General
  - Remote Control
- Conclusion

State Est: Inertial Navigation

- **Performance of Tactical Grade Inertial Nav:**
  - **Governed by velocity aiding**
  - **Wheel slip corrupts those measurements.**



g

f, ω

Inertial Navigation

R,V,Ψ

δf, δω

δR, δV, δΨ

z=δV

Complementary Kalman Filter

Key

R - Position

V – Velocity

Ψ – Orientation (Euler)

f – Non-gravitational

a – acceleration

g – Gravity

ω – Angular rate

δ R – Position error

δ V – Velocity error

δ Ψ – Orientation error

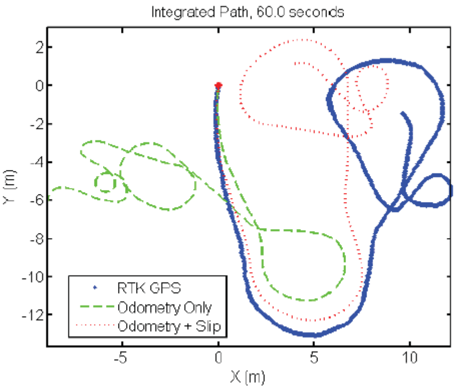
δ f – Accelerometer bias

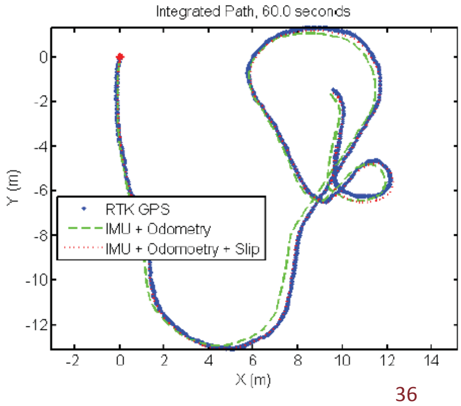
δ ω – Gyro bias

z – Kalman measurement

INS Results: Performance

- **Unaided (free) inertial is not viable at all.**
- **Odometry + slip is far better than odometry alone.**
- **IMU + odometry + slip model somewhat better than IMU + odometry.**
  - **Azimuth error is the dominant component and gyro is already excellent.**





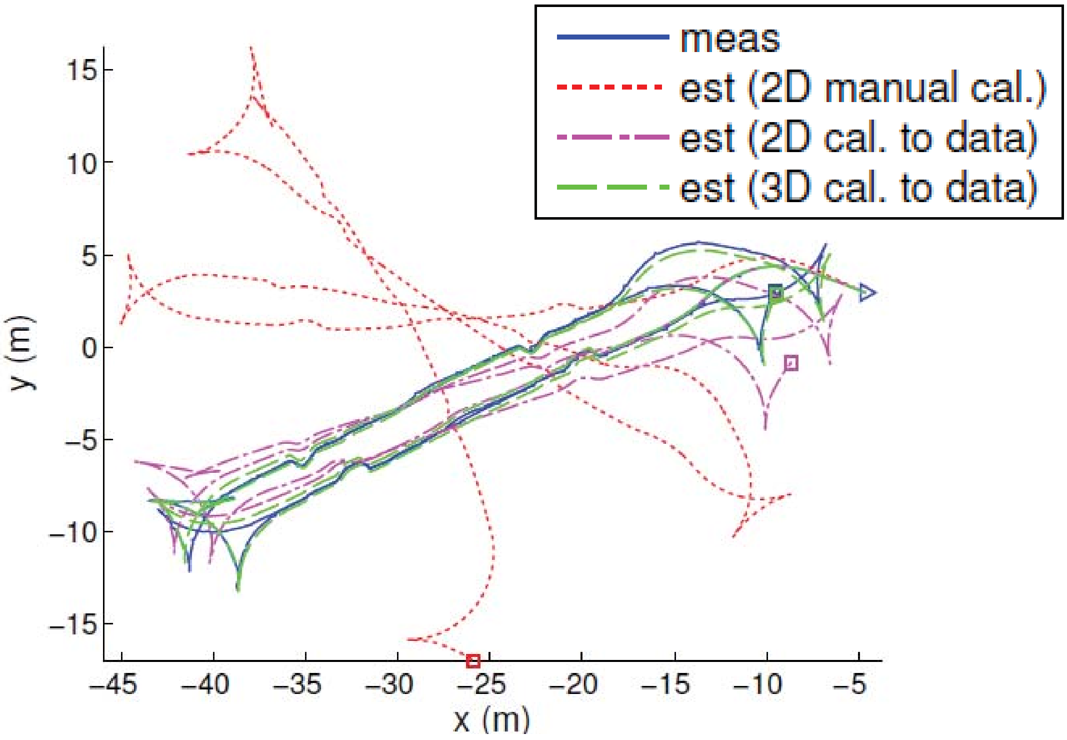
Calibrating Odometry in 3D

- Calibrate
  - Kinematics
  - Slip
- Results after travelling 200 meters + 4 three-point turns
  - 0.25 m (0.1%)
  - 2.3° yaw

Zoe Rover Traversing Ramps Repeatedly

Seegmiller and Kelly, Enhanced Kinematic Models, RSS 2014.

Results



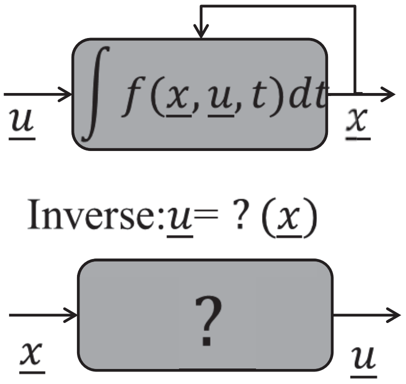
Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Inverting Dynamics
  - Trajectory Generation
  - Motion Planning in Traffic
  - Motion Planning in General
  - Remote Control
- Conclusion

Inverting Dynamics

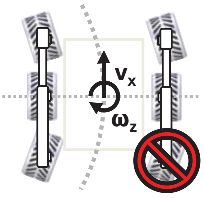
- The equivalent of inverse kinematics in a manipulator is…
- Invert a differential equation. Yikes!!
- In general, there is no solution.
  - For arbitrary trajectory $\underline{x}$.
- In practice, you need one anyway.

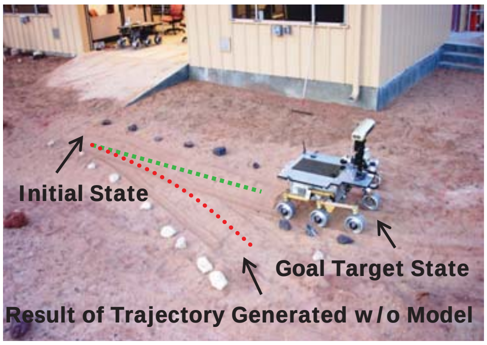
Forward: $\dot{\underline{x}} = f(\underline{x}, \underline{u}, t)$

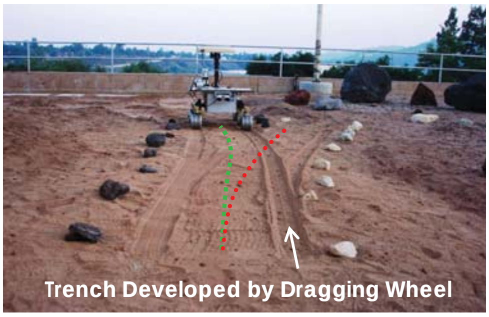

$\underline{u}$
$\int f(\underline{x}, \underline{u}, t)dt$
$\underline{x}$
Inverse: $\underline{u} = ?(\underline{x})$
$\underline{x}$
?
$\underline{u}$

JPL Field Experimentation

Impaired Mobility and Wheel Slip Models (May/June 2007)







Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Inverting Dynamics
  - Trajectory Generation
  - Motion Planning in Traffic
  - Motion Planning in General
  - Remote Control
- Conclusion

Trajectory Gen

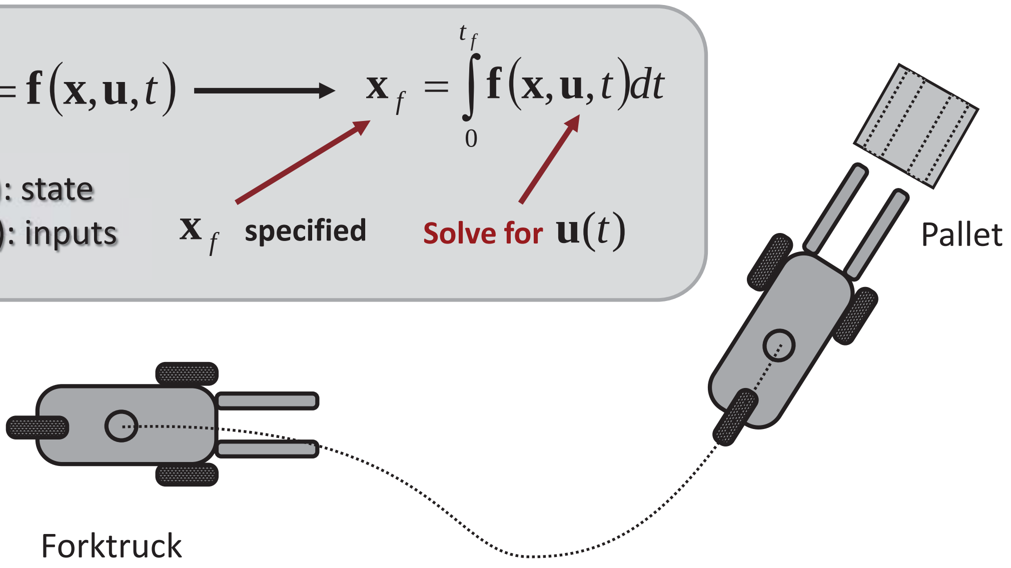
- Original motivation was to get a rover/fork truck to a particular terminal pose.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \longrightarrow \mathbf{x}_f = \int_0^{t_f} \mathbf{f}(\mathbf{x},\mathbf{u},t)dt$$

x(t): state
u(t): inputs

$\mathbf{x}_f$ specified

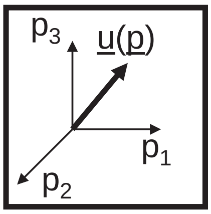
Solve for $\mathbf{u}(t)$



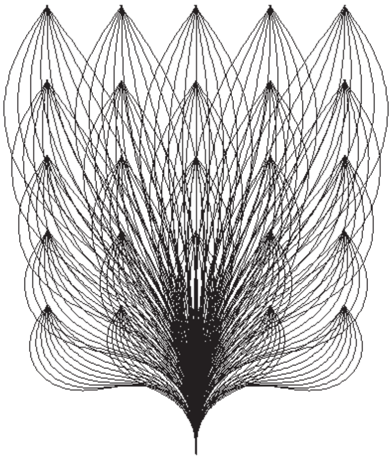
E.g. Polynomial Spirals

- Parameterization:

$$p \rightarrow \mathbf{u}(t) \rightarrow \mathbf{x}(t)$$



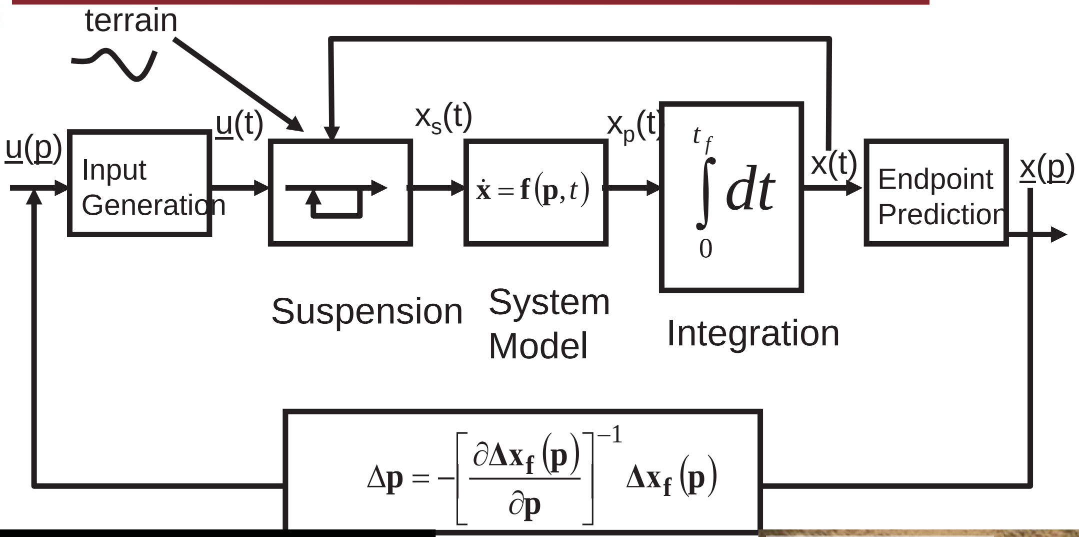
Optimal Control	Nonlinear Programming
$\mathbf{u} = \mathbf{u}(t)$	$\mathbf{u} = \mathbf{u}(\mathbf{p},t)$
$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},t)$	$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{p},t)$



Kelly & Nagy, Parametric Optimal Control, IJRR.

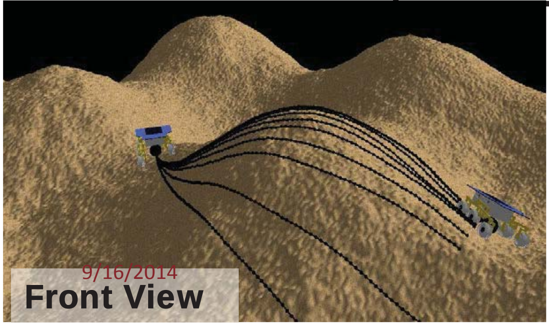
$$\kappa(s) = a + bs + cs^2 + ds^3 + es^4$$

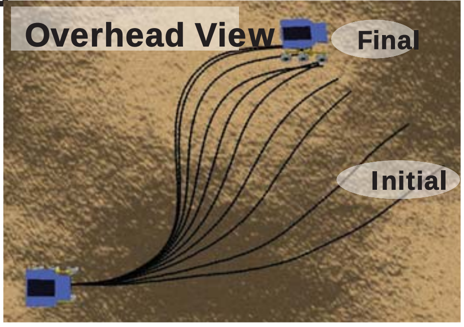
Architecture: 3 Loops



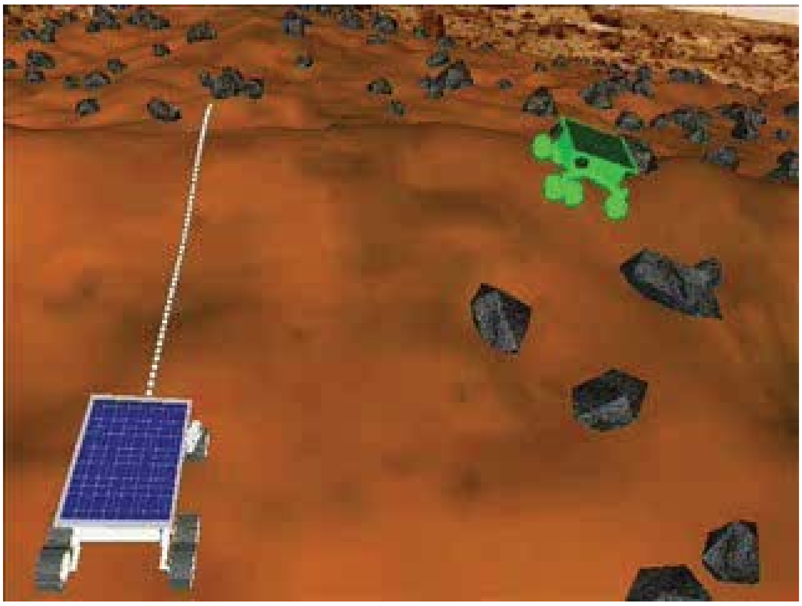
$$\Delta \mathbf{p} = -\left[\frac{\partial \Delta \mathbf{x_f}(\mathbf{p})}{\partial \mathbf{p}}\right]^{-1} \Delta \mathbf{x_f}(\mathbf{p})$$

Parameter Update



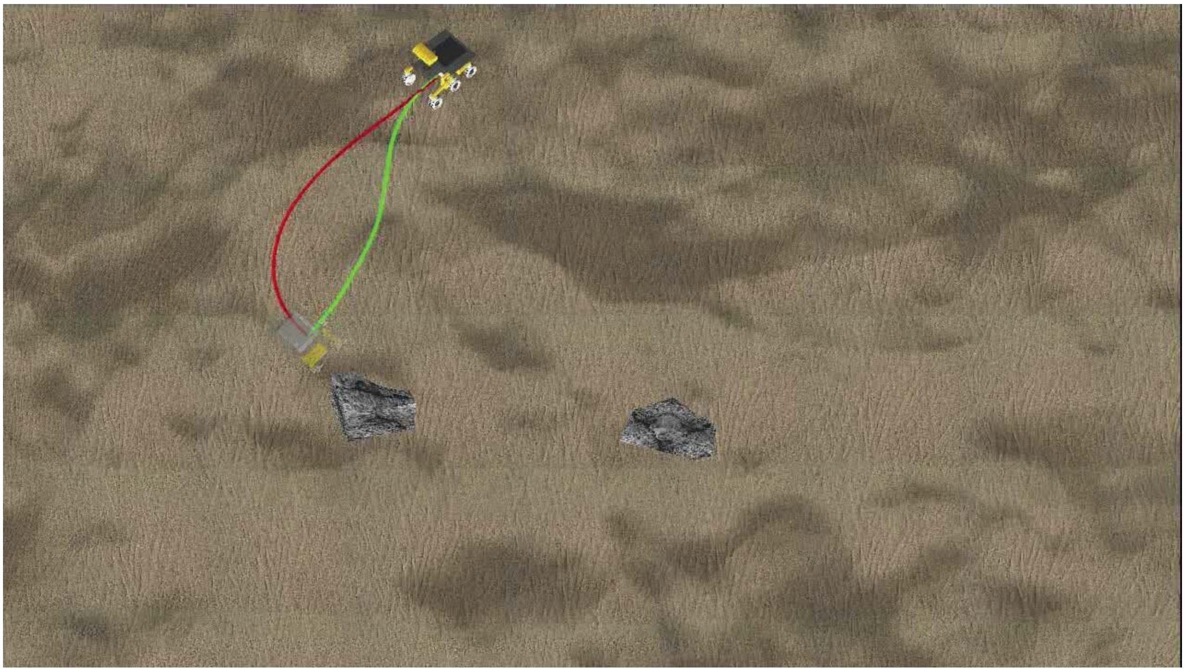


Convergence & Solution



Howard & Kelly, Rough Terrain Trajectory Generation, IJRR.

Trajectory Control

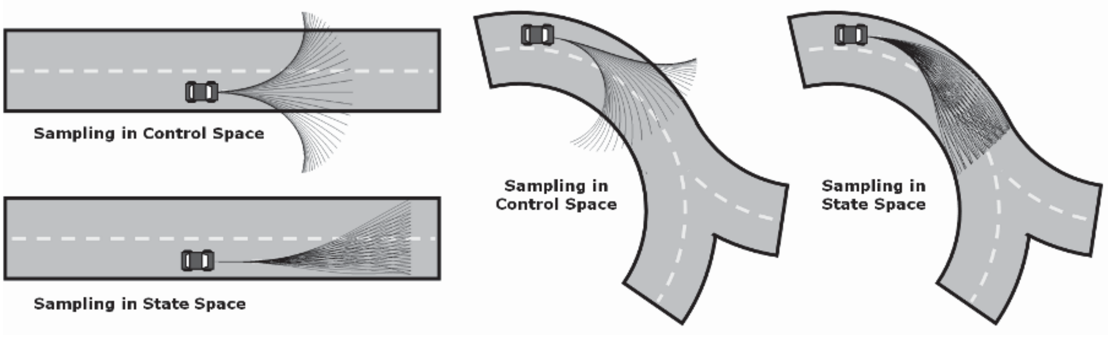


Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Inverting Dynamics
  - Trajectory Generation
  - Motion Planning in Traffic
  - Motion Planning in General
  - Remote Control
- Conclusion

State Space Sampling

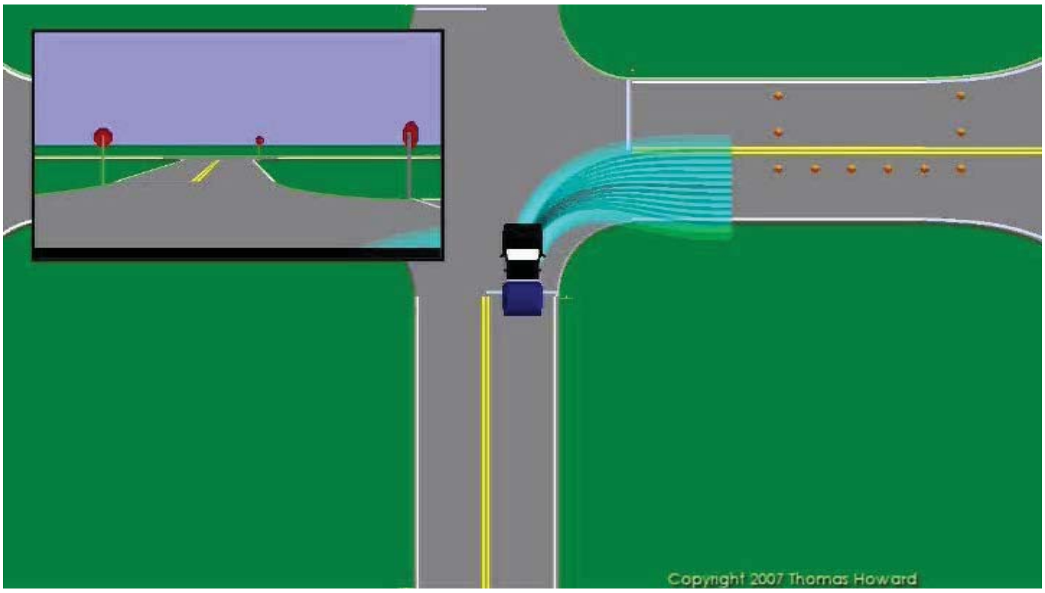
- Road Navigation example:
  - Controls satisfy terminal pose constraints.
  - Search available option for safe and feasible trajectory.



Howard, Green, & Kelly State Space Sampling, FSR 2007

49

In Lane Traffic Planner



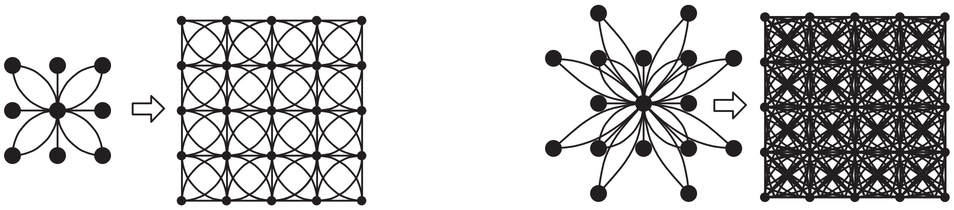
Page 50

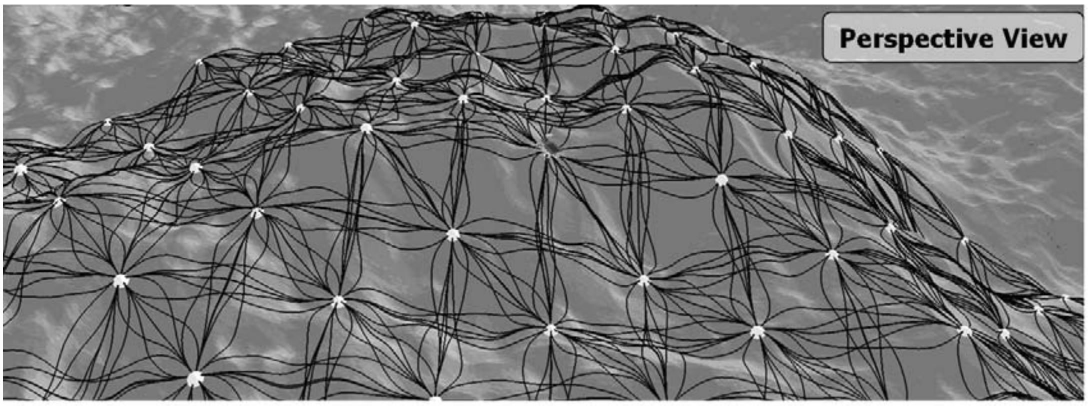
Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Inverting Dynamics
  - Trajectory Generation
  - Motion Planning in Traffic
  - Motion Planning in General
  - Remote Control
- Conclusion

Symmetric, Feasible Controls

- Forms the basis of a symmetric *reachability graph*.



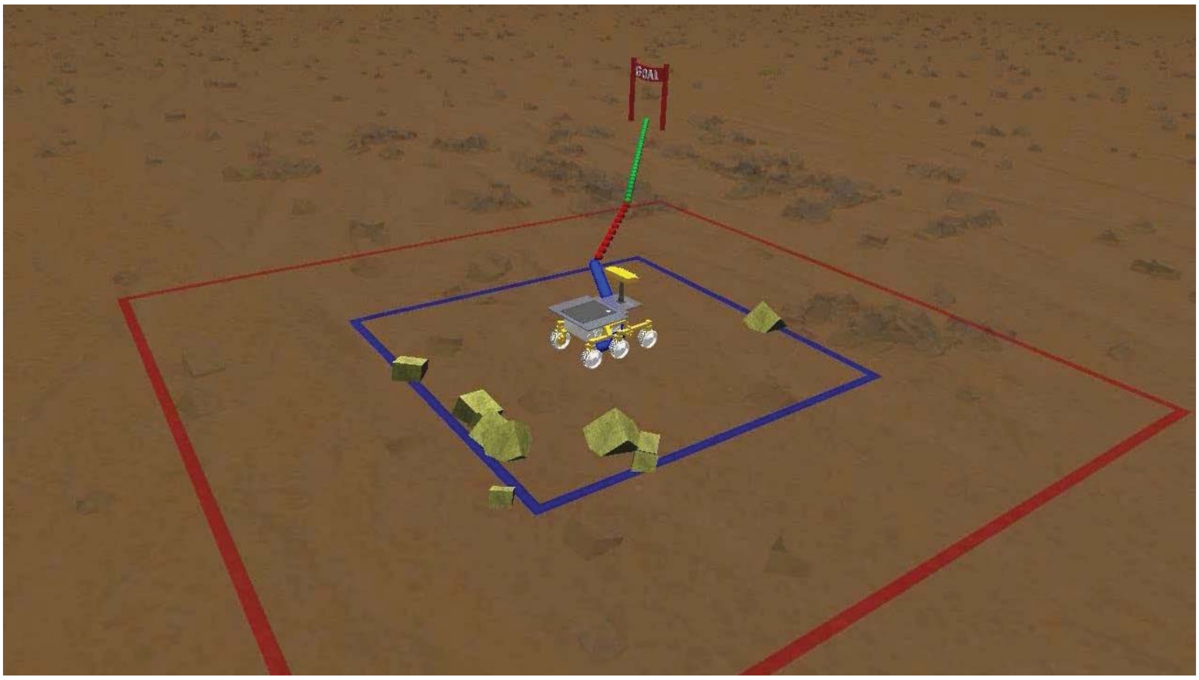


For Real

Pivtoraiko, Kelly & Knepper, Planning in State Lattices, JFR 2009

55

Graduated Fidelity



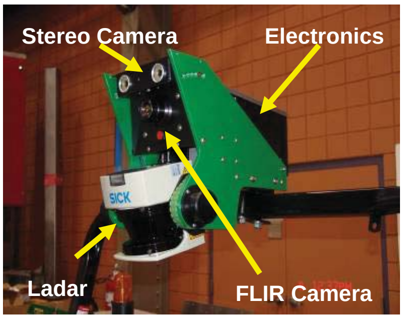
Pivtoraiko & Kelly, Graduated Fidelity, IROS 2008

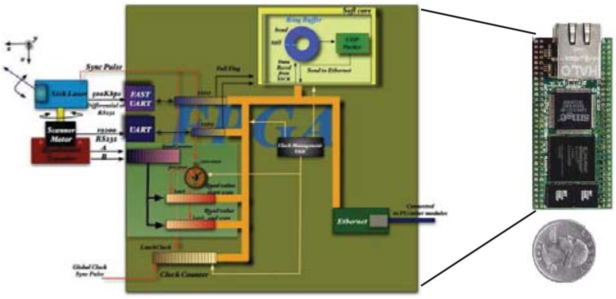
56

Outline

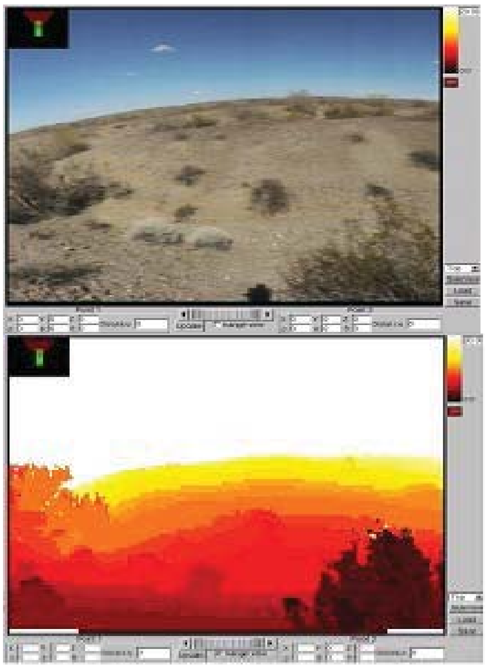
- Autonomy
- WMR Models
- Applications
  - – State Estimation
  - – Inverting Dynamics
  - – Trajectory Generation
  - – Path Following
  - – Motion Planning
  - – Remote Control
- Conclusion

Hardware and Synthetic Imagery

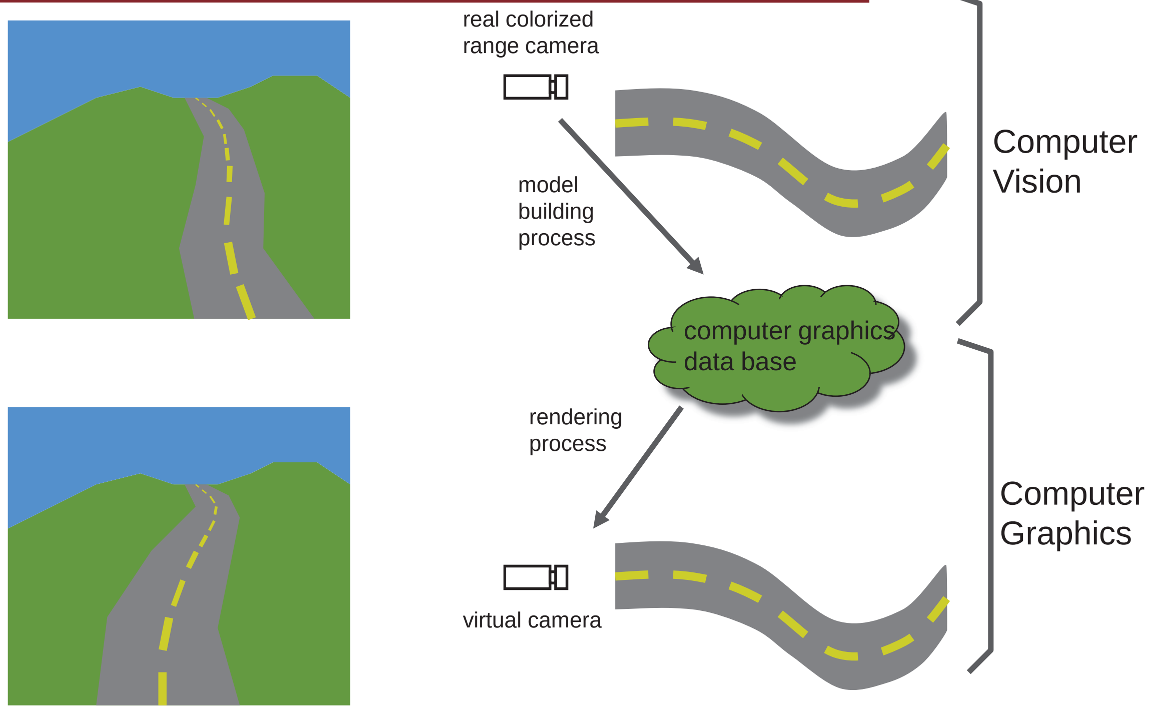




Xilinx Spartan 3 FPGA Based Processing Unit



Video Reprojection

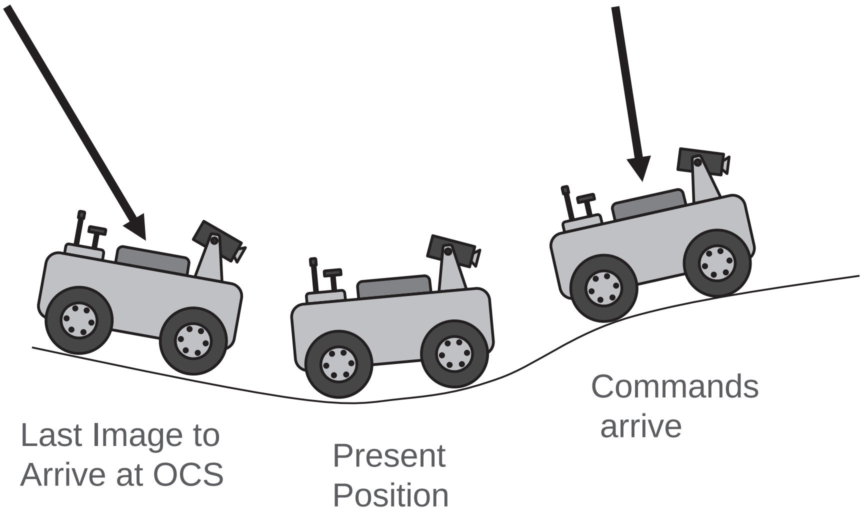


Rendered Information Comes from Real Video
So its highly realistic

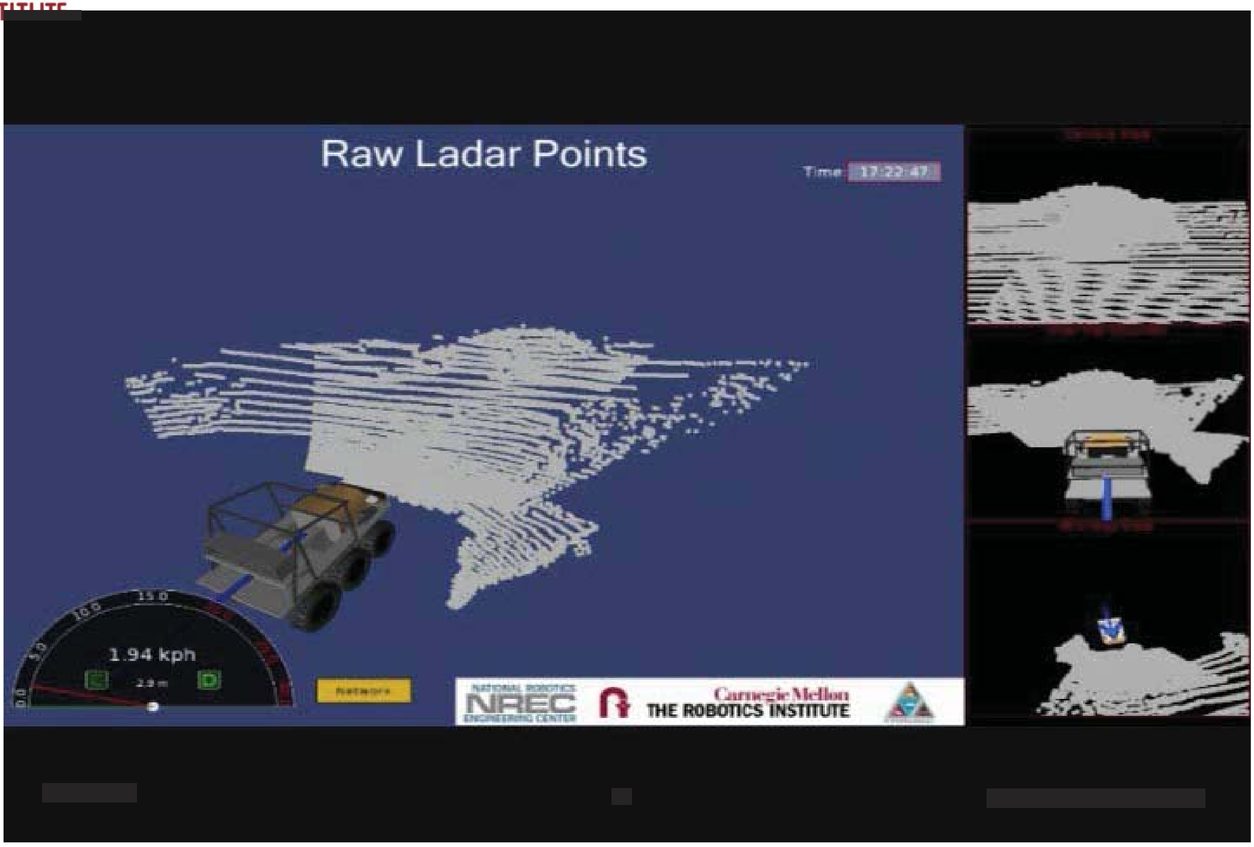
Latency Compensation via Motion Prediction

Distort video gathered at posn 1

To produce video that would be sensed at posn 3

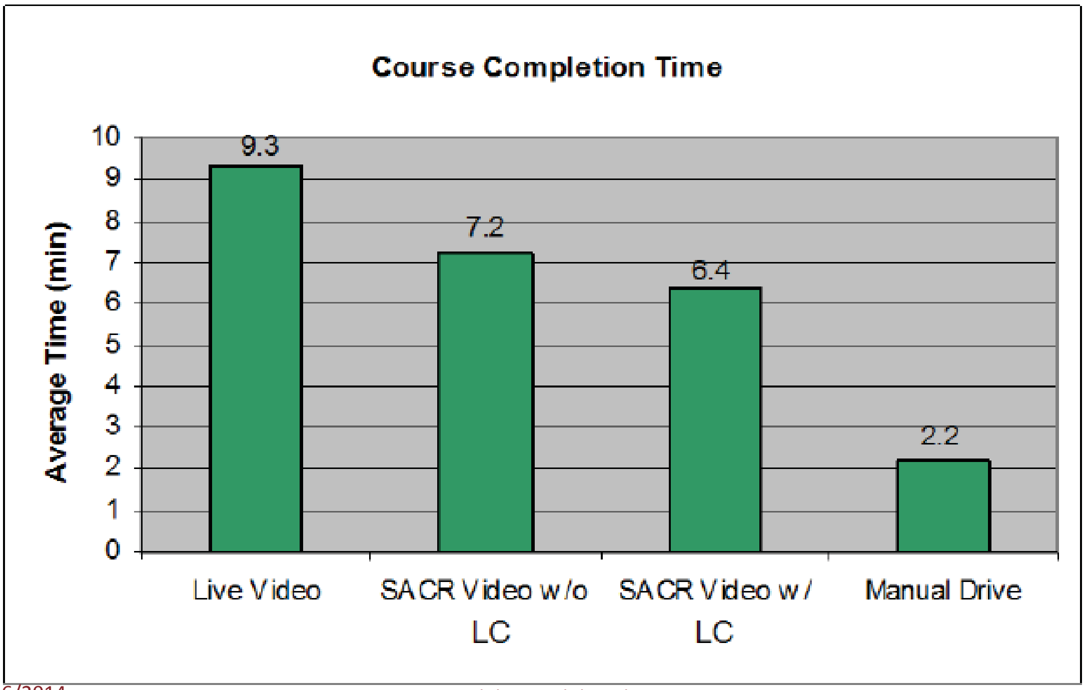


3D Video


Raw Ladar Points
Time 17:22:47
1.94 kph
0.0
5.0
10.0
15.0
2.3 m
Network
NATIONAL ROBOTICS NREC ENGINEERING CENTER
Carnegie Mellon THE ROBOTICS INSTITUTE

Test Results

First Result : 30% Reduction in Test Course Completion Time

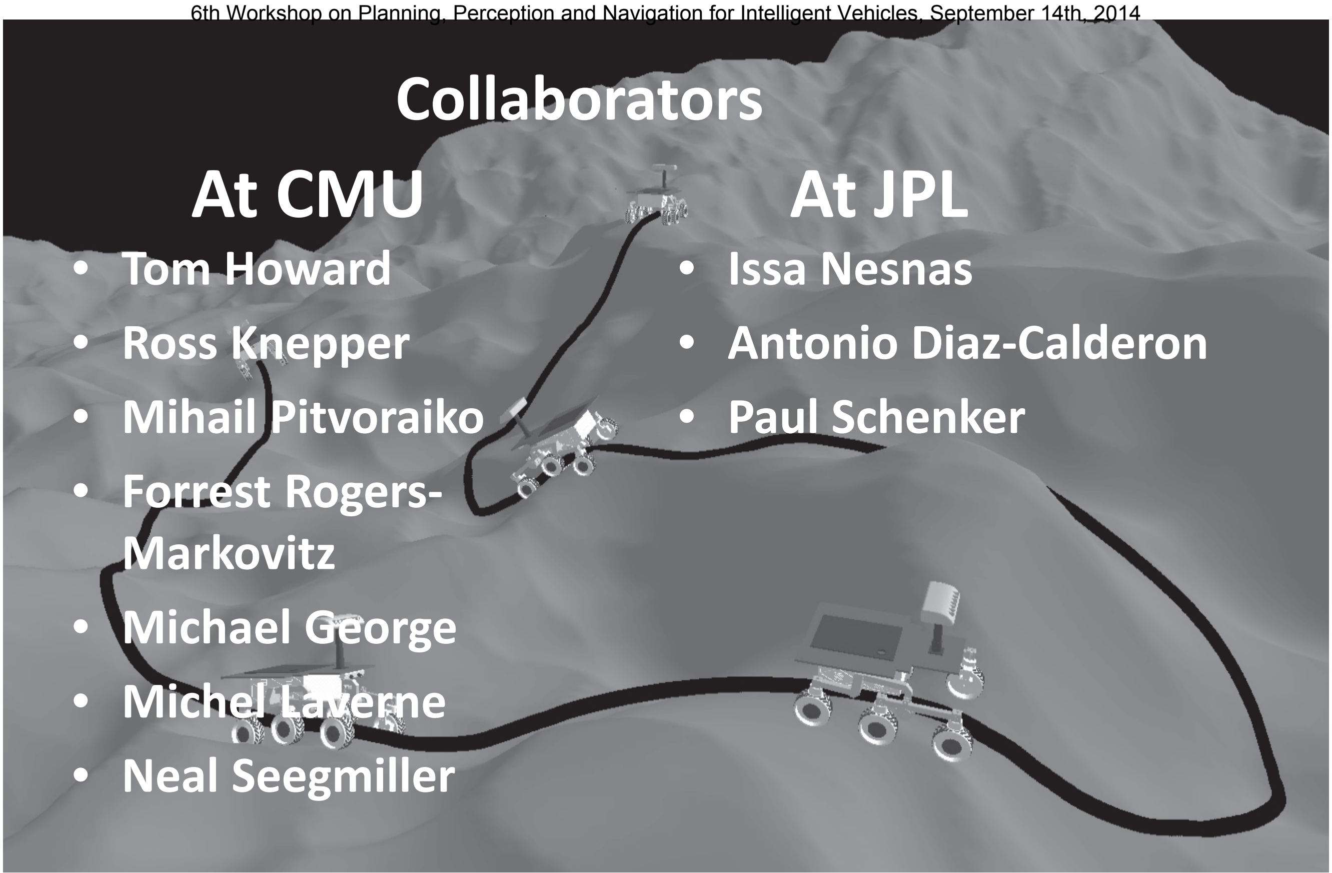


Outline

- Autonomy
- WMR Models
- Applications
  - State Estimation
  - Inverting Dynamics
  - Trajectory Generation
  - Path Following
  - Motion Planning
  - Remote Control
- Conclusion

Conclusion

- (Self) Modeling is the most basic ingredient in predictive control.
- Formulated correctly, it is a DAE.
- WMRs much harder than manipulators.
  - But doable!
- Once done, leads to capacity to act much more intelligently in real applications.



Collaborators

At CMU

- Tom Howard
- Ross Knepper
- Mihail Pitvoraiko
- Forrest Rogers-Markovitz
- Michael George
- Michel Laverne
- Neal Seegmiller

At JPL

- Issa Nesnas
- Antonio Diaz-Calderon
- Paul Schenker

http://www.frc.ri.cmu.edu/~alonzo/resume/detailedresearchinterests.html

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Session IV

Navigation, Control, Planning

- **Title: Modified flatbed tow truck model for stable and safe platooning in presences of lags, communication and sensing delays**
  Authors: A. Ali, G. Garcia, P. Martinet

- **Title: Global Robot Ego-localization Combining Image Retrieval and HMM-based Filtering**
  Authors: Cedric Le Barz, Nicolas Thome, Matthieu Cord, Stephane Herbin, Martial Sanfourche

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Modified flatbed tow truck model for stable and safe platooning in presences of lags, communication and sensing delays

Alan ALI[1], Gaëtan GARCIA[2] and Philippe MARTINET[1]

Abstract— **Many ideas have been proposed to reduce traffic congestion problems. One of the proposed ideas is driving in platoon. Constant spacing policy is the most important control policy. It increases traffic density, but it needs very reliable communication channel. Driving with a constant time headway between vehicle is also well known policy and robust control law, but the inter-vehicle distances are very large. We have proposed in [1], [2] a modification for the constant time headway policy. This modification reduces the inter-vehicle distances largely using only one information shared between all vehicles.**

In this work we propose an additional modification of our control law. This modification makes our control law similar, in form, to the classical constant spacing policy, but it only uses the same shared information. This modification improves the stability of the platoon. We proved the robustness of the control law in presence of parasitic actuating lags, sensing and communication delays.

This prove can be also used for proving the stability of classical spacing policy in presence of all previous delays, contrary to what have been proved in some papers in the literatures.

I. INTRODUCTION

Many ideas have been proposed to solve traffic congestions. Platooning using automated car seems to be promising idea. It increase traffic density and safety, at the same time it decrease fuel consumption and driver tiredness [14]. There are many projects on highways platooning, such as the platooning project in the PATH program (Partners for Advanced Transit and Highways) [15], SARTRE Project [6], and CHAUFFEUR 2 project [7]. Nevertheless, research is still going on for highways and urban areas platooning.

It was concluded that for high capacity traffic the constant spacing policy is necessary at the price of inter-vehicle communication [17].

Using communication may cause instability due to transmission delays or data drop. In [8] the effect of communication delays on string stability has been studied. It has been proved that the platoon becomes un stable for any propagation delays in the communicated leader informations. A solution was proposed in [18] by synchronizing all the vehicles to update their controllers at the same time and using the same leader information, it was shown that string stability can be maintained if the delay in preceding vehicle information is small. The effects of clock jitter, which may cause instability, was briefly mentioned. [10] proved string stability for the leader-predecessor and predecessor-successor framework neglecting information delays between vehicles. The effect of losing the communication is presented in [17]. It has been proved that string stability can be retained, with limited spacing error, by estimating lead vehicle's state during losses.

Another parasitic time delays and lags may be introduced in the physical systems due to actuating and sensing times. This delays may have also significant effects on stability if they are not taken into account. Stability conditions for many control laws, in presence of lags and parasitic delays, can be founded [10], [13], [16], [19]. A detailed study of the effect of delays and lags when using classical time headway policy for homogeneous and heterogeneous platoons is found in [9]. The results show that the time headway policy is more immune, to parasitic sensing and communication delays and actuating lags, than the constant spacing policy. But the large spacings between vehicle make it less important.

In [1], [2] we have proposed a modification of the time headway policy, which reduces the inter-vehicle distances largely to become nearly equal to the desired distance. These works were generalized to urban platoons [3], [4]. In lateral control, we used sliding mode control to ensure stability and robustness. Safety of platoon, when using this control law, was briefly studied in [1], [4] and deeply treated during critical scenarios in [5] . These scenarios include leader and followers hard braking taking into account even in case of communication loss.

In this paper, we continue our previous work. We concentrate on controlling identical tourist cars on nearly flat highways. We propose a modification to our control law. This modification enhances the robustness of the control and increase the immunity to parasitic actuating lags, sensing and even larges communication delays.

This paper is organized as follows: in section II we present a model for the vehicle with and without taking the lags and delays into account, In addition we will give a model for the platoon. The control law will be given in section III. String stability is proved in section IV. Then in section V, we show simulation results. Conclusion and perspective are done in the final section.

II. MODELING

A. Longitudinal Model of the Vehicle

We take a simplified longitudinal dynamic model [2], [9]:

$$\ddot{x} = \dot{v} = W \tag{1}$$

[1] A. ALI and P. MARTINET are with Institut de Recherche en Communications et Cybernétique de Nantes (IRCCYN), Ecole Centrale de Nantes (ECN), Nantes, France

[2] G. GARCIA is with Ecole Centrale de Nantes (ECN), Nantes, France

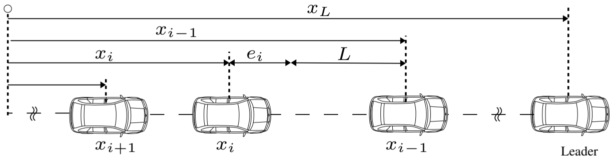


Fig. 1. A platoon

where x: Position of the vehicle, W: is the control input.

B. Vehicle model taking into account parasitic time delays and lags

The model given in (1) is ideal modem and is not sufficient in reality. Using it may lead to unstable control due to presence of parasitic delays and lags. Lags make the net engine torque not immediately equal to the desired torque computed by the controller. Another source of instability is the delay in the communicated data. This delay is due to heavy communications or data drops.

A system model taking into account actuating lags and sensing delays is found in [9]. We extend this model to take into account communication delay, this give us the following vehicle model:

$$\tau_i \; \ddot{v}_i(t) + \dot{v}_i(t) = u(t - \Delta_i, \tau_{c_i}) \tag{2}$$

where τ_i is the combination of the all the lags taken as a *lumped* lag, Δ_i is the combination of the all the sensing time delays taken as a *lumped* delay, τ_{c_i} is communication delay.

C. Platoon Model

The platoon is a set of vehicles moving together at the same speed and keeping a desired distance L between each two consecutive vehicles.

The spacing error of the i-th vehicle, assuming a point mass model for all vehicles, is defined as follow:

$$e_i = \Delta X_i - L \tag{3}$$

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.
- $\dot{e}_i = \dot{x}_{i-1} - \dot{x}_i = v_{i-1} - v_i$:the kinematic evolution of the spacing error
- $v_i = \dot{x}_i$ represents the speed of the i-th vehicle.

The longitudinal model of the platoon, shown in fig. 2 is called flatbed tow track model [1]. It is a set of vehicles virtually connected by one-directional spring-damper systems, and a virtual truck which is set to drive at a speed V, the value of V being known to all vehicles of the platoon. In this paper, we proposed to add new virtual spring between each vehicle and the virtual truck. This enhanced the stability and made our control law similar to constant spacing policy. The main difference is that in our case all the vehicles receive only the speed of the virtual truck V then each vehicle compute the position of the virtual truck X_V by integration.

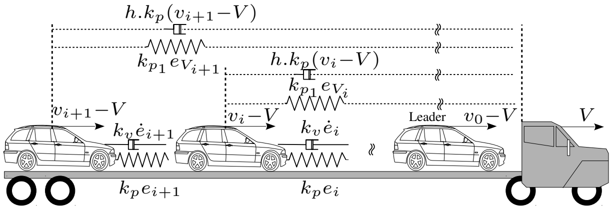


Fig. 2. Enhanced flatbed tow truck model

III. CONTROL LAW AND SPACING ERROR DYNAMICS

A. Control Objectives

The main objectives of the control law are:

1) Make the inter-vehicle distance equal to L so $e_i \to 0$.
2) All vehicles must move at the same speed so $v_i \to v_L$.
3) Stable platoon (String stability).
4) Increase traffic density.
5) Safety (collision free).
6) Stability and safety in case of communication losses.
7) Stability and safety even in presence of sensor time delay, actuator lags and communication delays.

Objectives from 1 to 6 are deeply studied in [1]–[5]. In this work we deal with Objectives 7.

B. Longitudinal Control

Introducing the virtual truck in the new longitudinal model enable us to deal with relative speed instead of the absolute speed, this enhances the performance of the longitudinal control by reducing the distance required to ensure string stability. This model is a modification of the classical time headway policy by subtracting a new term V form all speeds.

Spacing error becomes [2]:

$$\delta_i = e_i - h\ (v_i - V) = e_i + h\ \dot{e}_{V_i} \qquad i = 1...N \tag{4}$$

We add new term $\frac{\lambda_1}{h}\ e_{V_i}$ to our control law given in [2]. The new term is proportional to the distance between the i-th vehicle and the truck:

$$W_i = \frac{\dot{e}_i + \lambda\ \delta_i + \lambda_1\ e_{V_i}}{h}, \quad i = 1...N \tag{5}$$

Where $e_{V_i} = X_{V_i} - x_i - i\ L$,

N: is the total number of vehicles in the platoon.

V: is a common speed value shared by all vehicles of the platoon, it must be the same value for all the vehicles at the same sampling time [1], [2].

X_{V_i}: is the position of the virtual camion, it can be computed by accumulating V.

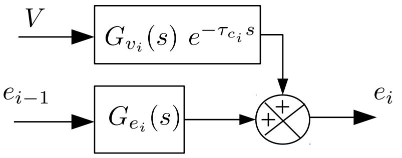


Fig. 3. Spacing error model taking delays and lags into account

C. Longitudinal Control With Delays and Lags:

The control law of the platoon when taking into account delays and lags becomes the following:

$$W_i(t,\Delta_i,\tau_{c_i}) = \frac{\dot{e}_i(t-\Delta_i) + \lambda\delta_i(t,\Delta_i,\tau_{c_i}) + \lambda_1 e_{V_i}(t,\Delta_i,\tau_{c_i})}{h} \tag{6}$$

Where:

$$\delta_i(t,\Delta_i,\tau_{c_i}) = e_i(t-\Delta_i) - h\ (v_i(t-\Delta_i) - V(t-(\Delta_i+\tau_{c_i}))) \tag{7}$$

$$e_{V_i}(t,\Delta_i,\tau_{c_i}) = X_V(t-(\Delta_i+\tau_{c_i})) - x_i(t-\Delta_i) - i\ L \tag{8}$$

With no loss of generality, we assume that $v_N(0) = v_i(0) = ... = v_0(0)$, $a_N(0) = a_i(0) = ... = a_0(0)$, $\delta_N(0) = \delta_i(0) = ... = \delta_0(0)$ $(0 \leq i \leq N)$ at the initial conditions.

We define $\Delta_{c_i} = \tau_{c_{i-1}} - \tau_{c_i}$ the propagation delay, form vehicle i to vehicle $i-1$, of the leader's transmitted data.

For homogeneous platoon we have :

$\Delta_i = \Delta_{i-1} = ... = \Delta, \tau_i = \tau_{i-1} = ... = \tau$, $\tau_{c_{i-1}} - \tau_{c_i} = \Delta_{c_i} = ... = \Delta_{c_1} = \Delta_c$ so $\tau_{c_i} = i\ \Delta_c$. Hence, $G_{e_i} = G_{e_i}(s) = G_{e_{i-1}} = ... = G_{e_1} = G_e$, $G_{V_i} = G_{V_{i-1}} = ... = G_{V_1} = G_V$.

Using (6), (7) and (2) and then by calculating Laplace transformation taking into account the previous assumptions we get:

$$e_i(s) = G_e(s)e_{i-1}(s) + G_V(s)e^{-\tau_{c_i}s}V(s), \qquad i = 2...N \tag{9}$$

Where

$$G_e(s) = \frac{(s+\lambda)\ e^{-\Delta\ s}}{h\tau s^3 + h\ s^2 + ((1+h\lambda)s + \lambda + \lambda_1)e^{-\Delta\ s}} \tag{10}$$

$$G_V(s) = \frac{(\lambda\ h\ s + \lambda_1)\ e^{-\Delta\ s}(e^{-\Delta_c\ s} - 1)}{s(h\tau s^3 + h\ s^2 + ((1+h\lambda)s + \lambda + \lambda_1)e^{-\Delta\ s})} \tag{11}$$

Equation (9) shows that the error of the i-th vehicle is not just a function of e_{i-1} but it is also a function of the shared speed $V(s)$ as shown in fig. 3, this is due to presence of communication delay.

It is very important to compute the dynamics of e_1. This dynamics has an important effect on the stability and the safety of the platoon. By using (6), (7) and (2) and by adding and subtracting $(\tau h\ddot{v}_0 + h\dot{v}_0 + \lambda h v_0 + \lambda_1 x_0)$ we get the dynamics of e_1 as a function of the speed of the leader v_0 and V :

$$\begin{aligned}&\tau h\dddot{e}_1(t) + h\ddot{e}_1(t) + (1+\lambda h)\dot{e}_1(t-\Delta) + \lambda e_1(t-\Delta) = \\ &\tau h\ddot{v}_0(t) + h\dot{v}_0(t) + \lambda h v_0(t-\Delta) - \lambda h V(t-(\Delta+\tau_{c_1})) \\ &+\lambda_1 x_0(t-\Delta) - \lambda_1 X_V(t-(\Delta+\tau_{c_1}))\end{aligned} \tag{12}$$

We compute Laplace transformation:

$$e_1(s) = F_e v_0(s) - F_V V(s) \tag{13}$$

$$F_e = \frac{\tau h s^3 + h\ s^2 + (\lambda h s + \lambda_1)e^{-\Delta\ s}}{s(\tau h s^3 + h\ s^2 + ((1+\lambda h)s + \lambda + \lambda_1)e^{-\Delta\ s})} \tag{14}$$

$$F_V = \frac{(\lambda h s + \lambda_1)e^{-(\Delta+\Delta_c)s}}{s(\tau h s^3 + h\ s^2 + ((1+\lambda h)s + \lambda + \lambda_1)e^{-\Delta\ s})} \tag{15}$$

IV. STABILITY

A. String Stability of Longitudinal Control

The general string stability definition in the time domain is given in [15], in essence, it means all the states are bounded if the initial states (position, speed and acceleration errors) are bounded and summable.

In [12] we find a sufficient condition for string stability:

$$\|e_i\|_\infty \leq \|e_{i-1}\|_\infty \tag{16}$$

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:

$$G_i(s) = \frac{e_i(s)}{e_{i-1}(s)} \tag{17}$$

A sufficient condition for string stability in the frequency domain is derived:

$$\|G_i(s)\|_\infty \leq 1 \quad and \quad g_i(t) > 0 \quad i = 1, 2..N \tag{18}$$

where $g_i(t)$ is error propagation impulse response of the i-th vehicle.

We proved the stability of the platoon in two steps: firstly by finding stability conditions taking into account only parasitic sensing time delay and lags, Then we add the communications delays and we checked stability.

B. System Stability With Parasitic Time Delay and Lags:

We neglect communication delays. All the equation and the condition which will be found here will be also used when taking into account the communication delay.

Neglecting communication delays makes $G_V(s) = 0$ and we get:

$$e_i(s) = G_e(s)e_{i-1}(s) \tag{19}$$

In this case we can use (18) to check the stability so we have to verify that $\|G_e\| < 1$.

We have:

$$||G_e(\omega)|| = \sqrt{\frac{a}{a + \mu + \lambda_1^2 + 2\lambda\lambda_1}} \tag{20}$$

A sufficient condition to ensure the stability is $\mu \geq 0$. This gives a group of conditions that verify the stability of the platoon in presence of lags and sensor delays:

$$\left\{ \begin{array}{l} \lambda \leq \frac{h-2(\Delta+\tau)+2\lambda_1\tau\Delta}{2(h(\Delta+\tau)-\Delta\tau)} \quad \& \quad \frac{\lambda_1}{\lambda} < \frac{h}{2} \quad \& \\ \lambda \geq \frac{\lambda_1\tau-1}{h-\tau} \quad \& \quad h \geq 2(\Delta+\tau) \end{array} \right\} \tag{21}$$

The last condition is to ensure that $\lambda \geq 0$.

C. System Stability with Communication Delays:

Stability can be verified easily using condition (18) when the error is only a function of the previous error. When the errors become a function of additional variables we have to check the maximum limits of the state variables (spacing, speed and acceleration errors). The system is stable if state variables in the platoon are always bounded [15].

Using (9) we can get progressively the relation between $e_i(s)$ and $e_1(s)$:

$$e_i(s) = G_e^{i-1}\ e_1 + G_V\ e^{-i\Delta_c\ s} \frac{1-(G_e\ e^{-\Delta_c s})^{i-2}}{1-G_e\ e^{-\Delta_c s}}\ V(s) \tag{22}$$

So we have:

$$\|e_i\| \leq \underbrace{\|G_e\|^{i-1}\ \|e_1\|}_{\xi_1} + \underbrace{\|G_v\| \left\| \frac{1-(G_e\ e^{-\Delta_c s})^{i-2}}{1-G_e\ e^{-\Delta_c s}} \right\| \|V\|}_{\xi_2} \tag{23}$$

In the following we study the limits of spacing error of the vehicle i when $i \longrightarrow \infty$.

The first term ξ_1 is bounded ($\forall\omega$ and $i \longrightarrow \infty$) if $||G_e|| \leq 1$ and $||e_1||$ is bounded.

The conditions that keep $||G_e|| \leq 1$ are already given in (21).

From (13) we can prove that $||e_1||$ is also bounded because the norm of $||F_e||$ and $||F_V||$ converge toward zero for high frequencies. For low frequencies e_1 becomes equal to $\lambda h(V - v_i) + \lambda_1(X_V - x_0)$, this can be bounded if we choose V correctly. For all other frequencies, the nominator of $||F_e||$ and $||F_V||$ is always larger or equal to $\sqrt{\omega^2(\lambda+\lambda_1)^2}$ (we already proved that $\mu \geq 0$). This means that the nominators are larger than zero $\forall\omega \neq 0$; So $\exists\ \alpha_1 \leq \infty$ and $\alpha_2 \leq \infty :$ $||F_e|| \leq \alpha_1$ and $||F_V|| \leq \alpha_1\ \ \forall\omega \neq 0$. Then the first term always converge toward zero $\forall\omega$ when $i \longrightarrow \infty$.

For the second term ξ_2 we have $0 < ||1 - G_e\ e^{-\Delta_c s}|| \leq 2$, $0 < ||1 - (G_e\ e^{-\Delta_c s})^i|| \leq 2$. We can also prove that $||G_V||$ is always limited for all ω; so ξ_2 is limited for all frequencies even when $\omega = 0$ hence the platoon is stable for limited communications delays.

So we can conclude that the platoon is stable in presence of lags, sensing delays and even communication delays. The conditions of stability in presence of lags and sensing delays are given in (21). While the maximum acceptable communication delay $\Delta_{c_{max}}$ can be defined by safety conditions.

V. SIMULATIONS

Simulation has been done using Matlab. A large platoon, consisted of 60 vehicles, is created. In reality, the platoons are much more smaller, but we use this big platoon just to verify that the error is not increasing even for the vehicle with a big index ($i \longrightarrow \infty$). The desired inter-vehicle distance $L = 10\ m$. The leader accelerates from stationary state to reach a speed of 140 km/h and then it make emergency stop. We take parasitic sensing delay equal to $\Delta = 200\ ms$, the actuating lags equal $\tau = 200\ ms$ and a communication delay between each consecutive vehicles equal to $\Delta = 50\ ms$. We take $h = 2, \lambda = 0.7, \lambda_1 = 0.2$. To ensure safety, the maximum acceptable acceleration/deceleration to keep safety is $\mp 4.5\ m.s^{-2}$. For clarity, we only show one speed from each ten consecutive vehicles.

We can see in fig.4 that the platoon is stable because the errors are not increasing through the platoon. In addition, we can see that the spacings between vehicles are always larger than zero so the platoon is safe. Previously in [2] we chose $L = 5m$, we notice here that we have doubled the desired inter-vehicle distance to accommodate the errors generated from lags and delays. We tested the system with the worst cases (acceleration from zero to maximum speed with maximum acceleration and then we applied the emergency stopping) to verify the stability and safety in its limits. In practice we add additional safety distance in the desired distance to ensure more safety.

VI. DISCUSSION

- The new modification improves the performance of our control law, without requiring new data from other vehicle. Each car can compute the current position of the truck using the shared speed V. So X_V is always the same for all vehicles.
- In case of communication loss, all vehicles switch to autonomous stable mode by making $V \to 0$ and $X_v \to x_i$ (for the i-th vehicle). This enable the vehicles to switch to classical time headway policy.

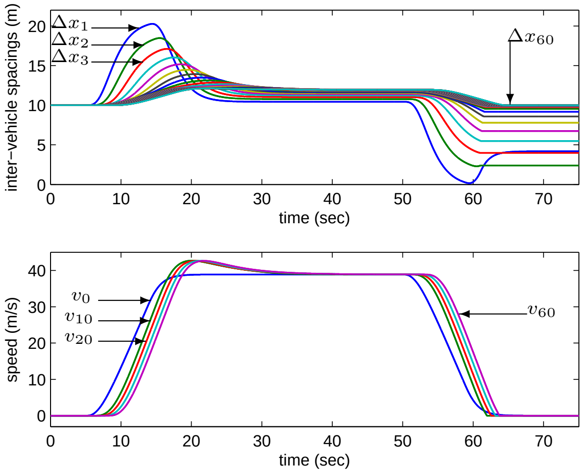


Fig. 4. Inter-vehicle spacings in presence of lags, sensing and communication delays

- No need to transmit the value of X_V because it is computed in each vehicle by integrating V. Losing the communication will prevent the vehicles to have the same value for X_V; so it is necessary to update X_V for all the vehicles after each communication loss.
- The error of integration will have no effect on the stability because this error will be the same for all the vehicles. But the most important condition is to keep $X_v - x_0$ limited.

  We can see that this control law with our control law proposed in our previous works [2] and the classical time headway represent integrated frame work for controlling the platoon with decreasing communication rate respectively and we can switch from one law to another smoothly in case of communication difficulties.

VII. Conclusion

In this paper we have addressed the control of platoons on highways. The longitudinal dynamics is modeled using modified flatbed tow truck model. We proved the robustness of this control law to lags, parasitic delays and even for communication delays. Sufficient stability conditions was given in (21). In the future work, passenger comfort and the safety of the platoon will be studied.

References

[1] Ali, A.; Garcia, G. and Martinet, P., The flatbed platoon towing model for safe and dense platooning on highways, IEEE Intelligent Transportation Magazine, to be published.

[2] Ali, A.; Garcia, G. and Martinet, P., Minimizing the inter-vehicle distances of the time headway policy for platoons control in highways, 10th International Conference on Informatics in Control, Automation and Robotics (ICINCO13), pp. 417-424. SciTePress, Reykjavik, Iceland, July 29-31, 2013.

[3] Ali, A.; Garcia, G. and Martinet, P., Minimizing the inter-vehicle distances of the time headway policy for urban platoon control with decoupled longitudinal and lateral control, 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC), pp. 1805-1810, The Hague, The Netherlands, 6-9 Oct. 2013.

[4] Ali, A.; Garcia, G. and Martinet P.,Urban platooning using a flatbed tow truck model, the 53rd IEEE Conference on Decision and Control CDC, submitted for publication.

[5] Ali, A.; Garcia, G. and Martinet, P., Safe platooning in the event of communication loss using flatbed tow truck model, the 13th International Conference on Control, Automation, Robotics and Vision, ICARCV 2014, submitted for publication.

[6] Ricardo (2009), *Cars that drive themselves can become reality within ten years*, [Online]. Available: http://www.ricardo.com/en-GB/News–Media/Press-releases/News-releases1/2009/Cars-that-drive-themselves-can-become-reality-within-ten-years/

[7] H. Fritz, A. Gern, H. Schiemenz and C. Bonnet, *CHAUFFEUR Assistant: a driver assistance system for commercial vehicles based on fusion of advanced ACC and lane keeping*, IEEE conf. Intell. Vehicles Symposium, pp. 495-500, 14-17, Jun. 2004.

[8] Hedrick, J. K.; Chen, Y.; and Mahal, S., Optimized Vehicle Control/Communication Interaction in an Automated Highway System, Institute of Transportation Studies, Research Reports, Working Papers, Institute of Transportation Studies, UC Berkeley.2001

[9] Lingyun, Xiao; and Feng, Gao, Practical String Stability of Platoon of Adaptive Cruise Control Vehicles, IEEE Transactions on Intelligent Transportation Systems, vol.12, no.4, pp.1184,1194, Dec. 2011

[10] Ling-yun, Xiao; and Feng, Gao, Effect of information delay on string stability of platoon of automated vehicles under typical information frameworks,Journal of Central South University of Technology, Vol.17, no.6, pp 1271-1278, Dec.2010

[11] Lingyun, Xiao; Darbha, S.; and Feng, Gao, Stability of String of Adaptive Cruise Control Vehicles with Parasitic Delays and Lags, 11th International IEEE Conference on Intelligent Transportation Systems ITSC 2008, vol., no., pp.1101,1106, 12-15 Oct. 2008

[12] Rajamani, R., Vehicle dynamics and control, Springer science, ISBN 0387263969, 2006.

[13] Rajamani, R.; and Shladover S., An experimental comparative study of autonomous and co-operative vehicle-follower control systems, Transp. Res. Part C, vol. 9, no. 1, pp. 15–31, Feb. 2001.

[14] Ricardo, http://www.ricardo.com/en-GB/News–Media/Press-releases/News-releases1/2009/Cars-that-drive-themselves-can-become-reality-within-ten-years/, 2009.

[15] 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.

[16] Swaroop, D.; and Rajagopal, K., A review of constant time headway policy for automatic vehicle following. In Proceedings IEEE Intelligent Transportation Systems, pp. 65-69, 2001.

[17] Teo, R.; Stipanovic, D.M.; and Tomlin, C.J., Decentralized Spacing Control of a String of Multiple Vehicles Over Lossy Datalinks,, IEEE Transactions on Control Systems Technology, vol.18, no.2, pp.469,473, March 2010.

[18] Xiangheng, Liu; Goldsmith, A.; and Mahal, S.S.; Hedrick, J.K., ”Effects of communication delay on string stability in vehicle platoons,” Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE , vol., no., pp.625,630, 2001

[19] Yanakiev, D.; and Kanellakopoulos, Ioannis, Longitudinal control of automated CHVs with significant actuator delays, IEEE Transactions on Vehicular Technology, vol.50, no.5, pp.1289,1297, Sep 2001

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Global Robot Ego-localization Combining Image Retrieval and HMM-based Filtering

Cédric Le Barz[1], Nicolas Thome[2], Matthieu Cord[2], Stéphane Herbin[3] and Martial Sanfourche[3]

***Abstract*— This paper addresses the problem of global visual ego-localization of a robot equipped with a monocular camera that has to navigate autonomously in an urban environment. The robot has access to a database of geo-referenced images of its environment and to the outputs of an odometric system (Inertial Measurement Unit or visual odometry). We suppose that no GPS information is available. The goal of the approach described and evaluated in this paper is to exploit a Hidden Markov Model (HMM) to combine the localization estimates provided by the odometric system and the visual similarities between acquired images and the geo-localized image database. It is shown that the use of spatial and temporal constraints reduces the mean localization error from 16 m to 4 m over a 11 km path evaluated on the Google Pittsburgh dataset when compared to an image based method alone.**

I. Introduction

The problem tackled in this paper is the visual autonomous navigation of a robot operating in an urban environment [1]. A typical target application could be the delivery of goods using unmanned ground or aerial vehicles where the robot trajectory has been defined before hand on a given map, and must be followed to reach its final destination (Fig. 1). Absolute localization system like GPS may be shadowed or completely unavailable in several areas of the trajectory and substitute localization means must be used.

Visual information is an appealing alternative because cameras and densely sampled geo-referenced images are now commonly available. Nevertheless, the localization of a robot exploiting only image content is challenging because two images of the same place acquired at different times and with different cameras may show huge appearance differences due to illumination and colorimetry variations (e.g. sunny or cloudy days), camera viewpoints changes, scene modifications (e.g. seasonal changes, building construction) and occlusions (e.g. by cars) (Fig. 2). Standard image retrieval (IR) methods such as k Nearest Neighbour (kNN) votes or Bag Of Visual Words (BoVW) [2] produce noisy results that necessitate filtering to be robustly exploited as primary global localization information source.

Odometric systems, IMU based or visually based, provide localization information at low cost: however this information is only relative to a given position and suffers from drift especially on complex trajectories. It can only be used reliably on small portions of the followed route and can't be the only source of measurement for absolute localization.

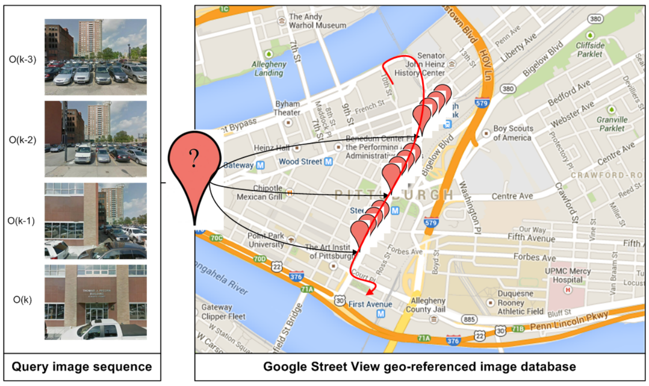


Fig. 1. Visual ego-localization system: Our system aims at matching a sequence of images with geo-referenced database images in order to determine accurate geo-localization from noisy odometric information.

The main contribution of this paper is to describe a general framework enabling to combine these two sources of noisy localization information: local odometry and visual similarity. More precisely, the solution we propose uses an IR algorithm applied to a database of geo-referenced images integrated into a Hidden Markov Model (HMM) accounting for odometry uncertainty. The role of the HMM is to exploit spatio-temporal constraints in order to filter out erroneous IR results.

The effectiveness of our approach has been evaluated over a 11 km path using two kinds of images: Google Streetview images [4] simulating images acquired online by the robot camera and Google Pittsburgh image dataset [3] as geo-referenced image database.

II. Related work

Visual place recognition problems have been addressed recently thanks to the availability of image databases. Most of them rely on the extraction of 2D and/or 3D features, that are compared to a geo-referenced feature database. Unlike [5] [6] that are loop closure algorithms developed for Simultaneous Localization and Mapping (SLAM) systems, we focus on position tracking.

Zamir et al. propose in [7] a hierarchical method to localize a group of images. SIFT descriptors from database images are indexed using a tree. A nearest neighbour tree search is then done for each SIFT query image feature. Weak votes are removed and each reliable feature votes for

[1] Cédric Le Barz is with Theresis department, THALES company, 91767 Palaiseau, France cedric.lebarz@thalesgroup.fr

[2] Nicolas Thome and Matthieu Cord are Sorbonne University, UPMC University, Paris 06, UMR 7606, LIP6, 75005 Paris, France nicolas.thome@lip6.fr, matthieu.cord@lip6.fr

[3] Stéphane Herbin and Martial Sanfourche are with the French Aerospace Lab, ONERA, 91123 Palaiseau, France stephane.herbin@onera.fr, martial.sanfourche@onera.fr

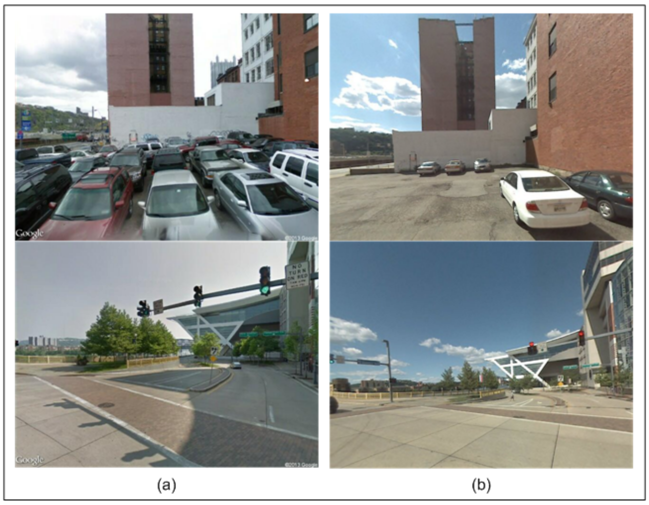


Fig. 2. Google Streetview images (a) and Robot images (b). Note the impact of different focal lenses, weather conditions, viewpoint changes and the presence/absence of cars in the scene.

a location. All accumulated spatial votes are then filtered by a Gaussian kernel. The geo-referenced image with the highest number of votes determines the location. In [8], the method described in [7] is improved by interpreting the 2D map votes as a likelihood. This likelihood is then used in a Bayesian tracking filter to estimate the temporal evolution based on the previous state. Both solutions are dedicated to web video annotation, and localization is not realized on the fly which makes it useless for navigation.

In [9], the vehicle localization algorithm uses simple visual features and 3D features. The solution requires in a preliminary phase to build a compact map described as a graph. Nodes include vehicle position at fixed distance interval and visual and 3D features. At runtime, a Bayesian filter is used to estimate the probability of the vehicle position by matching features extracted from sensors with database features. Their solution uses two lateral cameras and two lateral LIDARs. Same sensors are used during the map building step and the localization step. In contrast, our solution is monocular and uses different cameras for acquisition and reference database.

The solution proposed in [10] is based on the match of visual odometric measurements with a 2D road-map. The map is represented by a directed graph and a probabilistic approach is defined in order to navigate within this graph. They are able to localize themselves after a few seconds of driving with an accuracy of 3 m on a 18 km^2 map containing 2150 km of roads. Our navigation solution does not use any 2D road-map. It uses only visual features from images along the specified trajectory combined with odometric information.

In [11], the localization is achieved by recognizing temporal coherent sequences of local best matches. These local best matches are based on a Sum of Absolute Difference (SAD) on resolution-reduced and patch-normalized images between last acquire image and M previous images. They make the assumption that the robot velocity is constant between all image sub-sequences. The proposed solution is robust to extreme perceptual changes, but sensitive to point of view.

In [12], authors work on visual similarity for UAV ego-localization. They propose to generate artificial views of the scene in order to overcome the large view-point differences. Nevertheless, spatio-temporal constraint is not taken into account.

Another type of approach is to cast the problem as a classification task, as in [13]. A classifier for each image in the database is trained using per-exemplar SVM approach. The main contribution of the mentioned paper is the calibration of all SVM classifiers using mainly negative examples in order to be able to compare all classifiers scores.

As in [7] [8] [9] [11], our solution uses spatio-temporal coherency. Along with this, our solution uses a HMM enabling to take into account in a more flexible way robot dynamics. No assumption is done concerning the constant velocity of the robot, but as in [14] we consider coarse position estimates provided by an odometric sensor and their uncertainties. Furthermore, in contrast to [7] [8] [9], the latest part of the trajectory is re-estimated for each new acquisition.

III. Proposed solution

Preliminary experiments made clear that IR approaches are not selective enough for urban areas because the same features tend to be shared by several neighbour images and produce erroneous matches (Fig. 4). That is why we propose to exploit the spatio-temporal coherency in order to filter out the wrong matches provided by standard IR algorithms. This is achieved by combining the similarities supplied by an IR algorithm with a HMM where hidden states represent places. The idea is to find the trajectory that best explains the M past observations and therefore the current position. The definition of a HMM for each new image acquired by the robot will enable to re-estimate the latest part of the trajectory so that past errors are corrected on a long term basis. Furthermore, taking into account odometric information reduces online the number of database images used in the IR task.

A. General principle

At each time t the robot acquires an image O_k and receives an estimate of its current position $\tilde{S}_k$ from the odometric system. The goal of the global localization algorithm is to produce a better estimate $\hat{S}_k$ of the current robot position from the past observations and odometric estimate (Fig. 3). The estimator is a function of the M past observations $\mathbf{O}_k = \{O_{k-M+1}, \dots O_k\}$ (i.e. the current location estimate exploits a set of observations in a sliding window based approach of length M) and the estimated position $\tilde{S}_k$.

Estimation is realized in a classical random variable setting where the robot location at time t is considered as a random variable q_t taking values in a discrete set of possible location S_j $j \in \{1 \dots N\}$. The main modeling hypothesis is that its random behaviour is represented by a HMM.

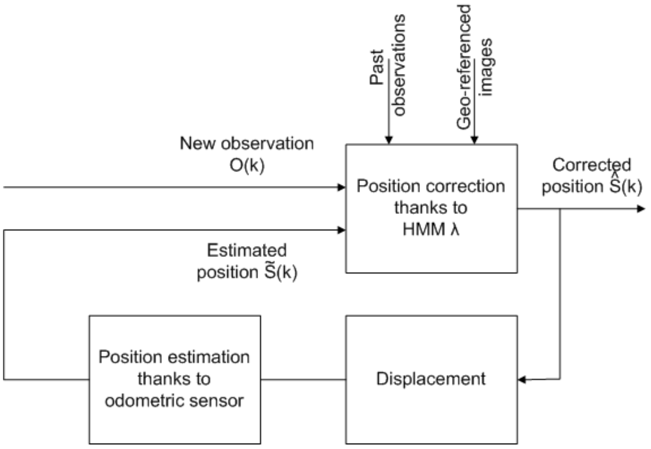


Fig. 3. System overview: for each new observation O_k, an odometric sensor provides a rough position estimate $\tilde{S}_k$ that is corrected thanks to a new HMM λ combining visual information and spatio-temporal constraints. This corrected position is noted $\hat{S}_k$.

Using the classical notations of [18], the use of a HMM requires the definition of the adequate model $\lambda = \{N, M, \Pi, A, B\}$ where N is the number of states, M is the number of observations, Π is the prior on the initial state, A is the transition probability matrix between the states and B is the observation probability matrix given some states.

The HMM approach provides a standard way to estimate the most likely state sequence $\hat{\mathbf{S}}_k$, i.e the M successive places, explaining the sequence of observations $\mathbf{O}_k = \{O_{k-M+1}, ..., O_{k-1}, O_k\}$ (Viterbi algorithm):

$$\hat{\mathbf{S}}_k = \arg\max_{\mathbf{S}} P(\mathbf{S}|\mathbf{O}_k, \lambda) \qquad (1)$$

The question is now to design the HMM adapted to the global estimate of the robot location. This will be detailed in two steps: construction of the state transition matrix A and initial state vector, and computation of the conditional observation matrix B.

B. State transition matrix and initial state vector

The state transition matrix A and initial state vector are built from knowledge of the odometric system behaviour, robot kinematics and quality of the available database of geo-referenced images.

From the robot kinematics, images are approximately acquired every D meters with an odometric uncertainty of Δ meters. The image database consists of overlapping images acquired every $D^{'}$ meters with $D^{'} \leq D$. In this setting, the database is therefore assumed to have a bigger sampling rate than the online image acquisition rate.

Each possible state location S_j is uniquely defined by a geo-referenced database image I_j.

The filtering capacity of the HMM depends on the number M of past observations. This control parameter is free and its influence will be studied in the experiments.

One critical parameter is the localization uncertainty U which defines the area where the robot is supposed to be. This localization uncertainty can be for example the initial position uncertainty when the robot starts its mission.

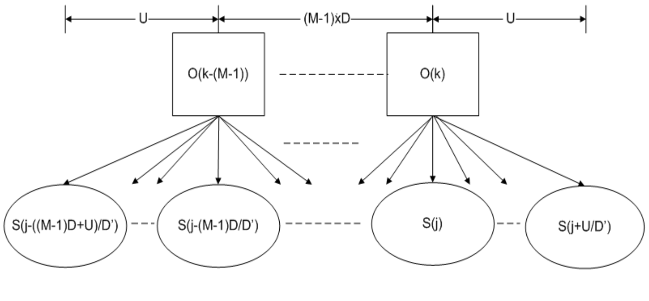


Fig. 5. Relation between states to consider and localization uncertainty.

The number of states N, i.e. the number of potentially matching images in the database, the initial state probability Π and the state transition probability matrix A depend on U, D, Δ and M. They are defined the following way:

- N : Given the putative position of the robot $\tilde{S}_k = S_j$, the localization uncertainty U, the approximative displacement D, and the observation number M, the potential states, i.e. the set of database images considered for matching is defined according to the schema on Fig. 5.
- Π: $\Pi = \{\pi_j\}_{j=1}^{j=N}$ where $\pi_j = P[q_1 = S_j]$. It depends on initial position estimate (i.e. estimated position by previous HMM) and localization uncertainty U. We use uniform uncertainty on interval of size $F = 1 + 2 \cdot \lceil U/D^{'} \rceil$.
- A: $A = \{a_{ij}\}$ where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1 \leq i, j \leq N$: To take into account odometric uncertainty for a displacement D, we defined A as $a_{ij} = \frac{D^{'}}{\Delta} rect_{\Delta/D^{'}}(j - i - (D/D^{'}))$. [1]

C. Observation matrix

The observation matrix B is computed from visual similarity between the M observations and the set of potentially matching database images as shown in Fig 5.

Visual similarity measurement is based on a state of the art IR solution. During the navigation phase, SIFT descriptors for all interest points detected by a SIFT detector [15] are extracted in a similar way as during the off-line phase. A kNN voting algorithm is then performed: 1) For each descriptor of a query image the k nearest neighbours are found from a subset of database descriptors, i.e. those that are near to the putative robot position. This subset is determined thanks to the estimated robot position $\tilde{S}_k$, D, U, Δ and M. 2) As noisy interest points are usually detected in an image, a filtering process based on the ratio of the distance between the query descriptor and the first and second nearest neighbours is used [15]. 3) Query descriptors that match with multiple database descriptors are removed, and finally 4) Outliers are rejected through a geometric verification, i.e. a

[1] The function $rect_\alpha(x)$ is the rectangular function defined by $rect_\alpha(x) = 1$ if $|x| \leq \alpha$ and else $rect_\alpha(x) = 0$.

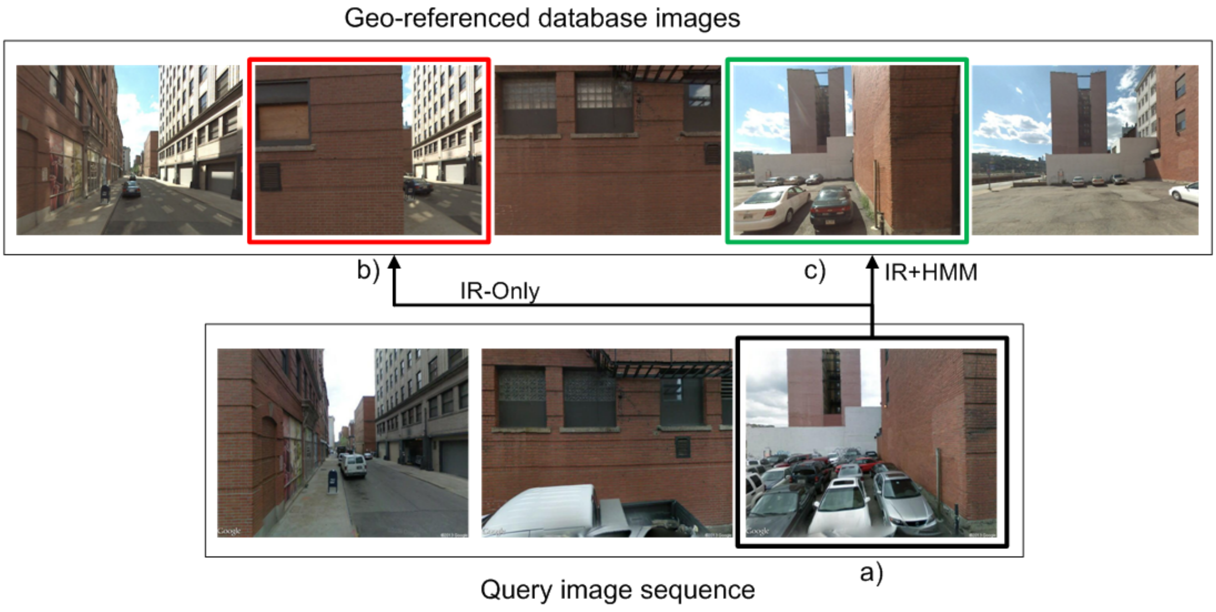


Fig. 4. Top images: Sequential database images considered for IR task - Bottom images: Sequential query images acquired by the robot - (a) Latest query image, (b) image returned by an IR algorithm only and (c) by an IR algorithm combined with a HMM.

RANSAC [16] 4-point algorithm [17] (homography). Hence, we get the number of descriptor correspondences between the descriptors of the query image O_k acquired by the robot and the descriptors of the database image I_j. This is our similarity measure, noted $f(O_k, I_j)$.

The observation matrix $B = \{b_j(k)\}$, where $b_j(k) = P[O_k \ at \ t|q_t = S_j]$, $1 \leq j \leq N$ and $1 \leq k \leq M$ is the probability of observing O_k when location is S_j. We propose to compute this probability from the similarity measure using the following formula:

$$b_j(k) = P[O_k \ at \ t|q_t = S_j] = \frac{\alpha}{1 + exp(a \cdot (f(O_k, I_j) + b))} \tag{2}$$

where a and b are two constants, $f(O_k, I_j)$ is the visual similarity measure and α is a normalization constant to impose $\sum_{j=1}^{j=N} b_j(k) = 1$.

A summary of the general estimation scheme is presented in algorithm 1.

Given $\lambda = \{N, M, \Pi, A, B\}$, (3) can be solved.

$$\begin{aligned} \hat{\mathbf{S}}_k &= \arg\max_{\mathbf{S}} P(\mathbf{O}_k|\mathbf{S}, \lambda) \cdot P(\mathbf{S}, \lambda) \\ &= \arg\max_{\mathbf{S}} \left(\prod_{k=1}^{k=M} P(O_k|\mathbf{S}, \lambda) \right) \cdot \left(\pi_1 \cdot \prod_{k=2}^{k=M} a_{k-1,k} \right) \end{aligned} \tag{3}$$

The first term of (3) refers to visual similarities between observations and the image database, whereas the second term refers to the dynamics of the robot and models spatio-temporal constraints. We study in section IV the achieved performances by mixing these two complementary aspects.

IV. Experimental results

To evaluate our solution in a realistic situation, we conducted our experiments on a 11 km trajectory. The dataset used has been acquired at different times (more than one year between acquisitions) and with different camera fields of view resulting in visual changes for the same scenes (Fig. 2).

```
Algorithm 1: Vision based global localization from odometric estimates
  Input: Estimated robot position $\tilde{S}_k$, Localization
         uncertainty $U$, Estimated displacement $D$ with
         odometric uncertainty $\Delta$, $M$ last past
         observations, Geo-referenced image features
         database.
  Output: Corrected robot position $\hat{S}_k$.
1 HMM initialization (A and $\Pi$) from $D$, $U$, $\Delta$ and $M$ as
  explained in section III-B;
2 Select relevant database images from estimated position
  $\tilde{S}_k$, $D$, $U$, $\Delta$, and $M$ (Fig. 5);
3 Compute similarities between the $M$ past observations
  and relevant database images as explained in
  section III-C;
4 Compute B from similarities with (2);
5 Apply Viterbi algorithm to solve (3) to estimate the
  latest state $\hat{S}_k$;
```

A. Image datasets and settings

We performed experiments on the Google Pittsburgh dataset as image database [3], and Google Streetview images as query images [4]. Pittsburgh dataset images have been resized to 640x480, so that their resolutions match the query image resolution. About 1160 SIFT descriptors are extracted and stored per image. From the original corpus, we keep one image every $D^{'} = 5$ m and remove non-informative images (e.g. images acquired in tunnels) resulting in a corpus of 2215 images. Query images are downloaded from the Internet via a HTTP request with the following settings: a resolution of 640x480, a field of view of 100^o and a camera tilt of 5^o. We requested one image every D = 10 m resulting in 1105 query images. For (2), a was set to 1 and b was set to -4.

B. Results

First, we compared our method (noted IR-HMM) to a state of the art IR algorithm based on visual similarities only (noted IR-Only). The meaningful metrics used are mean localization error and recall rate[2]. The recall rate increases from 36% to 84%, and the mean error localization decreases from to 16 m to 4 m (Tab. I). This considerable improvement confirms that exploiting the spatio-temporal constraint is essential. Our solution corrects ambiguous image matches (Fig. 4) thanks to spatio-temporal constraints imposed via the A matrix.

TABLE I

MEAN ERROR DISTANCE AND RECALL RATE FOR AN IR ALGORITHM BASED ON VISUAL SIMILARITIES ONLY (IR-ONLY), FOR AN IR ALGORITHM FOLLOWED BY A SPATIAL-TEMPORAL FILTER (IR-ST) AND FOR THE SAME IR ALGORITHM COMBINED WITH A HMM (IR-HMM) ON PITTSBURGH DATASET FOR M = 15 AND U = 50 M.

	IR-Only	IR-ST	IR-HMM
Mean error distance	15.8m	7.7m	3.9m
Recall	36.1%	71.2%	84.0%

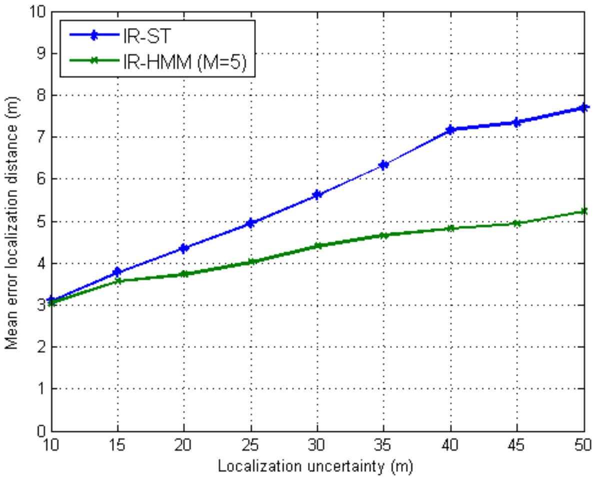


Fig. 6. Mean localization error distance vs. Localization uncertainty U. Spatio-temporal constraints reduce significantly false match that may appear with an IR algorithm, improving the mean error localization distance.

Then, we compared our solution with a method similar to the one described in [7] and reminded in section II: a Gaussian spatial filter is applied on putative positions obtained by query descriptor votes. Like ours, this method (noted IR-ST) takes into account spatio-temporal information. We noticed that for a localization uncertainty of 50 m, the use of the HMM enables to decrease the mean localization error from 8 m to 4 m (Fig. 6). Furthermore, when localization uncertainty increases, performance differences between both solutions increase. The trajectory estimate with a HMM is more precise than with a spatio-temporal filter that tends to smooth the trajectory. HMM removes impossible matches, whereas in a spatio-temporal filter false matches are used for position estimates.

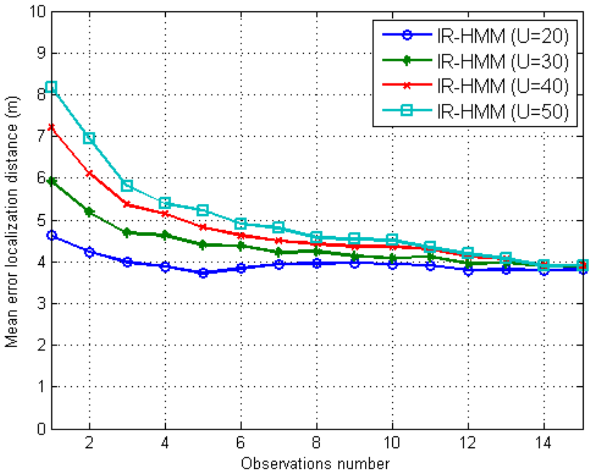


Fig. 7. Mean localization error distance vs. observation number M. When localization uncertainty increases, M must be also increased to guarantee a given mean localization error.

Finally, we studied the sensitivity of our solution to the number of past observations M used, according the localization uncertainty U (Fig. 7). The higher U, the more observations number have to be considered to keep the mean error localization under a threshold. As M approaches 0, only dynamics included in the A matrix is significant (3). In this case, our definition of A (possible transitions have equal probabilities) and Π (possible initial states have equal probabilities) implies equal probabilities for different states (3). The random selection performed among possible states explains the mean error localization increase.

Therefore, using dynamics only, or using visual similarity only, are insufficient in our context. Combining both improves significantly results.

V. CONCLUSION

We have proposed a general approach for global ego-localization able to combine noisy location estimates provided by an odometric system and visual place recognition. No GPS information is used. The solution exploits a Hidden Markov Model whose structure is adaptively defined from knowledge of the odometric system behaviour. Each new image acquisition by the robot allows a complete re-estimate of the M past observation locations ensuring odometric error correction on a long term basis.

The approach has been evaluated on the Pittsburgh Google dataset. We demonstrated the benefits of combining simple visual similarities and dynamics modelling: the proposed

[2] True positive images are defined as geo-referenced images whose distance with ground truth image is less than 5 m.

solution improves significantly the mean error localization which decreases from 16 m to 4 m for a localization uncertainty of 50 m.

Improved image retrieval solutions can be easily integrated in the system without substantial structural modifications: this is the avenue of future work.

ACKNOWLEDGMENT

This work results from a collaboration between UPMC University, Onera and Thales Services SAS.

REFERENCES

[1] J. Ibañez-Guzmán, C. Laugier, J. Yoder, and S. Thrun, "Autonomous driving: Context and state-of-the-art," in *Handbook of Intelligent Vehicles*, ser. Springer Reference, A. Eskandarian, Ed. Springer, Mar 2012, pp. 1271–1310.

[2] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, vol. 2. IEEE, 2003, pp. 1470–1477.

[3] (2011) Pittsburgh dataset website (provided by google for research purposes). [Online]. Available: http://www.icmla-conference.org/icmla11/challenge.html

[4] (2014) Google street view API website. [Online]. Available: http://developers.google.com/maps/documentation/streetview

[5] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *International Journal of Robotics Research*, vol. 30, pp. 1100–1123, August 2011.

[6] W. Maddern, M. Milford, and G. Wyeth, "Cat-slam: probabilistic localisation and mapping using continuous appearance-based trajectory," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 429–451, April 2012.

[7] A. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *Proceedings of the European Conference on Computer Vision*. IEEE, 2010, pp. 255–268.

[8] G. Vaca-Castano, A. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *Proceedings of the Computer Vision and Pattern Recognition conference*. IEEE, 2012, pp. 1186–1193.

[9] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2012, pp. 1635–1642.

[10] M. Brubacker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *Proceedings of the Computer Vision and Pattern Recognition conference*. IEEE, 2013, pp. 3057–3064.

[11] M. Milford and G. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[12] A. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "Mav urban localization from google street view data," in *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3979–3986.

[13] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proceedings of the Computer Vision and Pattern Recognition conference*. IEEE, 2013, pp. 907–914.

[14] J. Zhang, A. Hallquist, E. Liang, and A. Zakhor, "Location-based image retrieval for urban environments," in *Proceedings of the International Conference on Image Processing*. IEEE, 2011, pp. 3677–3680.

[15] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, November 2004.

[16] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[17] A. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. New York, NY, USA: Cambridge University Press, ISBN 978-0-521-54051-3, 2006.

[18] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, 1989, pp. 257–286.

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems

Panel Session

Towards driverless vehicles

Christian Laugier (INRIA, Grenoble, France)

***Abstract*:** The purpose of this panel session is to discuss the hot topics of "Advanced Driving Assistance Systems" and of "Driverless Vehicles". All the related Technical, Socio-economic and Legal issues were addressed and discussed.

Participants:

- **Seiichi Mita** (Toyota Technological Institute Nagoya, Japan)
- **Fawzi Nashashibi** (INRIA, France)
- **Christoph Stiller** (KIT, Germany)

IROS'14
PPNIV'14
6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles

2014 IEEE/RSJ International Conference on Intelligent Robots and Systems