



IROS11 International workshop on Perception and Navigation for Autonomous Vehicles in Human Environment

**Full Day Workshop
September 30th, 2011 San Francisco, California, USA**

<http://www.lasmea.univ-bpclermont.fr/Control/WorkIROS11/PNAVHE11.html>

Organizers

**Christian Laugier (INRIA, France),
Philippe Martinet (IRCCYN and LASMEA, France),
Urbano Nunes (ISR, Portugal)**

Contact

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



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Foreword

Autonomous driving and navigation is a major research issue which would affect our lives in near future. 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. These new technologies can be applied efficiently for other application fields such as unmanned vehicles, mobile service robots, or mobile devices for motion assistance to elderly or disabled peoples. Technologies related to this area, such as autonomous outdoor vehicles, achievements, challenges and open questions would be presented, including the following topics: 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, Driver assistance systems, Collision prediction and avoidance, Environment perception, vehicle localization and autonomous navigation, Real-time perception and sensor fusion, SLAM in dynamic environments, Real-time motion planning in dynamic environments, 3D Modelling and reconstruction, Human-Robot Interaction, Behavior modeling and learning, Robust sensor-based 3D reconstruction, Modeling and Control of mobile robot, Multi-agent based architectures, Cooperative unmanned vehicles (not restricted to ground transportation), Multi autonomous vehicles studies, models, techniques and simulations

Previously, six 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 [SNODE'07](#) in San Diego during IROS'07 (around 80 attendees), the third [PPNIV'08](#) in Nice during IROS'08 (more than 90 registered people), the fourth edition [SNODE'09](#) in Kobe during ICRA'09 (around 70 attendees), the fifth edition [PPNIV'09](#) during IROS'09 in Saint-Louis (around 70 attendees), and the last one [RITS'10](#) was organized in the last ICRA'10 in Anchorage (around 35 attendees). A special issue in IEEE Transaction on ITS, mainly focused on Car and ITS applications, has been published in September 2009.

This workshop is composed with 4 invited talks and 13 selected papers (8 selected for oral presentation and 5 selected for interactive session. Five sessions have been organized:

- Session I: Path Planning & Navigation Systems
- Session II: Perception & Situation Awareness
- Session III: Interactive session
- Session IV: 2D and 3D Mapping & Localization
- Session V: Mobile robot modeling and control

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 and Urbano Nunes



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Session I

Path Planning & Navigation systems

- **Keynote speaker: Rafael Toledo-Moreo (Technical University of Cartagena, Spain)**
Title: Why can't road positioning and integrity be friends?
- **Title: Autonomous Navigation in Crowded Campus Environments**
Authors: Z. J. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, E. S. Rankin, M. H. Ang Jr., E. Frazzoli, D. Rus, D. Hsu, K. H. Low
- **Title: Integration of visual and depth information for vehicle detection**
Authors: A. Makris, M. Perrollaz, I. Paromtchik, C. Laugier



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Session I

Keynote speaker: **Rafael Toledo-Moreo**
(**Technical University of Cartagena, Spain**)

Why can't road positioning and integrity be friends?

Abstract : Today's positioning systems work quite well in many situations. However, they lack some robustness, what hinders its exploitation for safety-critical and liability-critical applications. The warranty of the quality of the positioning service would open the way for further services, but for the community of the field it is unclear whether or not integrity is achievable for road positioning. There are several good reasons for that hesitation: The representation of the integrity of such a complex system by means of projecting all its possible errors onto an integrity parameter is difficult; the use of assumptions may cause that a non-modeled event disrupts the consistency of the error estimates; and also, the different ways in which the concept of integrity is approached in the literature lead to confusion and contradictions. This talk focuses on road positioning and its integrity, discussing aspects that play a role in this problem such as Global Navigation Satellite Systems, aiding sensors, the vehicle environment and its model, data fusion methods and map-matching algorithms.

Biography:

Rafael Toledo-Moreo received the M.S. degree in automation and electronics engineering from the Technical University of Cartagena (UPCT), Cartagena, Spain, in 2002 and the Ph.D. degree in computer science from the University of Murcia (UMU), Murcia, Spain, in 2006. He is a Professor with the Department of Electronics, Computer Technology, and Projects, UPCT. He is also a Research Member of the Intelligent Systems and Telematics Group, UMU. His main field of interest is road navigation systems. Dr. Toledo-Moreo is a member of the International Federation of Automatic Control Technical Committee on Transportation Systems and the IEEE Robotics and Automation Society Technical Committee for Intelligent Transportation Systems. He is an Associate Editor and member of the International Program Committees of several journals and conferences related to Intelligent Transportation Systems.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

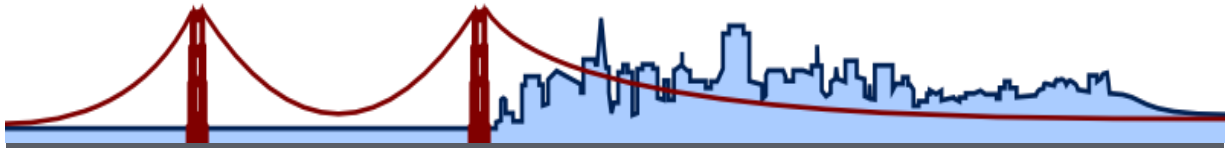


Session I

Path Planning & Navigation systems

Related paper for the invited speaker

- **Title:** Lane-Level Integrity Provision for Navigation and Map Matching With GNSS, Dead Reckoning, and Enhanced Maps,
Authors: Rafael Toledo-Moreo, David Bétaille, François Peyret
Published in : IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL.11, NO.1, MARCH2010
Abstract: Lane-level positioning and map matching are some of the biggest challenges for navigation systems. Additionally, in safety applications or in those with critical performance requirements (such as satellite-based electronic fee collection), integrity becomes a keyword for the navigation community. In this scenario, it is clear that a navigation system that can operate at the lane level while providing integrity parameters that are capable of monitoring the quality of the solution can bring important benefit to these applications. This paper presents a pioneering novel solution to the problem of combined positioning and map matching with integrity provision at the lane level. The system under consideration hybridizes measurements from a Global Navigation Satellite System (GNSS) receiver, an odometer, and a gyroscope, along with the road information stored in enhanced digital maps, by means of a multiple-hypothesis particle-filter-based algorithm. A set of experiments in real environments in France and Germany shows the very good results obtained in terms of positioning, map matching, and integrity consistency, proving the feasibility of our proposal.
Website : <http://wsdetcp.upct.es/Personal/rToledo/home.htm>



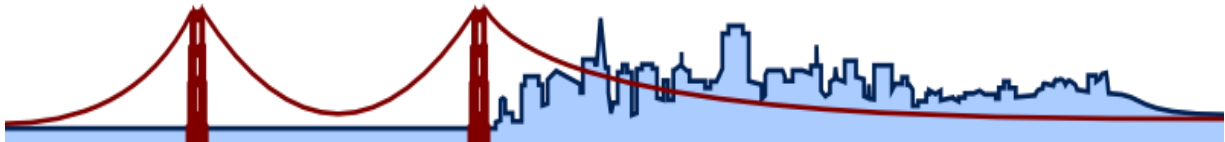
**2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011**



Session I

Path Planning & Navigation systems

- **Title:** Autonomous Navigation in Crowded Campus Environments
Authors: Z. J. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, E. S. Rankin, M. H. Ang Jr., E. Frazzoli, D. Rus, D. Hsu, K. H. Low
- **Title:** Integration of visual and depth information for vehicle detection
Authors: A. Makris, M. Perrollaz, I. Paromtchik, C. Laugier



**2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011**

Autonomous Navigation in Crowded Campus Environments

Z. J. Chong^{*}, B. Qin^{*}, T. Bandyopadhyay[‡], T. Wongpiromsarn[‡], E. S. Rankin[¶],
M. H. Ang Jr.^{*}, E. Frazzoli^{§‡}, D. Rus^{§‡}, D. Hsu^{*}, K. H. Low^{*}

^{*} National University of
Singapore,
Kent Ridge, Singapore

[‡] Singapore-MIT Alliance for
Research and Technology,
Singapore

[§] Massachusetts Institute
of Technology,
Cambridge, MA., USA.

[¶] DSO National
Laboratories,
Singapore

Abstract—This paper considers autonomous navigation in crowded city environments. An autonomous vehicle testbed is presented. We address two challenges of pedestrian detection and GPS-based localization in the presence of high-level buildings. First, we augment the localization using local laser maps and show improved results. A pedestrian detection algorithm using a complementary vision and laser system is proposed. We implement this algorithm in our testbed and evaluate its performance using purely off the shelf components and open source software packages provided by ROS. We also show how utilizing existing infrastructural sensors can improve the performance of the system. Potential applications of this work include fully automated vehicle systems in urban environments typical in megacities in Asia.

I. INTRODUCTION

One of the long-standing research activities in robotics has been towards increasing the level of autonomy in both manned and unmanned systems. In transportation systems, automation has been employed, for example, in traffic light management and congestion avoidance services and has attracted numerous research interests in the transportation science and logistics community. Intelligent vehicle/highway systems (IVHS) technologies have also been developed to enhance operational safety and efficiency [14].

In this paper, we focus on putting more autonomy into the transport vehicles to deal with the problem at a more local level. Interests in autonomous driving, especially in urban environments, were largely stimulated by the launch of the 2007 DARPA Urban Challenge (DUC). In this competition, the autonomous vehicles have to navigate, in a fully autonomous manner, through a partially known urban-like environment populated with (static and dynamic) obstacles and perform different tasks such as road and off-road driving, parking and visiting certain areas while obeying traffic rules. As the emphasis of the competition is geared more towards military applications, the vehicles have to be fully self-contained in every aspect including sensing, perception, computing, power, and control. Although this leads to an elegant setup, the situations encountered in DUC do not closely represent those faced in a real-world crowded city like Singapore, London, etc. In addition, the cost of hardware components on these autonomous vehicles is extremely high, making them imprac-

tical to be employed in social or commercial applications. The majority of the cost comes from expensive, high-performance sensors (e.g. Velodyne LIDAR) and localization units (e.g. Applanix Inertial Navigation System) needed so that the vehicles can effectively handle all the possible (even adversarial) environments they may encounter.

This paper focuses on crowded city situations where automation can significantly improve the throughput as compared to on ground human decisions. As opposed to adversarial environments faced in military applications, city environments faced in social or commercial applications are typically equipped with infrastructure including cellular networks, traffic cameras, loop detectors and ERP (Electronic Road Pricing) gantries. In addition, the detailed road network and many features of the environment in which the vehicles operate can be obtained a priori. In fact, autonomy in transport vehicles has been attained at various levels of interaction with the infrastructure. Systems that are driven with heavy infrastructure dependence such as mono-rails typically require significant setup and are more suitable for a fully controlled environment such as warehouses and docks. FROG (Free Ranging on Grid) Navigation Systems¹ is an example of such heavily structure-dependent systems and has been deployed for container handling operations in factory sites. Their system consists of a team of autonomous vehicles and a centralized supervisory system. The autonomous vehicles localize themselves using magnets on the floor and GPS for point to point navigation and use laser and sonar for obstacle avoidance. An advantage of this approach is that the global viewpoint of the environment can be obtained while the main disadvantages include delayed information, communication burden and cost of specialized infrastructure setup.

On the other hand, Google driverless car, for example, is fully autonomous [9]. Their converted Toyota Prius uses cameras to detect traffic lights and radars to detect pedestrians. Their autonomous navigation also relies on laser range finders, cameras, radars, inertial sensors and high-detail maps. An advantage of these fully autonomous systems is the ability to handle dynamic and adversarial scenarios. However, as

¹<http://www.frog.nl/>

opposed to heavily structure-dependent systems, only limited, local viewpoint can be obtained as on-board sensors on each vehicle have the same limitation of limited viewpoint and occlusion by urban structures and other vehicles. Each vehicle also bears the burden of sensing and perception. Additionally, being fully autonomous and self-contained comes with a heavy price tag.

In this paper, we investigate the integration of the heavily structure-dependent and the fully autonomous approaches in order to build an autonomous vehicle system at reasonable cost as well as keep the high level of autonomy even in a crowded scenario. As an initial step, we focus on campus environments. We exploit an abundance of infrastructural sensors available to the road network that can provide very important information about the presence of other entities on the road in real time. This information can help in planning collision-free optimal trajectories for each vehicle beyond visual range. Exploiting the infrastructure also helps reduce the sensing burden on the vehicle and reduce the cost of the vehicle, making such systems more economical and accessible to normal people. The main contributions of the paper are the following.

- 1) Typical localization techniques heavily depend on GPS (Global Positioning System). These techniques do not work when dealing with high-level buildings in city environments. We augment the localization using local laser maps and show improved results.
- 2) We implement a more robust pedestrian detection system using a complementary vision and laser system. We show that such a set up works well even for a moving vehicle.
- 3) Our approach is to incorporate existing infrastructural sensors whenever possible into the motion planning of the autonomous vehicle. To demonstrate the setup, we show a scenario incorporating the infrastructural sensors that significantly improves the performance of the vehicle operation.

The remainder of the paper is organized as follows. An overview of our testbed is provided in Section II. Section III describes our dynamic obstacle detection component and, in particular, the pedestrian detection algorithm implemented in our testbed. Its performance is evaluated, both for single and multiple pedestrian cases. The localization and navigation components are described in Section IV. Section V describes the role of infrastructural sensors in our system. Finally, Section VI concludes the paper and discusses future work.

II. AUTONOMOUS VEHICLE TESTBED

Our autonomous vehicle testbed is a Yamaha G22E golf car mounted with various sensors (Fig. 1a). It was modified and equipped with actuators to achieve drive by wire capabilities. Two servo motors are used to control the steering angle and amount of brake applied separately. Since it is an electric vehicle, the throttle control is readily accessible through a varying PWM voltage signal that can be regulated by a low level controller. To fulfill the power requirement for a wide variety of sensors, a 1350W inverter was used. For the

sensors, a wheel encoder is fitted at the left front wheel. The steering angle and brake are inferred implicitly from the motors encoder. To receive GPS signal, Ublox EVK-6R is used. The module comes with Enhanced Kalman Filter to give an estimate of global location using integrated data input from wheel encoders tick count and onboard gyroscope. The main sensors are the laser range finders that consist of two SICK LMS 291 and Hokuyo UTM 30LX. The SICK lasers have a range of 80 m with 180° field-of-view (FoV). The Hokuyo sensor, on the other hand, has 270° FoV with 30 m range. An off the shelf webcam is fitted on one of the SICK lasers to provide visual feedback and to perform vision processing.

Software Architecture: We have developed a modular software architecture for ease in incorporating additional functionality without modifying the core system. The system has been integrated into ROS (Robotic Operating System), which provides a standard form of communication among different software modules [11]. The main modules that have been implemented on the current system include perception, mapping-and-prediction, localization, planner and controller as shown in Fig. 1b.

The perception module takes as an input the raw sensed data from the sensors. Detection and tracking algorithm is then applied to extracted features (e.g., pedestrian and other moving and stationary objects) from the raw sensed data. As discussed in Section V, the sensors we have utilized include not only onboard cameras and laser range finders but also infrastructure cameras installed, for example, at an intersection. The data from these infrastructure cameras are transmitted through a WiFi network. To reduce the amount of data that needs to be transmitted, the raw sensed data may be processed so that only important features (e.g., other vehicles and pedestrians approaching the intersection) are transmitted.

The mapping-and-prediction module maintains the global map of the environment in which the system is operating. In addition, it predicts the motion of moving objects such as pedestrians. The localization module incorporates the data received from the GPS (Global Positioning System), IMU (Inertial Measurement Unit), laser range finder and vehicle's odometer to provide an estimate of the current state (position, velocity and heading) of the vehicle.

The planner module is responsible for computing an obstacle-free path, satisfying certain traffic rules, to the goal. In the case where the user is onboard, the goal may be specified as the destination through the user interface. Alternatively, the scheduling system that computes the pick-up and drop-off position for each autonomous vehicle may send the origin and destination to the system through a cellular network. Finally, the controller module is responsible for computing the actuator commands, including the speed, gear and steering angle, to the physical actuators so that the vehicle closely follows the planner-generated path.

III. DYNAMIC OBSTACLES DETECTION

For autonomous navigation, we need to pay special attention to dynamic objects, like pedestrians and other vehicles on the

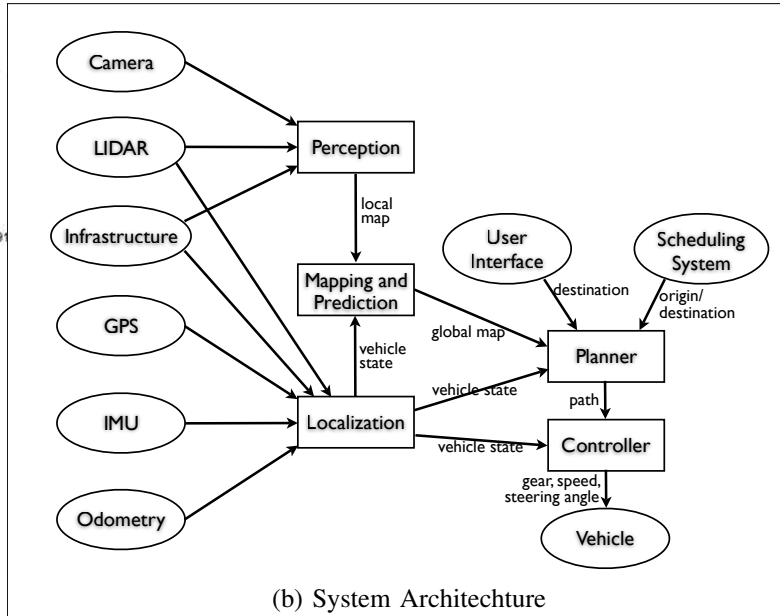
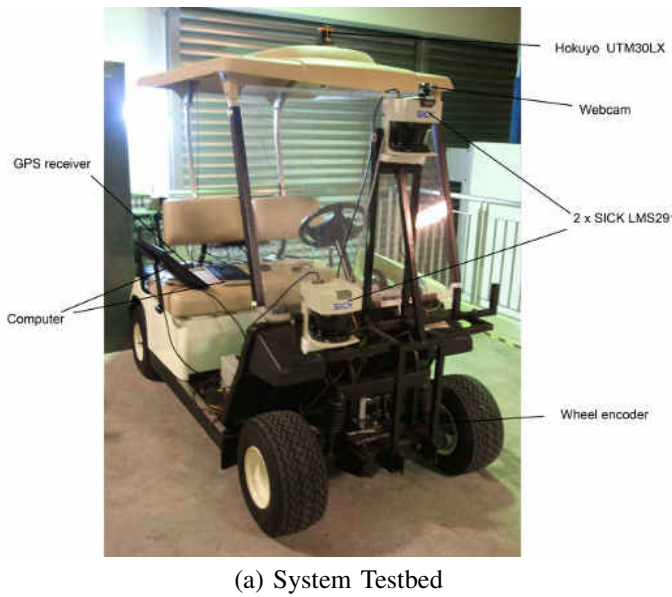


Fig. 1. Autonomous Vehicle Testbed



Fig. 2. Pedestrians and other dynamic vehicles need to be detected and handled. (a) NUS campus road, (b) Golf-cart operating in the presence of pedestrians

road in addition to static environmental features like kerbs, drains, traffic lights, *etc* (Fig.2). Usually the presence of static objects are known a-priori from a traffic database or built during and initial phase in an offline manner. However, dynamic objects on the road can only be handled while the autonomous vehicle is driving. Pedestrians, as a key factor in a campus environment, deserve more attention. On-board cameras are one of the most effective ways of identifying objects in the environment. However, the computation requirement and dependence of ambient light conditions limit their utility. Alternatively, laser based approaches can detect the presence of an object more reliably but have problem disambiguating different types of objects. In our project, we built an onboard pedestrian detection system by hierarchical fusing of a single-layer LIDAR and a simple off the shelf webcam. We combine the advantages of LIDARs in detecting an object with the simplicity of disambiguating objects from the camera images. It proves to be fast, computationally efficient and robust in our operations.

Significant research has been done on pedestrian detection and tracking with LIDARs and vision. The LIDARs provide accurate position information of scanned points in the surroundings, which will be segmented, clustered, and then classified into different objects. Pedestrians can be extracted

with certain criteria or features, such as static features of shape and size founded in [5], [10], or dynamic features of gait founded in [15], [2], and so on. These algorithms perform well with multiple LIDARs placed off-board and in relatively structured environment, but would probably fail in real urban traffic, due to severe occlusion and complex surroundings. In the final analysis, limitation of these algorithms comes from sparsity of information of LIDARs. The idea of multi-sensor fusion arises to counter this limitation. The most common type that can be found is a combination of LIDAR and camera. While some related algorithms have been introduced in [4], [6], few of them are suitable to autonomous vehicles, with considerations to the demanding working environment. An algorithm similar to our approach is proposed in [3]. It depends on a four-layer LIDAR to track pedestrians and do preliminary classification, and then use camera to refine the classification belief. In our project, a similar algorithm is proposed. While we also rely on a single-layer LIDAR to track objects, we do not try to classify them in this part, but leave that to the following part of vision verification. Our algorithm proves fast, computationally efficient, robust in operation, and easy to implement.

A. Pedestrian detection algorithm

In our implementation the moving object tracking is realized with a single-layer LIDAR. While the approach is general to any dynamic object, like vehicles, pedestrian and other objects, we take the pedestrians as a representative class to talk about in details in this paper. Fig.3 shows the flow of the algorithm while, Fig.4 shows the result of the detection algorithm for a single data frame. The algorithm runs in two phases, *pedestrian candidate detector* and *pedestrian verification*.

Pedestrian candidate detector: The LIDAR data is segmented and clustered based on their position and relative

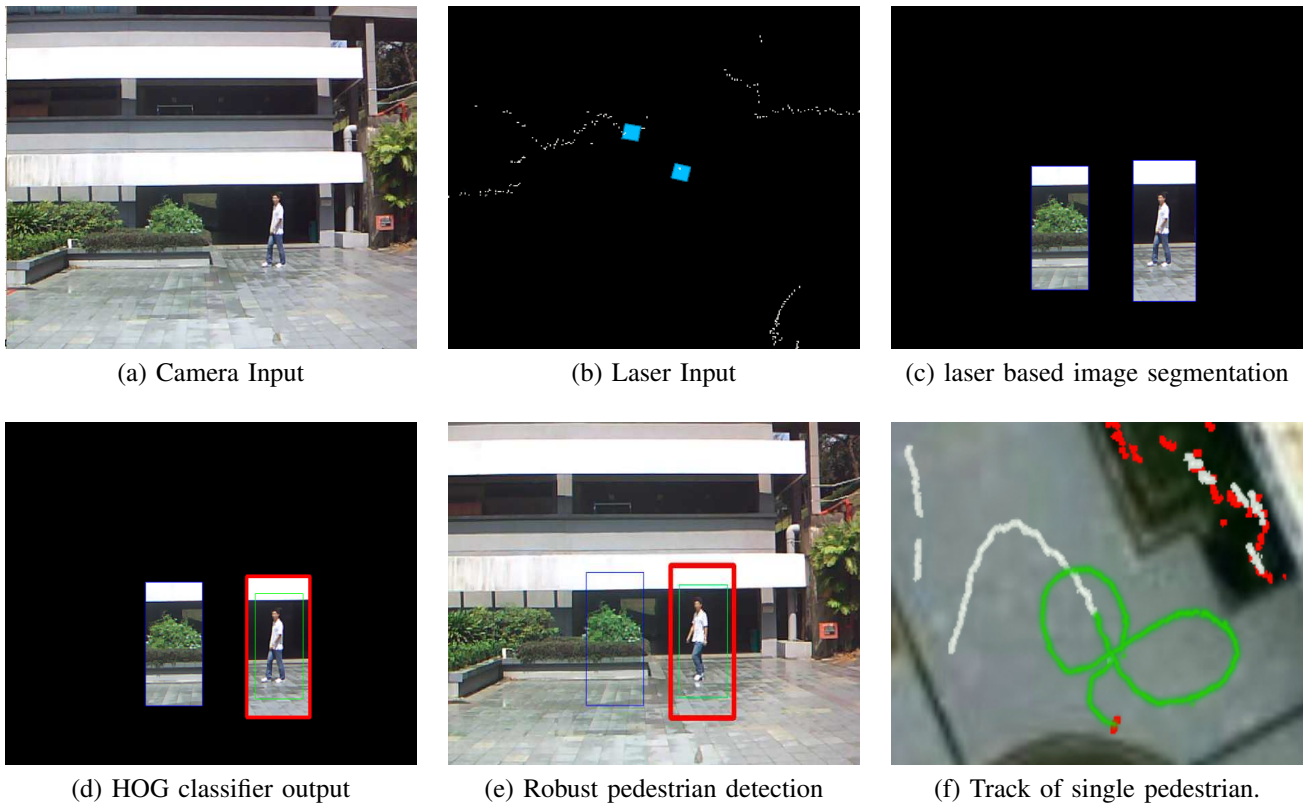


Fig. 4. Pedestrian detection module

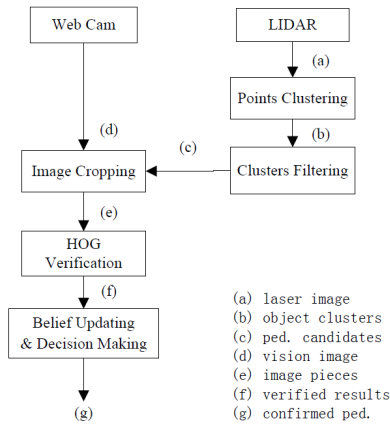


Fig. 3. Pedestrian detection module.

velocity (Fig.4b). Potential candidate clusters for pedestrians are filtered out based on their size and velocity. We use a simple linear velocity model in our implementation. However advanced model checks could also be used for higher accuracy. In fact keeping a more relaxed and conservative filter decreases the rate of false negatives in the subsequent pedestrian verification phase.

Pedestrian verification: In this part, we use a common webcam to verify whether extracted objects are pedestrians or not. Extrinsic calibration of webcam and LIDAR is done beforehand. These candidates are projected onto certain areas

of webcam image correspondingly. The whole image is then cropped into several smaller sub images (Fig.4c). Since only a small number of sub-images are processed, we decrease the computational time in image processing. The vision verification algorithm used here is histogram of oriented gradient object detection (HOG). In this work, a default trained people detector Support Vector Machines (SVM) from OpenCV was chosen. To enable fast verification, GPU accelerated HOG algorithm was used. HOG classifier identifies each sub image containing pedestrians (Fig.4d) and we label the LIDAR tracks accordingly. This helps us in avoiding running vision based pedestrian detectors on the whole image (Fig.4a,e) and significantly reduces the computational load. The reduced computation allows us to run such detectors in real time on the vehicle.

Note that the pedestrian detection allows us to improve the motion planning for the autonomous vehicles by reasoning about the motion models of the pedestrian obstacles. In the case of false negatives due to vision errors and FoV limitations, the pedestrian clusters are still tracked by the laser, treated as a generic dynamic object and avoided accordingly.

B. Performance evaluation

Single pedestrian detection: Fig.4 shows how this system detects and track a single pedestrian. Fig.4(f) shows the track of this pedestrian. At first, the pedestrian gets tracked by LIDAR, and labeled as a dynamic object shown in white. After it enters the FoV of webcam, it gets verified as a pedestrian

and the track turns green. Other potential dynamic objects in the image are correctly rejected.

Multiple pedestrian detection: Fig.5 shows the result of our pedestrian detection module during an segment of an autonomous run. Fig.5a shows a snapshot of a typical scene with multiple pedestrians. We see that when the pedestrians are too close to each other the laser signatures get merged and they are detected as a single cluster. However, in the view of motion planning it does not matter how many actual pedestrians are close by, our autonomous vehicle avoids the pedestrians effectively. In the test, most pedestrians got detected, whether as an individual, or as a group, making safe autonomous driving of our vehicle. Frequency of this detection system is up to 37Hz, limited by scan frequency from LIDAR. Range of effective detection is about 14 meters, limited by resolution of webcam. Our system works well within 14 meters with system precision around 0.8, From Fig.5b, Vision verification improves precision of detection system based on laser extraction. Even when much noise exists, as what happened at the distance between 4 to 6 meters in our collected data, while precision of laser part is only about 0.20, vision part improves it to about 0.80. Note that most of the errors if any in detection as shown in Fig.5b are corrected at around 4m, at which distance the autonomous vehicle is able to come to an emergency stop under the influence of high frequency low level obstacle avoidance. We will be re-visiting this detection rate at higher speeds once we implement the system on a faster vehicle.

IV. LOCALIZATION AND AUTONOMOUS NAVIGATION

Most of the popular approaches in autonomous navigation outdoors depend heavily on GPS based localization. In fact the DARPA challenge was based on GPS based waypoints as input. However GPS is not very reliable in urban areas due to multi-path, limited satellite view in tall sky-scrapers. Such a scenario is shown in Fig.6a. A cloudy sky coupled with tall buildings can attenuate the GPS signals, resulting in erroneous localization as shown by the GPS track in Fig.6b. As the vehicle moves, its GPS erroneously estimates the vehicle location inside buildings and a pure GPS based localization and control could lead to failed navigation.

Interestingly one of the main reasons of GPS limitation i.e, the proximity of buildings, itself provides a good opportunity to utilize range based localization algorithms. In our work we use the laser based maps to augment in regions where GPS underperforms.

A. Localization

We compare our approach with the popular approach used by most of the participants successfully in the DARPA Urban Challenge of using the integrated GPS + Inertial based bayesian estimate. While such an approach can help localize in a-priori unknown environments as was the case in the DARPA Challenges, the presence of cloudy sky and high rise buildings give very poor estimates.

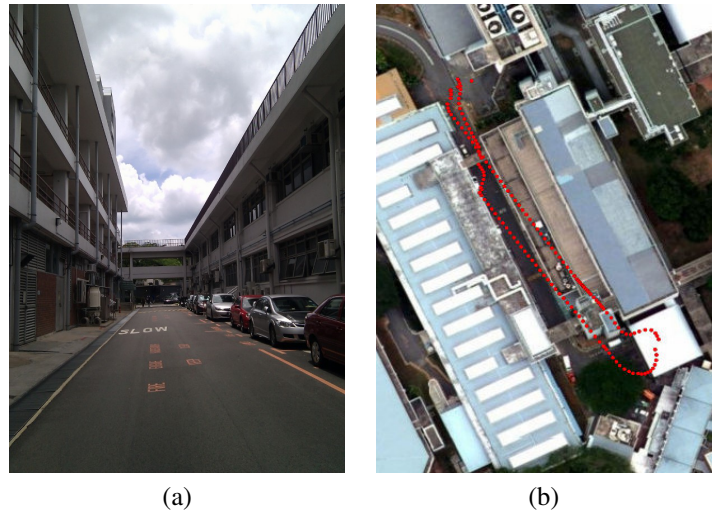


Fig. 6. Bad GPS with integrated gyro based odometry

Recently, [8], showed significant improvements in autonomous vehicle localization using the powerful Velodyne 3-D omnidirectional HD-LIDAR. The cost of building a precomputed map, is greatly offset by improvements in the online localization accuracy. We used a 2-D single plane laser 180 FoV Hokuyo UTM 30 LX mounted at the roof of the autonomous vehicle for mapping and localization. To build an occupancy grid based map, we used off-the-shelf package in ROS, OpenSLAM [1]. We performed the standard Adaptive Monte Carlo Localization (AMCL) [12] during the autonomous run. AMCL requires odometry which we fused from the IMU/LIDAR scan matching data streams. The purpose of the exercise was to see how our system fared qualitatively, with such a simple setup.

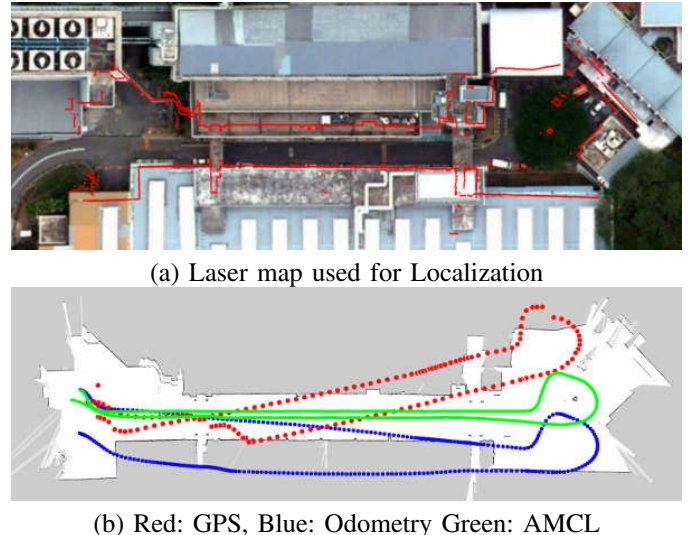
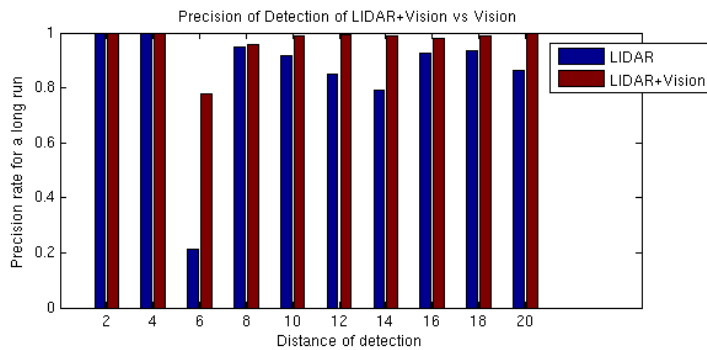


Fig. 7. Qualitative comparison of various localization schemes

We show the results of an autonomous run in Fig.7. Fig.7a shows the boundary of occupancy map overlaid on a satellite map showing sufficient reliability. Fig.7b shows the localization results of various algorithms on the same test run. The red track shows the GPS logs while the blue track shows the



(a) Sample pedestrian detection



(b) Pedestrian precision

Fig. 5. Pedestrian detection from an autonomous run of our testbed.

location of the vehicle using the odometry alone. The green track is computed by the AMCL algorithm. We see that for the purposes of our sample test run, our laser based localization was sufficient. This is also borne out by various runs of our autonomous vehicle one of which is shown in Fig.7b, in the presence of temporary occlusions and other dynamic vehicles.

B. Cost based Navigation function

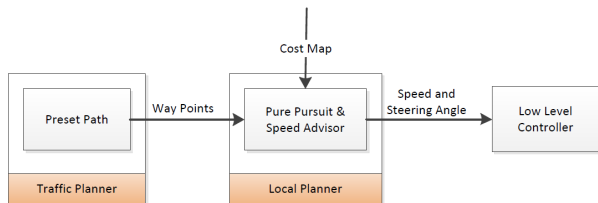


Fig. 8. Navigation module of the autonomous vehicle

Fig.8 shows the structure of our navigation module. During each run, the vehicle maintains its own map based on rolling basis, with the vehicle centered on the map. A map of 50m x 50m with a grid cell size of 0.2m is maintained at all time. Each cell in the map can have a 1 byte value. Initially, the cells in the map are marked as unknown with a value of 255. Whenever an obstacle is observed, the map is updated with a cost value of 254, with the cells now marked as obstacles, the cost is propagated radially outward with an exponential function. At the low level, speed and steering control are separated. For the speed control, the vehicle considers the following input before planning for next action: the average cost function that is present within a defined area in front of itself and the curvature of the path. To ensure stability, a conservative approach is utilized. First, an exponential function is used to calculate the safe speed given the steering angle of the golf car. Then, from the normalized average cost along the projection of the golf car within a fix distance, another safe speed is obtained. Between these 2 values, the minimum one is chosen as the final set point for the speed. The implementation of waypoint follower uses pure pursuit control [7]. Since the golf car's maximum speed is limited to 2 m/s, the look-ahead vector is fixed to 3 meters.

V. EXPLOITING INFRASTRUCTURE SENSORS

An important feature that distinguishes urban environments from those considered in military applications is the technological advances that we can exploit in order to increase safety and efficiency of the system without imposing much additional cost. Consider, as an example, the scenario where an autonomous vehicle has to traverse an intersection. In many cases, other vehicles approaching the intersection from other directions may not be detected properly by the onboard sensors due to limited sensing coverage and occlusions caused by structures and other environmental features. In [13], the authors mitigate this problem by using two pointable long-range sensors and propose a methodology for computing the optimal angles of the sensors to maximally cover relevant unobserved regions of interest. A method for detecting occlusions is also presented. A phantom vehicle is then placed in the occluded area, leading to a safe but potentially conservative result.

In this work, we consider utilizing infrastructure cameras installed, for example, at an intersection, rather than completely relying on the onboard sensors. These infrastructure cameras can provide information about whether there are pedestrians or other vehicles approaching the intersection. The information can then be transmitted through a WiFi or cellular network. An advantage of this approach is that more accurate information can be obtained as the infrastructure cameras may be mounted to avoid occlusions. In addition, as the number of autonomous vehicles in the system exceed the number of intersections, the cost can be substantially reduced. In fact, in many modern cities, cameras are already installed at many intersections to detect traffic violations. Hence, this approach may incur almost no additional cost.

A. Avoiding Unobserved Pedestrians

To show the effect of additional information, we simulate an infrastructural sensor as a wifi-node broadcasting specific information. The infrastructural camera detects the presence of pedestrians and gives a binary information to the golf-cart whether there are pedestrians about to cross the road or if the region is pedestrian free. Currently we are not building models of pedestrian intentions to analyze whether the pedestrian is facing the road or whether s/he is just waiting rather than trying to cross the road. Any pedestrian detection would

trigger the autonomous vehicle to slow down in anticipation for the pedestrian to cross the road. The rate of pedestrian detection is 5 Hz. However, since the algorithm only depends on the pedestrian detection alone and not a more detailed analysis based on the pedestrian position and heading, it would also work well with modern traffic/security cameras operating around 1Hz.

Fig.9(a,b) show the view of the scene from onboard as well as a mock infrastructure sensor. The detection of the pedestrian in the left of the image in Fig.9a is quite difficult due to the occlusion from pillars and railings. The autonomous vehicle has to communicate with an existing sensor (security camera) to get more information to plan its path. The pedestrian detection is much easier in Fig.9b. The autonomous vehicle gets the pedestrian information from the infrastructure pedestrian detector sensor and modifies its motion plan, as shown in Fig.9c. The videos of the operation can be accessed at <http://dl.dropbox.com/u/20792983/pedestrianVisual1.mp4> and <http://dl.dropbox.com/u/20792983/pedestrianVisual2.mp4>.

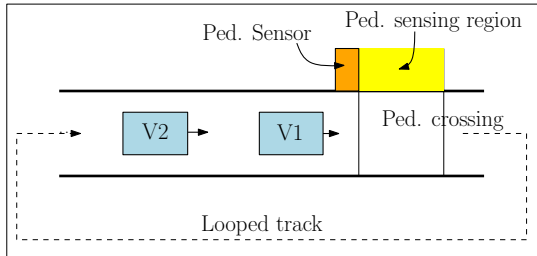


Fig. 10. Simulation environment setup.

1) *Control comparative experiments:* Clearly, we would expect to see an improvement in the navigation performance while incorporating the information from an infrastructural sensor. To test the amount of improvement in a stochastic pedestrian arrival scenario, we run a control experiment in simulation as shown in Fig.10. One or more vehicles move in a loop that has a pedestrian crossing. A pedestrian detection sensor (i.e., a traffic camera) detects the presence of pedestrians and sends information to the autonomous vehicles $V1, V2, \dots$. Depending on the presence of pedestrians appearing in a stochastic poisson process of mean rate $\lambda = 1ped/sec.$, the vehicles slow down or keep moving. The vehicles are also constrained to maintain a minimum distance between them to avoid collision. We run the experiment both for a single vehicle as well as multiple vehicles. We compare the performance with the baseline case where there is no infrastructural sensor and the autonomous vehicle has to come to a stop before detecting pedestrians on road, something similar to a regular stop and yield traffic sign. We run the simulation for various vehicle speeds and various number of vehicles. Let T_{base} be the time taken to reach the pedestrian crossing by the baseline algorithm, while $T_{infra.}$ be the same measure for our algorithm getting additional information from the infrastructure sensor. We compute the difference in the time taken to complete each lap, as the time gained by using

the infrastructural sensor, $T_{gain} = T_{base} - T_{infra.}$.

Single Vehicle: Fig.11a, shows the plot of T_{gain} vs the number of laps the vehicle completes. We see clearly that the cumulative time gained by using the infrastructural sensor improves with time. We also note that such a gain is more significant when the vehicle moves at a higher speed. The blue plot is the gain for vehicle moving at 2m/s while the red at 1m/s. This shows that the traffic flow at pedestrian crossings where the vehicles are able to move at higher speeds can be significantly improved by using infrastructural sensors.

Multiple Vehicles: Fig.11b, shows the plot of T_{gain} vs the number of laps the vehicle completes for multiple vehicles. We see that as the number of vehicles increases, T_{gain} also increases. This is because in the baseline algorithm, each vehicle has to stop for pedestrians whether or not they are present. Additionally they have to stop to maintain a minimum distance to the vehicle in front when the front vehicle stops. The number of vehicle stops increases significantly when the number of vehicles increases raising the discrepancy between the proposed and the baseline algorithm.

T_{gain} is the difference in cumulative improvement in performance. In a stochastic setting, T_{gain} varies and so although we see a clear mean increase in performance, the plot is not strictly monotonic. Moreover for the multiple vehicle scenario we assumed that pedestrian take longer duration to complete the crossing than the car making the stop and go sequence. Hence over the long run, more often than not the lead car (Vehicle 1) takes the burden of stopping for the pedestrians showing a smaller rate of performance improvement.

During the operation and our experimental runs, we noticed that the infrastructure sensor rarely had any failure in detecting the appearing pedestrians. As the detection is a boolean i.e, either there is a pedestrian or not, a very reliable detection rate is achieved by carefully tuning the parameters of the static infrastructure sensor to the exact detection region. Moreover, the autonomous vehicle testbed moves relatively slowly under 3m/s and performs a proximity test with its onboard range sensors at a very high frequency. These two factors in conjunction avoids any future collisions of the vehicle with incoming pedestrians. We are currently investigating the effects of noisy infrastructure sensor, information delay, and high speed vehicles for more complex scenarios.

VI. CONCLUSION AND FUTURE DIRECTIONS

We considered three main challenges in autonomous navigation in crowded city environments: localization, pedestrian detection and limited onboard sensing capability. We showed that in the proximity of tall buildings, popular GPS-based localization can be extremely erroneous. Odometry-based localization was shown to perform slightly better. In order to achieve acceptable performance, we augmented the localization using local laser maps based on Adaptive Monte Carlo Localization technique and showed significantly improved results. We also integrated the use of vision and LIDARs to achieve more robustness in pedestrian detection and tracking. Finally, we

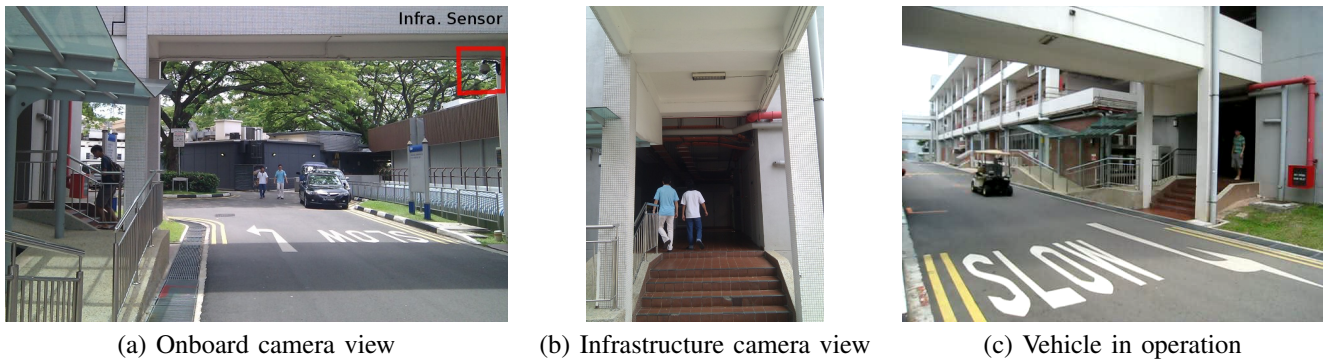


Fig. 9. Pedestrian crossing experiment

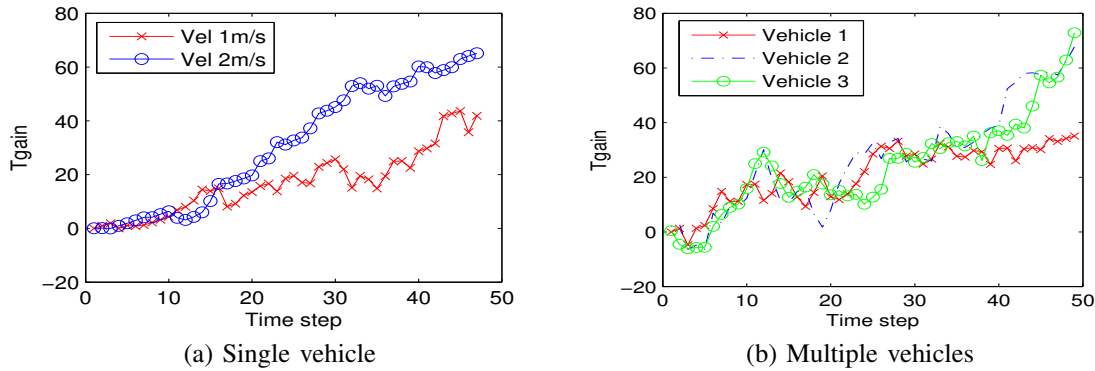


Fig. 11. Improvement in traffic flow due to incorporating infrastructural sensing in simulation.

exploited existing infrastructural sensors to improve the onboard sensors visibility. The performance of the overall system was evaluated.

Future work targets at augmenting the current system to a fully automated campus vehicle system. To this end, we are currently investigating the use of WiFi-based localization as a complementary approach to GPS-based and laser-based localization. We also plan to incorporate pedestrian intentions in motion planning. Implementation of high-level logics to ensure that the autonomous vehicle obeys traffic rules, properly handles pedestrian and responds to faults and failures is also of interest. Finally, the system needs to be verified for safety both for nominal operations and in the presence of faults and failures.

REFERENCES

- [1] OpenSLAM. <http://openslam.org/>.
- [2] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Robust tracking of multiple people in crowds using laser range scanners," in *Proceedings of ICPR*, vol. 4, 2006, p. 857.
- [3] F. N. G. Gate, A. Breheret, "Fast pedestrian detection in dense environment with a laser scanner and a camera," in *Proceedings of IEEE Vehicular Technology Conference*, 2009.
- [4] R. S. H. Zhao, "A real-time system for monitoring pedestrians," in *Proceedings of IEEE WACV/Motions*, vol. 1, 2005, p. 378.
- [5] V. W. K. Fuerstenberg, K. Dietmayer, "Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner," in *Proceedings of Intelligent Vehicle Symposium*, 2002.
- [6] Y. Kobayashi and Y. Kuno, "People tracking using integrated sensors for human robot interaction," in *Proceedings of 2010 IEEE International Conference on Industrial Technology*, 2010, p. 1617.
- [7] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How. Real-time motion planning with applications to autonomous urban driving. <http://dspace.mit.edu/openaccess-disseminate/1721.1/52527>.
- [8] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2010.
- [9] J. Markoff, "Google cars drive themselves, in traffic," *The New York Times*, October 9, 2010.
- [10] T. H. O. Mozos, R. Kurazume, "Multi-part people detection using 2d range data," in *Proceedings of 2009 IEEE ICRA*, 2009.
- [11] 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.
- [12] D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic Robotics*. MIT Press, 2005.
- [13] Y.-W. Seo and C. Urmson, "A perception mechanism for supporting autonomous intersection handling in urban driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1830–1835.
- [14] S. E. Shladover, "Potential contributions of intelligent vehicle/highway systems (IVHS) to reducing transportation's greenhouse gas production," *Transportation Research Part A: Policy and Practice*, vol. 27, no. 3, pp. 207 – 216, 1993.
- [15] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser-range scanners," in *Proceedings of 2005 IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 2, March 2005.

Integration of visual and depth information for vehicle detection

Alexandros Makris, Mathias Perrollaz, Igor E. Paromtchik, Christian Laugier

Abstract—In this work an object class recognition method is presented. The method uses local image features and follows the part based detection approach. It fuses intensity and depth information in a probabilistic framework. The depth of each local feature is used to weigh the probability of finding the object at a given scale. To train the system for an object class only a database of annotated with bounding boxes images is required, thus automatizing the extension of the system to different object classes. We apply our method to the problem of detecting vehicles from a moving platform. The experiments with a data-set of stereo images in an urban environment show a significant improvement in performance when using both information modalities.

I. INTRODUCTION

The state-of-the-art visual object class recognition systems operate with local descriptors and codebook representation of the objects. Various local features (e.g. gradient maps, edges) are used to create the descriptors. Then kernel based classifiers are commonly employed to classify the detected features in one of several object classes [1][2][3][4]. The recognition of vehicles or pedestrians from sensors mounted on a moving platform is achieved by different approaches using various types of sensors, e.g. stereo camera, laser [5][6][7][8]. The approaches that perform data fusion from various sensors have proven to be the more robust in a variety of road conditions [9][10].

This work focuses on the development of an object class recognition system which follows the part based detection approach [2]. The system fuses intensity and depth information in a probabilistic framework. To train the system for a specific object class, a database of annotated with bounding boxes images of the class objects is required. Therefore, extending the system to recognize different object classes is straightforward. We apply our method to the problem of detecting vehicles by means of on-board sensors. Initially, depth information is used to find regions of interest. Additionally, the depth of each local feature is used to weigh its contribution to the posterior of the object position in the corresponding scale. In the following we provide a brief review of the methods related to our approach.

In the object recognition literature there is a large amount of works that follow the part-based approach. In [2], a codebook of object part appearance is constructed using interest point detector-descriptor pairs. The detected features are grouped into clusters and linked to the center of the object. A method that builds upon the aforementioned approach is presented in [11]. An approach to discriminatively

learn a mapping between image patches and Hough votes is presented. Random trees are used to learn the above mapping in a supervised way (instead of clustering). In [12] shape and appearance information is used to perform object class recognition based on part detection and Hough transform. The codebook entries are selected using the boosting algorithm according to their significance, which is related to its discrimination capacity and the precision of the localization information for the object's centroid. In [1], a grouping of local features into pairs is proposed in order to increase their discriminative power. Selecting features connected by lines ensures finding features pairs with high repeatability.

Stereo-vision is widely used in the field of intelligent vehicles, mainly for generic obstacle detection [13][14]. A different approach for vehicles recognition is presented in [15], where the authors detect possible cars using 3D points provided by stereo-vision, and confirm the recognition of cars through a symmetry criterion. In [16], the author generates hypotheses of pedestrian as connected areas of constant disparity, and uses the aspect ratio of the corresponding regions as a clue to recognize pedestrians.

Lately, several methods that combine intensity with depth information have been proposed. In [17], vehicle and pedestrian detection is performed following the approach of [2] but also filtering the search regions by using the ground plane constraints. In [18], a method for pedestrian detection from a moving vehicle is presented. Stereo cues and a clustering algorithm are used to find candidate areas. In the following several detection windows are constructed around each area. The detection takes place in these windows using multiple features applied in manually predetermined sub-regions. In [9], a pedestrian classification method using depth and intensity features is developed. In this method the holistic detection approach is used extracting features from the whole region and feeding a classifier. The authors demonstrate that using both depth and intensity information outperforms any single modality method. Integration of stereo-vision with visual recognition has been proposed in [19], for estimating the road surface, reducing the hypotheses for a sliding window approach. In [20][21][22], video and laser data are fused to achieve robust vehicle and pedestrian detection.

The novelty of our approach is the fusion of depth and intensity information to form a probabilistic part-based detector. Firstly, we develop a framework to estimate the probability of finding an object at a position given all the available information. The depth of the detected local features is used to weigh (w.r.t. the corresponding distance) their contribution for the scale of the object. Using the depth information in this way takes into account the context in which we expect to

INRIA Grenoble Rhône-Alpes, 38334 St. Ismier, France
(alexandros.makris, mathias.perrollaz,
igor.parmotchik, christian.laugier)@inria.fr

find the objects (e.g. distant view, close-up). This is beneficial for the robustness of the approach, by avoiding for example many noisy detections resulting from false matches between features of different scales. Additionally, the computational gain from filtering out regions is very important for the on-line operation of the system which is required in the intelligent vehicles application. The method is tested with stereo video sequences captured in an urban environment.

The paper is structured as follows. Section II provides the theoretical background for our method. Section III gives the implementation details, providing a description of the stereoscopic sensor, the depth calculation algorithm, and the training and detection algorithms. The experimental evaluation of our method follows in Section IV and finally the conclusions in Section V.

II. METHOD DESCRIPTION

The proposed method is a probabilistic part-based object recognition method fusing intensity and depth information. The aim is to find the occurrences of a specific object category and viewpoint. Let o_n denote the object category/viewpoint with state vector $\mathbf{x} = [i_x, i_y, i_s]^T$ comprised of the image coordinates of the object's center and its scale. The method estimates the probability distribution $p(o_n, \mathbf{x} | \mathbf{I})$ where \mathbf{I} denotes the image measurements.

The measurements are a set of N image features $\mathbf{I} = \{\mathbf{f}_j, \mathbf{d}_j\}_{j=1}^N$, where \mathbf{f}_j and \mathbf{d}_j are the intensity and depth descriptor of feature j respectively. The features are linked to the object through a codebook representation denoted by $\mathbf{C} = \{C_j, \mathbf{x}_{cj}\}_{j=1}^N$ where C_j is a random variable over the possible codebook labels of feature j and $\mathbf{x}_{cj} = [i_{xj}^c, i_{yj}^c, i_{sj}^c]^T$ its position and scale. The possible labels are the M clusters of the codebook $\{c_i\}_{i=0}^M$ where c_0 is the possibility that no cluster is observed. For each codebook cluster c_i we calculate during training the associated descriptor \mathbf{f}_{c_i} , and the conditional probability distribution $p(C_j = c_i, \mathbf{x}_{cj} | o_n, \mathbf{x})$. This distribution enables us to estimate the position and scale of the cluster knowing the position and scale of the object \mathbf{x} . If the camera parameters are known, the distance between the camera and observed cluster and thus the object can also be inferred. The graphical model depicting the conditional independence assumptions that we make is shown in Fig. 1.

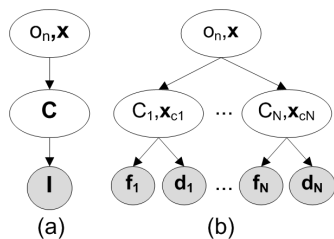


Fig. 1. Graphical Model of the method. (a) Model using \mathbf{C} variable to denote the cluster labels and positions and \mathbf{I} for all the available image measurements. (b) Analytic form showing the decomposition when multiple features are present. Each feature has an intensity \mathbf{f}_j and a depth descriptor \mathbf{d}_j and is associated with the possible clusters labels through C_j .

The probability of the object o_n at position \mathbf{x} given all the

available measurements is given by:

$$p(o_n, \mathbf{x} | \mathbf{I}) = \sum_{\mathbf{C}} p(o_n, \mathbf{x} | \mathbf{C}) p(\mathbf{C} | \mathbf{I}) \quad (1)$$

where the marginalization is over the values of \mathbf{C} .

The first term of (1) is the probability of having the object at a position given the set of observed clusters:

$$p(o_n, \mathbf{x} | \mathbf{C}) = p(o_n, \mathbf{x}) \prod_{j=1}^N \frac{p(C_j, \mathbf{x}_{cj} | o_n, \mathbf{x})}{p(C_j, \mathbf{x}_{cj})} \quad (2)$$

where we make the assumption that each cluster is independent from the others given the object. The second term of (1) is given by:

$$p(\mathbf{C} | \mathbf{I}) = \prod_{j=1}^N p(C_j, \mathbf{x}_{cj} | \mathbf{f}_j, \mathbf{d}_j) \quad (3)$$

$$\propto \prod_{j=1}^N p(\mathbf{f}_j | C_j) p(\mathbf{d}_j | C_j, \mathbf{x}_{cj}) p(C_j, \mathbf{x}_{cj})$$

where the probability of observing a feature given the corresponding cluster is considered independent from the rest of the features. The terms of (3) are:

- $p(\mathbf{f}_j | C_j)$ is the intensity likelihood calculated by comparing the observed feature descriptor \mathbf{f}_j with the cluster's descriptor.
- $p(\mathbf{d}_j | \mathbf{x}_{cj}, C_j)$ is the depth likelihood computed by comparing the distance of the feature calculated using the depth information δ_d with the distance calculated using the scale of the cluster δ_s .
- $p(\mathbf{x}_{cj}, C_j)$ is the prior for observing the cluster C_j at a position \mathbf{x}_{cj} .

By replacing (3), (2), in (1) we get:

$$p(o_n, \mathbf{x} | \mathbf{I}) \propto p(o_n, \mathbf{x}) \prod_{j=1}^N \sum_{(C_j, \mathbf{x}_{cj})} p(C_j, \mathbf{x}_{cj} | o_n, \mathbf{x}) p(\mathbf{f}_j | C_j) p(\mathbf{d}_j | C_j, \mathbf{x}_{cj}) \quad (4)$$

We consider the prior $p(o_n, \mathbf{x})$ as uniform. Additionally, for each possible object position we consider only the contribution from the clusters observed within the object region. The possible detections are the local maxima of the posterior. The clusters observed outside the object region cannot affect the position of these maxima. In Section III-C, we describe the algorithm we use to estimate the posterior.

III. VEHICLE DETECTION SYSTEM IMPLEMENTATION

A. Stereo System

The vision system used in this paper is a stereoscopic sensor. It is considered as perfectly rectified. Cameras are supposed identical and classically represented by a pinhole model, $(\alpha_u, \alpha_v, u_0, v_0)$ being the intrinsic parameters. The length of the stereo baseline is b_s .

For further geometrical developments, let us define a *Vehicle Coordinate System* (VCS). For simplicity in notations,

and without loss of generality, the yaw, pitch and roll angles of the camera, relative to the VCS, are set to zero. If it is not the case, homographies can be applied to the images in order to retrieve an equivalent configuration. In the VCS, X axis is parallel to the stereo baseline, Y is parallel to the optical axes and Z is oriented toward increasing height. (X_o, Y_o, Z_o) denotes the center of the stereo baseline in the VCS. Arbitrarily, we use the left camera of the stereo pair for the recognition task. Thus the coordinates $[i_x, i_y]$ will refer to the left image coordinates.

The stereo images are processed in order to retrieve depth information, following the approach described in [23]. First, a semi-dense matching algorithm is used in order to estimate a disparity value i_d for each pixel. During this stage, pixels are classified as road or obstacle by considering vertical and horizontal objects hypotheses. We use this information to discard the regions which correspond to the road surface or to objects that are not of interest (e.g. buildings, sky) using an arbitrarily chosen threshold for the height of the objects. An example of the mask resulting from this procedure is shown in Figure 2. With this step typically about 75% of the image is discarded thus the computational cost of the approach is reduced by the same ratio. For the obstacle pixels we retain the depth information. The distance of each pixel into the VCS is given by:

$$\delta_d = Y_o + \frac{\alpha_u b_s}{i_d} \quad (5)$$



Fig. 2. Depth mask example. The mask filters out the road surface and the objects that are over a prespecified height.

B. Detector Training

The training of the visual object recognition system follows the codebook based approach of [2]. For each object category/view we want to detect, a database of positive images is used to train the system. During the training phase we calculate the local SIFT [24] or SURF [25] features in a dense grid of image positions and different scales. A clustering step in the feature space using k-means is then performed to create a codebook of local appearances for each object class. For each cluster c_i we store: a) its appearance represented by the mean feature vector \mathbf{f}_{c_i} , b) its relative position to the center of the object. The latter is used to estimate $p(C_j = c_i, \mathbf{x}_{c_j} | o_n, \mathbf{x})$. Figure 3 shows an example of several clusters for the side-view of vehicles object class.

C. Detector Implementation using Depth-Vision Integration

In this section we describe the detection algorithm we use to estimate the probabilities defined in Section II. The overall

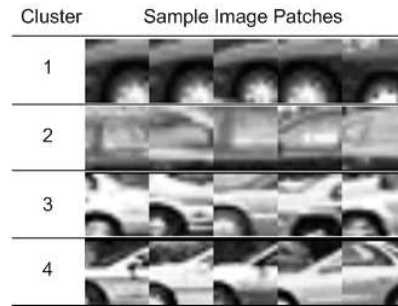


Fig. 3. Car-Side codebook clusters. Several image patches belonging to four clusters are shown. The clusters have been created with features extracted from the UIUC car database.

approach is shown in Figure 4. Algorithm 1 summarizes the steps of the approach.

In the detected regions of interest features are extracted from a dense grid and the respective descriptors are computed. The features are then matched to the clusters of the codebook. The likelihood of an intensity descriptor given a cluster $p(\mathbf{f}_j | C_j)$ is calculated by comparing the cluster's descriptor to the feature's descriptor. For the depth likelihood the scale in which the cluster is observed has to be taken into account. Let i_s^f be the scale in which a feature is detected and i_s^{cin} the initial scale of the matched cluster in the codebook. Then the feature will be assigned with a cluster of scale:

$$i_s^c = \frac{i_s^f}{i_s^{cin}} \quad (6)$$

Knowing the scale of the cluster assigned to the feature we can determine the scale of the object. Using the predetermined size of the object class and the camera parameters we convert this scale into distance δ_s . For the same image patch we calculate the distance information we get from the stereo δ_d . As shown in equation 5, δ_d is obtained from a disparity value i_d . This value is estimated by taking the median disparity value in the neighborhood associated to the feature. Using the two distances the depth likelihood is calculated according to:

$$p(\mathbf{d}_j | C_j, \mathbf{x}_{c_j}) = \exp \left\{ -\frac{(\delta_s - \delta_d)^2}{2\sigma_d^2} \right\} \quad (7)$$

where σ_d^2 is the variance parameter and is a linear function of δ_d . As the distance grows the uncertainty of the stereo distance estimation grows as well so a larger variance is required in order to have a non-negligible likelihood even with significant difference between δ_d and δ_s . The above technique allows us to group together features of the same scale, verified by the depth information. This way we filter out the noise resulting from false positive matches between different scales.

When the contribution of all features is taken into account, the mean-shift algorithm is used to find the local maxima in the \mathbf{x} space. The maxima represent the positions and scales of the possible detections.

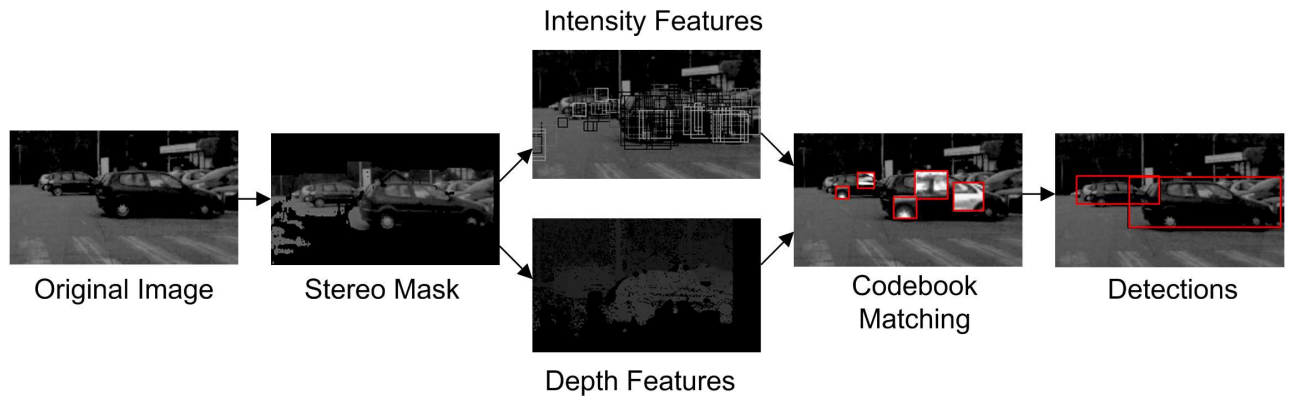


Fig. 4. The steps of the detection procedure are shown. The stereo information is used to define the regions of interest for the subsequent steps. Intensity and depth features are extracted from a dense grid within these regions. In the following the features are matched with the codebook clusters which are in turn used to estimate the posterior for the object in each position. The detections are the local maxima of the posterior.

Algorithm 1 Detection Algorithm

Input: Stereo pair: \mathbf{I} , pdf: $p(C_j = c_i, \mathbf{x}_{c_j} | o_n, \mathbf{x})$.

Filter image using stereo.

Extract intensity/depth feature pairs from each of the N positions of a dense scale-space grid.

for For Feature $j = 1$ to N **do**

Calculate Intensity likelihood $p(\mathbf{f}_j | C_j)$.

Calculate Depth likelihood $p(\mathbf{d}_j | C_j, \mathbf{x}_{c_j})$.

Posterior update with the contribution of the feature using (4).

end for

Locate local maxima of the posterior using mean-shift.

Output: A set of K detections $\left\{ o_n^{(k)}, \mathbf{x}^{(k)} \right\}_{k=1}^K$, with associated probabilities: $p(o_n^{(k)}, \mathbf{x}^{(k)} | \mathbf{I})$.

IV. EXPERIMENTS

In this section, we describe the experiments we conducted to evaluate the performance of our method. We apply our method to car detection and we demonstrate the improvement in robustness and computational efficiency of the complete system compared to the system using only intensity information.

For testing purposes we created a data-set using our experimental platform. The platform is a Lexus LS600h vehicle equipped with a TYZX stereo camera placed behind the windshield (Fig. 5). The stereo camera baseline is 22cm, with a field of view of 62°. Camera resolution is 512x320 pixels with a focal length of 410 pixels. The data-set contains 150 stereo images taken in an urban environment. We annotated the cars in these images with bounding boxes. The data-set includes challenging images, with poor illumination conditions, partial occlusions, and significant scale variations. For instance, the height of the annotated cars varies from 20 to 100 pixels.

For evaluation we compare the full method with the one using only intensity. To train both methods we used the UIUC car database. This database contains 550 images of



Fig. 5. Our Experimental Platform. Lexus LS600h vehicle equipped with a TYZX stereo camera placed behind the windshield.

side views of cars. Using this data-set we created a codebook of 2000 clusters. For the full method we set the variance parameter of the depth likelihood in (7) to $\sigma_d = 0.05\delta_d$. We tested the system with both SIFT and SURF descriptor. The difference in performance was negligible therefore in the experiments we used the SURF descriptor because it can be computed much faster. For the fairness of comparison we used the depth mask to find regions of interest for both methods.

In Fig. 6 we show some example detections. The proposed method detects side-views of cars in various scales, in cases with partial occlusions, and under significant background clutter. Part-based methods in general are more robust with partial occlusions. The use of depth information increases further the robustness as the features of each object are associated with a scale which in general is different from the scale of the occluding objects. An example of such situation can be seen in Fig. 7. We show a detection with and without depth information along with the features that contributed to that detection. As can be seen, in the case where no depth information is used (Fig. 7(c), (d)), many features that belong to a part of another vehicle in the background interfere with the detection resulting in inaccurate scale and position. With the use of depth information most of the features that are not on the object have been filtered out, thus resulting in a much better detection.

To perform a quantitative comparison we used a subset of our data-set, containing 60 images, where we detected the side-views of cars. For evaluation, we followed the single frame scheme which is adopted by the PASCAL object detection challenges [26]. For each frame we ran our



Fig. 6. Car-side detection examples. True and false positive detections are represented with red and yellow bounding boxes respectively. (a) Cars in different scales with significant background clutter and significant occlusions are detected. (b) Precise detection of the un-occluded vehicle, whereas a vehicle that is heavily occluded in the left is not detected. (c) Difficult detection of a vehicle which is far and partially occluded and a false detection in the region between the road surface and the trees. (d) Detection with partial occlusion. (e) Partial detection of a taller than normal vehicle (on the left). The training dataset does not contain vehicles of this type. (f) Successful detection of a partially occluded car and a false positive arising from a bus and a van. Training separate detectors for these types of vehicles as well will help to avoid these false alarms.

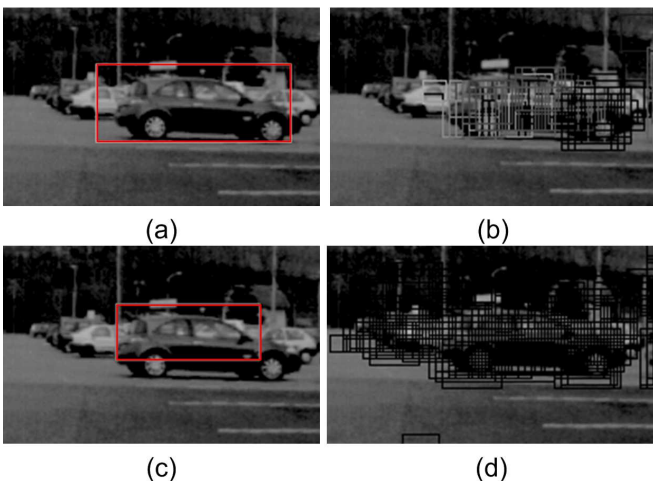


Fig. 7. Comparison of a vehicle detection. (a) Detection using depth-intensity. (b) Features that contributed to the detection. The depth information filters out the features that belong to background clutter. (c) Detection with intensity information. (d) Features that contributed to the detection.

multiscale detector resulting in a set of detected bounding boxes B_{dt} and using the ground-truth bounding boxes B_{gt} we accept a detection if:

$$\alpha = A(B_{dt} \cap B_{gt}) / A(B_{dt} \cup B_{gt}) > 0.5 \quad (8)$$

where $A()$ denotes the area of the box. We associate only one detection with each ground-truth bounding box, if other detections intersect with it we count them as false positives. The output of our algorithm is a set of detection with

probabilities. By adjusting the threshold to accept a detection we obtain the precision-recall curve.

In Fig. 8, the precision-recall curves are shown for our method with and without using depth information. We can see that using depth information we have a considerable increase in the performance. Additionally, this information enables us to create a mask and discard about 75% of the image thus decreasing the computational cost. As can be seen from the precision-recall curves, the challenging nature of the data-set poses difficulties for both methods. In particular, cars with poor illumination are difficult to detect with features based on image gradients. Using other type of features (e.g. based on shape) that perform better under poor illumination is expected to increase the performance. The variability in the scales of the objects is another factor that meets the limits of the used descriptors considering that they were trained using the UIUC database. This database contains cars from a single scale. Additionally, the American cars contained in the UIUC data-set have a different shape from the European cars that we have in our data-set. Nevertheless, as shown in [7], most of the state-of-the-art methods experience great difficulties in data-sets of this type (captured from a moving platform, urban environment). Under these circumstances however the increase in performance using depth information is significant. For instance the proposed method detects about one third of the vehicles, with 60% precision while the method using only intensity cannot even achieve this recall rate.

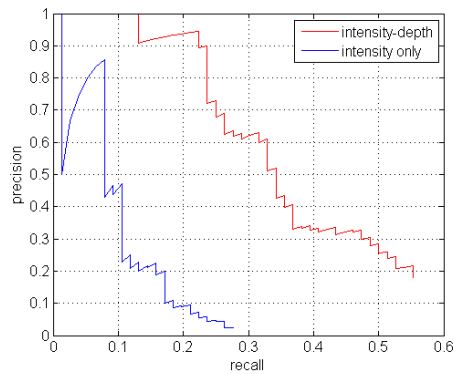


Fig. 8. Precision-Recall curves for the method using depth-intensity compared with the method using only intensity.

V. CONCLUSIONS

In this work we presented a method that fuses intensity with depth information to create a robust part-based detector. We applied the method to create a system for car detection from a moving vehicle. We tested it in a real urban environment using a data-set collected from our experimental platform. The comparison with the system using only intensity information shows a significant increase in performance.

As a first future work we consider using the stereo images dataset to train the system with intensity and depth information. This way we will be able to better estimate the parameters for the calculation of the depth likelihood. We will also be able to test the system with new types of features extracted from the depth images. As another future extension we consider to use the output probability densities of several detectors to do higher level reasoning in order to disambiguate between different object type detections for the same image region. Depth information, can be beneficial in such situations because it facilitates the reasoning in cases of occlusions.

ACKNOWLEDGMENT

This work has been done within the framework of the ArosDyn Project of INRIA. The authors thank Toyota Motor Europe for their support on our experimental work.

REFERENCES

- [1] M. Awais and K. Mikolajczyk, "Feature pairs connected by lines for object recognition," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, 2010, pp. 3093–3096.
- [2] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.
- [3] E. Seemann, M. Fritz, and B. Schiele, "Towards robust pedestrian detection in crowded image sequences," in *CVPR*. IEEE Computer Society, 2007.
- [4] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [5] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, 2006.

- [6] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, 2010.
- [7] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009, pp. 304–311.
- [8] M. Enzweiler and D. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.
- [9] M. Rohrbach, M. Enzweiler, and D. M. Gavrila, "High-level fusion of depth and intensity for pedestrian classification," in *DAGM-Symposium*, ser. Lecture Notes in Computer Science, J. Denzler, G. Notni, and H. Süße, Eds., vol. 5748. Springer, 2009, pp. 101–110.
- [10] D. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007.
- [11] J. Gall and V. S. Lempitsky, "Class-specific hough forests for object detection," in *CVPR*, 2009, pp. 1022–1029.
- [12] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 16–44, 2008.
- [13] R. Labayrade, D. Aubert, and J. Tarel, "Real time obstacles detection on non flat road geometry through v-disparity representation," in *IEEE Intelligent Vehicles Symp.*, Versailles, France, 2002.
- [14] A. Broggi, C. Caraffi, P. Porta, and P. Zani, "The single frame stereo vision system for reliable obstacle detection used during the 2005 DARPA Grand Challenge on terramax," in *Proc. of the IEEE Intelligent Transportation Systems Conf.*, Toronto, Canada, 2006.
- [15] G. Toulminet, M. Bertozzi, S. Mousset, A. Benschrair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Transactions on Image Processing*, vol. 15, no. 8, August 2006.
- [16] T. Veit, "Connexity based fronto-parallel plane detection for stereo-vision obstacle segmentation," in *IEEE Int. Conf. on Robotics and Automation, Workshop on Safe Navigation in Open and Dynamic Environments: Applications to Autonomous Vehicles*, Kobe, Japan, 2009.
- [17] B. Leibe, K. Schindler, N. Cornelis, and L. J. V. Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, 2008.
- [18] I. P. Alonso, D. F. Llorca, M. Á. Sotelo, L. M. Bergasa, P. R. de Toro, J. Nuevo, M. Ocaña, and M. A. G. Garrido, "Combination of feature extraction methods for SVM pedestrian detection," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 292–307, 2007.
- [19] D. Geronimi, A. D. Sappa, A. Lopez, and D. Ponsa, "Adaptive image sampling and windows classification for on-board pedestrian detection," in *Proc. of the 5th Int. Conf. on Computer Vision Systems*, Bielefeld, Germany, 2007.
- [20] W. R. M. Mahlich, R. Schweiger and K. Dietmayer, "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection."
- [21] R. T. L. Spinello and R. Siegwart, "A trained system for multimodal perception in urban environments," in *IEEE Int. Conf. on Robotics and Automation, Workshop on Safe Navigation in Open and Dynamic Environments: Applications to Autonomous Vehicles*, 2009.
- [22] P. P. M. S. L. Oliveira, U. Nunes and F. Moita, "Semantic fusion of laser and vision in pedestrian detection," in *Pattern Recognition*, vol. 43, no. 10, 2010, pp. 3648–3659.
- [23] M. Perrollaz, A. Spalanzani, and D. Aubert, "A probabilistic representation of the uncertainty of stereo-vision and its application to obstacle detection," in *Proc. of the IEEE Intelligent Vehicles Symp.*, San Diego, CA, USA, 2010.
- [24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [25] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in *ECCV (1)*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951. Springer, 2006, pp. 404–417.
- [26] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang, and A. Zisserman, "Dataset issues in object recognition," in *Toward Category-Level Object Recognition*, LNCS, vol. 4170. Springer, 2006, pp. 29–48.



Session II

Perception & Situation Awareness

- **Keynote speaker: C. Laugier (Inria Grenoble, France)**
Title: Situation awareness & Risk based navigation in dynamic environments
Co-Authors: I. Paromtchik, M. Perrollaz, J.D. Yoder, C. Tay, K. Makhnacha, C. Fulgenzi, A. Spalanzani
- **Title: From Structure to Actions: Semantic Navigation Planning in Office Environments**
Authors: K. Uhl, A. Roennau, R. Dillmann
- **Title: Situation Assessment and Trajectory Planning for AnnieWAY**
Authors: C. Stiller, J. Ziegler



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Session II

Keynote speaker: **C. Laugier (Inria Grenoble, France)**

Situation awareness & Risk based navigation in dynamic environments

Co-Authors: I. Paromtchik, M. Perrollaz, J.D. Yoder, C. Tay, K. Makhnacha, C. Fulgenzi, A. Spalanzani

Abstract : This talk address the problem of safe navigation in dynamic environments, with a focus on intelligent vehicle application. After a global overview of the problem and of the state of the art, several key aspects of this problem will be addressed: Bayesian perception and sensor fusion, Motion prediction for sensed mobile obstacles (including maneuvers prediction at road intersections), Probabilistic collision risk assessment, and Risk based navigation. Results obtained with our equipped Lexus hybrid vehicle will be presented and discussed.

Biography: Dr. Christian Laugier is Research Director at INRIA and Scientific Leader of the *e-Motion* team-project. He is also responsible at the International Affairs Department of INRIA of the Scientific Relations with Asia & Oceania. He received the Ph.D degree in Computer Science from Grenoble University, France in 1976. His current research interests mainly lie in the areas of *Motion Autonomy, Intelligent Vehicles and Probabilistic Robotics*. He has co-edited several books in the field of Robotics and several special issues of scientific journals such as IJRR, Advanced Robotics, JFR, or IEEE Trans on ITS. In 1997, he was awarded the Nakamura Prize for his contributions to “Intelligent Robots and Systems”. Dr. Christian Laugier is a member of several scientific committees such as the Steering/Advisory Committees of the IEEE/RSJ IROS, FSR, and ICARCV conferences. He is also co-Chair of the IEEE RAS Technical Committee on AGV & ITS. He has been General Chair, Program Chair or co-Chair of international conferences such as IEEE/RSJ IROS’97, IROS’02, IROS’08, IROS’10, IROS’12, or FSR’07. In addition to his research and teaching activities, he co-founded four start-up companies in the fields of Robotics, Computer Vision, Computer Graphics, and Bayesian Programming tools. He also served as Scientific Consultant for the ITMI, Aleph Technologies, and Probayes companies.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Collision Risk Assessment to Improve Driving Safety

Christian Laugier*, Igor E. Paromtchik*, Christopher Tay†, Kamel Mekhnacha†,
Gabriel Othmezouri‡, Hiromichi Yanagihara‡

*INRIA Grenoble Rhône-Alpes, 38334 Saint Ismier, France

†ProBayes, 38334 Saint Ismier, France

‡Toyota Motor Europe, B-1930 Zaventem, Belgium

Abstract—Robust analysis of dynamic scenes in urban traffic environments is needed to estimate and predict collision risk level during vehicle driving. The risk estimation relies on monitoring of the traffic environment of the vehicle by means of on-board lidars and a stereo camera. The collision risks are considered as stochastic variables. Hidden Markov Model and Gaussian process are used to estimate and predict collision risks and the likely behaviors of multiple dynamic agents in road scenes. The proposed approach to risk estimation is tested in a virtual environment with human-driven vehicles and during a highway driving. The obtained results have proven the feasibility of our approach to assist the driver in avoiding potentially dangerous situations.

Index Terms—Collision risk, urban road, driver assistance, Hidden Markov Model, Gaussian process

I. INTRODUCTION

A. Problem statement

The urban traffic environment with multiple participants contains risks of potential collision and damage. The vehicle safety technologies (e.g. seat belts, airbags, safety glass, energy-absorbing frames) mitigate the effects of accidents. The advanced technologies will be capable of monitoring the environment to estimate and predict collision risks during vehicle driving, in order to help reduce the likelihood of accidents occurring. The risk management by traffic participants is an efficient way to improve traffic safety toward *zero-collision* driving. The key problem is to correctly interpret the traffic scene by means of processing information from a variety of sensors [1].

A collision risk level can be predicted for a few seconds ahead to warn the driver about unnoticed potential risks. The estimated risk of collision can also be used to select a trajectory that minimizes the risks for an autonomous vehicle. The estimation of collision risk relies on the sensor information about the surrounding environment and the driver's behavior. The obtained risk values must be interpreted by the dedicated application. The following set of processed sensor inputs is assumed to be available.

Road geometry. The road width and the road curvature are obtained by processing raw information from camera images and lidars or, alternatively, from a Geographic Information System (GIS) with a pre-built map and a localization device such as Global Positioning System (GPS).

Object tracking. The detection and tracking of moving objects is accomplished by the dedicated algorithms, i.e. positions and velocities of the objects are available.

Auxiliary sensors. Information about the signal light status of other vehicles is an indicator of the motion intention. Additional “virtual” sensors are capable of detecting distances between the vehicles and the lane borders, which might indicate an intention to perform a lane change.

We use a term “ego-vehicle” to distinguish our vehicle from other vehicles. The ego-vehicle is assumed to be equipped with the appropriate sensors for obtaining a set of inputs mentioned above. The estimated risk is a numerical value which expresses quantitatively the collision risk of the ego-vehicle with another vehicle during the next few seconds.

Estimating the future collision risk requires the models which describe the vehicle motion in the sensor visibility range of the ego-vehicle. This model must be capable of reasonably predicting the future states of the vehicle in terms of the probability. We present a fully probabilistic model of the vehicle's motion evolution for obtaining and inferring beliefs on the future states of vehicles in an urban traffic environment. Consequently, the estimated risk of collision is obtained from the models in terms of the probability in a theoretically consistent manner.

B. Related work

Current commercially available crash warning systems aim at preventing front, rear, or side collisions. Such systems are usually equipped with radar based sensors on the front, rear or sides to measure the velocity and distance to obstacles. The algorithms for determining the risk of collision are based on variants of time-to-collision (TTC) [2], giving the time remaining before one vehicle collides with another one, assuming the both vehicles are maintaining their linear velocities.

Some systems are capable of directly controlling the brakes and possibly the steering to perform the necessary corrective actions. Systems based on TTC use the observations made at a reasonably high frequency in order to adapt to a dynamic environment. Current commercial systems work reasonably well on highways or straight sections of the city roads. However, the linearity assumption does not hold on curved roads, as shown in figure 1, where the risk level tends to be overestimated.

Several research projects overcome this problem by taking into account the structure of the environment, especially at intersections where the rate of accidents is higher. These projects aim at providing the collision warning systems, which use wireless communication either between the vehicles or between the vehicle and a road infrastructure, such as traffic

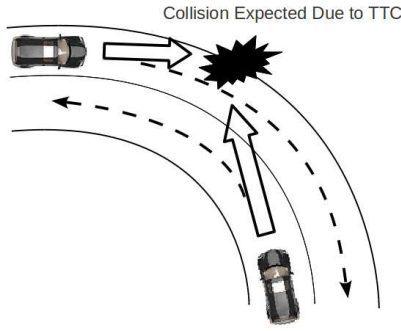


Figure 1: An example of triggering a false alarm about collision because of the invalid linearity assumption on curved roads in the TTC based systems.

lights [3], [4], [5], [6], [7]. These systems are equipped with a pair of detectors (radars or laser scanners) at the left and right front corners of the vehicle in order to detect cross-traffic vehicles at intersections. The obtained speeds of the objects and the respective TTC are then evaluated to determine the collision risk. Although the environmental structures are taken into consideration, yet the collision risk calculation assumes the straight motion. The time horizon of risk prediction is short, and the crucial environmental information and sensor data are not fully employed.

C. Outline of the approach

The knowledge about an object being at a certain location at a specific time does not provide sufficient information to assess its impact on the safety of the ego-vehicle. In addition, environmental constraints should be taken into account, especially on urban roads. We propose a framework for understanding behaviors of other vehicles and present our approach in the next section.

The relevant sensors include stereo vision, lidars, an inertial measurement unit (IMU) combined with the GPS, and odometry. The local environment is represented by a grid. The fusion of sensor data is accomplished by means of the Bayesian Occupancy Filter (BOF) [8], [9], that provides to assign probabilities of *cell occupancy* and *cell velocity* for each cell in the grid. The collision risks are considered as stochastic variables. Hidden Markov Model (HMM) and Gaussian process (GP) are used to estimate and predict collision risks and the likely behaviors of multiple dynamic agents in road scenes [10], [11].

Consider vehicle A and ego-vehicle B traveling in the same direction on the adjacent lanes, as shown in figure 2. The collision risk must be estimated for vehicle B. From the driver’s viewpoint, the road structure is implicitly described by such maneuvers as: move straight, turn right/left, or change a lane. These maneuvers are referred to as behaviors, and a set of the possible behaviors is predefined. However, some behaviors are unavailable at all instances, e.g. it might be impossible to turn left at an intersection because of the road geometry.

The lane following for a given behavior is represented by means of a GP, i.e. a probability distribution over the possible future realizations of the paths with the mean corresponding

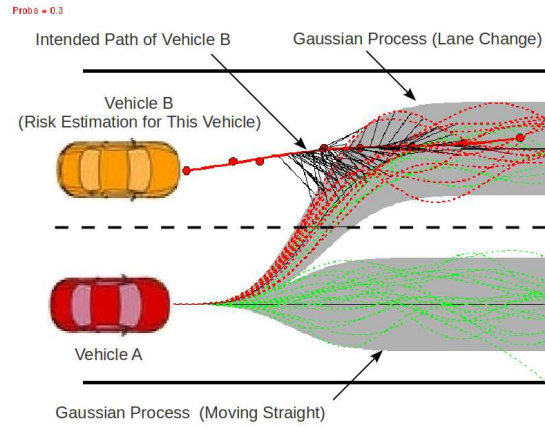


Figure 2: Collision risk estimation for vehicle B relies on predicting the path of vehicle A by sampling from the GP for two possible behaviors in this example: “moving straight” or “lane change”. The collision risk is obtained as a weighted sum of paths leading to collision.

to the exact following of the lane middle. The GP samples for such behaviors as “lane change” and “moving straight” are depicted in figure 2, where the dotted lines represent the paths sampled from the GP. For a lane turning on a curved road, the GP is adapted to reflect the road geometry. The set of GPs for each feasible behavior, in combination with the probability of vehicle A executing a certain behavior, gives a probabilistic model of the future evolution of vehicle A in the scene.

Similar to the TTC approach, the evaluation of collision risk is performed for vehicle B against vehicle A. In contrast to the TTC’s linearity assumption about the future paths for the vehicles, we evaluate the collision risk of the intended path of vehicle B against all possible paths to be taken by vehicle A. The weights are assigned according to the probabilistic model of the behaviors’ evolution of vehicle A.

II. COLLISION RISK ESTIMATION

An overall architecture of the risk estimation module is shown in figure 3. It comprises three sub-modules, such as: driving behavior recognition, driving behavior realization, and collision risk estimation [11], [12].

Driving behavior recognition. The behavior recognition aims at estimating the probability distribution of feasible behaviors, e.g. $P(\text{turn_left})$ represents the probability of turning left by the vehicle. The behaviors provide an implicit high-level representation of a road structure, which contains semantics. The probability distribution over behaviors is obtained by HMM. Our current model includes the following four behaviors: move straight, turn left, turn right, and overtake.

Driving behavior realization. The collision risk evaluation requires the road geometry. Driving behavior realization takes the form of GP, i.e. a probabilistic representation of a possible evolution of the vehicle motion for a given behavior [10].

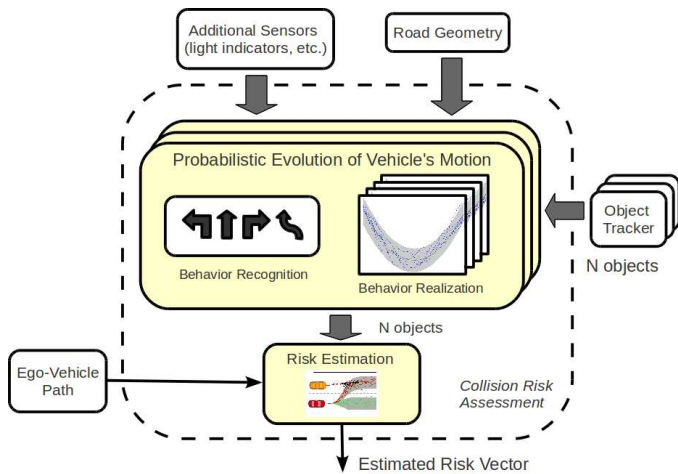


Figure 3: Architecture of the risk assessment module.

The adaptation of GP according to the behavior is based on the geometric transformation known as the Least Squares Conformal Map (LSCM) [13].

Collision risk estimation. A complete probabilistic model of the possible future motion is given by the probability distribution over behaviors from the driving behavior *recognition* and the driving behavior *realization*. The collision risk is calculated from this model. Intuitively, the result can be explained as “collision risk for a few seconds ahead”. However, the precise mathematical definition of risk depends on the meaning and interpretation of risk [11].

Behavior recognition and modeling

The behavior recognition aims at assigning a label and a probability measure to sequential data, i.e. observations from the sensors. Examples of sensor values are: distance to lane borders, signaling light status, or a proximity to an intersection. The objective is to obtain the probability values over behaviors, i.e. the behaviors are hidden variables.

The behavior modeling contains two hierarchical layers. The upper layer is a single HMM, where its hidden states represent high-level behaviors, such as: move straight, turn left, turn right, and overtake. For each hidden state or each behavior in the upper layer HMM, there is a corresponding HMM in the lower layer to represent the sequence of the finer state transitions of a single behavior, as depicted in figure 4.

Let us define the following hidden state semantics in the lower layer HMMs for each of the following behaviors of the higher layer HMM:

- *Move straight (1 hidden state):* move forward.
- *Turn left or turn right (3 hidden states):* Decelerate before a turn, execute a turn, and resume a cruise speed.
- *Overtake (4 hidden states):* lane change, accelerate (while overtaking a vehicle), lane change to return to the original lane, resume a cruise speed.

In order to infer the behaviors, we maintain a probability distribution over the behaviors represented by the hidden states of the upper layer HMM. The observations of vehicles (i.e.

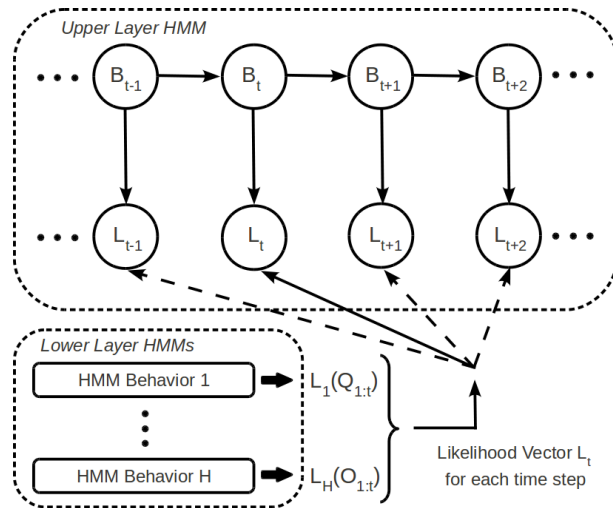


Figure 4: Layered HMM, where each lower layer HMM’s observation likelihood is the upper layer HMM’s observation.

sensor data) interact with the HMM in the lower layer, and the information is then propagated to the upper layer.

Driving behavior realization

A behavior is an abstract representation of the vehicle motion. For a given behavior, a probability distribution over the physical realization of the vehicle motion is indispensable for risk estimation. The GP allows us to obtain this probability distribution by assuming that usual driving is represented by the GP, i.e. lane following without drifting too far off to the lane sides. On a straight road, this is a *canonical* GP with the mean corresponding to the lane middle.

To deal with the variations of lane curvature or such behaviors as “turn left” or “turn right”, we propose an adaptation procedure, where the canonical GP serves as a basis and it is deformed according to the road geometry. The deformation method is based on LSCM. Its advantage is a compact and flexible representation of the road geometry. The canonical GP can be calculated once and, then, can be reused for different situations, thus, resulting in a better computational efficiency. An example is shown in figure 5 for a curved road.

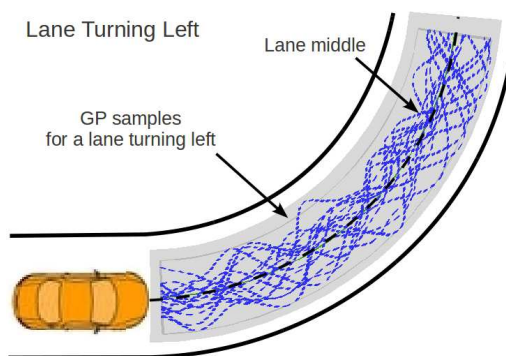


Figure 5: Deformed GP model example for a lane turning left.

Collision risk estimation

The layered HMM approach assigns a probability distribution over behaviors at each time instance, and a GP gives the probability distribution over its physical realization for each behavior. Because the behavioral semantics are propagated from the layered HMM down to the physical level, it is now possible to assign semantics to risk values.

One should note that the definition of risk can take a variety of forms, which is largely dependent on how the risk output is going to be used. A risk scalar value might be sufficient for a crash warning system, or an application might require the risk values against each vehicle in the traffic scene.

The risk calculation is performed by first sampling of the paths from the GP. The fraction of samples in collision gives the risk of collision, which corresponds to the behavior represented by the GP. A general risk value is obtained by marginalizing over behaviors based on the probability distribution over behaviors obtained from the layered HMM. It is possible to calculate risk of taking a certain path, a certain behavior, or a general risk value of a certain vehicle against another vehicle. A systematic framework for evaluation of different types of risk can be found in [11].

III. EXPERIMENTS

A. Driving simulation

The simulation of crash situations in a virtual environment is used instead of dealing with them in real experiments. The virtual environment is a 3D geometric model of a road network with vehicles, where each vehicle is driven by a human driver. The simulator was developed by Toyota Motor Europe (TME). Each human driver controls his or her virtual vehicle by means of a steering wheel, the acceleration and brake pedals. Recording a scenario with multiple vehicles, which are driven concurrently, requires a large number of human drivers. An alternative is to generate the scenario iteratively, with one human-driven vehicle at a time and “adding” human drivers iteratively, with a replay of the previously recorded human-driven vehicles. The resulting virtual environment allows us to simulate crash situations safely.

The layered HMM evaluates the behavior of every vehicle in the scene for different time horizons, except the ego-vehicle. The training data are obtained by collecting sequences for a series of human-driven cases, where each driver uses the steering wheel as an interface to the virtual environment of the simulator. The driving sequences are then annotated manually by means of an annotation tool of ProBayes. Subsequently, the annotated data are used to train the layered HMM.

The TME simulator provides a 3D road view for the driver and a 2D view of the road network, as shown in figure 6. The collision risk is calculated for a yellow vehicle, while other vehicles are shown by red rectangles. The relevant area of the scene is inside a large yellow circle. The right-hand traffic rule is assumed. The trail behind the yellow vehicle in 2D view indicates the risk levels estimated previously. At each instant, the probabilities of the possible behaviors of the nearest neighbor (red vehicle) are estimated by the layered

HMM and are displayed by the vertical white bars. The speed of the yellow vehicle is shown in 3D view, where the right-side vertical bar shows the risk encoding by color from “low” (green) to “high” (red). The left-side vertical bar in 3D view indicates the current risk value for the yellow vehicle.



Figure 6: Virtual environment of the TME simulator.

The speed warning in the case of a potential danger of frontal collision is available in most commercial systems. Additionally to this functionality, our algorithm evaluates risk at intersections, where the linearity assumption about the vehicle motion would result in underestimated values of collision risk. The combination of the behavior estimation by the layered HMM and the use of semantics (e.g. turn right or move straight) at the geometric level allows us to obtain the appropriate risk values.

The training data for the layered HMM were collected with ten human drivers who were asked to show different driving behaviors. The collected data is split uniformly distributed into the training data and the test data (30% of total data examples). The behavior recognition is trained on the training data and is evaluated against the test data.

Figure 7 summarizes the recognition performance of the layered HMM. The results are presented as a confusion matrix, where the columns correspond to the true class and the rows correspond to the estimated class. The diagonal values of the confusion matrix give the correctly predicted class, while non-diagonal values show the percentage of mislabeling for each class. The highest recognition rate is for “move straight” behavior (91.9%) as well as “turn right” or “turn left” ones (82.5% and 81.1%, respectively). The “overtake” behavior has a relatively low recognition rate of 61.6%. Intuitively, this lower rate can be explained by a composite structure of the overtaking maneuver because it consists of such behaviors as: accelerating, lane changing, returning to the original lane, and resuming a cruise speed. Consequently, it also takes longer than a three-second period (current prediction horizon) to complete an overtaking maneuver.

The approach to risk assessment is illustrated by figure 8, where the probability of collision is estimated for a period of three seconds ahead of each collision for ten different traffic scenarios. The rapid increase in the probability of collision and its certainty are observed when the collision instant approaches.

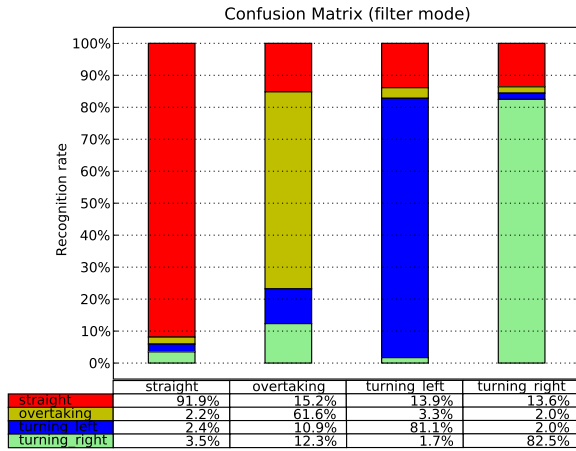


Figure 7: Performance summary of the behaviors recognition with layered HMM.

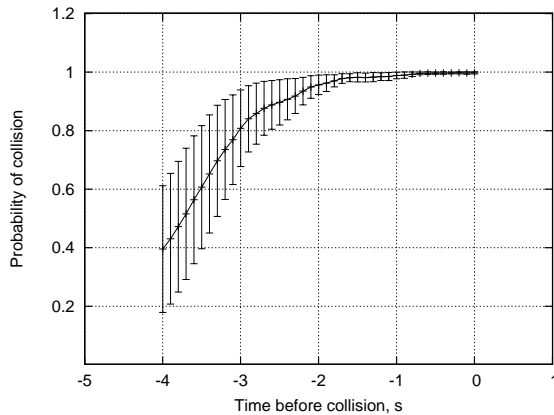


Figure 8: Example of collision risk assessment for ten human-driven scenarios and a three-second prediction horizon.

B. Behavior estimation on a highway sequence

The first phase is to gather experimental data when driving on a highway to estimate behaviors of other vehicles. The experiments have been conducted jointly by the TME and ProBayes. The data acquisition was performed for four scenarios on a highway, with each scenario lasting for ten minutes approximately and the sensor data (stereo camera images, vehicle odometry, and GPS information) being recorded. The behaviors to be estimated are: move straight, a lane change to the left, and a lane change to the right. An example of the behavior estimation on a highway is shown in figure 9.

The detection of vehicles is performed by clustering of the disparity points obtained from the stereo camera mounted behind the windshield. The clustering is performed in the image areas, which are indicated by the image based detection using support vector machines (SVMs). The positions of vehicles are tracked on the road plane by means of the BOF [8], [9].

The observation variables for behavior recognition include

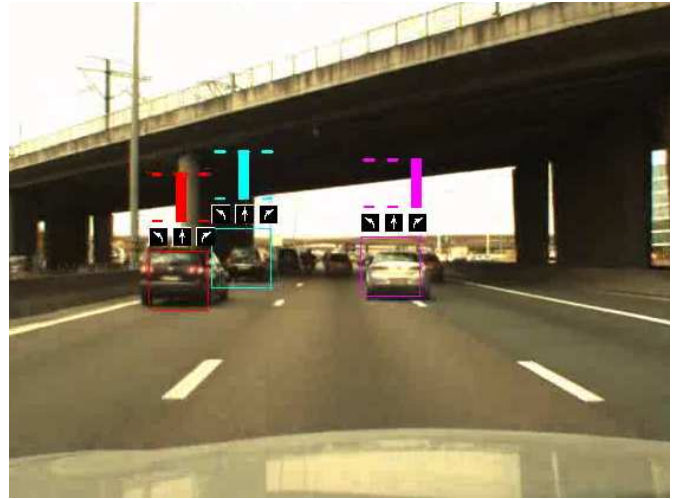


Figure 9: Example of a highway scenario where a vehicle on the middle lane performs a lane change to the right.

the vehicle’s speed, the distances to the lane borders, and the information about the presence of other vehicles on the adjacent lanes. In order to obtain the observation variables in a global reference frame, a particle filter is used for localizing the vehicle on the highway map obtained from the Geographic Information System (GIS). The particle filter allows us to estimate the position and direction of the vehicle at each time instant and to employ the observations from stereo-vision (lanes detection), GPS and vehicle odometry. A similar approach is used for the training phase, when the acquired data are divided into the training and evaluation sets annotated manually to indicate the current behavior for each time instance of the data acquired.

An example of the behavior estimation on a highway is shown in Fig. 9. The positions of the tracked vehicles are projected onto the image plane and are represented by the rectangles. The probability distribution of the estimated behaviors is shown by the height of the color bars above the vehicles, e.g. the “lane change to the right” behavior of the vehicle on the middle lane and the “move straight” behavior of the two vehicles on the left lane are evaluated correctly. These results illustrate the validity of the proposed approach for behavior estimation. The different probability decomposition of the observation variables, the selection of the observation variables and the reactivity of the behavior estimation are topics of our ongoing work to generalize the approach.

The results on behavior estimation are currently preliminary. The current work involves experimenting with different probability decomposition of observation variables and observation variable selection. Furthermore, for purposes of risk evaluation, we will also be able to evaluate the reactivity of behavior estimation. The ongoing work will allow us to generalize the results and evaluate them quantitatively.

IV. CONCLUSION

Collision risk estimation and prediction will be mandatory for future vehicles. A fraction of a second of the driver's reaction time can help save human lives. Our data processing approach, sensor models and software modules allow us to monitor the urban traffic environment. The analysis and interpretation of traffic scenes rely on evaluation of driving behaviors as stochastic variables to estimate and predict collision risks for a short period ahead. Our initial experiments on behavior estimation during vehicle driving allowed us to verify the validity of the approach. Our future work will deal with its integration and evaluation on a Lexus vehicle equipped with sensors and shown in figure 10.



Figure 10: Experimental platform on a Lexus LS600h equipped with two IBEO Lux lidars, a TYZX stereo camera, and an Xsens MTi-G inertial sensor/GPS.

REFERENCES

- [1] I. E. Paromtchik, C. Laugier, M. Perrollaz, A. Nègre, M. Yong, and C. Tay, "The ArosDyn project: Robust analysis of dynamic scenes," in *Int. Conf. on Control, Automation, Robotics, and Vision*, (Singapore), December 2010.
- [2] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.
- [3] J. Pierowicz, E. Joco, M. Lloyd, A. Bittner, and B. Pirson, "Intersection collision avoidance using its countermeasures," Tech. Rep. DOT HS 809 171, NHTSA, U.S. DOT, 2000.
- [4] "Vehicle-based countermeasures for signal and stop sign violation," Tech. Rep. DOT HS 809 716, NHTSA, U.S. DOT, 2004.
- [5] K. Fuerstenberg and J. Chen, "New European approach for intersection safety - results of the EC project INTERSAFE," in *Proc. International Forum on Advanced Microsystems for Automotive Application*, 2007.
- [6] S. Lefèvre, J. Ibanez-Guzmán, and C. Laugier, "Context-based prediction of vehicle destination at road intersections," in *Proc. of the IEEE Symp. on Computational Intelligence in Vehicles and Transportation Systems*, (France), 2011.
- [7] S. Lefèvre, J. Ibanez-Guzmán, and C. Laugier, "Exploiting map information for driver intention estimation at road intersections," in *Proc. of the IEEE Intelligent Vehicles Symp.*, (Germany), 2011.
- [8] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: An automotive application," *Int. J. Robotics Research*, no. 1, 2006.
- [9] M. K. Tay, K. Mekhnacha, C. Chen, M. Yguel, and C. Laugier, "An efficient formulation of the Bayesian occupation filter for target tracking in dynamic environments," *Int. J. Autonomous Vehicles*, vol. 6, no. 1-2, pp. 155–171, 2008.
- [10] C. Tay and C. Laugier, "Modelling smooth paths using Gaussian processes," in *Proc. of the Int. Conf. on Field and Service Robotics*, 2007.

- [11] C. Tay, *Analysis of Dynamics Scenes: Application to Driving Assistance*. PhD Thesis, INRIA, 2009.
- [12] C. Laugier *et al.*, "Vehicle or traffic control method and system," *Patent application no. 09169060.2-1264*, August 2009.
- [13] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," in *ACM SIGGRAPH conference proceedings*, 2002.



Session II

Perception & Situation Awareness

- **Title: From Structure to Actions: Semantic Navigation Planning in Office Environments**
Authors: K. Uhl, A. Roennau, R. Dillmann
- **Title: Situation Assessment and Trajectory Planning for AnnieWAY**
Authors: C. Stiller, J. Ziegler



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

From Structure to Actions: Semantic Navigation Planning in Office Environments

Klaus Uhl and Arne Roennau and Rüdiger Dillmann

Abstract—The use of meaning in mapping and navigation is inevitable if a robot has to interact with its environment in a goal-directed way. Moreover, a semantic environment model makes navigation planning more efficient and simplifies the review and communication of the robot’s knowledge. Existing work in this area decomposes the environment into places, which can be distinguished using the robot’s sensors. However, if important features of the environment cannot be detected by the robot’s sensors a different approach is needed.

This paper introduces the *Semantic Region Map*, an environment model with complex metric, topological and semantic features. It shows how navigation points, so-called *semantic positions*, can be deduced from the map using a semantic description of the environment. Furthermore, the semantic positions are connected to a reachability graph, whose edges are labelled with robot actions, using a semantic description of the robot’s capabilities. An ontology consisting of a taxonomy and a set of rules are used to implement the semantic models.

The concept of the *Semantic Region Map* is applied to a robot operating in an office environment.

I. INTRODUCTION

If a service robot has to interact with its environment in a goal-directed way, the use of meaning in mapping and navigation is inevitable. This is especially true if a service robot is designed to assist humans in everyday tasks. The surroundings in which humans live and work are usually divided into discrete spatial regions, such as corridors, offices, bedrooms etc. A robot which can reason about the meaning and relations between those regions is able to more easily and naturally communicate with the people it has to assist as it can understand commands like “bring this batch of letters to the secretary of the public relations department” (cf. [1]). Aside from that, a semantic environment model makes navigation planning more efficient, navigation execution more robust and simplifies the review and communication of a robot’s knowledge.

Existing work in this area decomposes the environment into places which can be distinguished using the robot’s sensors and uses those places as navigation points for the robot. However, taking the price, dimensions and weight of sensors into account, there will always be important features in an environment which cannot be detected by the robot because it is not equipped with enough sensors to detect them.

If a robot is able to detect and classify regions, perceive relations between the detected regions and determine relations between its own position and the detected regions,

K. Uhl, A. Roennau and R. Dillmann are with FZI Research Center for Information Technology, Intelligent Systems and Production Engineering (ISPE), 76131 Karlsruhe, Germany {uhl, roennau, dillmann}@fzi.de

more flexibility is possible. Navigation points can suddenly be independent of distinguishable places and can be located in a density only limited by the granularity of distinguishable relations. Additionally, regions which cannot be detected by the robot’s sensors can be handled indirectly via inference. The dense navigation points, in turn, give the planner fine control over the motion behaviour of the robot on a semantic level.

This paper introduces the *Semantic Region Map* as the basis for abstract, semantic navigation planning for robots operating in indoor environments. It shows how an environment can be modelled using complex region features consisting of metric, topological and semantic information. It shows how the *Semantic Region Map* can be combined with a generic region algebra and a semantic model of a concrete environment to deduce abstract navigation points, so-called *semantic positions*, from the map. By adding a semantic model of a concrete robot, the semantic positions can be connected to a reachability graph whose edges are labelled with the actions the robot has to perform in order to move from one semantic position to the next. This gives the planner fine control over the exact robot behaviour along its path. Using this semantically enriched environment model, planning a navigation path is reduced to determining the current and goal semantic positions of the robot using queries to the ontology, extracting the reachability graph and finding the shortest path between the two semantic positions.

This paper is organised as follows. First, we briefly describe related work and describe the semantic mission control system to which this work belongs. Then we introduce the semantic navigation planning concepts, followed by a general procedure for modelling an environment and a robot for a specific application. We apply the modelling procedure to a robot operating in an office environment and show experimental results. Finally, we conclude and give an outlook to future work.

II. RELATED WORK

Belouaer et. al. [2] describe an ontology-based, semantic representation of spatial entities, spatial relations and imprecise spatial information. Spatial entities are modelled as axis-aligned rectangles and an algebra of topological relations allows to deduce relations between distant entities. Although the system is designed to support path planning, path planning is limited to the purely geometric level and the system cannot handle different driving strategies like wall following, door traversal and straight driving as navigation actions.

Galindo et. al. [3] have developed a semantic map framework in which spatial information is anchored to semantic labels which are, in turn, connected to a conceptual ontology of the environment. The system is tailored to deriving the existence of spatial entities which have not yet been seen and to refine the classification of spatial entities by deduction. A semantic-level planning algorithm uses the semantic map and the conceptual ontology to start planning on the conceptual level. Motion planning, however, is restricted to moving the robot from one spatial area to the next without fine control over the actual motion behaviour.

Guillon and Farges [4] combine a general task planner with a specialised path planner to a hybrid mission planning system. Navigation tasks are modelled as preconditions to other actions. As there can be behavioural as well as geometric constraints for the path planner, it is possible to enforce a specific driving behaviour. However, this behaviour cannot be tied to a single spatial area and cannot be changed in different areas.

Mozos et. al. [5] propose a multi-hierarchical map which links a metric map, a topological navigation map, a topological area map and a conceptual map. The ontology which backs the conceptual map has similar deduction capabilities and limitations as the work of Galindo et.al. [3].

Shi et. al. [6] propose an algorithm to create a semantic grid map from laser range data. Each cell of the grid map is semantically labelled to be either a room, corridor or doorway. By using a grid map Shi et. al. are able to classify subregions of a single laser scan to different semantic classes. However, they do not currently use their maps for navigation planning.

III. SYSTEM CONTEXT

The semantic navigation planning system described in this paper is part of a larger semantic mission control system [7]. The system architecture is shown in Fig. 1. It consists of nine modules in four layers which are distinguished by the kind of data processed.

The *semantic level* consists of the *User Interface* which communicates with the system's user. It also contains the semantic navigation planning system (*Semantic Navigation*).

The *symbolic-semantic level* contains the *Semantic Mapping* which computes and updates the *Semantic Region Map* of the environment. It uses a semantic SLAM algorithm with complex features that capture metric, topological and semantic properties [8]. Also located on this level are the *Semantic Localisation* which determines and tracks the robot's current semantic position and the *Execution Unit* which decomposes plans from the *Semantic Navigation* into individual symbolic actions and monitors the plan's execution.

The *subsymbolic-symbolic level* contains the *Navigation Data Analysis* and the *Basic Control*. The *Navigation Data Analysis* continuously locates and classifies regions in the robot's sensor data and determines their parameters and relations. The *Basic Control* receives a single symbolic action from the *Execution Unit* at a time, passes it as subsymbolic commands to the sensor and actor interfaces and monitors

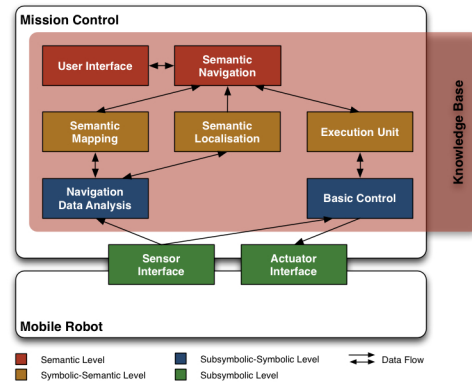


Fig. 1. The semantic navigation planning system is integrated into a semantic mission control system.

its execution. The *subsymbolic level* contains the sensor and actor interfaces to the robot.

The implementation of the mobile robot is mostly independent of the semantic mission control. In the case of our mobile research robot *Odete* (see Fig. 4) and our autonomous shopping trolley *InBot* (see Fig. 5), behaviour-based robot control systems have been implemented. They are capable of executing a set of complex behaviours which are mapped to subsymbolic commands in the actor interface. The detection and tracking of dynamic and semi-dynamic obstacles is also implemented in the robot control software as it has to be tightly integrated with the robot's safety functions. Several different obstacle avoidance behaviours, which can be activated independently, use this tracking information to safely navigate in crowded environments.

IV. SEMANTIC NAVIGATION PLANNING

The semantic navigation planning system consist of two parts: an *ObjectLogic* [9] ontology, which contains knowledge about the application domain, the robot and the environment, as well as a planner, which extracts knowledge from the ontology and creates navigation plans.

A. Semantic Region Map

The first major concept of the semantic navigation planning system is the *Semantic Region Map*. It segments an environment into a set of regions with metric, topological and semantic features. Each region is an instance of a subclass of the *Region* concept in the ontology. The region class represents the semantic meaning of a region. Regions are topologically connected to their neighbours via the *neighbourOf* relation. They can also be fully contained in other regions, in which case they are connected via the *containedIn* relation. Additionally, the relative orientation of neighbouring regions is specified via one of the four relations *northOf*, *eastOf*, *southOf* and *westOf*.

The metric feature of a region depicts its approximate geometric extent in a global coordinate system. It consists of a centre rectangle and two connected sub-rectangles, which can be moved along the left and right edges of the centre rectangle. Therefore the region geometry can be described

via the tuple $(x, y, w, h, \gamma, w_l, h_l, y_l, w_r, h_r, y_r)$. The left and right sub-rectangle can be omitted if they are not needed to describe a region's geometry. In this case the values w_l, h_l and y_l or the values w_r, h_r and y_r are set to 0. The region geometry is attached to a region instance in the ontology via the `hasShape` relation.

The *ObjectLogic* specification for the `Region` concept is defined as follows:

```
Region[eastOf{0:*,inverseOf(westOf)} *=> Region,
northOf{0:*,inverseOf(southOf)} *=> Region,
southOf{0:}* *=> Region, westOf{0:}* *=> Region,
hasShape{1:1} *=> Shape, containedIn{0:}* *=> Region,
neighbourOf{0:*,symmetric} *=> Region].
```

The semantic navigation planning system expects a *Semantic Region Map* as its input. This map has to specify the regions in the environment, their shape as well as the `neighbourOf` and `containedIn` relations. For regions which are connected with a `neighbourOf` relation, it must also specify one of the `northOf`, `eastOf`, `southOf` and `westOf` relations.

B. Semantic Positions

The second major concept of the semantic navigation planning system are *semantic positions*. They are fuzzy navigation points that are defined by semantic relations to regions in their surroundings.

Modelling the semantic positions for an application domain is a complex task. It can become even more tedious when different orientations of regions and neighbouring regions have to be considered because the number of possible combinations explodes. To counteract this, each region has a local coordinate system which is rotated against the global coordinate system of the *Semantic Region Map* according to the orientation of the region. Most relations between a region and its implied semantic positions are specified in this local coordinate system.

The *ObjectLogic* specification for the `SemanticPosition` concept is defined as follows:

```
SemanticPosition[
impliedBy{0:*,inverseOf(implies)} *=> Region,
inRegion{0:}* *=> Region,
spAtStartOf{0:}* *=> Region,
spAtCentreOf{0:}* *=> Region,
spAtEndOf{0:}* *=> Region,
near{0:}* *=> Region, visAVis{0:}* *=> Region,
localEastOf{0:}* *=> Region,
localNorthOf{0:}* *=> Region,
localSouthOf{0:}* *=> Region,
localWestOf{0:}* *=> Region,
local(East|North|South|West)SideOf{0:}* *=> Region,
local(East|North|South|West)mostIn{0:}* *=> Region,
local(East|North|South|West)mostAlong{0:}* *=> Region,
neighbourOf{0:*,symmetric} *=> SemanticPosition].
Region[implies{0:}* *=> SemanticPosition].
```

These relations specify the relative orientation (north, east, south or west of a region) and location (at the start, centre or end of a region) of semantic positions. They also define if semantic positions are in a region, outside but near a region or vis-à-vis. And they form local neighbourhood graphs between the semantic positions that are implied by the same region.

C. Semantic Navigation Algebra

The semantic navigation system contains an algebra, i. e. a set of rules in the ontology which performs calculations that reduce the complexity of the application domain model and the number of facts that have to be explicitly asserted in the *Semantic Region Map*.

Although the region geometry contains a rotation angle, most of the time we only deal with a discrete set of semantic orientations. The ontology, therefore, introduces the `Orientation` concept and the four orientation instances `East`, `North`, `South` and `West`. The semantic navigation algebra derives the semantic orientation of each region from its rotation angle by assigning a 90° segment to each orientation instance.

As the relations between semantic positions and the regions by which they are implied are specified in the regions' local coordinate system, the semantic navigation algebra converts them into the global coordinate system. If semantic positions have relations to other regions, the algebra converts from the global coordinate system back into the local coordinate systems of those regions. In order to make the modelling of robot actions easier, the semantic navigation algebra also contains rules which connect the semantic positions to a global neighbourhood graph.

D. Robot Actions

The final component is the model of robot actions. It is built around two concepts in the ontology. The actions which the robot in a specific application domain can perform are modelled as instances of the `Action` concept. The `Reachability` concept is used to connect pairs of neighbouring semantic positions with actions using the ternary `ReachableByAction` function symbol, thus generating a directed *reachability graph* labelled with robot actions:

```
Action[].
Reachability[].
ReachableByAction(?StartSP, ?EndSP, ?Action):Reachability
```

E. Navigation Planning

The *Semantic Region Map*, the model of semantic positions, the semantic navigation algebra and the model of robot actions yield a reachability graph by deduction through the ontology system. This graph is extracted from the ontology by retrieving all semantic position instances and all instances of the `Reachability` concept. The semantic positions form the nodes of the graph while the reachability instances form the edges, labelled with the action to be performed. A weight is assigned to each edge by calculating the Euclidean distance between the approximate coordinates of the start and end semantic positions.

Navigation goals are specified as a set of relations between the desired target semantic position and regions in its surroundings. Using the reachability graph, navigation planning consists of the following steps:

- 1) Determine the current semantic position of the robot by querying the ontology with the set of relations to regions the robot has currently detected.

- 2) Determine the target semantic position by querying the ontology with the set of goal relations.
- 3) Plan the shortest path between the current and target semantic position in the reachability graph.

Navigation goals may be ambiguous and yield multiple semantic positions when querying the ontology. In this case it is assumed that reaching any of the resulting semantic positions achieves the goal, so the nearest of them is taken.

V. MODELLING METHODOLOGY

When designing the domain ontology for a specific application (i.e. environment and robot) a number of steps have to be performed. First of all, the navigation actions which the robot can perform have to be added to the ontology as instances of the `Action` concept. Secondly, the relevant region classes that occur in the environment have to be identified. They have to be added to the ontology as subclasses of the `Region` concept. If applicable, e.g. if regions of a specific class are always longer than wide, a preferred orientation of the local coordinate system has to be defined for some region classes.

Then the interesting navigation points and the conditions in which they are relevant have to be determined. This is done following a three-step procedure:

- 1) Identify interesting navigation points for each region class.
- 2) For each pair of region classes and each possible topological relation of the two, identify additional navigation points that are of interest in this special combination.
- 3) Determine which navigation points generated by the same region should be considered neighbours.

For each identified interesting navigation point a rule which derives a semantic position has to be added to the ontology. The body of this rule has to contain the condition under which the semantic position should be derived. The head of the rule has to assert a semantic position with a unique name. A unique name can be created by choosing a unique function symbol and adding the region from which the semantic position is implied as a function argument. The head of the rule also adds relations to regions and other semantic positions (see Sec. VI for an example).

Finally, the robot actions have to be considered in order to connect the semantic positions to form a reachability graph. This follows a procedure similar to identifying the interesting navigation points:

- 1) For each region class look at the implied semantic positions and check if the robot can move between two adjacent semantic positions with a specific action.
- 2) For each pair of region classes look at the implied semantic positions and check if the robot can move between two adjacent semantic positions with a specific action.
- 3) For each action check if there are generic conditions in which the robot can use this action to reach an adjacent semantic position.

Each identified reachability rule has to be added to the ontology. The rules have to assert a `Reachability` individual in their head, using the ternary `ReachableByAction` function symbol.

VI. MODELLING AN OFFICE ROBOT APPLICATION

To validate the semantic navigation system and the modelling methodology, an office robot application has been chosen. The target platform is our mobile research robot *Odeto*, but the model can be easily transferred to any robot that can execute the same abstract actions. *Odeto*'s task is to navigate through an office environment conducting transports. To make the robot's behaviour more predictable for people in the office, the robot should always stick to the right wall in the direction of travel when driving in corridors.

Following the methodology of Sec. V we first list the actions which the robot can perform:

```
DriveStraight:Action.      TurnFromDoor:Action.
FollowWall:Action.        TurnToDoor:Action.
TransitDoor:Action.
```

We assume that the `FollowWall` action is able to follow walls around corners, although we could easily factor this behaviour out into a separate action if the robot implementation would require it.

Now, we identify the relevant region classes that occur in the office environment:

```
Corridor::Region.      Door::Region.      Room::Region.
```

We define that the longer sides of doors and corridors have to face north in their local coordinate systems.

The next step is to identify interesting navigation points. For rooms we want to have a single navigation point inside the room, which means "the robot is somewhere in the room". For doors, the robot has to be able to traverse the door. Therefore we need a navigation point at the centre of each side of the door. We also want to be able to stop the robot at the beginning and end of a doorway. Corridors need no additional navigation points as the navigation points that are derived from doors are located in the adjacent regions.

Next, we look at each pair of region classes. The only combination that is of interest here is a door at the side of a corridor. As the robot should always drive along the right wall in corridors it must be able to turn to a door from the opposite side. Therefore, if a door is at the side of a corridor we place two additional navigation points at the start and end of the doorway vis-à-vis the door.

Now, we can add rules to the ontology, which assert the identified navigation points as semantic positions with appropriate relations in their head. The condition for deriving the navigation points goes into the body of these rules. The following rule derives the three semantic positions east of a door. Similar rules have to be written for the other interesting navigation points.

```
SP1(?Door):SemanticPosition[impliedBy->?Door,
    localEastOf->?Door, near->?Door, atStartOf->?Door,
    inRegion->?Other, neighbourOf->SP2(?Door)] AND
SP2(?Door):SemanticPosition[impliedBy->?Door,
```

```

localEastOf->?Door, near->?Door, atCentreOf->?Door,
inRegion->?Other, neighbourOf->?SP3(?Door)] AND
SP3(?Door):SemanticPosition[impliedBy->?Door,
localEastOf->?Door, near->?Door, atEndOf->?Door,
inRegion->?Other]
:- ?Door:Door[neighbourLocalEast->?Other:Region].

```

Having identified the relevant semantic positions, the next step of the modelling methodology is to look at the robot actions. First, we look at each region class and their implied semantic positions. We find that doors have to be traversed using the `TransitDoor` action. Therefore the semantic positions at the centre of each side of the door have to be connected in both directions. We further have defined that the `FollowWall` action should always be used when the robot drives along the right side of a corridor. If a door is at the end of a corridor the semantic positions at its side also have to be connected to the last semantic positions at the corresponding sides of the corridor using `FollowWall`.

In the second step we have to look at each pair of region classes and their implied semantic positions:

- If the robot is in a room and needs to pass through a door, we define that it has to drive to the door using the `TurnToDoor` action, first.
- If the robot has entered a room through a door, it should drive further into the room using the `DriveStraight` action.
- If the robot turns left after having traversed a door into a corridor, it has to cross the corridor and proceed along the opposite wall. This is accomplished by performing the `TurnFromDoor` action.
- If the robot drives in a corridor and has to traverse a door on the opposite side of the corridor, it has to cross the corridor using the `TurnToDoor` action.

All identified reachability rules have to be added to the ontology. The following rule connects semantic positions along the east side (in the corridor's local coordinate system) of a corridor with the `FollowWall` action:

```

ReachableByAction(?StartSP,?EndSP,FollowWall):Reachability
:- ?Corridor:Corridor AND
?StartSP:SemanticPosition[inRegion->?Corridor,
localEastSideOf->?Corridor] AND
?EndSP:SemanticPosition[inRegion->?Corridor,
localEastSideOf->?Corridor, neighbourOf->?StartSP] AND
LocalNorthAlongRegion(?EndSP, ?StartSP, ?Corridor).

```

VII. EXPERIMENTS

To test the validity of our models we conducted several experiments with two different maps. The first map represents a very simple, artificial office environment (see Fig. 2(a)). This simple map has been used to validate and visualise the individual conceptual steps the semantic navigation system performs.

Fig. 2(b) shows the deduced semantic positions. Remark that the semantic positions are not characterised by their geometric position, although the visualisation might suggest otherwise. Fig. 2(c) shows how the model of the robot's actions connected the semantic positions to a reachability graph. Notice that the arrows along the right side of the corridor point upwards while the arrows along the left side

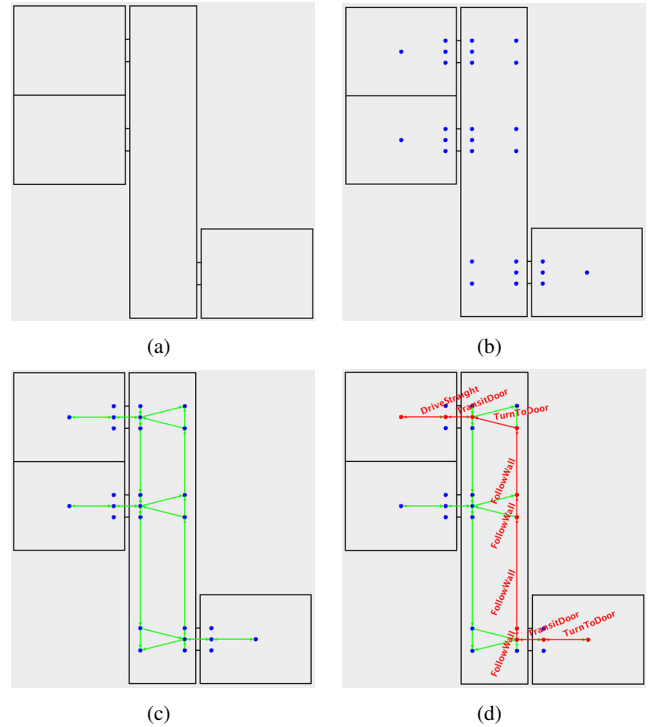


Fig. 2. Experiments with a *Semantic Region Map* of a simple office environment. (a) The *Semantic Region Map* models the environment in an abstract way. (b) Semantic positions have been implied. (c) The semantic positions have been connected with actions to a reachability graph. (d) A path has been planned from the bottom right room to the top left room. Its edges are labelled with the actions the robot has to perform.

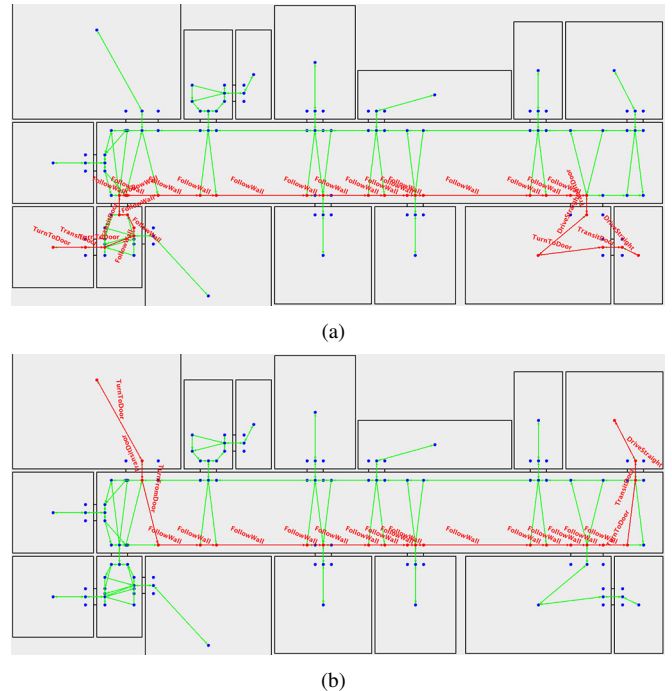


Fig. 3. Experiments with a *Semantic Region Map* of a more complex office environment.

of the corridor point downwards. Therefore, the robot always drives along the right side of the corridor.

Finally, Fig. 2(d) shows a path that has been planned by the semantic navigation system from the bottom right room to the top left room. The robot starts by turning to the door and passing through it. Then the robot follows the eastern wall of the corridor until the south end of the target room's door. It turns to the door thereby crossing the corridor, passes through the door, and lastly drives straight into the room.

We also conducted experiments with the *Semantic Region Map* of a larger office environment. Fig. 3(a) and 3(b) show two paths that have been planned by the semantic navigation system, along with the reachability graph. A remarkable result of our model can be seen in Fig. 3(a) in the small corridor on the bottom left side of the map: The robot strictly adheres to the “always drive right” policy, although one might argue that it would be more efficient to drive straight between the two doors in this situation. This could be achieved by introducing a *NarrowCorridor* region class and modelling semantic positions and robot actions accordingly.

VIII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

This paper introduced the *Semantic Region Map*, an environment model with complex metric, topological and semantic features. It presented how navigation points, so-called *semantic positions*, could be deduced from the map using a semantic description of the environment. Furthermore, it showed how a semantic description of the robot's actions can be used to connect the semantic positions to a *reachability graph*, whose edges were labelled with robot actions. An ontology consisting of a taxonomy and a set of rules was used to implement the semantic models. The paper introduced a methodology to model concrete environments as well as robot actions. It applied this methodology to a service robot operating in an office environment. Experiments with the map of a small office showed that the reachability graph was deduced as expected and that paths could be planned by determining the current and goal semantic positions using abstract queries to the ontology, extracting the reachability graph and finding the shortest path between the two semantic positions. Further experiments with a more complex map showed that the approach scales well.

B. Future Works

In future work we will extend the model of the office robot application with additional region classes and actions in order to make it more capable and flexible. It will also be possible to mark regions as “not passable” so that temporarily blocked regions (e.g. closed doors) can be handled. Additionally, we will integrate the semantic navigation planning system with the semantic SLAM algorithm from [8] and a semantic localisation system. This will enable us to test the entire loop from mapping to path planning to path execution on our mobile research robot *Odete* (see Fig. 4).



Fig. 4. The mobile research robot *Odete* carrying letters and soda.



Fig. 5. The autonomous shopping trolley *InBot*.

We are also planning to port the semantic navigation system to *InBot*, our autonomous shopping trolley (see Fig. 5), and an automatic guided vehicle (AGV), which transports goods in hospitals.

Moreover, we will add dynamic obstacles to the *Semantic Region Map* by introducing a *DynamicObject* concept into the ontology. This information will be used in the planner to adjust the driving behaviour of the robot depending on the types, quantity and motion of dynamic obstacles within a region.

IX. ACKNOWLEDGMENTS

The authors thank *ontoprise GmbH* for providing research licences for their *ObjectLogic* ontology products *OntoStudio* and *OntoBroker* at no charge.

REFERENCES

- [1] F. Dellaert and D. Bruemmer, “Semantic slam for collaborative cognitive workspaces,” in *AAAI Fall Symposium Series 2004: Workshop on The Interaction of Cognitive Science and Robotics: From Interfaces to Intelligence*, 2004.
- [2] L. Belouaer, M. Bouzid, and A. Mouaddib, “Ontology Based Spatial Planning for Human-Robot Interaction,” in *Proceedings of the 2010 17th International Symposium on Temporal Representation and Reasoning*, 2010, pp. 103–110.
- [3] C. Galindo, J.-A. Fernández-Madrugal, J. González, and A. Saffiotti, “Robot task planning using semantic maps,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.
- [4] J. Guillon and J.-I. Farges, “Geometric and symbolic reasoning for mobile robotics,” in *Proceedings of the 3rd National Conference on Control Architectures of Robots*, Bourges, 2008, pp. 76–97.
- [5] Ó. M. Mozes, P. Jensfelt, H. Zender, G.-J. M. Kruijff, and W. Burgard, “From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots,” in *Proceedings of the ICRA-07 Workshop on Semantic Information in Robotics*, Rome, Italy, 2007, pp. 33–40.
- [6] L. Shi, S. Kodagoda, and G. Dissanayake, “Laser Range Data Based Semantic Labeling of Places,” in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 5941–5946.
- [7] M. Ziegenmeyer, K. Uhl, J. M. Zöllner, and R. Dillmann, “Autonomous Inspection Of Complex Environments by Means of Semantic Techniques,” in *Proceedings of the Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI-2009)*, Thessaloniki, Greece, 2009, pp. 303–310.
- [8] J. Oberländer, K. Uhl, J. M. Zöllner, and R. Dillmann, “A region-based SLAM algorithm capturing metric, topological, and semantic properties,” in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, pp. 1886–1891.
- [9] ontoprise GmbH, *ObjectLogic Tutorial*, 2010.

Situation Assessment and Trajectory Planning for AnnieWAY

Christoph Stiller and Julius Ziegler

Abstract—This contribution addresses machine perception of *a priori* unknown environment, situation recognition, and automated trajectory planning in urban traffic. We discuss how to represent and acquire metric, symbolic and conceptual knowledge from video and lidar data of a vehicle. A hardware and software architecture tailored to this knowledge structure for an autonomous vehicle is proposed. Emphasis is laid on methods for situation recognition employing geometrical and topological reasoning and Markov Logic Networks. Trajectory planning is conducted in spatiotemporal state lattices. The computational effort of the planning method is almost independent of the number of moving objects as these simply disable spatiotemporal nodes. The planning optimizes a quality measure that considers safety, efficiency, and comfort. Results are shown from the autonomous vehicle AnnieWAY that is able to autonomously travel in urban and platooning scenarios.

I. INTRODUCTION

Autonomous Vehicles that perceive their environment, communicate with each other, understand the current traffic situation and may by themselves or cooperatively with others plan and conduct appropriate driving trajectories are an intense field of international research. This contribution outlines the concept and architecture of the 'Cognitive Automobile AnnieWAY' that has successfully participated in international competitions such as the 2005 Grand and the 2007 Urban Challenge, and recently won the 2011 Grand Cooperative Driving Challenge [1], [2], [3]. The vehicle constitutes an experimental basis for automated machine behaviour [4], [5]. Within a few years, large improvements in traffic safety is expected from such technologies [6].

A major goal of the scientific research is to advance knowledge acquisition and representation as a basis for automated decisions. As illustrated in Figure 1, driving - whether by a human or by a cognitive machine - involves knowledge representation in various forms. *Metric knowledge*, such as the lane geometry and the position or velocity of other traffic participants is required to keep the vehicle on the lane at a safe distance to others. *Symbolic knowledge*, e.g. classifying lanes as either 'vehicle lane forward', 'vehicle lane rearward', 'bicycle lane', 'walkway', etc. is needed to conform with basic rules. Finally, conceptual knowledge, e.g. specifying a relationship between other traffic participants allows to anticipate the expected evolution of the scene to drive foresightedly.

C. Stiller and J. Ziegler are with Institut für Mess- und Regelungstechnik, KIT - Karlsruher Institut für Technologie, 76131 Karlsruhe, Germany
 stiller, ziegler@kit.edu

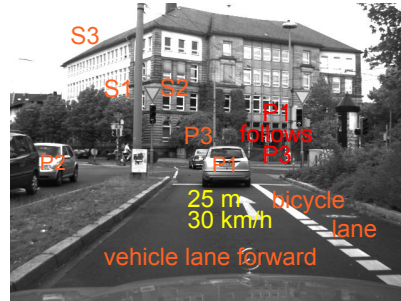


Fig. 1: Metric (yellow), symbolic (orange), and conceptual (red) knowledge for cognitive automobiles

II. ANNIEWAY SYSTEM OVERVIEW

A. AnnieWAY Hardware Architecture

Embodiment is widely considered a crucial element in cognitive systems research. To assess and validate theoretical findings we have adopted the unified hardware and software framework of the Karlsruhe-Munich *collaborate research center 'cognitive automobiles'* [7], [8]. Based on the architecture depicted in Figure 2, meanwhile some ten experimental cognitive automobiles were set up [9], [5], [10]. To ensure

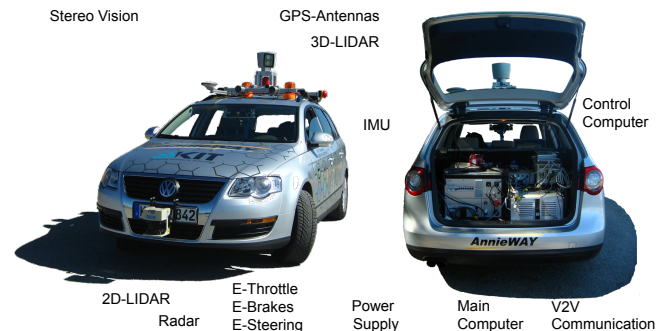


Fig. 2: Hardware setup for the cooperative cognitive automobile AnnieWAY.

real-time capabilities, vehicle control is performed on a dedicated dSpace AutoBox which directly communicates with the actuators over the vehicle CAN. All other perception and planning modules as well as sensor data acquisition are performed by a single multicore multiprocessor computer system which delivers sufficient computing power to host all processes providing low latencies and high bandwidth for inter-process communication.

B. AnnieWAY Software Architecture

The hardware is complemented with a real-time capable software architecture as depicted in Figure 3. The framework has been proposed, implemented, and made publicly available by [7], [11]. Its central element is a real-time database for information exchange. The various driving and perception tasks run in separate processes that communicate via the database and share a centralized view on all available information at every time. The framework supports parallel operation of processes at variable update rates and ascertains hard real-time performance where needed.

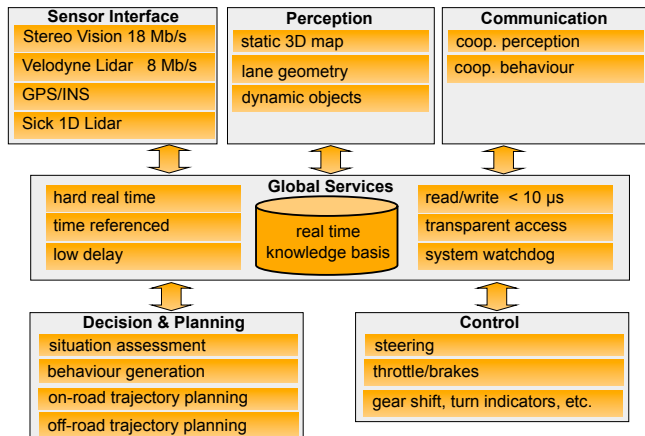


Fig. 3: Software setup for the cooperative cognitive automobile AnnieWAY.

III. SITUATION RECOGNITION

A. Simple geometric and topological reasoning

In this section, we will assume that a representation of the road network is available. This representation has to contain the geometry of single lanes as well as a topology, *i.e.* their interconnectedness within the network. Formally, this representation is a special *geometric graph*, *i.e.* a graph whose edges describe a distinctive road geometry, expressed by a planar curve. Such a representation was available during the Urban Challenge in form of a so called *road network definition file* (RNDF). As has been shown in [12], such a

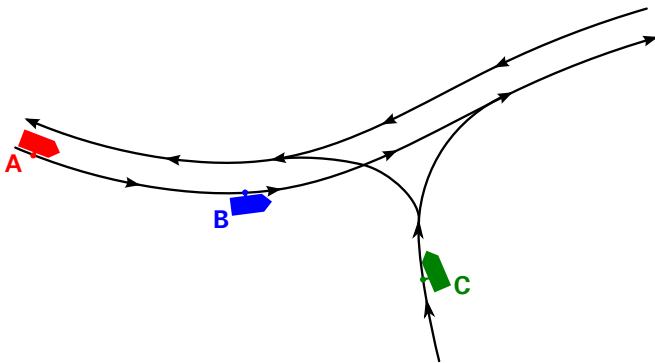


Fig. 4: Geometric graph for road representation and situation recognition.

representation can also be derived from vision cues using formal logic reasoning. Figure 4 shows an example for such a graph. The depicted situation is that of a one-way road forming a T-type-junction towards a road which allows two-way traffic.

Other road users are embedded into the graph using purely geometric reasoning. They are assigned to that edge in the graph which best explains their position and heading. A simple, orientation-aware point-to-curve distance function can be used for this task. Figure 4, depicts three vehicles and their association to edges in the graph.

The graph provides a rich description that readily allows to determine roles of and relations among other road users. From Figure 4, *e.g.*, the relations “A follows B” and “B must yield to C” can be derived *ad-hoc*.

B. Markov Logic Networks

Markov Logic Networks (MLNs) refer to a class of probabilistic logical models combining first-order predicate logics with Markov random fields [13]. An MLN is defined through a set of formulas $\{F_1, \dots, F_n\}$ in first-order predicate logics on a random field with random variables $\mathbf{X} = (X_1, \dots, X_q)$ and a set of scalar weights $\{w_1, \dots, w_n\}$ such that one weight is attributed to each formula.

The joint distribution of the random field is then defined by a Gibbs distribution

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{k=1}^n w_k F_k(\mathbf{x}) \right), \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_q)$ denotes a realization of the random field \mathbf{X} , and Z is a normalizing constant. The logical formulas F_k are instantiated by the realizations \mathbf{x} rendering each formula either *true* or *false*. Typically, each formula will depend on a small subset of variables in \mathbf{x} only that forms a clique of the Gibbs distribution.

Table I shows a simple example for an MLN with two generic formulas. The first formula is applied to each vehicle O_i while the second formula is applied to each pair of vehicle and lane (O_i, R_j) detected in the scene. For a specific scene

	w_i	F_i
1	1.4	$\forall o \text{ hasDirection}(o, \text{Same}) \Rightarrow \text{car}(o)$
2	0.6	$\forall o \forall r \text{ on}(o, r) \wedge \text{road}(r) \wedge \text{hasSpeed}(o, \text{Low}) \Rightarrow \text{car}(o)$

TABLE I: Formulas and weights specifying an MLN

with, *e.g.* two vehicles $\{O1, O2\}$ and one lane $\{R1\}$, one is left with the Markov random field shown in the graph of Fig. 5. This simple example supports the classification of cars through context information [14]. The formulas of an MLN can thus be considered as probabilistic rules with the weights quantifying our degree of belief in these rules. The Gibbs distribution (1) models world configurations as most probable the more they conform with rules that possess large weights.

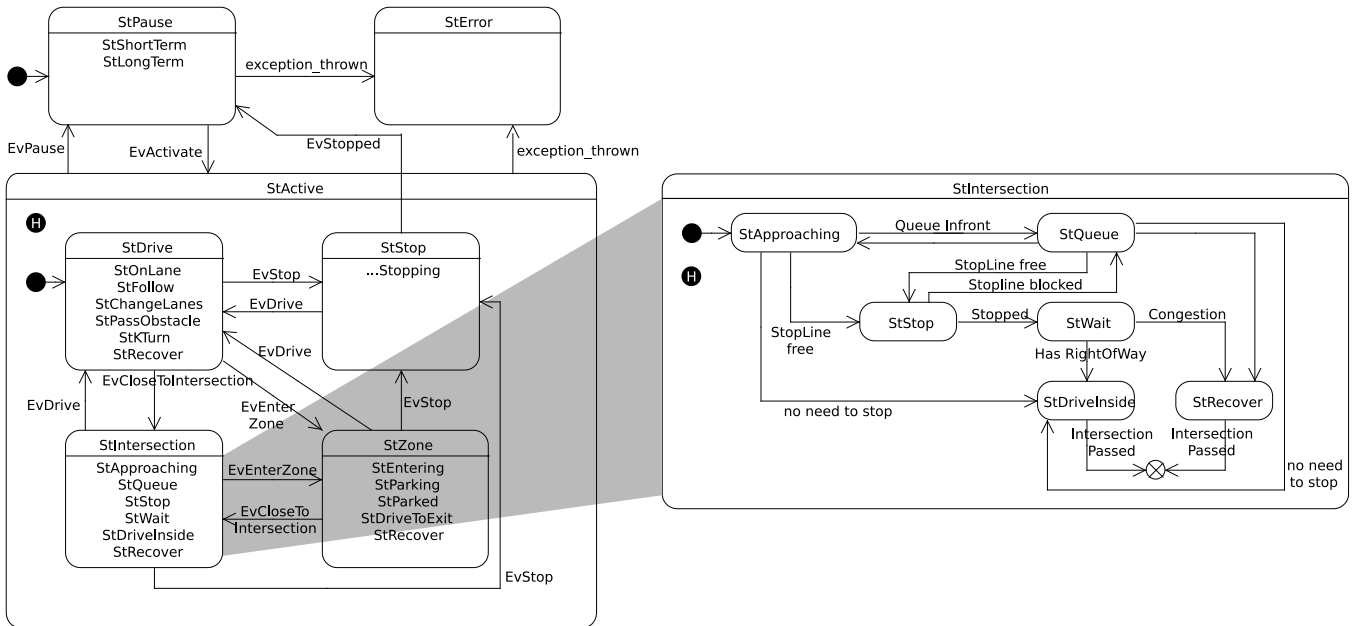


Fig. 6: AnnieWAY’s hierarchical state automaton.

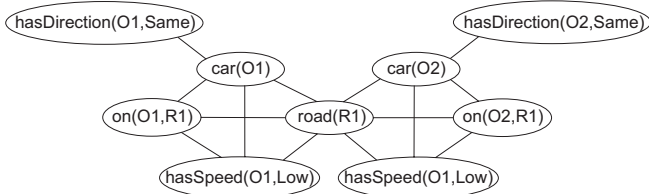


Fig. 5: Graphical representation of the Markov Logic Network defined through the generic formulas and weights from Table I and a scene with two vehicles $\{O1, O2\}$ and one lane $\{R1\}$.

IV. BEHAVIOUR GENERATION

Building on the information provided by the situation recognition module, the behavioural layer makes decisions on actions which need to be carried out in the current situation. Actions are communicated downstream to the trajectory generation stage in the form of center and boundary lines for the driving corridor, or as hard constraints which are imposed onto the generated trajectories (like forcing a stop at a stop line, or obeying a speed limit). Some simplistic tasks, like flashing an indicator, are passed on to the vehicle hardware directly. All these actions are generated using a state automaton which is organised in a hierarchical fashion. The possibility to describe state automata hierarchically has first been described by David Harel in [15] (*Harel state charts*). Figure 6 shows the state automaton which has been used on board ANNIEWAY during the Urban Challenge. Descriptors of states and events are prefixed by St...and Ev..., respectively. Substates are, for the most part, displayed in short form, e.g. the state StDrive contains sub states StOnLane, StFollow, StChangeLane etc. The principle of hierarchal organisation is illustrated by an exemplary “zoom”

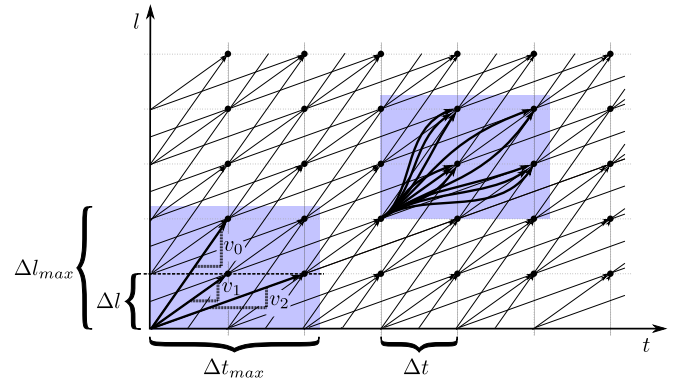


Fig. 7: A spatiotemporal state lattice over a one dimensional workspace. The lower left shaded area depicts a control set for paths in C^0 while the upper right one depicts one designed for higher order continuity, consisting of quintic polynomials.

into the state StIntersection, which shows the detailed structure of the relation of StIntersection’s sub states. For a detailed treatment of state charts and their graphical notation cf. [15]. For a more detailed discussion of the specific use on board ANNIEWAY we refer interested readers to [16].

V. TRAJECTORY PLANNING

After the situation has been recognized and an appropriate behaviour has been identified a specific trajectory is planned. The planning concept described in the sequel belongs to the class of state lattice planers which has been adapted for on road driving in the presence of moving obstacles. A more complete description of the methodology can be found in [17].

A. Spatiotemporal state lattices

Static state lattices result from appropriate sampling of the continuous configuration space and are known as efficient representations for path planning in static environments [18], [19]. Spatiotemporal augment the configuration space of a standard state lattice with time into a single manifold, followed by discretization. To illustrate this concept, we will first consider the simplistic case of a one dimensional spatial configuration space.

Consider a vehicle traveling with varying velocity in \mathbb{R} . Its state is described by its distance from the origin, l and time t . In the spirit of the state lattice approach, we constrain the state space to an equidistantly sampled subset of \mathbb{R}^2 with sampling interval $\Delta l, \Delta t$.

Figure 7 depicts a spatiotemporal state lattice over the described workspace. The figure sketches state transitions for piecewise constant, positive velocities and C^2 continuous paths achieved by quintic polynomials, respectively. Quintic polynomials are attractive for planning dynamic driving manoeuvres, because they minimize squared jerk [20] and allow for fast computation of their coefficients for given boundary conditions. Closed form expressions exist to describe the integral of squared jerk and for maximum speed, acceleration and speed along the trajectory [21]. Quintic splines have been used for automotive motion planning before [22], albeit only to describe kinematic paths without time parametrization.

B. Motion planning using spatiotemporal state lattices

In order to account for moving obstacles their future positions are predicted. Obstacles can then readily be transferred to the space-time manifold, as shown in Figure 8. The shaded area is occupied by a small object that moves with velocity $\frac{1}{2} \frac{\Delta l}{\Delta t}$. A trajectory is found within the spatiotemporal lattice that does not collide with the obstacle.

To deal with obstacles efficiently, we create a mapping between a discrete space-time obstacle map and the set of all edges in the graph. This can be done in the offline graph generation phase. Then, edges blocked by obstacles can be invalidated quickly by a single run over the obstacle map. This method scales well with the number of obstacles maintaining an almost constant overall processing time.

Edge costs consider the integral of the squared jerk of their geometric representations, as opposed to simply considering arc length. This improves safety, controllability and driving comfort.

Graph-based motion planning algorithms usually employ shortest path algorithms that maintain vertices visited in a partially ordered data structure. Algorithms belonging to this class include A* search, as well as Stentz' D* [23] and focused D* [24]. Spatiotemporal lattices belong to the class of directed acyclic graphs (DAG). Hence, sorting vertices by time yields a topological ordering in advance, and vertices can be just processed in this order. The resulting algorithm is linear in the number of vertices n , as opposed to Dijkstra's general scheme which is in $\mathcal{O}(n \log n)$.

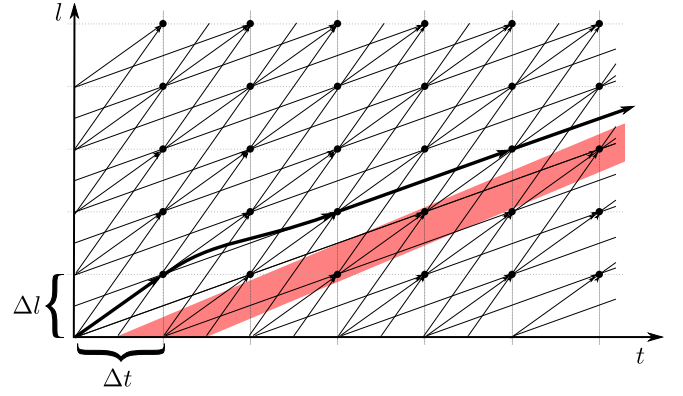


Fig. 8: Planning with a moving obstacle in the space-time manifold. The shaded area is covered by a moving object. A trajectory is shown that is composed of elements of the control set. Shortest paths can be found by relaxing vertices from left to right.

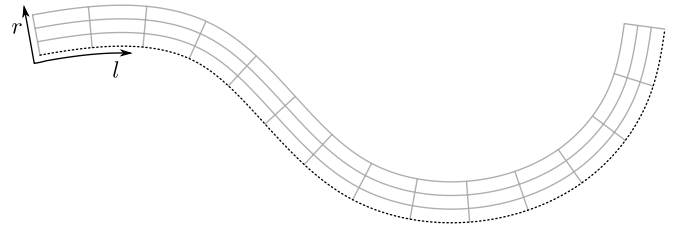


Fig. 9: Reparametrization of the Cartesian plane. The dotted line indicates the original run of the road, (X, Y) . The grey structure illustrates the discrete reparametrization in l and r .

C. Lane-adapted reparametrization

The principle of spatiotemporal state lattices developed in the preceding sections generalizes naturally to two dimensions. Doing this naively, however, produces dimensionality problems due to the required dense sampling of the state space. Note that, in comparison with [19] the dimensionality of the sampling space for the state lattice rises from 3 (2D position and orientation, in [18], curvature is considered additionally) to 7 (2D position, 2D velocity, 2D acceleration and time), due to moving from a kinematic to a higher order dynamic model and the incorporation of time. With dimensionality rising, coverage of the configuration space requires an exponentially growing number of samples. Hence, an efficient way of sampling the configuration space is needed that is adapted to the special case of navigating on a road whose run is known *a priori*, e.g. from digital map data.

Given a continuous, piecewise twice differentiable, arc length s parametrized representation $(X(s), Y(s))$ of the course of the road, we define the following reparametrization (l, r) of the 2D workspace, where (x, y) denote Cartesian coordinates, $l(t)$ is the distance travelled along the road, and $r(t)$ is the lateral offset towards the road centre:

$$x(t) = X(l) - rY'(l) \quad (2)$$

$$y(t) = Y(l) + rX'(l). \quad (3)$$

This is a base change towards a local orthogonal coordinate

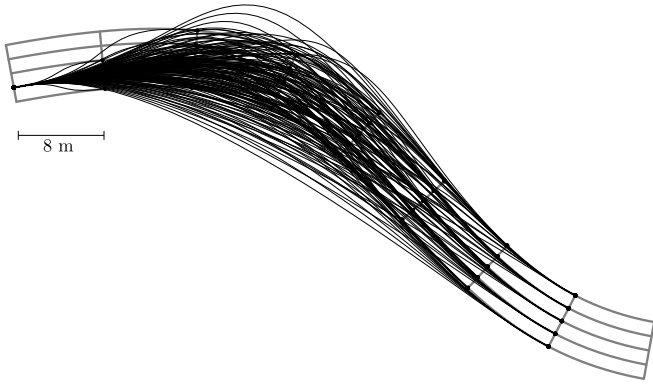


Fig. 10: State transitions on the transformed grid. The successors of one vertex are shown in black.

system that has its abscissa aligned with the road for any l . It defines a two dimensional manifold as depicted in Figure 9. As described earlier, differential boundary conditions of up to second order are required for edge generation. We therefore need to transform them through equations (2) and (3): Given \dot{l} , \dot{r} , \ddot{l} and \ddot{r} , by application of the chain rule we obtain

$$\dot{x} = \dot{l}X'(l) - \dot{r}Y(l) - r\dot{l}Y'(l) \quad (4)$$

$$\dot{y} = \dot{l}Y'(l) + \dot{r}X(l) + r\dot{l}X'(l) \quad (5)$$

and

$$\ddot{x} = \ddot{l}X' + \dot{l}^2X'' - \ddot{r}Y - (2\dot{r}\dot{l} + r\ddot{l})Y' - \dot{r}\dot{l}^2Y'' \quad (6)$$

$$\ddot{y} = \ddot{l}Y' + \dot{l}^2Y'' - \ddot{r}X - (2\dot{r}\dot{l} + r\ddot{l})X' - \dot{r}\dot{l}^2X'' \quad (7)$$

We now restrict parameters $l, r, \dot{l}, \dot{r}, \ddot{l}$ and \ddot{r} to a discrete, grid like set (the vertices of the search graph) and transform them through equations (2) - (7). The resulting $x, y, \dot{x}, \dot{y}, \ddot{x}$ and \ddot{y} , together with discrete values for time t , are used as boundary values to calculate quintic polynomial trajectories as described in section V-A. To assert dynamic and kinematic feasibility, a respective edge is only added to the graph, if velocity, acceleration and jerk stay within bounds defined in advance. In the effort to further reduce the number of vertices, some ad hoc reductions can be applied to the sets of discrete parameters: r is constrained to an interval so as to restrict all vertices of the lattice to be within the bounds of the road. We set $\dot{r} = 0$ and constrain \dot{l} to be positive, since we wish the vehicle to make progress along the road, while crosswise motion is to be avoided. Second derivatives \ddot{l} and \ddot{r} of the untransformed coordinates are set to zero at the grid points.

Figure 10 gives an impression of the graph we used for our experiments by displaying successor edges of a single vertex. The outdegree of vertices is approximately 200.

VI. EXPERIMENTAL RESULTS

Figure 11 shows an exemplary result of the proposed trajectory planning method. The scenario selected for this example is that of merging into running traffic at a T-junction. As can be seen, the proposed method yields a trajectory that is smooth in the sense of minimum mean squared jerk and

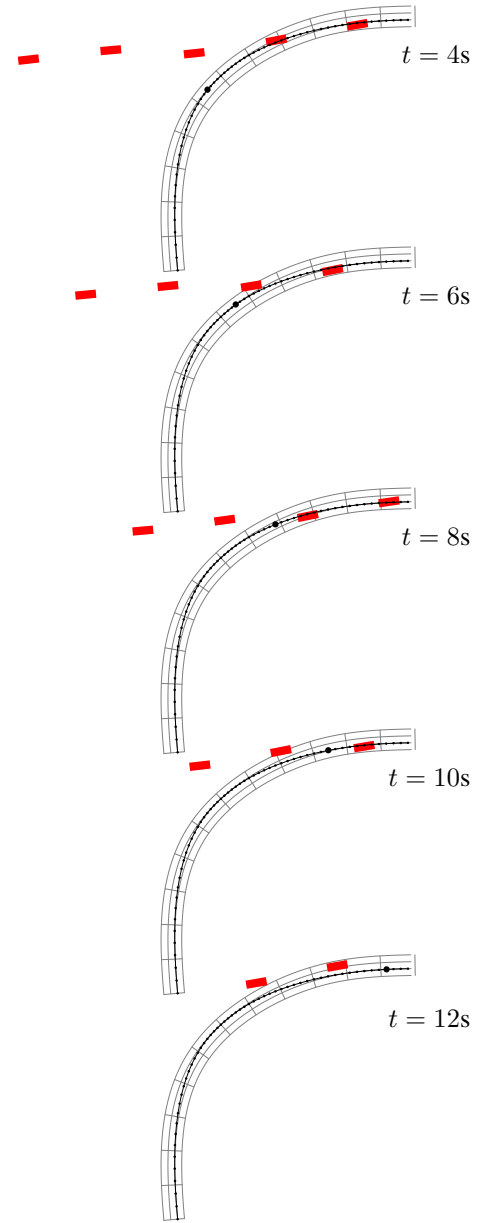


Fig. 11: Merging into traffic. The planner selects a comfortable and safe trajectory into a convoy of five vehicles traveling towards a T-junction.

safe in the sense of entering the gap at a safe distance to all other vehicles with a velocity of the gap itself. The proposed method inherently selects the optimum gap to cut in. The planner is also able to find trajectories under more complex traffic conditions.

Figure 12 shows AnnieWAY performing in the Urban Challenge of 2007. The graph based representation of the road network can be seen. In the top right of each frame, the sequence of states the hierarchical state automaton traverses is displayed. More recently, the vehicle has won the Grand Cooperative Driving Challenge in May 2011.

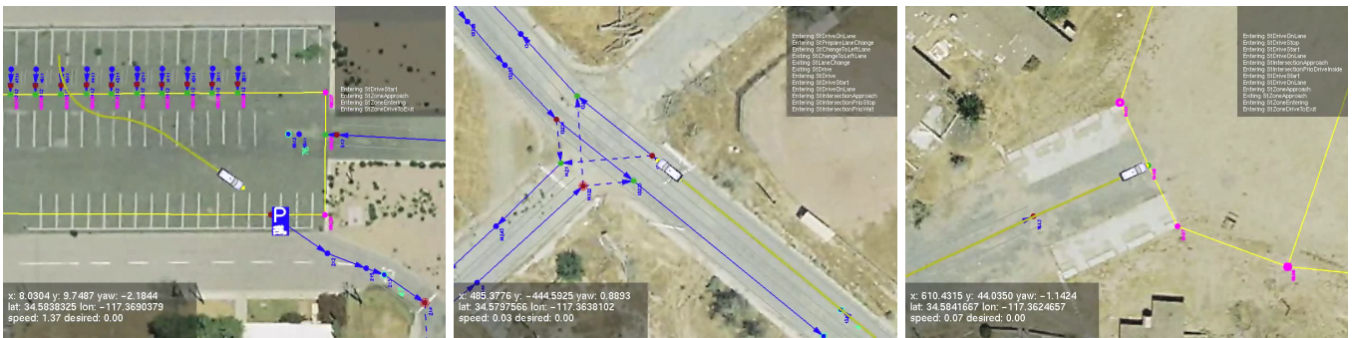


Fig. 12: AnnieWAY in the Urban Challenge of 2007

VII. CONCLUSIONS

AnnieWAY is an experimental autonomous vehicle that perceives *a priori* unknown environments, recognizes situations, plans appropriate trajectories, and controls its actuators to follow these. It acquires metric, symbolic and conceptual knowledge from video and lidar data and employs prior knowledge from maps. The current driving situation is recognized employing geometrical and topological reasoning. Furthermore, we experiment with Markov Logic Networks for automated inference of pair relations among objects. Spatiotemporal state lattices have been shown to allow for efficient trajectory planning without severe restrictions on optimality and generality. Moving vehicles are accounted for by disabling spatiotemporal nodes along their paths. Thus the method maintains its computational load irrespective of scene complexity. The quality criterion optimized during planning can be chosen with large freedom. Currently, we minimize mean squared jerk, but extended measures could account for safety, energy efficiency, or psychokinetic comfort in the future. Results are shown from the autonomous vehicle AnnieWAY that is able to autonomously and cooperatively travel in urban and platooning scenarios.

REFERENCES

- [1] Ü. Özgüner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," *IEEE Proceedings*, vol. 95, no. 2, pp. 1–16, Feb. 2007.
- [2] Sören Kammel, Julius Ziegler, Benjamin Pitzer, Moritz Werling, Tobias Gindele, Daniel Jagzent, Joachim Schröder, Michael Thuy, Matthias Goebel, Felix von Hundelshausen, Oliver Pink, Christian Frese, and Christoph Stiller, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615 – 639, Sept. 2008.
- [3] F. v. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640 – 673, Sept. 2008.
- [4] C. Stiller, G. Färber, and S. Kammel, "Cooperative cognitive automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 215–220.
- [5] M. Thuy, M. Althoff, M. Buss, K. Diepold, J. Eberspächer, G. Färber, M. Goebel, B. Heiöing, S. Kraus, R. Nagel, Y. Naous, F. Obermeier, F. Puente León, F. Rattei, C. Wang, M. Schweitzer, and H.J. Wünsche, "Kognitive Automobile - Neue Konzepte und Ideen des Sonderforschungsbereiches/TR-28," in *3. Tagung Aktive Sicherheit durch Fahrerassistenz*, Garching bei München, 2008.
- [6] European Union, "E-Safety," http://www.ec.europa.eu/information_society/activities/esafety, November 2010.
- [7] Matthias Goebel and Georg Färber, "A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles," in *Proc. IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007, pp. 734–739.
- [8] Moritz Werling, Matthias Goebel, Oliver Pink, and Christoph Stiller, "A hardware and software framework for cognitive automobiles," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2008*, Eindhoven, Niederlande, 2008, pp. 1080 – 1085.
- [9] M. Goebel, M. Althoff, M. Buss, G. Färber, F. Hecker, B. Heiöing, S. Kraus, R. Nagel, F. Puente León, F. Rattei, M. Russ, M. Schweitzer, M. Thuy, C. Wang, and H.J. Wuensche, "Design and Capabilities of the Munich Cognitive Automobile," in *Proc. IEEE Intelligent Vehicles Symposium*, Eindhoven, the Netherlands, June 2008, pp. 1101–1107.
- [10] J. Schröder, T. Gindele, D. Jagszent, and R. Dillmann, "Path planning for cognitive vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Holland, 2008, pp. 1119–1124.
- [11] Matthias Goebel, *Eine realzeitfähige Architektur zur Integration kognitiver Funktionen*, Dissertation, Technische Universität München, München, 2009.
- [12] Britta Hummel, Werner Thiemann, and Irina Lulcheva, "Description logic for vision-based intersection understanding," in *Proc. Cognitive Systems with Interactive Sensors (COGIS)*, 2007.
- [13] Matthew Richardson and Pedro Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [14] C. Stiller, S.Kammel, I. Lulcheva, and J.Ziegler, "Probabilistic methods for environment perception of cognitive automobiles," *at - Automation*, vol. 11, pp. 568–574, 2008.
- [15] David Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, June 1987.
- [16] Tobias Gindele, Daniel Jagzent, Ben Pitzer, and Rüdiger Dillmann, "Design of the planner of team annieway's autonomous vehicle used in the darpa urban challenge 2007," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, Die Niederlande, 2008.
- [17] Julius Ziegler and Christoph Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *International Conference on Intelligent Robots and Systems*, 2009.
- [18] M. Pitvoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, 2005.
- [19] Maxim Likhachev and Dave Ferguson, "Planning long dynamically-feasible maneuvers for autonomous vehicles," in *Proceedings of Robotics: Science and Systems IV*, 2008.
- [20] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local path planning and motion control for AGV in positioning," in *International Workshop on Intelligent Robots and Systems*, 1989.
- [21] R.L Andersson, "Aggressive trajectory generator for a robot ping-pong player," *Control Systems Magazine*, vol. 9, pp. 15–21, 1989.
- [22] A. Piazzi, C.G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G^2 -splines for the iterative steering of vision-based autonomous vehicles," *Transactions on Intelligent Transportation Systems*, vol. 3, pp. 27–36, 2002.
- [23] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *International Conference on Robotics and Automation*, 1994.
- [24] A. Stentz, "The focussed D* algorithm for real-time replanning," in *International Joint Conference on Artificial Intelligence*, 1995.



Session III

Interactive session

- **Title: Proposition for propagated occupation grids for non-rigid moving objects tracking**
Authors: B. Lefaudeux, G. Gate, F. Nashashibi
- **Title: Probabilistic Road Geometry Estimation using a Millimetre-Wave Radar**
Authors: A. Hernandez-Gutierrez, J. I. Nieto, T. Bailey, E.M. Nebot
- **Title: Safety robotic lawnmower with precise and low-cost L1-only RTK-GPS positioning**
Authors: J.M. Codol, M. Poncelet, A. Monin, M. Devy
- **Title: Odometry from Planar landmarks**
Authors: K. Narayana, B. Steux
- **Title: Probabilistic autonomous navigation using Risk-RRT approach and models of human interaction**
Authors: J. Rios-Martinez, A. Spalanzani, C. Laugier



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Proposition for propagated occupation grids for non-rigid moving objects tracking

Benjamin Lefaudeux
IMARA - INRIA Rocquencourt
benjamin.lefaudeux@inria.fr

Gwennael Gate
Aldebaran Robotics
Email: ggate@aldebaran-robotics.com

Fawzi Nashashibi
IMARA - INRIA Rocquencourt
CAOR - Mines Paristech
Email: fawzi.nashashibi@inria.fr

Abstract—Autonomous navigation among humans is, however simple it might seem, a difficult subject which draws a lot of attention in our days of increasingly autonomous systems. From a typical scene from a human environment, diverse shapes, behaviours, speeds or colours can be gathered by a lot of sensors; and a generic mean to perceive space and dynamics is all that is more needed, if not easy. We propose an incremental evolution over the well-known occupancy grid paradigm, introducing grid cell propagation over time and a limited neighbourhood, handled by probabilistic calculus. Our algorithm runs in real-time from a GPU implementation, and considers completely generically space-cells propagation, without any a priori requirements. It produces a set of belief maps of our environment, handling occupancy, but also items dynamics, relative rigidity links, and an initial object classification. Observations from free-space sensors are thus turned into information needed for autonomous navigation.

I. OCCUPANCY GRIDS - EXTENDED

A. General considerations

1) *Perception of non-structured moving objects*: Amid a vast set of localisation and perception algorithms currently developed, among which various SLAM variants have taken the greater share for the last years, some challenges are still not addressed in the perception field. Detection and tracking of moving objects, without pre-requisites on rigidity, structured environment or immobility remains a tough question, and is not yet addressed by many approaches. Different SLAM-based approaches have been used for the last years, some of them being very successful in building a map of the environment from a non-associative sensor, while dealing with the identification of some moving objects amid the map. FastSLAM, using particle filtering and initially presented by Thrun et al. ([1]) or other iterative optimisation techniques (ICP, [2] [3]) are very powerful algorithms which can deal with some level of movement amid tracked points, as long as fixed points of the map are enough to estimate robot pose and environment mapping. This proved effective, for example by Wang et al. ([4]), which showed that moving points could be separated from a still structure with a good accuracy, and tracked accordingly. Overall scene requisites are however still strong in this setup, and would probably not be enough for reliable moving object perception and tracking in a highly unstructured environment, without a majority of solid points to observe, which can easily happen outdoor in densely populated areas. Another possibility is

exploitation of detection before tracking, using information from visual or speed measures sensor, in order to remove moving objects from the traditional SLAM calculus. This has notably been demonstrated by Agrawal ([5]), which proved effective to detect individuals while still needing fixed points as a reference. Contrary to SLAM, our aim is thus not primarily to localise oneself in the environment along with building a map of its static features, but instead to be able to reliably detect and monitor position and dynamics of surrounding moving objects, possibly outdoor and without much visible solid infrastructure. Those observations will drive the following developments, and while the proposed algorithm is nowhere near best SLAMs in terms of accuracy, we believe it could be useful in typical human environments.

2) *Interests and limits of the simple grid-based approach*: Occupancy grids are very common in robotics, since their first introduction by Moravec and Elfes ([6]), initially related to sonar based mapping. Principles are simple and effective, relying on the information storage and information source spatial localisation similarities, thus allowing to keep any spatial relations between cells at a minimal cost. This is still used nowadays, although not being any more the only processing step of the algorithm, and can still be viewed as a very-capable mean of storing spatially related information or sharing information between several subjects. It is also specially well-suited to deal with promising new massively-parallel computing capabilities. Occupancy grids were indeed present in most of the latest SLAM propositions (based on filtering or optimisation), and in many of broadly speaking perception systems (Badino et al [7] showed for example a free-space perception system based on vision and occupancy grids in 2007, as did more recently by Yguel et al. [8]). In those examples, grids are however only used for information representation and storage, most of the processing being due to external algorithms. Sensor filtering over time is indeed rarely considered within the grid formalism (grid update rule often relies on mere accumulation). Links between cells are also rarely considered while updating grids, spatial and temporal independence between measures being at the heart of the initial simplicity of the method. Obviously a big step from accurate world description, this is basically of little consequences provided the aim is, as it was in the original Moravec article [6], the cartography of still environment. In

case we chose the occupancy grid formalism to track moving objects, this is however a major limitation, as one could expect the displacement of physically related points to be correlated.

3) *Optimal update to occupancy grids*: Considering a hypothetical occupancy grid propagation, one could wonder why an optimal Bayesian propagation would not be possible. The probability of any possible cell displacement could be computed, thus computing the most-probable map prediction from a given set of measure, allowing both temporal filtering and a link between new measures and known information. Trouble is, computing the formal probability (provided all the needed information is known) is extremely demanding, if one is to consider every possible move. A complete probability calculus, considering a set of S possible state in any cell of a $N * M$ grid, would imply the consideration of each and every possible map, which is S^{N*M} . Taking the smallest possible state sampling (2 states), and a ridiculously small map of $10*10$ cells, and the amount of maps to compute to get to the full solution is already beyond any reasonable range (2^{100}). The key to this “absurd” complexity when keeping every probability on track, which could seem somewhat odd from a human perception (we don’t usually need much time to get a good grasp on a typical scene involving moving objects), surely relies in a lot of useless probabilities being taken into account, even possibly beyond the causality principle. Anything at a given place have little to no chance of influencing the very next future of a very remote location, and this is the idea behind dealing with neighbourhood-restricted probabilities.

Another possible approach, demonstrated by Coue et. al in [9], would be the use of a Bayesian network coupled with a motion model for dynamic objects mapping. In this case, no actual association needs to be done, this being handled by the Bayesian framework computing transition probabilities between states. The main difference with our approach is that they do not attempt to compute the most probable next place for a given occupancy, but rely on a given motion model (constant velocity in this example). This is very fine in most cases, and definitely is an improvement over static occupancy maps, but we believe that this can lead to the wrong prediction in some cases, among which heavy occlusion or colliding courses. We however certainly share a lot of the abstraction presented in this article, although a few more notions are present in our algorithm.

B. Tackling inter-dependence within computing boundaries

We propose the use of propagated occupancy grids, able to deal with some of the interactions between cells, in a common prediction/measure Bayesian cycle. Our algorithm aims at taking into account both temporal and spatial relations between measures, while keeping computing costs low enough to conceive a real-time use and concurrent use to other more specialised algorithms.

Firstly, we introduce the probability for every cell to move to its neighbourhood, given previous knowledge of the scene (occupation, speed, classification) and specific heuristics (separate cells cannot converge, nor can cells from the same object diverge). Secondly, we compare this prediction to a new measure, and compute the most probable estimate given prediction and latest measure. Thirdly, we update associated knowledge used in the prediction step, namely occupancy of every cell of the grid, speed, relation between cells (in a neighbourhood) or object classification used for different sensor models. Those principles were presented by Gate in [10], initially on a standard CPU implementation, and showed very promising results despite a high computing cost making it prohibitive for any real-time application.

1) *Initial definitions* : : The probability mass functions (*pmf*) modelled in the following algorithm concern a set of notions that we’d like to define :

- Mapping (occupancy) probability $M_k(x_i) \in [0, 1]$ of the cell x_i in the spatial environment E at the time k , provided the measures $Z_{0:k} = \{z_0, ..z_k\}$:

$$P(M_k(x_i) = 1 | Z_{0:k}) \quad \forall x_i \in E \quad (1)$$

- Vehicle localisation (including position and speed in $E \times V$), at the iteration k . This is not yet addressed by the algorithm, and in the examples below every speed and position is relative to the vehicle.

$$P(L_k = l_j | Z_{0:k}) \quad \forall l_j \in (E \times V) \quad (2)$$

- Association, ie the probability for a given cell from the iteration $k-1$ to be associated with another given cell at the iteration k . In our case, only associations coming from a restricted neighbourhood are taken into account, which cuts the number of evaluated map candidates from an exponential dependence on the number of cells to a more reasonable dependence on neighbourhood scale.

$$P(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_i) = 1, Z_{0:k}) \quad \forall (x_i, x_j) \in E^2 \quad (3)$$

- Velocity probability of a cell, given its occupancy and previous measures :

$$P(V_k(x_i) = v | M_k(x_i) = 1, Z_{0:k}) \quad \forall (x_i, v) \in E \times V \quad (4)$$

- Detection probability, to handle the probability that two given cells x_i and x_j are part of the same object. The neighbourhood constraint limiting interactions to a finite part of the map is used once more, to limit the intricateness and heavy computing cost. Affiliation to a given object can however be “propagated” further than one cell’s neighbourhood, although our span is limited and this could prove to be a problem. Detection probability, $D_k(x_i, x_j) \in [0, 1]$ is 1 if x_i x_j are from the

same object.

$$P(D_k(x_i, x_j) = 1 | M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k}) \\ \forall (x_i, x_j) \in E^2 \quad (5)$$

Several criteria could be used to determine this probability from measures, among which a constant relative configuration, fit with a given shape, or a set of distinct characteristics (speed, shape, colour,..). In this first implementation, a simple geometrical criteria is used : $D_k(x_i, x_j) = 1$ if the distance between x_i and x_j is conserved over iterations, with a Gaussian decrease elsewhere (detailed below).

- Classification designs the probability of this set of cells to be part of a given class of objects (car, pedestrian, still object,..). We attempt to model this by matching extended characteristics of a set of cells (beyond geometrical characteristics for example) to a model. This conditions the update rule of association calculus, envisaged future positions of a cell being for example adapted from its class motion model. Classification probability are simultaneously kept from a different set of classes, a cell being capable of a partial fit with different classes. This ensures a more robust classification, initially prone to errors.

$$P(C_k(x_i) = c_j | M_k(x_i) = 1, Z_{0:k}) \quad \forall (C \times E) \quad (6)$$

2) *Update rules - proposed algorithm* : Having set these definitions, the proposed update rules, implemented as such in the parallel execution we present as a last-part example, are as follows :

- Associations are updated in a several pass mechanism, making an extra initial assumption of independence between cells behaviour, which we attempt to correct in a second part. This behaviour was already present in the [10] proposition, and is a key to the possible use of massively parallel computing. Approximations are obviously primordial in our attempt to make the calculus feasible in real-time, but we believe most of the interactions between cells are still modelled with this proposition.

$$P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k}) = \\ \eta \cdot \underbrace{P_{local}(z_k | X_{k-1}^{next}(x_j), M_{k-1}(x_j) = 1, Z_{0:k-1})}_{Correction} \\ \cdot \underbrace{P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k-1})}_{Prediction} \quad (7)$$

η is here a normalisation constraint, to ensure that possible moves sum up to one for any given cell.

First we then compute the local associations prediction, which is to say that macroscopic interactions are not yet taken into account : cell previous speed and class are used to predict the asserted new positions. This could be seen, similarly to SLAM particle filters ([1]), as a new set of particles generated for every cell of the grid iteratively, representing this cell's occupancy possible next moves, depending on previous

knowledge and motion model. In our case, initial predictions are weighted by a Gaussian, whose standard deviation is function of the identified class of the object (thus representing the possible uncertainty in an object next move, this being different for a pedestrian or an identified bus). The centre of the Gaussian weight is relative to the object previous speed estimation.

$$P_{local}(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_j) = 1, Z_{0:k-1}) = \\ \Psi(x_j, x_i, V_{k-1}(x_j), C_{k-1}(x_j)) \quad (8)$$

which could be rewritten as, with g the standard Gaussian expression and σ its standard deviation dependent on the identified class :

$$\Psi(x_j, x_i, V_{k-1}(x_j), C_{k-1}(x_j)) = \\ g(x_j + \mathbf{V}_{k-1}(x_j) \cdot dT - x_i, \sigma_{C_{k-1}(x_j)}) \quad (9)$$

Last measure is then taken into account to produce an estimated local association, still without macroscopic constraints to alter these predicted associations. Predictions are weighted according to the sensor model occupancy new measure, while keeping the normalized sum of all possible displacements :

$$P_{local,weighted}(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_j) = 1, Z_{0:k}) = \\ \gamma P_{local}(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_j) = 1, Z_{0:k-1}) \\ \cdot P(M_k(x_j) = 1 | Z_k) \quad (10)$$

with γ such as $\sum_{a_k \in A} P_{local,weighted}(a_k) = 1$ (A being the set of investigated associations).

Associations initially local estimation are then altered according to additional constraints : unlikely moves are penalised according to different heuristics (different cells cannot converge to the same place, cells from the same rigid object cannot diverge). Rigidity and non inter-penetration constraints are modelled by the potential function $\Phi_{association}$, which is currently based on two Gaussian-window weight functions.

$$P(X_{k-1}^{next}(x_j) = \hat{x} | M_{k-1}(x_j) = 1, Z_{0:k-1}) \\ \simeq \sum_{a_k \in A} \left\{ \prod_{1 \leq j \leq N} P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_i) = 0, Z_{0:k}) \right\} \\ \cdot \Phi_{association}(a_k, E, Z_{0:k}) \quad (11)$$

were A is again the set of all possible associations. The process here described can be differently factorised, but was split into several summations in an attempt to increase its readability.

- Mapping is computed taking into account the two cases : in the cell is a newly observed presence, or the displacement

(possibly null) of a previously seen cell. In the first case, the sensor model is the only input taken into account, in the second case contributions from all the possible associations are summed up to compute the predicted occupancy. Cell interactions have in this case already been taken into account in the association computation. The two cases are dissociated by a random variable S_k , which can take two values : 0 if the cell has never been seen, 1 if the cell has already been seen. Its probability is computed with association computation results : to sum up, if the considered cell corresponds to a local association maximum, $P(S_k(x_i) = 1|Z_{0:k})$ takes the value $\gamma \in [0, 1]$, else it takes the value $1 - \gamma$. The value of γ is chosen depending on the “renewal“ rate of the map, that is to say ”how often do we think a new object can appear from nowhere“ ? A value around 0.5 has proven to work well in practice.

$$\begin{aligned}
P(M_k(x_i)|Z_{0:k}) = & \\
& P_{seen}(M_k(x_i)|S_k x_i = 1, Z_{0:k}) \cdot P(S_k(x_i) = 1|Z_{0:k}) \\
& + P_{unseen}(M_k(x_i)|S_k x_i = 0, Z_{0:k}) \\
& \cdot P(S_k(x_i) = 0|Z_{0:k}) \quad (12)
\end{aligned}$$

with the corresponding calculus (A being the cell neighbourhood) :

$$\begin{aligned}
P_{seen}(M_k(x_i)|Z_{0:k}) = & \\
\sum_{j \in A} \{ & P(X_{k-1}^{next}(x_j) = x_i | M_{k-1}(x_j) = 1, Z_{0:k-1}) \\
& \cdot P(M_{k-1}(x_j) = 1 | Z_{0:k-1}) \} \quad (13)
\end{aligned}$$

P_{unseen} is in this case typically related to the sensor occupancy model.

- Velocities are computed taking into account the same two possibilities, depending if the observed cell is considered a new one, or the association of an already-observed cell to a new position :

- considering the velocity of already-observed cells, velocity is simply computed from the associations, summing up speed values stemming from all the possible contributors.

- considering appearing cells, the probability distribution of velocities had been proposed by Gate in [10] as follows, and kept in this proposition :

$$P(V_k(x_i)|S_k(x_i) = 0, Z_{0:k}) = \frac{1}{card(V)} \quad (14)$$

Merging of the two possibilities is done similarly to eq. 12. In current implementation, we only retain the most probable *a posteriori* value, and a confidence value, instead of retaining the whole probability distribution.

- Detection update needs a broader view from previous information, in order to be able to detect structures and links between updated cells. In this paper we propose a simple (and fast) mechanism to handle this detection, but we expect to

adopt a maybe more ”large scale” approach in the future. The mechanism proposed in this initial algorithm consists in measuring the auto-correlation between consecutive cells associations after filtering, in the vicinity of their neighbourhood. We then measure the compatibility of their predicted moves with the ”rigid body” hypothesis. With the approach used in eq. 12, we dissociate in this calculus cells which are believed to have been seen before from ”new” ones, for which no rigidity information can be guessed.

$$\begin{aligned}
P(D_k(x_i, x_j) = 1 | M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k}) = & \\
\sum_{a_k, a'_k \in A_k \times A'_k} & \\
\{ P(X^{next}(x_i) = a_k | Z_{0:k}) \cdot P(X^{next}(x_j) = a'_k | Z_{0:k}) \} & (15)
\end{aligned}$$

with a_k, a'_k being in fact the same associations in each cell respective referential (we go through every possible association for the (x_i, x_j) cells. Although not very illustrative, eq. 15 calculus is fast to compute, but relationships further than a cell neighbourhood are not taken into account (typically a few meters radius). This could prove insufficient for the tracking of big objects, and other methods could be investigated in the future. A simple k-means clustering could for example be used outside of the prediction/update cycle to emphasize object detection for an external navigation task.

- Classification updates can similarly be done using every gathered information (mapping, velocity, rigidity links,...) correspondence to a given sensor model, which would on the other hand improve prediction steps of the algorithm. This is not yet present in our implemented algorithm, and presented results can thus be seen as perfectible. There is however no theoretical constraints on this calculus, which should be in place in our implementation algorithm in a short time for several classes of objects (pedestrian, cars,...). The performance impact is to be investigated, but should not theoretically prevent the algorithm to run in real-time, every added class acting in this organisation as another ”layer” of probabilities to be computed. Worst impact could thus be a linear cost in terms of the number of identified classes, which should not be increased inconsiderately.

II. PRACTICAL IMPLEMENTATION AND RESULTS

A. Performance considerations

As stated in eq. I-A3, complexity constraints on calculus are not to be neglected in occupancy grid update rules, many thinkable algorithms being simply not realistic in terms of computing needs. The complexity of the mechanism we propose can be summarised as follows :

- for the sake of simplicity, we state a $N * N$ grid, every cell being able to move in a $M * M$ neighbourhood.

- considering the propagation of one cell, every possible move of every of its neighbours ($O(M^4)$) are to be investigated for each individual envisaged propagation ($O(M^2)$), which translates in $O(M^6)$ complexity.

- cells updates being independent except for conformation rules already taken into account in the previous step, overall cell considerations is finally of $O(N^2 * M^6)$ complexity for the full update step.

Heavy approximations are of course still present when compared to an full propagation calculus, namely that each and every cell moves are initially computed individually, although being later filtered to take some interactions into account. Computation is thus still not intricate, which keeps the complexity “low”, although $O(N^2 * M^6)$ remains a heavy burden for any realistic dimensions. Several points can be emphasised from this simple complexity calculus : Firstly, although the computation remains slow by all means, the relative independence of most calculus make it a plausible candidate for a parallel implementation, which is nowadays common and lifts some of the computing time constraints. Secondly, digging into specific complexity aspects, our algorithm is also linear in complexity as regards the number of cells for a given neighbourhood (which translates into a squared dependence as regards the size of the grid, naturally). This seems quite a burden, but one must remember the usual exponential complexity of algorithms meaning to explore any possible point move onto a map. This relative lightness in the grid size complexity is a key benefit for sharing applications : extending the size of the grid translates into linear increase in complexity instead of an exponential increase in case of a propagation calculus on the whole map. Extending the domain of tracked speed is however more complicated : tracked speed depends on the span of possible moves taken into account at every iteration, literally $V_{max} = \frac{M/2*\Delta}{T_{iteration}}$ with Δ the spatial extension relative to one grid cell. Considering a given iteration maximum computing time (limited for example by the Lidar frequency for real-time operations), the maximum tracked speed is rapidly capped by the computing power at disposal relatively to the maximum computing time. We’ll see with our preliminary results that this translates to very acceptable maximum speeds for our initial implementation on current hardware.

B. Some results

As usual when dealing with grid-based algorithms, sensor occupancy models are a key factor in our proposition. A standard Lidar occupancy model is used in this initial implementation, computed on GPU. Occupancy of areas in the shadow of laser impacts are chosen to 0.5 out of 1, neutral in our occupancy ratio. Laser impacts are otherwise set to an occupancy probability of 1, while empty spaces between the vehicle and impacts are set to 0, as it is common using Lidars. Real data gathered in an urban environment are used. Logging and replay framework is RTMaps software from Intempora.

Exhaustiveness and generic nature of the algorithm is important : every mapped cell of the environment is considered equal, and no *a priori* is ever made on geometrical

bounding, preferred positions, structures, stationary or moving parts. Although still quite demanding on computing resources, the algorithm works in real-time on current state-of-the-art hardware. The first scene computed below needs 120ms to compute on a GF100 GPU from nVidia counting 448 cores, which translates without additional work to below 100 ms on higher-end offerings currently available. As regards raw performance, this algorithm is also an initial draw, and pure implementation could certainly be greatly improved as it is often the case with hardware-sensitive programming.

1) *Algorithm memory*: In this example, we emphasise a temporary occlusion situation, where a pedestrian shadow hides another previously seen pedestrian (figure eq. 1). Only the Lidar sensor is used in this case, camera captures being presented for illustrative purpose, along with bounding boxes. On the *pmf* representations (figures eq. 2 and eq. 3), the point of view is from above, in a common “bird-view” perspective. All the boxes are drawn for illustrative purpose, we don’t present here the output of a detection algorithm. The resolution of grid mapping is 15cm, speeds up to 4.5m/s being theoretically tracked. This last value can be improved without any computing cost by simultaneously degrading the spatial resolution and increasing the range of the measures, which could be a dynamic trade-off depending on the vehicle speed.

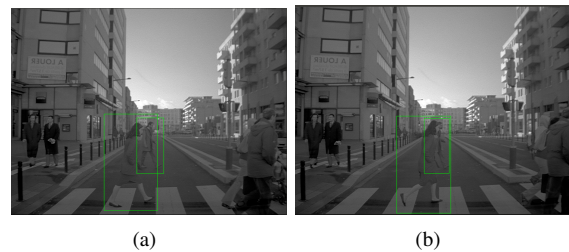


Fig. 1. Successive camera views

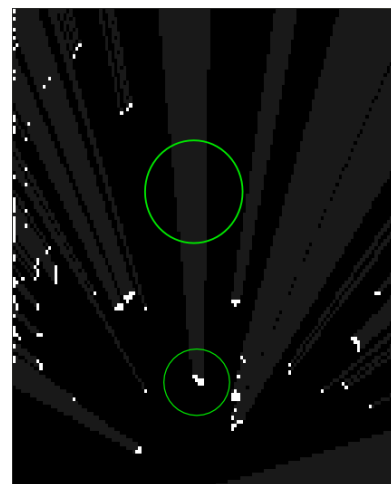


Fig. 2. Lidar output when occluded

The shadowed pedestrian is still clearly visible on the

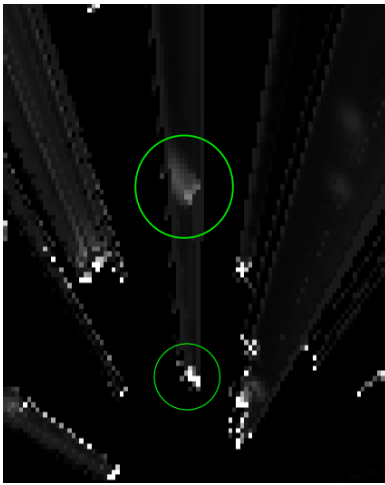


Fig. 3. Occupancy evaluation from our algorithm

occupancy map, although the Lidar cannot get through the first occluding person. Its position spreads over time, which shows that our knowledge decreases with the age of the data. Extensive propagation of cells once perceived occupied naturally leads to this result : occupied cells have not disappeared, although not being visible on the sensor. We thus stress the importance of sensor filtering, taking into account possible spatial and temporal correlation. In this case, consequences of the pedestrian not being visible without filtering has no practical consequences, but this is not always the case and we believe that such filtering would be compulsory for autonomous vehicle navigation in an urban area.

2) *Dynamics estimation*: This example plans to emphasise dynamics estimation capabilities of the algorithm, along with segmentation of the scene. Although this is already possible via bounding boxes in the case of clearly separated persons, those matchings often miss when persons are too close to each other, or when groups change in size due to some people joining and leaving. The exhaustive approach that we carry on a per-cell basis provides an estimation of the probable speeds. All boxes presented on figures *eq. 4 eq. 5 eq. 6* are here on illustrative purpose, and do not come from a detection algorithm.



Fig. 4. Camera capture

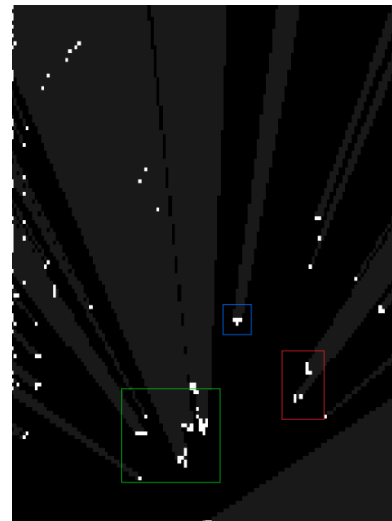


Fig. 5. Lidar sensor output

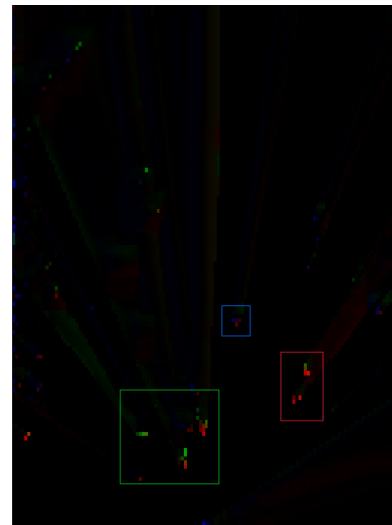


Fig. 6. Output of the algorithm - speed map

Figure *eq. 6* shows the speed map maintained by the algorithm, speed orientation being coded by colour, while speed value (in the car referential) is represented by the brightness. Three boxes have been overlaid by hand on the figure, to emphasise specific cases. In green are the crossing pedestrians, which may have been difficult to track on sensor data alone (cf figure *eq. 5*), due to the pushchair and their proximity. In blue is the road sign, which is obviously standing still, but which shows the residual speed of the vehicle (end of the braking sequence) and noise on Lidar data. It is barely visible on the speed map, due to the very low residual speed of the car. Segmentation of the scene after the algorithm process between moving and still parts proves effective. In red is the car coming on the other way, which is also going slow due to the crossing pedestrians, and have very few Lidar impacts (laser beam was oriented upwards, maybe too much). Tracking on geometrical grounds on Lidar

data alone may have been difficult in this case.

C. Future works

Alternative source of occupancy, velocity or classification are planned, mainly based on vision processing. Stereo-vision is for example a proven source of free-space measure, as shown by *Moravec* once again in [11], or more recently *Badino et al.* using dense disparity calculus and occupancy grids ([7] and articles following). Motion detection and evaluation have also been proven to be a valuable output from stereo-vision capture, *Argrawal et al.* demonstrating in [5] that platform ego-motion could also be removed from the initial optical flux in order to track moving objects. Initial theoretical work on this use would be from *Adiv et. al* ([12]), although this field has received a lot of attention in the past years, notably since dense stereo-vision processing is now possible in real-time. To finish with, the state of the art as regards vision-based SLAM (notably *Davison* [13] [14])) leads us to believe that laser-based sensors could possibly be replaced in the near future for most perception tasks, this being a strong incentive for us to develop visual inputs. Another source of possible improvements would be an evolution to handle collaborative perception, by means of merging grid-based beliefs in our algorithm.

III. CONCLUSION

Perception of moving objects, such as pedestrians, with minimal requirements on their size, moves or behaviour is a difficult task ; which will nevertheless be a key to enable autonomous navigation in urban environments, or even comprehensive assistance on current automotive devices. We proposed a novel technique based on an extension of the traditional occupancy grids registration, using probabilist propagation and extensive consideration of displacements and interactions over a restricted neighbourhood.

An evolution from an initial ambitious proposition, our work shows promising results while being capable of real-time execution, although still being a work-in-progress. An interesting evolution would be its extension towards collaborative perception, which should be easier than for some other approaches due to the grid-based principle being kept. Many challenges would still need to be resolved in order for separated vehicles to take part into one another perception of the environment, among which localisation or synchronisation inaccuracies, but we believe that autonomous transportation and road safety would benefit a lot from such developments.

REFERENCES

- [1] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, vol. 4, no. 3, pp. 380–407, 2004. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.6632&rep=rep1&type=pdf>
- [2] C.-c. Wang and C. Thorpe, "A Hierarchical Object Based Representation for Simultaneous Localization and Mapping A Hierarchical Object Based Representation for Simultaneous Localization and Mapping," *Robotics*, 2004.
- [3] B. Steux and O. El Hamzaoui, "CoreSLAM: a SLAM Algorithm in less than 200 lines of C code," *Submission ICARCV*, 2010. [Online]. Available: http://www-roc.inria.fr/imara/dw/_media/users/oussamaelhamzaoui/coreslam.pdf
- [4] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364907081229>
- [5] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Motion and Video Computing, 2005. WACV/MOTIONS'05 Volume 2. IEEE Workshop on*, vol. 2. IEEE, 2007, pp. 207–214. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4129607
- [6] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 116–121. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087316
- [7] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*. Rio de Janeiro: Citeseer, 2007, pp. 1–12. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.4161&rep=rep1&type=pdf>
- [8] M. Yguel, O. Aycard, D. Raulo, and C. Laugier, *Grid based fusion of off-board cameras*. IEEE, [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1689641
- [9] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364906061158>
- [10] G. Gate and F. Nashashibi, "An approach for robust mapping, detection, tracking and classification in dynamic environments," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5174739
- [11] H. Moravec, "Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids," *Perception*, no. September, 1996. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.9045&rep=rep1&type=pdf>
- [12] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 384–401, Jul. 1985. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767678>
- [13] A. J. Davison and N. Kita, "3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain (PDF)," *Measurement*, vol. 00, no. C, 2001. [Online]. Available: <http://www.computer.org/portal/web/csdl/doi/10.1109/CVPR.2001.990501>
- [14] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," London, pp. 0–7, 2003.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Probabilistic Road Geometry Estimation using a Millimetre-Wave Radar

Andres Hernandez-Gutierrez, Juan I. Nieto, Tim Bailey and Eduardo M. Nebot

Abstract—This paper presents a probabilistic framework for road geometry estimation using a millimetre wave radar. It aims at estimating the geometry of unpaved and unmarked roads, and also provides the vehicle location with respect to the edges of the road. This road tracking system employs a radar sensor due to its robustness to weather conditions such as fog, dust, rain and snow. The proposed approach is robust to noisy measurements because the radar target locations are modelled as Gaussian distributions. These observations are integrated into a Kalman Particle filter to estimate the *posterior* distribution of the parameters that best describe the geometry of the road. Experimental results using data acquired on a highway road are presented. The effectiveness of the proposed approach is demonstrated by a qualitative analysis of the results.

I. INTRODUCTION

Road detection and tracking refers to the process of locating the road surface and its boundaries. Depending on the type of road and environment, this process can be accomplished by gathering information from sensors such as video cameras, laser rangefinders, GPS receivers together with digital maps, and more recently using radar sensors. Road tracking systems are of significant importance in both autonomous systems and in advance driving assistance systems. In these applications, the automobile has to localise itself with respect to the edges of the road in order to perform its next task. The first lane recognition system dates back 25 years when a vision-based approach using the Kalman filter was proposed in [1]. Subsequent versions of this work, which analysed spatio-temporal information and extended the curve estimation to 3D space, were presented in [2] and [3]. These implementations were then followed by the ARCADE algorithm (*Automated Road Curvature And Direction Estimation*) in [4]. It estimates the curvature and orientation of the road based on features extracted from monochromatic images in combination with a model of the road, providing appropriate results for marked roads. A vision-based system for lane recognition, using a particle filter approach, was developed in [5]. A method based on multiple features such as edges, texture and colour extracted from images was presented in [7].

Although the first implementations for road detection were vision-based systems, the detection and tracking of roads has also been attempted by the employment of laser rangefinders. Implementations of these systems are proposed in [8] and

[9]. Techniques that fuse information from multiple sensors are given in [10] and [11], and more recently in [12]. All these implementations provide reliable results in moderate weather conditions; nevertheless, their main disadvantages are the lack of robustness against environmental conditions such as dust, fog, snow and rain in which their performance could be degraded. Therefore, it is necessary to utilise a sensor that is capable of providing reliable information even if adverse environmental factors are present.

Radar sensors are less susceptible to these environmental conditions [13], [14]; hence, automobile manufactures have recently equipped vehicles with radar sensors to perform tasks such as collision avoidance, traffic scene interpretation and dynamic object detection. Unlike a laser rangefinder, which emits a coherent light beam, a radar sensor radiates a wide directive beam that illuminates part of a surface. In general, a measurement at a given bearing angle consists of various returns from targets detected in the radar footprint area; however, due to the sidelobes found in the radiation pattern, highly reflective objects located near the radar footprint area could generate an erroneous range measurement at that bearing angle. This is because all measurements are assumed to be detected at the centre line of the radar beam. Another concern when using a radar sensor refers to clutter signals. In a road tracking system, these signals correspond to targets located far from the edges of the road, radar returns corresponding to dynamic objects or multipath echoes due to multiple reflections. Therefore, a challenging task when using a radar sensor is being able to surmount this extra complexity while taking advantage of its immunity to most of the weather conditions.

The work presented in this paper aims at developing a probabilistic approach for detection and tracking of both unpaved and unmarked roads, with robustness to normal and harsh environmental conditions such as those found in mining roads. The *posterior* probability density function (pdf) of the parameters that best describe the geometry of the road and the vehicle location with respect to the edges of the road are the output of a Kalman particle filter.

This paper is organised as follows. An overview of the proposed algorithm is given in section II. A description of the model used to represent the road is detailed in section III. Section IV explains how the radar observations are modelled as Gaussian distributions, which are subsequently used in section V to estimate the road geometry parameters. Experimental results in section VI are followed by the conclusions in section VII.

Andres Hernandez-Gutierrez, Juan I. Nieto, Tim Bailey and Eduardo M. Nebot are with the Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney NSW 2006, Australia. {a.hernandez, j.nieto, t.bailey, nebot}@acfr.usyd.edu.au

II. ALGORITHM OVERVIEW

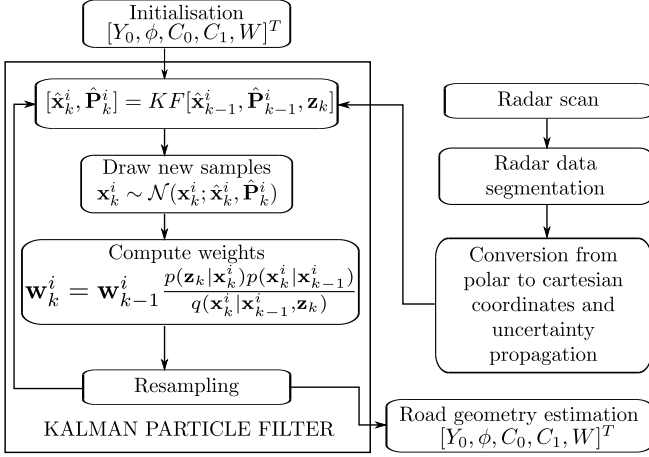


Fig. 1. Block diagram of the road geometry estimation process.

Fig. 1 shows a block diagram of the probabilistic road tracking system proposed in this work. It estimates the relative vehicle location, Y_0 , with respect to the left edge of the road as well as other road geometry parameters, such as the orientation angle of the road, ϕ , with respect to the radar coordinate frame, the curvature, C_0 , the rate of curvature, C_1 , and the width of the road, W .

A radar measurement comprises a 180-degree scan that may contain radar information belonging to objects detected on the road, edges of the road or reflective targets located near the borders. This information is first segmented in order to extract useful radar data corresponding to the boundaries. By analysing the intensity level of the radar returns and signal noise along the complete trajectory, it has been observed that valuable information can be extracted by thresholding, setting the threshold value at $65dB$. Subsequently, a conversion from polar to Cartesian coordinates is applied to this data. This stage is followed by the uncertainty propagation from polar to Cartesian coordinates, which takes into account the uncertainty in range and bearing of the radar sensor.

The road edges are modelled as a pair of clothoid curves parametrised by the elements of $\mathbf{x} = [Y_0, \phi, C_0, C_1, W]^T$. A Kalman particle filter is used to estimate these parameters by generating N hypotheses at each time step k . First, for each hypothesis $\hat{\mathbf{x}}_{k-1}^i$, $i = 1, \dots, N$, the Mahalanobis distance is used to associate the radar observations, \mathbf{z}_k , to the left or right clothoid curve, then after the prediction stage in the Kalman filter, each hypothesis is updated and its mean, $\hat{\mathbf{x}}_k^i$, and covariance matrix, $\hat{\mathbf{P}}_k^i$, are used to build a proposal distribution. This proposal distribution is used to sample new particles. Subsequently, using the transitional density function, the sensor likelihood function and the proposal distribution, the weight for every hypothesis is obtained. Particles having the highest weights are selected in the resampling stage and are used to represent the geometry of the road.

III. CLOTHOID CURVE FOR ROAD GEOMETRY ESTIMATION

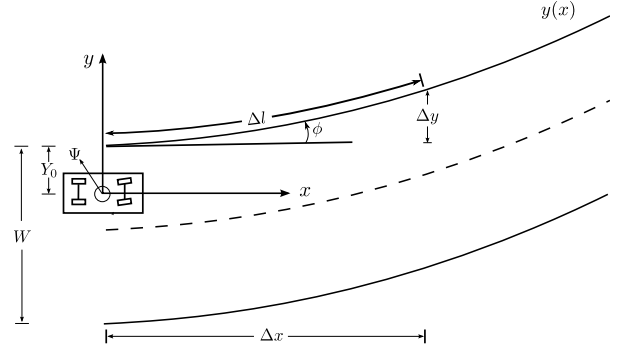


Fig. 2. Road model represented by a cubic approximation of a Clothoid function.

Clothoid curves have been used in civil engineering [15] for road design in order to geometrically join a straight road to a curve road. The main characteristic of a Clothoid function is that its curvature is proportional to the length of the curve measured from its origin [1].

$$C(l) = C_0 + C_1 * l \quad (1)$$

where C_0 and $C_1 = \frac{dC}{dl}$ are the curvature and rate of curvature of the road respectively. Fig. 2 shows a plane curve that is parametrised by the first integral of curvature given by the heading direction $\phi(l)$, and the second integrals represented by $x(l)$ and $y(l)$ listed below. Substituting l by τ for integration purposes, it yields

$$\phi(l) = \phi_0 + \int_0^l C(\tau) d\tau \quad (2)$$

$$x(l) = x_0 + \int_0^l \cos \phi(\tau) d\tau \quad (3)$$

$$y(l) = y_0 + \int_0^l \sin \phi(\tau) d\tau \quad (4)$$

substituting (1) in (2) and solving in terms of τ yields

$$\phi(\tau) = \phi_0(\tau) + C_0\tau + \frac{1}{2}C_1\tau^2 \quad (5)$$

If $\phi(\tau)$ is substituted in (3) and (4), it would require to solve Fresnel's Integrals. Instead, provided the vehicle heading changes are less than 10° , that is, $|\phi| < 10^\circ$, the approximation of $\cos \phi \approx 1$ and $\sin \phi \approx \phi$ remains valid. Assuming this constraint, and solving (3), the parametric function $x(l)$ can be rewritten as

$$x(l) = x_0(l) + l \quad (6)$$

setting $x_0(l) = 0$ leads to $x(l) = l$, and the parametric function $y(l)$ is then given by

$$y(l) \approx Y_0 + \phi l + \frac{1}{2}C_0 l^2 + \frac{1}{6}C_1 l^3 \quad (7)$$

The road model used to represent the left curve in terms of the longitudinal distance x is obtained by substituting $x(l) = l$ in (7) which yields

$$y(x) \approx Y_0 + \phi x + \frac{1}{2}C_0x^2 + \frac{1}{6}C_1x^3 \quad (8)$$

We assume that both the left and right curve are parallel; therefore, the left curve is offset to the right by the width of the road in order to obtain the Clothoid curve that represents the right boundary. The state vector \mathbf{x} to be estimated by the Kalman particle filter is given by

$$\mathbf{x} = [Y_0 \quad \phi \quad C_0 \quad C_1 \quad W]^T \quad (9)$$

where these parameters represent the lateral offset with respect to the left border, the orientation of the road with respect to the vehicle coordinate frame, the curvature, rate of curvature, and the width of the road respectively.

IV. PREPROCESSING RADAR INFORMATION

A single radar return is $\hat{\mathbf{a}} = (r, \theta)$, where r is range and θ is bearing, and is modelled as $\mathbf{a} \sim \mathcal{N}(\hat{\mathbf{a}}, \mathbf{P}_a)$, with mean $\hat{\mathbf{a}}$ and covariance matrix \mathbf{P}_a . For the road estimation process, every radar observation is converted from polar to Cartesian coordinates using the nonlinear transformation $\mathbf{g}(\cdot)$.

$$\hat{\mathbf{z}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \mathbf{g}(\mathbf{a}) = \begin{bmatrix} r \cdot \cos \theta \\ r \cdot \sin \theta \end{bmatrix} \quad (10)$$

The covariance matrix associated with every radar return in polar coordinates is

$$\mathbf{P} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (11)$$

where σ_r and σ_θ is the standard deviation of the uncertainty in range and bearing. Through a linearisation approach, the density of \mathbf{z} is modelled by a Gaussian distribution with mean $\hat{\mathbf{z}}$ and covariance matrix \mathbf{R} .

$$\mathbf{R} = \nabla \mathbf{g}(\mathbf{z}) \mathbf{P} \nabla \mathbf{g}^T(\mathbf{z}) \quad (12)$$

where $\mathbf{g}(\mathbf{z})$ is the Jacobian of $\mathbf{g}(\cdot)$ given by

$$\nabla \mathbf{g}(\mathbf{z}) = \frac{\partial \mathbf{g}(\mathbf{z})}{\partial \mathbf{a}} = \begin{bmatrix} \cos \theta & \sin \theta \\ -r \cdot \sin \theta & r \cdot \cos \theta \end{bmatrix} \quad (13)$$

V. ROAD EXTRACTION USING RADAR DATA

The Kalman particle filter is a recursive Bayesian estimator based on Monte Carlo sampling methods [16]. Particle filter based estimation methods have the advantage of not being subject to any linearity or Gaussianity constraint on both the prediction and observation models [5]. In the *Sequential Importance Resampling* filter *SIR*, the posterior probability distribution is represented by weights \mathbf{w}_k^i associated to a set of particles \mathbf{x}_k^i . The importance weights up to a normalising constant are evaluated as

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} \quad (14)$$

A common proposal distribution is given by the prediction model $p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)$, which gives importance weights as (15).

However, this proposal distribution does not explore the sample space efficiently. It has been demonstrated in [17] that the importance density distribution that minimises the variance of the importance weights, provided \mathbf{x}_{k-1}^i and \mathbf{z}_k , is given by (16).

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i) \quad (15)$$

$$q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k^i, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)} \quad (16)$$

substituting (16) back into (14) yields:

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{x}_k^i, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)} \quad (17)$$

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) p(\mathbf{z}_k | \mathbf{x}_k^i)}{p(\mathbf{z}_k | \mathbf{x}_k^i, \mathbf{x}_{k-1}^i)} \quad (18)$$

because \mathbf{z}_k is conditional independent of \mathbf{x}_{k-1}^i , we obtain

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) p(\mathbf{z}_k | \mathbf{x}_k^i)}{p(\mathbf{z}_k | \mathbf{x}_k^i)} \quad (19)$$

$$\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \quad (20)$$

which expresses that importance weights at time k can be computed before the particles are propagated to time k . This is an important aspect when the vehicle is approaching a road that has a small radius of curvature. In contrast, when using (15), the weights are computed using the propagated particles that were based on the previous estimation. Hence if the road has a small radius of curvature, the propagated particles (Clothoid curves) could probably diverge from where the actual radar observations are being detected. As a result, the filter would not provide an appropriate estimation of the parameters.

An option to estimate the optimal proposal distribution in (16) is applying the Kalman filter which encourages the particles to move into a region of high likelihood. That is, it corrects the particles and compute the covariance matrix for each hypothesis and a new particle is sampled from a Gaussian proposal distribution with mean and covariance given by the updated parameters in the Kalman filter. Although the computational cost when applying the KF to each particle increases, it is compensated by a reduction of the number of samples required to achieve a certain level of performance since the particles have been corrected. For this reason, we have adopted the Kalman Particle filter approach in this work.

The pseudo-code of the Kalman particle filter implemented here is listed in Algorithm 1. The resampling approach used in this algorithm is the stratified sampling. The number of particles N was set to 1000.

Algorithm 1 Kalman Particle Filter

```

1:  $[\{\mathbf{x}_k^j, \mathbf{w}_k^j, \mathbf{P}_k^j\}_{j=1}^N] = \text{KPF}[\{\mathbf{x}_{k-1}^i, \mathbf{w}_{k-1}^i, \mathbf{P}_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$ 
2: for  $i = 1$  to  $N$  do
3:    $[\hat{\mathbf{x}}_k^i, \hat{\mathbf{P}}_k^i] = \text{KF}[\mathbf{x}_{k-1}^i, \mathbf{P}_{k-1}^i, \mathbf{z}_k]$ 
4:    $\mathbf{x}_k^i \sim \mathcal{N}(\mathbf{x}_k^i; \hat{\mathbf{x}}_k^i, \hat{\mathbf{P}}_k^i)$ 
5:    $\tilde{\mathbf{w}}_k^i = \mathbf{w}_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$ 
6:   where  $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = \mathcal{N}(\mathbf{x}_k^i; \hat{\mathbf{x}}_k^i, \hat{\mathbf{P}}_k^i)$ 
7: end for
8:  $t = \text{SUM}[\{\tilde{\mathbf{w}}_k^i\}_{i=1}^N]$ 
9: for  $i = 1$  to  $N$  do
10:   $\mathbf{w}_k^i = t^{-1} \tilde{\mathbf{w}}_k^i$ 
11: end for
12:  $\tilde{N}_{eff} = \frac{1}{\sum_{i=1}^N (\mathbf{w}_k^i)^2}$ 
13: if  $\tilde{N}_{eff} < \frac{N}{2}$  then
14:   $[\{\mathbf{x}_k^i, \mathbf{w}_k^i\}_{i=1}^N] = \text{RESAMPLE}[\{\mathbf{x}_k^i, \mathbf{w}_k^i\}_{i=1}^N]$ 
15: end if

```

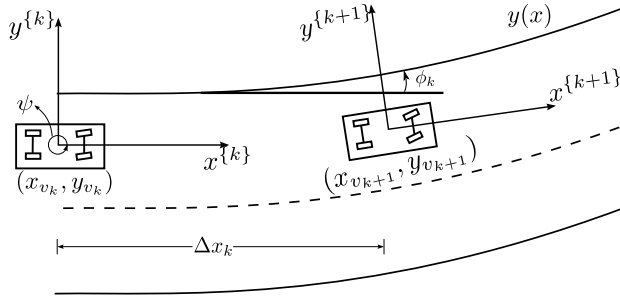


Fig. 3. Road model represented by a Clothoid curve along with vehicle motion representation.

A. Prediction Model

The vehicle model is approximated by the Ackerman bicycle model

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} \Delta x_k \cos \theta_k \\ \Delta y_k \sin \theta_k \end{bmatrix} \quad (21)$$

Fig. 3 shows the egomotion of the vehicle from the local coordinate frame $(x^{\{k\}}, y^{\{k\}})$ to the frame $(x^{\{k+1\}}, y^{\{k+1\}})$. Considering the longitudinal vehicle motion, Δx , and the Clothoid curve presented in (8), the prediction model can be obtained as follows

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\Delta x_k, \mathbf{x}_{k-1}) + \mathbf{u}_{k-1}(\psi_{k-1}) + \mathbf{v}_k \quad (22)$$

$$\mathbf{f}_{k-1}(\Delta x_k, \mathbf{x}_{k-1}) = \begin{bmatrix} 1 & \Delta x_k & \frac{\Delta x_k^2}{2} & \frac{\Delta x_k^3}{6} & 0 \\ 0 & 1 & \Delta x_k & \frac{\Delta x_k^2}{2} & 0 \\ 0 & 0 & 1 & \Delta x_k & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} \quad (23)$$

$$\mathbf{u}_{k-1} = [0 \quad -\psi_{k-1} \quad 0 \quad 0 \quad 0]^T \quad (24)$$

The additive noise \mathbf{v} is a zero mean Gaussian noise with covariance \mathbf{Q} . The vehicle location as well as the yaw rate ψ are obtained from the vehicle navigation system.

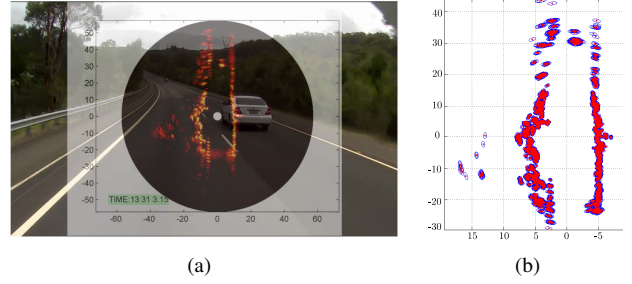


Fig. 4. (a) Raw radar image superimposed on a colour image of the environment. Brighter pixels represent returns from highly reflective targets. (b) Radar observations modelled as Gaussian distributions. Sigma ellipses error are represented by the blue ellipses around each radar target in red.

B. Observation model used in the Kalman Filter

Fig. 4(a) shows a raw radar image as well as a colour image of the environment acquired on a highway road. The radar sensor position is represented by the white circle at the coordinate $(0,0)$. Fig. 4(b) illustrates how the radar observations are modelled as Gaussian distributions after segmenting the raw radar data. Red points represent the radar measurements and the uncertainty of each radar return is represented by the sigma ellipses error in blue.

As can be seen in Fig. 4(a), there are some radar observations detected not only on the road, but also far from the borders. In order to remove these outlier measurements, we can perform gating and data association, which involves identifying the origin of measurements and associating them to a specific particle or Clothoid curve from which they could have been generated. As stated in [18], the ellipsoidal validation gate is optimal for a linear observation model with additive noise

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \quad (25)$$

where the additive noise \mathbf{w} is zero-mean Gaussian. The validity of the measurement \mathbf{z}_k is determined from its Mahalanobis distance from the predicted observation \mathbf{y}_k in the y -direction, and so outlier observations are removed and are not considered in the weighting process for that specific particle.

Because several radar returns detected within a region on the edge of the road are correlated, the assumption of treating them as independent measurements is not valid. Rather, each return detects a point on the surface of the roadside bump, and we must devise a model to map this to the clothoid line. Hence, the radar measurements are first clustered into regions of $5m$ long along each side of the road then those Gaussian distributions found within these regions are fused into single Gaussian distribution or pseudo-observation. Clusters represent a simple surface model that maps the shape of the bump onto the clothoid that runs along the centre line of the bump. It is illustrated by the blue ellipses error on each side of the road in Fig. 5(a) for the clothoid curve represented by the blue lines in Fig. 5(b).

The predicted observations in the Kalman filter are represented by a 1-dimensional vector \mathbf{y}_k . This vectors is constructed by the y -coordinates of the Clothoid curve which are

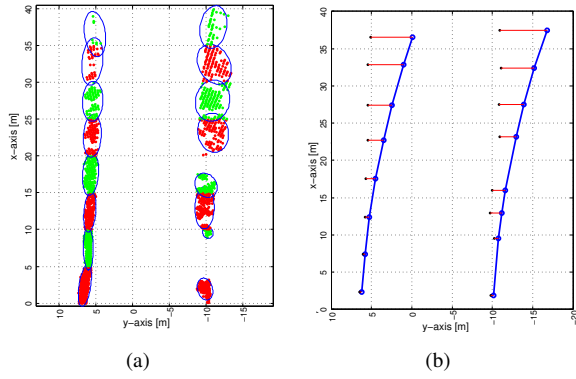


Fig. 5. (a) Fusion of multiple Gaussian distribution into a single Gaussian functions in blue. (b) Centroids of the combined Gaussian distributions represented by the black dots along with a Clothoid curve in blue.

aligned with the y -coordinate of the centroid of the pseudo-observations belonging to the left and right edges of the road. These centroids are represented by black points in Fig. 5(b). Their locations are defined by its x -and- y coordinates on the road plane, that is, $[(xL_j, yL_j), \dots, (xL_{ML}, yL_{ML})]^T$, $j = 1 \dots ML$ and $[(xR_j, yR_j), \dots, (xR_{MR}, yR_{MR})]^T$, $j = 1 \dots MR$. Where ML and MR are the number of pseudo-observations on each edge of the road respectively. The error considered here corresponds to the discrepancy in the y -coordinates. The complete predicted measurement vector \mathbf{y}_k is represented by

$$\mathbf{y}_k = [yL_1 \dots yL_{ML}, yR_1 \dots yR_{MR}]^T \quad (26)$$

The matrix \mathbf{H} that associates the propagated particle with the predicted observations in (25) is given by

$$\mathbf{H}_k = \begin{bmatrix} 1 & xL_1 & \frac{xL_1^2}{2} & \frac{xL_1^3}{6} & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & xL_{M_l} & \frac{xL_{M_l}^2}{2} & \frac{xL_{M_l}^3}{6} & 0 \\ 1 & xR_1 & \frac{xR_1^2}{2} & \frac{xR_1^3}{6} & -1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & xR_{M_r} & \frac{xR_{M_r}^2}{2} & \frac{xR_{M_r}^3}{6} & -1 \end{bmatrix} \quad (27)$$

The elements in the last column in \mathbf{H}_k , whose value is -1 , enable us to obtain the predicted observation on the right edge of the road.

VI. RESULTS

A. Experiment setup and radar specifications

The experiments presented in this work were conducted on a highway road. The driven trajectory comprises about 5km . Besides the radar scanner and a GPS sensor, a stereo camera was used for display purposes. These sensors were mounted on the test vehicle shown in Fig. 6(a). The radar sensor frame rate was 2Hz . The vehicle navigation system provided GPS position per second.¹

¹See available videos at www-personal.acfr.usyd.edu.au/ahernandez/roadEstimation.avi



Fig. 6. (a) Test vehicle used in the experiments. (b) Radar sensor and GPS mounted on the vehicle.



Fig. 7. Radar setup on the test-vehicle.

The specifications for the radar sensor are listed in Table I. Some of these parameters can be associated to the information presented in Fig. 7 which shows the configuration of the radar sensor mounted on the test-vehicle.

TABLE I
RADAR SENSOR SPECIFICATIONS

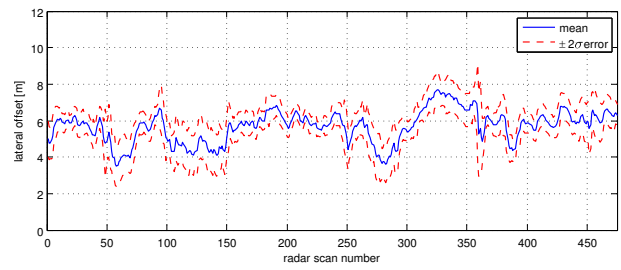
Field of view	360°
Maximum detection range R_{max}	60m
Operation frequency	94 GHz
Uncertainty in range σ_r	0.25 cm
Uncertainty in orientation angle σ_θ	1°
Horizontal beam width γ_h	2°
Vertical beam width γ_v	8°
Height of the radar on top of the vehicle H	2.4 m

B. Raw radar image interpretation

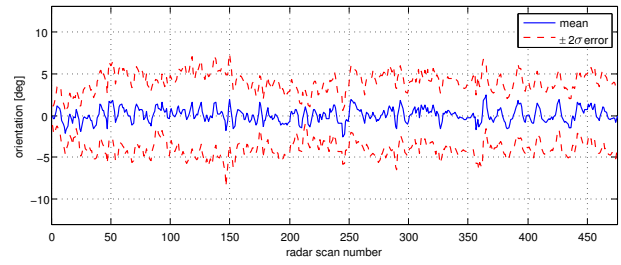
The brightness of features in a radar image depends on the fraction of the radiated energy that is returned back to the radar sensor from targets detected in the environment. The intensity of this backscattered energy is basically dependent on the following aspects:

- 1) Surface roughness of the target.
- 2) Radar viewing and surface geometry relationship.
- 3) Moisture content and electrical properties of the target.

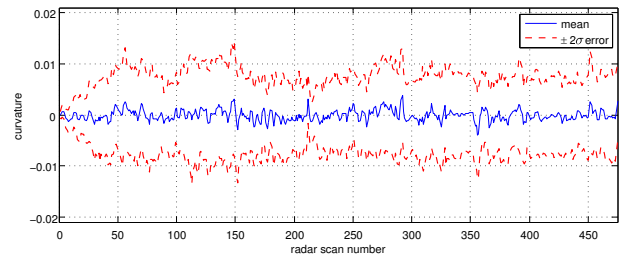
The surface of the road is considered as a smooth area if the height fluctuations are much smaller than the radar wavelength, which is about 3mm for the MMW radar used in the experiment. A smooth surface causes specular reflections of the incident energy, usually away from the sensor; hence, only a small amount of energy is returned to the radar. Therefore, the surface of the road in Fig. 4(a) appears as darker toned areas. In contrast, when the surface height variations approach the radar wavelength, the surface will appear 'rough', then some reflections will be sent back to the radar receiver. Berms located on the sides of the road will strongly reflect the radar signal intensity back to the radar receiver. In this case, the borders will appear as a brighter tone as shown on the right side of the road in Fig. 4(a).



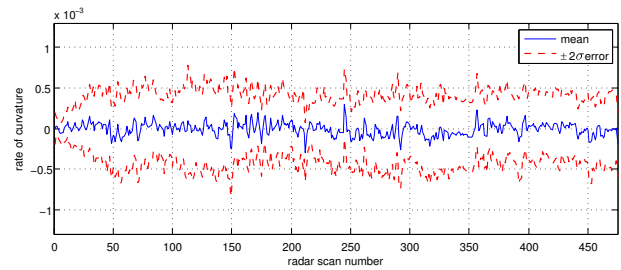
(a) Lateral offset to the left edge of the road along the driven trajectory.



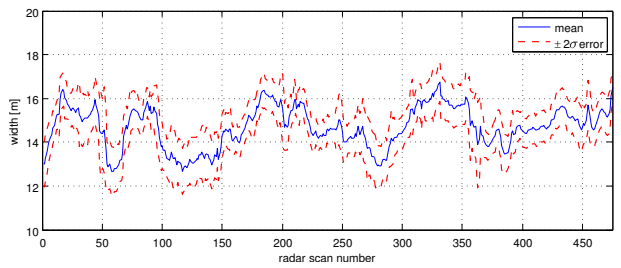
(b) Orientation angle of the road with respect to the radar.



(c) Curvature of the road.



(d) Rate of curvature of the road



(e) Width of the road

Fig. 8. Estimated parameters that best describe the geometry of the road ahead of the vehicle. The $\pm 2\sigma$ confidence interval is represented by the red lines in each Figure.

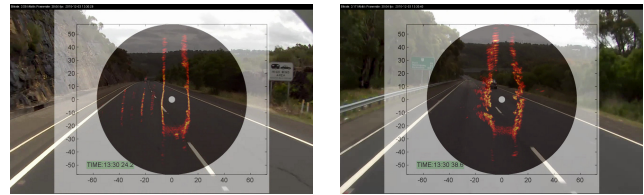
The results presented in Fig. 8 depict the estimated parameters that describe the geometry of the road ahead of the vehicle. Each figure shows the estimated mean as well as the $\pm 2\sigma$ uncertainty area. Fig. 8(a) shows the lateral offset that is measured from the radar sensor to the left edge of the road.

Variations in the lateral offset are smoother in sections of the road where there is either a berm, a rock wall or a guard rail located on the left edge of the road. The uncertainty in the estimation of this parameter decreases in these sections of the road because the radar targets locations are concentrated within a narrower spatial area as it can be seen from the scans number 10 to 50 in Fig. 8(a). The estimation of the lateral offset for this section of the road correspond to the environment illustrated in Fig. 9(a).

Fig. 8(b) illustrates the orientation of the road with respect to the radar coordinate frame. In this case the confidence interval increases when the vehicle is turning. An example of this issue can be seen from the scans number 100 to 150. Fig. 8(c) and Fig. 8(d) show the curvature and rate of curvature of the road. The values for these estimated parameters are very close to zero because the curves along this particular road are not highly curved. From Fig. 8(c), it is also observed that the confidence interval starts to increase when the vehicle is approaching a curve. An example of this aspect can be also seen from the scans number 100 to 150. Fig. 8(e) shows the width of the road. The actual width of the lane is approximately 8m. The width of each road shoulder on each side of the road is about 2m. However, the mean of the estimated parameters fluctuates from 12.5m to 16.5m because the road estimation was based on targets such as berms, guard rails, rock walls, and trees located on the sides of the road, rather than detecting the marking lanes.

Fig. 9(a) shows some multipath echoes on the left side of the road due to multiple signal reflections from the wall rock to the vehicle. By applying validation gate, the proposed algorithm discriminates these multiple signal reflections as they could affect the road estimation process if these returns are detected close to valid targets, such as those belonging to the wall rock.

Fig. 10 illustrates a sequence of four consecutive frames. As can be seen in Fig. 10(h), most of the radar returns on the left side of the road belong to the berm located about 8m on the left and trees detected beyond the road boundaries. This as can be seen in the colour image in Fig. 10(g). At this step, the algorithm provides a suitable representation of the road. However, in Fig. 10(f), the estimation of the road is slightly affected by the lack of radar measurements near the left edge of the road. In this case, the algorithm estimates the edges of the road based on observations detected on the right road boundary. As soon as radar targets are detected on



(a)

(b)

Fig. 9. (a) Radar measurements obtained when a rock wall is located on the side of the road. (b) Radar measurements that correspond to targets such as trees or vegetation near the edges of the road.

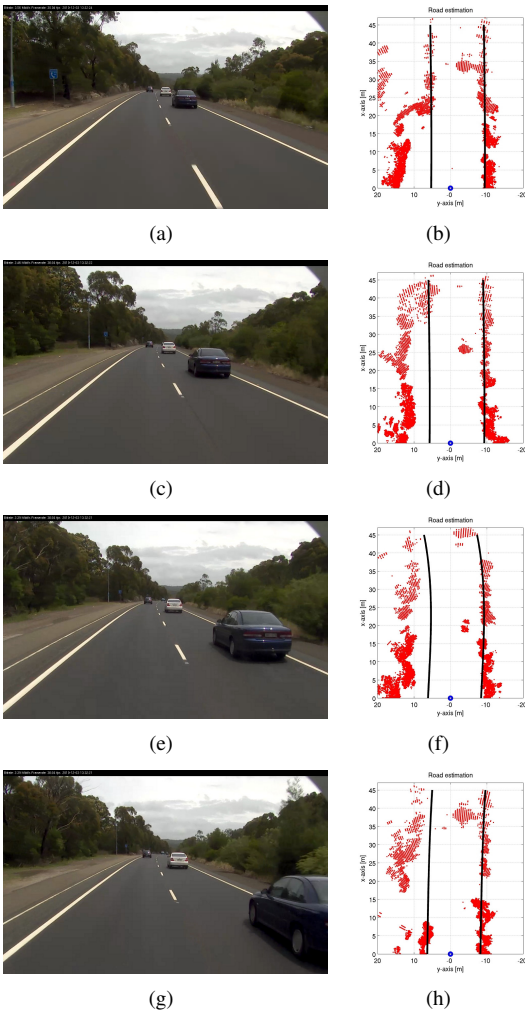


Fig. 10. Consecutive radar scans that show when the road estimation is slightly affected by the absence of a berm or a wall near on either edge of the road. However, the estimation of the road is corrected as soon as radar returns are detected on the left side again as shown in Fig. 10(d).

the left side of the road the estimation is corrected as shown in Fig. 10(d) and Fig. 10(b). When the radar sensor does not receive any return from either side of the road, the estimation of the road is based on the prediction model unless no radar information is detected for more than a number of scans. In such a case, the curvature, C_0 , and the rate of curvate, C_1 , are set to zero and the other parameters are initialised to their previous estimated values.

VII. CONCLUSIONS AND FUTURE WORKS

This paper presented a probabilistic approach for road estimation using a radar sensor. The radar measurements were modelled as Gaussian distributions and integrated to a Kalman particle filter, and it estimated the parameters that best described the geometry of the road using a Clothoid function to model the shape of road boundaries.

Experimental results with data acquired on a highway road validated the proposed approach. The obtained results are encouraging considering that no special work (such as adding infrastructure) was performed on the berms limiting

the road. This implies that results could be improved by proper maintenance of the road.

Although the assumption of considering both edges or the road as parallel curves provided encouraging results, in future work, we will propose to estimate the geometry of the road modelling each road boundary independently.

VIII. ACKNOWLEDGEMENTS

We would like to thank to Javier Martinez, Vijay Nichani and David Orchansky from the Australian Centre for Field Robotics for their valuable assistance with the experiments presented in this paper.

This work has been partially supported by CONACYT and SEP Mexico, the Australian Centre for Field Robotics and the New South Wales State Government.

REFERENCES

- [1] E. D. Dickmanns and A. Zapp, "A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision", *SPIE*, vol. 727, 1986, pp 161-168.
- [2] E. D. Dickmanns, B. Mysliwetz and T. Christians, "An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles", *Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, 1990, pp 1273-1284.
- [3] E. D. Dickmanns, B. D. Mysliwetz, "Recursive 3-D Road and Relative Ego-State Recognition", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, 1992, pp 199-213.
- [4] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features", *Proceedings of Intelligent Vehicles Symposium*, 1994, pp 109-114.
- [5] B. Southall and C. J. Taylor, "Stochastic road shape estimation", in *ICCV Eighth International Conference on Computer Vision*, vol. 1, 2001, pp 205.
- [6] M. Isard and A. Blake, "ICondensation: unifying low-level and high-level tracking in a stochastic framework", in *ECCV European Conference of Computer Vision*, 1998.
- [7] N. Apostoloff, "Vision based lane tracking using multiple cues and particle filtering", Master's thesis, Australian National University, 2005.
- [8] R. Aufrere, R. Chapuis, and F. Chausse, "A fast and robust vision based road following algorithm", *Proceedings of Intelligent Vehicles Symposium*, 2000, pp 192-197.
- [9] W. Wijesoda, K. Kodagoda, and A. P. Balasuriya, "Road boundary detection and tracking using lidar", *Transaction on Robotics and Automation*, vol. 20, 2004, pp 456-464.
- [10] R. Aufrere, C. Mertz, and C. Thorpe, Multiple sensor fusion for detecting location of curbs, walls, and barriers, *Proceedings of Intelligent Vehicles Symposium*, 2003, pp 126-131.
- [11] H. Loose, U. Franke, and C. Stiller, "Kalman Particle Filter for Lane Recognition on Rural Roads", *Proceeding of Intelligent Vehicles Symposium*, vol. 1, 2009, pp 60-65.
- [12] M. Tsogas, N. Floudas, P. Lytrivis, A. Amditis, and A. Polychronopoulos, "Combined lane and road attributes extraction by fusing data from digital map, laser scanner and camera", *Informat. Fusion*, 2010.
- [13] G. Brooker, R. Hennessey, C. Lobsey, M. Bishop, and E. Widzyk-Capehart, "Seeing through Dust and Water Vapor: Millimetre Wave Radar Sensors for Mining Applications", *Journal of Field Robotics*, vol. 24, no. 7, 2007, pp 527-557.
- [14] J. Ryde and N. Hillier, "Performance of Laser and Radar Ranging Devices in Adverse Environmental Conditions", *Journal of Field Robotics*, vol. 26, no. 9, 2009, pp 712-727.
- [15] R. Lamm, B. Psarianos, and T. Mailaender, *Highway design and traffic safety engineering handbook*, McGraw-Hill; 1999.
- [16] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House; 2004.
- [17] V. S. Zariitskii, V. B. Svetnik, and L. I. Shinelevich, "Monte Carlo Technique in Problems of Optimal Data Processing", *Auto. Remo. Cont.*, vol. 12, 1975, pp 95-103
- [18] T. Bailey, B. Upcroft, and H. Durrant-Whyte, *International Conference on Information Fusion*, 2006, pp 1-6.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Safety robotic lawnmower with precise and low-cost L1-only RTK-GPS positioning

Jean-Marie CODOL^{1,2,3}, Michèle PONCELET¹, André MONIN^{2,3}, Michel DEVY^{2,3}

Abstract—In this paper, we will introduce an autonomous robotic lawnmower, equipped by a safety and low-cost RTK-DGPS centimetric positioning system available also in semi-urban environment. The GPS-RTK sensors are a pair of L1-only GPS receivers (L1-only GPS receivers are cheaper than dual-frequency ones because of the existence of patents on the usage of the second frequency). This work is an extension of a collaboration between NAV ON TIME and BELROBOTICS, consisting on evaluate GPS replacement for the current mower area limit (a buried wire). The objective of the latest work is to ensure the GPS mission realization, keeping the same safety as the buried wire one. In this context, this paper will present a complete statistical approach to L1-only RTK-positioning system in urban environment. The result of this approach have been embedded into the mower machine, by using a Linux operating system equipped with an ARM-9 processor running at 400MHz, and an UHF radio-communication to the reference station, this one having the role of realize path planning, geographical database managing, remote and IHM communication.

I. INTRODUCTION

Navigation system is a critical work in autonomous robotic systems. Robotic critical applications need guarantees in terms of precision, integrity, safety and availability. To reach this goal, we propose here to take into account these aspects in a statistical approach for the design of a low-cost RTK navigation system. This work was supported by NAV ON TIME and the LAAS-CNRS. In a first part we will present the original machine and the available solutions for GNSS navigation. Then the positioning design who guarantees the previous concepts, in particular the integrity one. Before to conclude presenting a 24 hour experiment.

II. MOBILE ROBOT DESIGN

A. BIGMOW machine

BIGMOW lawnmower robot was designed by BELROBOTICS. The genuine robot evolves on area limited by a buried wire: the area limit is detected by a wire sensor embedded into the mower. This sensor allows the machine to perform a turn around when the area limit is reached, and automatic return to charging operation (by following the wire). This robot is widely used by professional gardeners. Additionally, this robot is safe, so it can evolve in human environment, the safety is guarantee by ultrasound and contact sensors, placed in the front of the machine. The hardware and primary software architecture are presented in figure 2. Our experiments were done on an semi-urban terrain in Toulouse,

France, presented in photos of figure 1. This terrain is ideal (because of it is a little more difficult to real final application) when we know that these final applications will be to mow golf courses, or sports terrains.



Fig. 1. The tests were performed on an urban terrain near Toulouse, France. The main difficulties was to navigate near the buildings.

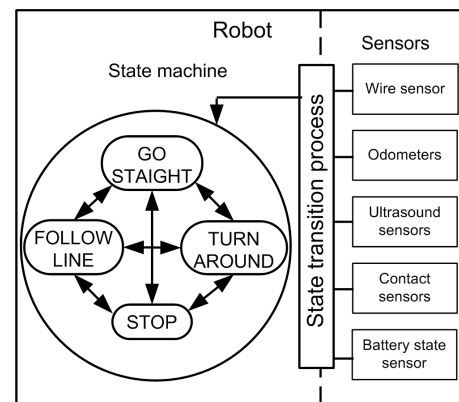


Fig. 2. The machine guidance internal states are presented on the left side of the figure. Transition between these states depends on the sensors measurements, by an algorithm (the state machine transition process).

III. DGPS FOR MOBILE ROBOTS

Standard GPS positioning precision in urban environment is incompatible with mobile robot operation, therefore we propose to use DGPS-RTK technique. DGPS consists on a couple of receiver, one static and the other embedded on the machine, the DGPS positioning process needs both GPS

¹NAV ON TIME, 42 Av. du Général Decrouette, 31100 Toulouse, France
²CNRS;LAAS; 7 Av. du colonel Roche, F-31077 Toulouse Cedex 4, France
³Université de Toulouse ; UPS, INSA, INP, ISAE ; UT1, UTM, LAAS ; F-31077 Toulouse, Cedex 4, France

measurements to perform measurements differentiation. RTK technique consists in solving integer ambiguities on each carrier phase measurement tracking loop. In this context, one access to centimetric positioning. This positioning is relative, Figure 3 presents the differential pseudorange measurement $\Delta\rho_t^i$ model with respect to geometrical constraints such as

$$\begin{aligned}\Delta\rho_t^i &= p_t \cdot u_t^i + \Delta b_t + \varepsilon_{\rho,t}^i \\ \Delta\varphi_t^i &= p_t \cdot u_t^i + \lambda a^i + \Delta b_t + \varepsilon_{\varphi,t}^i\end{aligned}$$

where $\Delta\rho$ and $\Delta\varphi$ stands for the differential pseudorange and carrier phase measurements (between mobile and reference), p stands for the position, u the unit vector supporting the line of sight, Δb the differential receiver clock bias, a the integer carrier phase tracking loop ambiguity, λ is the carrier phase wavelength, t and i the time and satellite indexes.

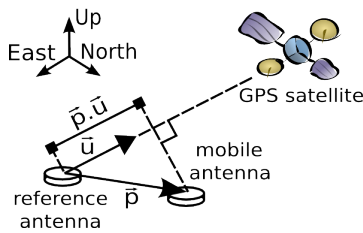


Fig. 3. Differential GPS positioning eliminates the atmospheric delays. By differencing the pseudorange measurements between the mobile and the reference receivers, we estimate the relative antenna positioning in an accuracy way. Here p is the relative positioning and u is the unit vector from the baseline to the satellite direction. The differential pseudorange measurement contains term $p \cdot u$ added to the clock bias, but not atmospheric delays.

For a short baseline (less than 10km), the pseudorange measurements give us access to metric positioning precision, and tens of meters integrity (the same precision is reached using EGNOS SBAS). The usage of carrier phase dual-frequency receiver in a DGPS way permit to reach centimeter accuracy and integrity[1][2][3]. Thus mono-frequency receivers can be used to perform the same precision performances as dual-frequency one, despite an initialization time (some minutes)[4]. During this initialization period, one have access to positioning with real typed ambiguity estimate (by evaluating integer ambiguities as real typed values, table I resume the GPS and DGPS precision and integrity capabilities. Finally Precise Point Positioning technique (PPP)[5], allows integer ambiguity resolution by using precise satellites monitoring, but this method need a worldwide expensive infrastructure.

TABLE I
EXPECTED POSITIONING PRECISION AND INTEGRITY

Method	Precision (m.)		Prot. level (m.)		delay (y/n)
	urban	open	urban	open	
GPS	50	5	100	10	n
DGPS	20	2	50	5	n
RTK L1/L2	0.02	0.02	0.05	0.05	n
RTK L1	0.02	0.02	0.05	0.05	y
PPP	0.02	0.02	0.05	0.05	y

Here protection level stands for a probability of being outside the given area being $1e^{-5}/hour$. The solution presented here use low-cost mono-frequency DGPS RTK, in urban environment. The improvement relies here in RTK integrity monitoring.

IV. INTEGRATE THE NAVIGATION SYSTEM INTO THE EXISTING PLATFORM

To integrate the DGPS RTK navigation helper into the BIG-MOW machine, we plugged on board an ARM-9 card running at 400MHz with an embedded Linux operating system. This navigation element contains also one mono-frequency GPS receiver, an interface to machine guidance system, and also a radio communication (to communicate with the reference station). Figure 4 presents the hardware and communication architecture. This architecture have been patented by NAV ON TIME.

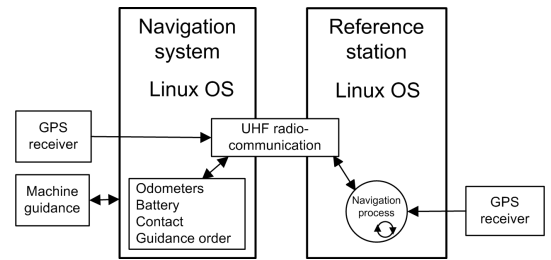


Fig. 4. Our navigation system architecture consist on performing the global navigation processes on the reference station. Measurement grabbing and transmission is done on the robot. This allow also a short time autonomy in the case of short connection lost time.

We have chosen to perform the navigation process (including path planning, positioning and monitoring) on the reference station for two reasons: first this allows to monitor multiple machines, then because the navigation process needs a lot of resources (processing power for carrier phase ambiguity resolution, for path planning, and memory for the geographical database ...). These elements does not need to be embedded on each machines. The reference station perform path planning, positioning, monitoring ... and send back to the embedded system, periodically, the navigation information. The navigation information contains the real-time guidance orders. Thus, on BIGMOW machine, the guidance orders are compliant with existing 'follow the wire' orders.

V. RTK-GPS AND INTEGRITY

By using integer carrier phase ambiguity resolution, we perform centimetric RTK positioning. The advantage of using mono-frequency receiver instead of a multi-frequency one is the price (we consider a factor 100 between them). Thanks to this advantage, mono-RTK (mono-frequency RTK) is a challenge in robot navigation. Here we perform a complete statistical approach of this method, allowing robustness capability of the navigation system. We will see, in a first part, how to construct an integer carrier phase ambiguity search space, and then how to perform the elimination test on each of these hypothesis. The contribution in this paper is the usage

of explicit simple statistical tests to perform the RTK process in presence of multipath.

A. Floating ambiguity for RTK

Construct integer carrier phase ambiguity search space construction is divided in two ones : first estimating the ambiguities as real typed values(in a Gaussian least square context), then the integer hypotheses set construction. The search space is the set of integer values in an interval. We will study this method. Let us consider the state vector X containing robot pose R and real types ambiguities A , as Gaussian random variables and let P the X corresponding to variance-covariance matrices.

$$\begin{aligned} X &= [R, A]^T \\ \hat{X} &\sim (\bar{X}, P) \end{aligned}$$

So we can define here our linear system

$$\begin{aligned} X_t &= FX_{t-1} + w_t \\ y_t &= HX_t + v_t \end{aligned}$$

with F the dynamic system matrix, H the observation matrix, y the measurement vector, w and v Gaussian noises respecting $\mathbb{E}[ww^T] = Q$ and $\mathbb{E}[vv^T] = R$. In [6], authors propose to estimate ambiguities by a least square minimisation procedure on pseudorange and carrier phase measurements $p(A_t|\varphi_{0:t}, \rho_{0:t})$. In urban environment, due to multipath effect on pseudorange, it is more robust to use only carrier phase measurements. Given real typed ambiguities estimates A_t , one access to integer values by choosing, for each ambiguity term a_i , an interval I where $\int_I p(a_i)da = P_i$. We construct, in this interval I , the list of integer hypothesis $\{\hat{a}_i\}$. Knowing $a_i \in \mathbb{Z}$, one have $\sum_i p(a_i \in \{\hat{a}_i\}) = \int_I p(a_i)da = P_i$. So the integer ambiguity search space is a N -combination of integers. The probability of containing the true ambiguities in our search space is $P = \prod(P_i)$. Note that considering N_i hypothesis on ambiguity term i , the search space contains $N = \prod_{i < N} N_i$ hypothesis.

In figure 5, we can see 2D case of the integer extraction. If the real typed estimated ambiguities are correlated, we can perform a smart ambiguity search (eliminating the ambiguity when the Mahalanobis distance exceed a rejection thresholds), we can see a 2D example in figure 5. If the test fails, we do not expand the sub-tree for this hypothesis.

This method is not the one used in our positioning process. Instead, we construct triple difference carrier phase measurement[7][8]. These measurement are ambiguities free. So the ambiguity search space is constructed by a different process.

B. Triple difference for RTK

Recently, some improved triple difference positioning filters[8][9]. These filters reach real-time performances, allowing embedded positioning process directly into robots. Triple difference filter does not deal with real typed ambiguity terms. Thus construct the integer search space is done by a variable change

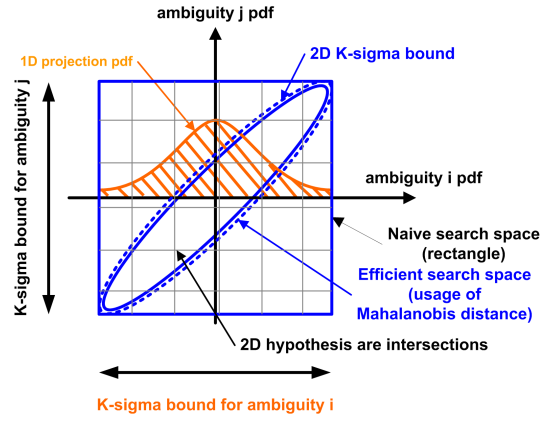


Fig. 5. The search space for a couple of ambiguities is, on each ambiguity 1D space, the integers values into an interval where the probability is controlled. If the real types estimates values of ambiguities are correlated, so a naive search is not the optimal one.

$$\begin{aligned} \Delta\varphi_t^i &= p_t \cdot u_t^i + a^i + \Delta b_t + \varepsilon_{\varphi,t}^i \\ a^i &= \Delta\varphi_t^i - p_t \cdot u_t^i - \Delta b_t - \varepsilon_{\varphi,t}^i \end{aligned}$$

We can deduce for each satellite double difference carrier phase measurement (difference between measurement i and a pivot p), a Gaussian estimate of the ambiguity term a^i . We construct the same search space as presented in previous title, about floating ambiguity. By this way, the probability of not containing the right ambiguity set in the search space is controlled statistically.

C. Integer ambiguity resolution : Gaussian noise case

The integer ambiguity resolution latency is a known problem, called the Time To Fix (TTF). Clearly, TTF is the time needed to eliminate all the wrong hypothesis in the search space (including removing of all the hypotheses if the true one is not in our search space). We chose to applied a simple hypothesis test (primary test) at each period T . The null hypothesis (H_0 is the right ambiguity set) is tested. We consider the primary tests independent, and so we construct a N -binomial test.

If the tested hypothesis contains the true ambiguities set, the residual norm statistical distribution is a centralized Chi-2 distribution. Given a PFA (Probability of False Alarm or 'first kind'), one can determine a threshold for primary test. We perform this test multiple times and we eliminate the hypothesis of the search space if this primary test fails more than F times on N tests. So, if the primary tests are independent, the probability of eliminating H_0 follows being in the search space is $p(\text{elim}(H_0))$ and probability of keep a wrong hypothesis H_i , $p(\text{keep}(H_i))$ are

$$\begin{aligned} p(\text{elim}(H_0)) &= \sum_{F < k < N} \mathbb{C}_N^k PFA^k (1 - PFA)^{N-k} \\ p(\text{keep}(H_i)) &= 1 - \sum_{F < k < N} \mathbb{C}_N^k PND_i^k (1 - PND_i)^{N-k} \end{aligned}$$

where PND is the probability of non-detection (or error of second kind). PND_i depends on PFA and the distribution of H_i residual norm (see figure 6(a)).

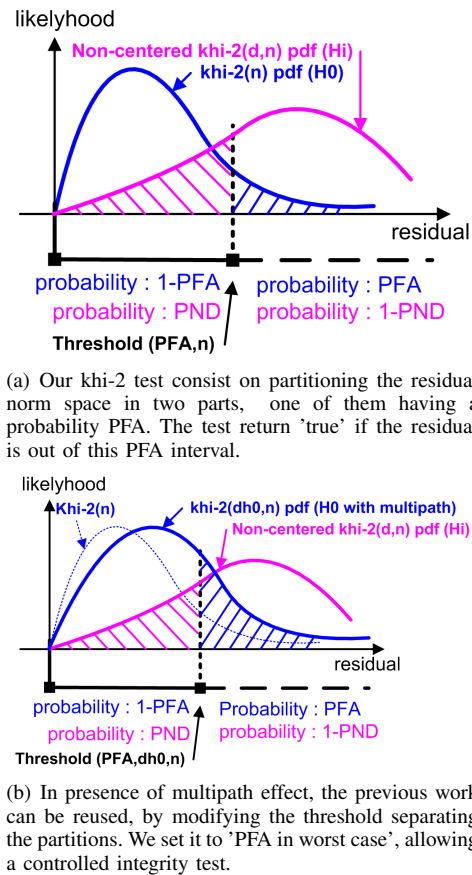


Fig. 6. Summary of Khi-2 test. Here $Khi-2(n)$ represent a n degree centered khi-2 distribution, and $Khi-2(d,n)$ represent a n degree, d center, non-centralized khi-2 distribution

D. Integer ambiguity resolution with Multipath

In the case of multipath, each carrier phase measurements residual norm distribution became a non-centralized khi-2 distribution, and the noises are time-correlated. The first problem is eliminated by taking a different threshold on the primary test (we fix PFA and deals with the worst-case distribution to keep an upper bound of the test power). The second problem is solved by modifying primary tests period, taking account time to prevent correlation in multipath effect (some seconds is sufficient in dynamic case). The figure 6(b) show the new primary test modelization.

To conclude, the key of our work is to define the PFA value and to use the worst case model for carrier phase noise spectrum. So the probability to not solve the true integer ambiguity (the risk) is controlled.

VI. RESULTS

The validation of the new robotic lawn-mower navigation system was done on some lands in France. A successful demonstration was done in October 2010, near the stadium of Toulouse, including a 'slalom path' around plots.

The terrain presented in figure 1 is entirely mowed by the new robot since many months. In figure 7 we can see a 24 hour squared trajectory (the not aligned lines are trajectory to return to charging operation).

Although the ground truth is not available, the 'return to charging' operation need the robot to reach an electrical plug in a range of ± 5 cm. This operation is realized with success during many weeks, at a frequency of one return to charging of two hours every two hours.

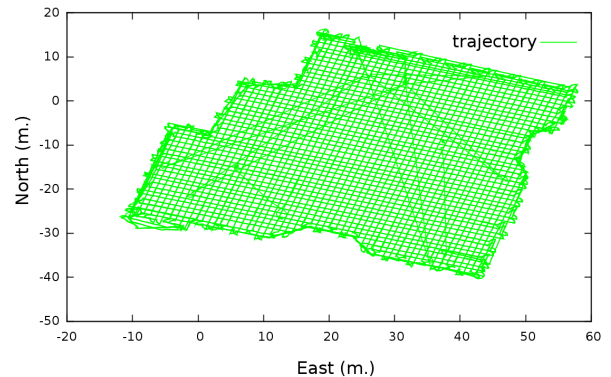


Fig. 7. The mowed terrain coverage during 24 hours, including the automatic return to station procedure.

VII. CONCLUSION

To conclude, we presented here a robotic lawn-mower system guided by a low-cost RTK-GPS navigation system. The positioning system have been seen as a stochastic process, this way permitting to control the integrity. Finally the main result was presented : a 24 hours autonomous mowing, including the 'return to charging' processes. This navigation system should be reused in many kind of robot, including area security or mobile transport systems.

REFERENCES

- [1] B. Parkinson, J. Spilker, P. Axelrad, and P. Enge, "Global Positioning System: Theory and Applications Volume II," *Progress in astronautics and aeronautics*, vol. 163, 1996.
- [2] D. Kim and R. Langley, "GPS ambiguity resolution and validation: methodologies, trends and issues," in *Proceedings of the 7th GNSS Workshop-International Symposium on GPS/GNSS, Seoul, Korea*, vol. 30, no. 2.12, 2000.
- [3] C. Tiberius and P. De Jonge, "Fast positioning using the LAMBDA method," in *Proceedings of the 4th International Symposium on Differential Satellite Navigation Systems DSNS*, vol. 95. Citeseer, 1995, pp. 24-28.
- [4] H. Kim and H. Lee, "Real-Time Implementation of L1 RTK System Based on Position-Domain Hatch filter."
- [5] S. Bisnath and Y. Gao, "Current state of precise point positioning and future prospects and limitations," *Observing our Changing Earth*, pp. 615-623, 2008.
- [6] P. De Jonge and C. Tiberius, "The LAMBDA method for integer ambiguity estimation: implementation aspects," *Publications of the Delft Geodetic Computing Centre*, vol. 12, 1996.
- [7] B. Remondi and G. Brown, "Triple differencing with Kalman filtering: making it work," *GPS Solutions*, vol. 3, no. 3, pp. 58-64, 2000.
- [8] M. Petovello, K. O'SKeefe, G. Lachapelle, and M. Cannon, "Consideration of time-correlated errors in a Kalman filter applicable to GNSS," *Journal of Geodesy*, vol. 83, no. 1, pp. 51-56, 2009.
- [9] J. Codol and A. Monin, "Improved triple difference GPS carrier phase for RTK-GPS positioning," *International Symposium on Signal Processing SSP-2011, Nice France*, 2011.

Odometry from Planar Landmarks

Keerthi Narayana and Bruno Steux

Abstract—This paper presents a new perception odometry approach using extracted stationary planar features to resolve 5 degrees of freedom of the robot motion. The approach exploits the geometrical properties of the extracted features to determine the transformation of the moving robot, which has perceived these landmarks. This way of localizing can help several applications in indoors and outdoors such as urban canyons, with plenty of planar features. The paper presents the concept and the algorithm, and validates them using a simulated scenario.

I. INTRODUCTION

Several approaches are proposed in the recent years to provide assistance and solutions to the 3D localization (pose estimation) problem. Most solutions use Global Navigation Satellite Systems (GNSS) and Inertial Measurement Units (IMU). However, pose computed from GNSS receivers degrade especially in urban and indoor environments, where satellite signal reception is perturbed by manmade structures. Consequently, Inertial Measurement Units (IMU) are a good alternative to fill the short gaps in GNSS-based pose. However, the pose computed from these sensors subjected to drift errors, and a good quality IMU costs a lot.

In the indoor robotic navigation, the technique of Simultaneous Localization and Mapping (SLAM) is used for localization. It uses perception sensors to create the map of the environment, and this information is used either to correct the estimated pose or compute the robot transformation. The first implementation is a Bayesian filter approach ([1], [2], [3]), where the predicted pose from the additional sensors such as odometers are corrected using the constructed maps, containing mainly a set of selected features (landmarks) from the scene. Alternatively, robot transformation can also be computed by comparing two overlapping scans taken at two different instants of time. A featureless scan correlation ([4], [5]) is performed to resolve transformation using an optimization formulation.

However, the Bayesian approach already incorporates the errors of the additional sensors. All the accumulated uncertainties over a period time, causes erroneous associations between landmarks in the constructed map and the new observations, resulting in the failures of the approach ([2]). On the other hand, featureless approaches are computationally heavy and the optimization formulation is nondeterministic.

K. Narayana was with the center of robotics (CAOR), Mines ParisTech, 60 Bld Saint Michel, 75006 Paris, France keerthi2narayan@gmail.com

B. Steux is with the center of robotics (CAOR), Mines ParisTech, 60 Bld Saint Michel, 75006 Paris, France Bruno.Steux@mines-paristech.fr

Therefore, we propose odometry (dead reckoning) approach, where certain landmarks are extracted from the scene, and from the geometrical properties of these stationary landmarks observed at two instants of time, the relative pose of the robot is computed. A similar approach is presented in [7], where a 2D transformation is estimated from the circular landmarks. These landmarks are extracted from the tree trunks in a parking area. We choose planar features as the landmarks, since they are one of the most recurring features of the manmade environments. Moreover, their time invariant geometrical properties can be used for both identifying the correspondence between two observation sets (Data Association), and to compute the undergone robot 3D transformation.

II. 3D TRANSFORMATION

Most SLAM problems are tackled only in the 2D space. This is mainly due to the associated complex 6 Degrees of Freedom (6DoF) in the 3D space, and the unknown associations between the landmarks. Moreover, there is a lack of 3D pose sensors, apart from GNSS receivers and Inertial Measurement Units (IMU). Our approach addresses the 3D pose resolution using the geometrically invariant properties of the extracted features, i.e. planar landmarks.

We tackle the localization problem in a 3D space with abundant stationary planar features. These features are observed by the moving observation platform such as a robot. A way to extract such planar features from the mobile platform is published in our previous work [6].

A. 3D Rotation

In order to compute a 3D rotation of a observing platform, a 3D reference frame is needed which can depict the undergone rotation. Planar features, such as building facades can be characterized by their normal vectors pointing in the direction perpendicular to their surface. By using two non parallel planes, the desired 3D reference frame can be constructed, as shown in figure 1.

Once the 3D rotation frame is determined at a given observation epoch \mathbf{k} , and the same can be constructed in another epoch $\mathbf{k} + \delta$, from the same pair of planes \mathbf{P}_1 and \mathbf{P}_2 , the undergone rotation $\mathbf{R}_{\mathbf{k}}$ of the observing moving platform can be determined, as shown in equation 1. The assumption is that the planes are stationary with respect to the moving mapping platform between the two observation epochs.

$$\left. \begin{aligned} \mathbf{U}_{\mathbf{k}+\delta} &= \mathbf{R}_{\mathbf{k}}^{-1} \mathbf{U}_{\mathbf{k}} \\ \text{where} \\ \mathbf{U} &= [\vec{n}_1 \quad \vec{n}_2 \quad \vec{n}_3] \end{aligned} \right\} \quad (1)$$

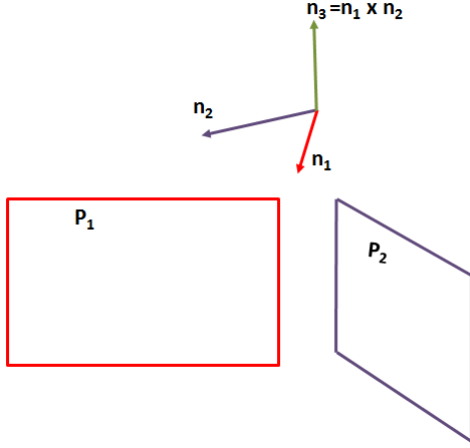


Fig. 1. Construction of 3D rotational frame from two non parallel planes, P_1 and P_2 using their respective normal vectors n_1 and n_2 . Third normal n_3 is computed by the vector cross product between the two normals.

$U_{k+\delta}$ and U_k given in equation 1, are constructed from the observed and associated pair of planes of the two observation epochs, as shown by U .

Manmade environments often have two or more non parallel planes in the observable scene. In such scenarios, the 3D rotation of the observing sensor platform can be computed from a pair of non parallel planes identified across the two observation epochs. The advantage of this rotation computation is, it remains independent to the translation of the platform. Therefore, it can be resolved independently.

B. 3D Translation

3D translation computation demands a stationary 3D reference point in a fixed reference frame. However, for the moving observing platform, this point appears to have undergone a translation same in magnitude but reverse in direction.

Planar landmarks are uniform surfaces and not necessarily comes with a reference 3D point. At many instants even the observed patches of these landmarks vary in their border length and surface areas, depending on the pose of the platform. A way to determine the reference point is to use the intersection points of the planar features. As shown in figure 2, a 3D reference point can be constructed only from limited combinations of the planar intersections.

Moreover, observing non parallel planes to the ground is feasible only in some rare scenarios. Most manmade environments have erect walls, and the horizontal roofs are not visible for a ground vehicle. However, an alternative way to determine the 3D reference point is to use a sensor such as a digital camera.

In our present approach which depends only on a 2D laser scanner setup, as presented in [6], we decided to opt to resolve the translation in 2D space. The 2D reference point ϖ is determined by projecting the intersecting line on to the horizontal plane as shown in the last sub figure in the

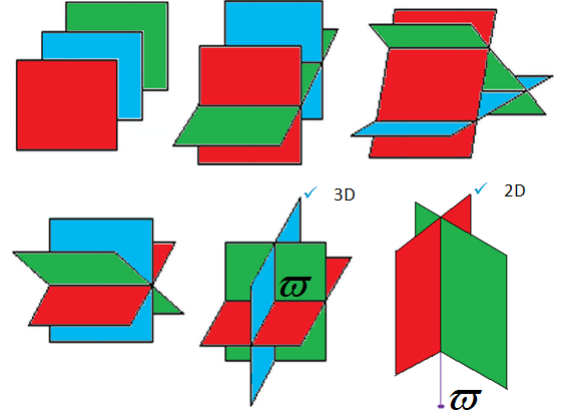


Fig. 2. Construction of 3D rotational frame from two non parallel planes, P_1 and P_2 using their respective normal vectors n_1 and n_2 . Third normal n_3 is computed by the vector cross product between the two normals.

figure 2. The translation vector T_k of the platform, can be expressed using equation 2.

$$\varpi_{k+\delta} = \varpi_k - T_k \quad (2)$$

Again, to determine the 2D translation, we need two non parallel planes, and such scenarios are often present in manmade environments. The critical issue is to determine the corresponding pair of non parallel planes across the two observation epochs.

III. DATA ASSOCIATION

As explained, to compute 5DoF transformation (3DoF rotation and 2DoF translation) in 3D space, a pair of non parallel planes needs to be identified across the two observation sets. The identification of the corresponding element across two sets is widely known as Data Association (DA).

Uniform planar features do not come with any uniquely identifiable tags. Moreover, planar features appear and disappear from the observable scene, as the robot moves, making the identification problem non trivial. Additionally, we opted to compute the robot pose independent of other sensors; therefore, restricting the association to be based only on the geometrical properties of the observed landmarks.

A. Relative Constraints

The dihedral angles between a pair of observed planes is conserved regardless the observing position of the robot. This property is computed for each pair of the observed planes in an epoch. The first level of association is done by comparing this property of the pair of planes for the two observation sets. All parallel plane pairs are discarded. However, results often contain several associations including some ambiguous ones, as it is likely to have the same angular relationships for more than one pair of planes in the scene.

B. Absolute Constraints

To minimize these erroneous associations, a constraint based on the knowledge that a land vehicle on a manmade roads do not move abruptly, can be applied. This implies, for the two observation sets done within a short time, the corresponding planes appear closer to each other. Moreover, if they are projected on one another, they have some overlapping regions. This conditions are tested using a distance criteria (spatial neighborhood) and a spatial alignment test for overlapping regions.

C. Lenient and Strict Data Associations

The planar features help separate the rotation computation from translation resolution. Once rotation is resolved the associations can be done relatively easier compared to the state, where the entire undergone transformation is unknown. Therefore, even the DA can be performed in two stages.

The absolute constraints are applied on observations across the two sets, therefore, the margin for the criteria varies depending upon the motion of the robot. If no information is available about the undergone motion of the robot, a higher magnitude threshold needs to be applied to provide margins for all the unknowns. In such case we call the DA, Lenient Data Association (LDA).

If the rotation is resolved prior computing translation, at this stage, a more stringent condition can be applied to reduce ambiguities in associations. This step we call it as Strict Data Association (SDA). It also employs algorithms to ensure injective associations, meaning, an associated plane is retained only with the most suitable match in the other set.

IV. HANDLING OUTLIER ASSOCIATIONS

As mentioned before, the proposed approach, simplifies the 6DoF pose problem by not only separating the rotation from translation, but also by splitting the Data Association in two, LDA and SDA. We term this as a Divide and Conquer (D&C) approach.

However, the lenient constraints can lead into several ambiguous associations. Additionally, different associated planar landmark pairs can compute results with varying precision, mainly due to the distance of these planes and their orientation to the observing platform. Therefore, a strategy to handle multiple associations is essential. In the presence of multiple associated planar pairs, several transformation solutions can be computed. However, in the presence of ambiguous or outlier associations the common choice of mean-based approach often fails. Therefore, we devised a new algorithm, to choose an associated pair of planes which optimally describes the undergone transformation of the robot.

The algorithms, Optimal Candidate Selection by Consensus (OCSC) and its weighted variant (WOCSC) are inspired from the RANSAC algorithm [8]. The algorithm OCSC like RANSAC tries to choose the best candidate solution by a maximizing function. However, RANSAC is nondeterministic, as it tries to select the best fitting model parameters to a given large population by randomly sampling a minimal set,

till a satisfactory solution can be computed using this random set. However, in the deterministic OCSC each candidate minimal set generates a solution, and if the solution is the one best fits all the members of the associated set then the solution is retained. The OCSC algorithm has two steps, an expectation step and a consensus step. The expectation step generates a candidate transformation from a pair of associated planes, and the consensus step, applies this result on all the associated planes, from the set of planes of one of the epoch. The optimal transformation is the one which produces the best fit (minimum error) for all the associated planes across two epochs.

The problem is formulated using equation 3.

$$\mathbf{S}_{\mathbf{k}+\delta} = \mathbf{K}_{\mathbf{k}} \nabla \mathbf{S}_{\mathbf{k}} \Rightarrow \forall i \in \{1 \dots p\}, \mathbf{x}_{\mathbf{k}+\delta}^i = \mathbf{K}_{\mathbf{k}} \nabla \mathbf{x}_{\mathbf{k}}^i \quad (3)$$

where, \mathbf{S} is a varying population observed at two epochs \mathbf{k} and $\mathbf{k} + \delta$. In our case, it corresponds to the set of planar landmarks. The variation can be described using an operand $\mathbf{K}_{\mathbf{k}}$ and operator ∇ . Each member of the population \mathbf{x} undergoes the same transformation. For the rotation, $\mathbf{K}_{\mathbf{k}}$ is the rotation matrix $\mathbf{R}_{\mathbf{k}}^{-1}$ and ∇ is matrix vector multiplication. For the translation, $\mathbf{K}_{\mathbf{k}}$ is the translation vector $-\mathbf{T}_{\mathbf{k}}$ and ∇ is vector addition.

Expectation step generates the candidate $\mathbf{K}_{\mathbf{k}}^j$ from a minimal subset, in our case, an associated pair of non parallel planes \mathbf{j} . In the consensus step, the optimal candidate $\mathbf{K}_{\mathbf{k}}^*$ is selected, as shown in equation 4.

$$\mathbf{K}_{\mathbf{k}}^* = \arg \min_{\mathbf{j} \rightarrow \mathbf{K}_{\mathbf{k}}^j} \sum_i ((\mathbf{K}_{\mathbf{k}}^j \nabla \mathbf{x}_{\mathbf{k}}^i) - \mathbf{x}_{\mathbf{k}+\delta}^i)^2 (\mathbf{w}_{\mathbf{k}}^i \cdot \mathbf{w}_{\mathbf{k}+\delta}^i) \quad (4)$$

The improvement of WOCSC with respect to OCSC is the term $(\mathbf{w}_{\mathbf{k}}^i \cdot \mathbf{w}_{\mathbf{k}+\delta}^i)$, which takes for each element, the confidence \mathbf{w} into account. Each planar landmarks are considered to have an associated confidence (refer CPEF on [6]). It shall be noted that only the associated pairs \mathbf{i} of observations $(\mathbf{x}_{\mathbf{k}}^i, \mathbf{x}_{\mathbf{k}+\delta}^i)$ are used in the consensus, and not all the observed planes from epochs \mathbf{k} and $\mathbf{k} + \delta$.

The algorithm, handles outliers and reduces noise by choosing the optimal solution. In its general form, it can be applied to solve any overdetermined system.

V. DIVIDE AND CONQUER APPROACH

The overall approach used is summarized in figure 3.

VI. EXPERIMENTAL RESULTS

A. Test Scenario

The concept and the algorithms are validated using a simulation platform. The platform helps to know the true trajectory traversed by the robot, and it provides the laser scanner range measurements (refer [6]) from each known pose of the robot, which is the input for our algorithms. Another advantage is that these inputs can be altered by adding configured noises. These algorithms are at present implemented in MATLAB. The used trajectory and the scene are shown in figure 4.

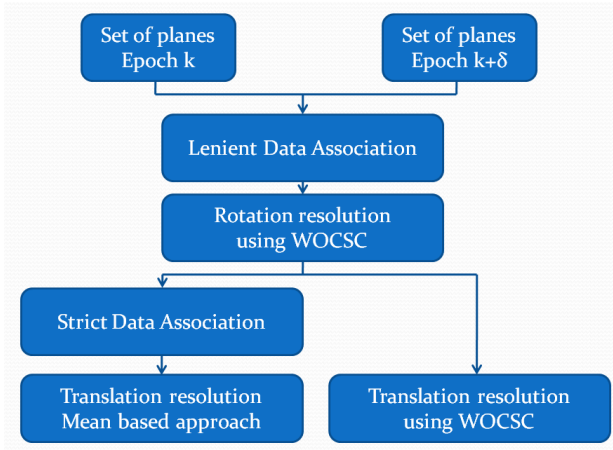


Fig. 3. This provides the overall view of Divide and Conquer (D&C) approach, where rotation is separated from translation. First a Lenient Data Association (LDA) is performed to tackle all unknowns of the undergone transformation. Therefore, rotation is resolved using Weighted Optimal Candidate Selection by Consensus (WOCS) algorithm. Translation can be resolved using two ways, either using a mean based approach or again by applying WOCS. If mean based approach is applied, it is a must to ensure that there is no outlier associations, and this is done by using Strict Data Association (SDA). SDA is optional if we apply WOCS algorithm.

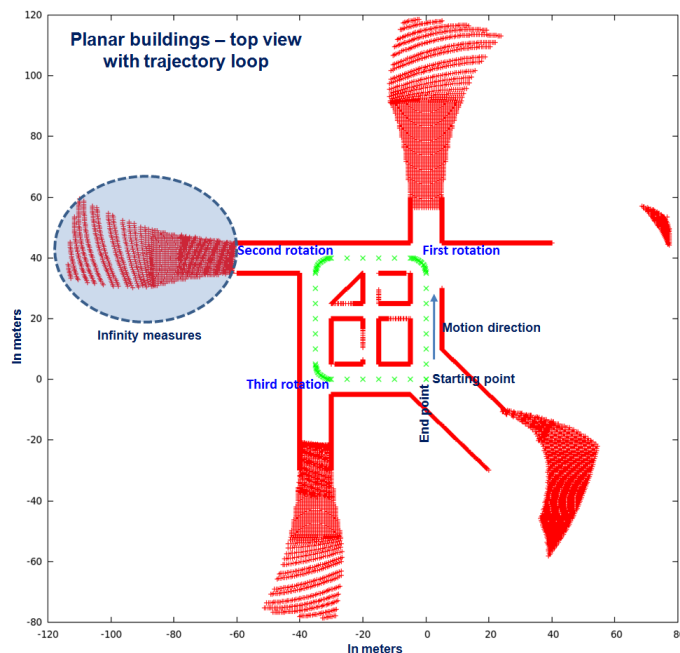


Fig. 4. Top view of the scene generated from the measured laser scans, shown in red dots. Infinity measures are the laser scan measurements for far distance, we discard in our algorithm. Straight red lines shows the walls, corresponding to planar features. Green dots, traversing a loop, is the true trajectory.

We generated test data with four different noise levels, which is added to each laser range measurement. First, a no noise case $\sigma = 0$. Second $\sigma = 6mm$, corresponding to the precision of the SICK LMS 221 laser scanners, we tested inside our laboratory. Third, $\sigma = 2cm$, corresponding to the precision of a low quality laser scanners, and the fourth $\sigma = 10cm$, a worst case scenario.

B. Test Results

Figure 5 shows the four trajectory results obtained for the four different input data generated using the same scene described in figure 4 with explanation in its caption.

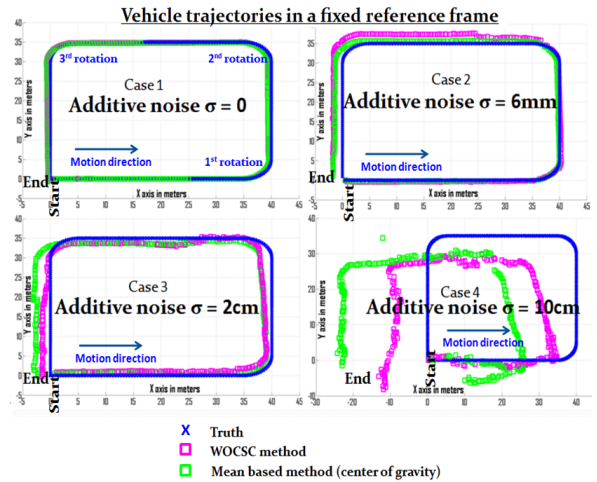


Fig. 5. Four trajectories generated using four data sets with different noise levels are shown in the four sub figures. All the trajectories follow the basic shape but as expected, higher the noise higher the trajectory error. In a fixed reference frame (trajectory) the accumulated error appears as a drift. Each of the four cases depicts the truth in blue cross, trajectory generated from the algorithm sequences (LDA,WOCS Rotation,SDA,Mean Translation) in green, and (LDA,WOCS Rotation,WOCS Translation) in magenta. As discussed before, WOCS approach (second sequence) performs better as the noise level increases, as shown by the plots in magenta. The error in the mean based approach in the last sub figure originates from the erroneous data association, result of the higher noisy data. Higher noises alters the estimated planar properties, which then contributes to the robot pose error. While computing the trajectory, these errors (non systematic if originates from DA ambiguities) gets accumulated resulting in drift errors.

The error characteristics obtained for the 5DOF relative pose estimation using our approach is summarized in figures 6 and 7. In the table the standard deviation (σ) is the important information providing the precision of the computed relative pose. σ for the pitch rate increases rapidly with the noise level, compared to the other two rotation rates. This is because, the planar landmarks in direction perpendicular to the vehicle motion (incidence angle 0° for the laser range measurements in our setup [6]), gets the full magnitude of the laser measurement noise. σ for the cog method gets higher compared to WOCS approach showing its sensitive to the outlier associations. A noise with $\sigma = 10cm$, added to each laser range measurement, alters the estimated planar landmark properties. This change in the planar properties leads to a higher σ for the computed pose, especially in the cluttered environments.

	Mean roll rate error (deg/epoch)	Standard deviation (deg/epoch)	Mean pitch rate error (deg/epoch)	Standard deviation (deg/epoch)	Mean yaw rate error (deg/epoch)	Standard deviation (deg/epoch)
No noise	-2.85506E-15	9.51929E-14	1.03948E-14	1.83304E-13	6.35649E-04	1.62722E-02
Noise $\sigma=6\text{mm}$	-5.53156E-03	9.68693E-02	-2.72707E-03	1.75711E-01	8.34166E-04	1.14980E-01
Noise $\sigma=2\text{cm}$	-2.12593E-02	3.18210E-01	-3.00451E-02	5.53417E-01	8.14208E-04	2.72763E-01
Noise $\sigma=10\text{cm}$	-7.16991E-02	1.12940E+00	-4.70674E-02	2.38289E+00	8.36368E-03	7.65798E-01

Fig. 6. Angular Rate Precision - 3DoF

	Mean velocity errors (meters/epoch)				Std deviation (meters/epoch)			
	WOCSC		Cog		WOCSC		Cog	
	X	Y	X	Y	X	Y	X	Y
No noise	1.46903E-03	1.75199E-03	1.44865E-03	2.16769E-03	1.48305E-02	1.38290E-02	1.54236E-02	1.52860E-02
Noise $\sigma=6\text{mm}$	5.97738E-03	-6.33657E-03	5.94436E-03	-1.41951E-04	9.64136E-02	1.09878E-01	7.19298E-02	7.19869E-02
Noise $\sigma=2\text{cm}$	4.98491E-03	7.96491E-04	7.79074E-03	7.90890E-04	1.68949E-01	2.00265E-01	1.98067E-01	1.68686E-01
Noise $\sigma=10\text{cm}$	4.02695E-02	2.79718E-02	7.82631E-02	5.10067E-03	6.99082E-01	7.16818E-01	1.33231E+00	1.27200E+00

Fig. 7. Translation Precision - 2DoF

At present, we have applied no noise reduction filters, neither for the laser range measurements nor for the trajectory estimation, except the estimation of the plane is done by a least squares sense (refer [6]). Additional noise reduction filters, higher precision and higher rate perception sensors can help in a better estimation of the trajectory.

VII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The concept of odometry approach using planar landmarks is promising to address the current short comings of the localization techniques. It can address the 5DoF 3D pose problem without using any other additional sensor information. To determine the 5DoF pose, it is sufficient to have a pair of non parallel planes, observed in both epochs between which transformation is estimated. We also presented a new algorithm called Optimal Candidate Selection by Consensus (OCSC) and its weighted variant (WOCSC). They can be applied to resolve ambiguities in associations by handling outliers and noisy associations, while doing so chooses the optimal transformation undergone by the robot. The algorithms in its general form can be applied to solve any overdetermined system. We validated the concept using a simulated data with different levels of additive noise.

B. Future Works

Additional validations of the algorithm can be done including real data scenarios. The computation of the pose, solely from the perception sensor information means a higher trust on them. Therefore, a better quality perception sensors can help. There are higher quality 3D laser scanners such as Velodyne High Definition Lidar (HDL), which can facilitate better landmark feature extractions, thus reducing the pose errors. Acceleration and vibration of the robot can be the other major sources of laser measurements errors. Higher perception rate or lowering the speed of the robot or a 'stop and go' motion, can help to minimize such errors.

For resolving the remaining vertical translation (6th DoF), an additional digital camera can be used. The typical drift error observed in the scenarios with noisy inputs can be addressed by using a higher value for $\delta > 1$. That means, the Data Association is performed not between the immediate two sets of data (i.e. $\delta = 1$) but with the data observed with a certain delay. This increased delay ensures a higher signal to noise ratio, which when averaged by the used time delay, acts as a filter, helping to reduce the noise, and therefore, the drift. To use a higher δ value, landmarks need to remain observable for a longer period. The two isoclinal pairs of laser scanner setup, pointing forward and backward direction of the vehicle achieves this (refer [6]). The other major obstacle for this approach is the absence of non parallel planar landmarks in the observing scene. The additional camera or a 3D laser scanner with ability to process landmarks other than the planes can be very useful. Even the empty spaces (where there is no detectable objects present) can be used as landmark patterns.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the Center of Robotics (CAOR) from Mines ParisTech for providing the platform and resources to realize this work, and the reviewers' comments.

REFERENCES

- [1] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, & M. Csorba. "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp 229-241, 2001.
- [2] M. Montemerlo, S. Thrun. "Simultaneous localization and mapping with unknown data association using FastSLAM", *In Proceedings of IEEE International Conference on Robotics and automation*, 2003, vol. 2.
- [3] J. Weingarten, R. Siegwart "EKF-based 3D SLAM for structured environment reconstruction", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005). Proceedings*, 2005, pp 2089-2094.
- [4] F. Lu & E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans", *Journal of Intelligent and Robotic Systems*, vol. 18, no. 3, pp 249-275, 1997.
- [5] B. Steux & O. El Hamzaoui, "CoreSLAM: a SLAM Algorithm in less than 200 lines of C code", *Proceedings of the 11th Control Automation Robotics & Vision (ICARCV)*, pp 1975-1979, 2010.
- [6] K. Narayana, F. Goulette & B. Steux, "Planar Landmark Detection Using a Specific Arrangement of LIDAR Scanners", *Proceedings of the Position Location and Navigation Symposium 2010*, USA, 2010.
- [7] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments", *PhD thesis, Australian Center for Field Robotics, University of Sydney*, 2002.
- [8] M.A. Fischler & R.C. Bolles. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp 381-395, 1981.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Probabilistic autonomous navigation using Risk-RRT approach and models of human interaction

Jorge Rios-Martinez¹, Anne Spalanzani², Christian Laugier¹

¹INRIA Rhone-Alpes, Grenoble, France

²UPMF-Grenoble 2 - INRIA - LIG

{jorge.rios-martinez, anne.spalanzani, christian.laugier}@inrialpes.fr

Abstract—Autonomous transportation in human environments must follow social conventions. An autonomous wheelchair, for example, must respect proximity constraints but also respect people interacting, it should not break interaction between people talking, unless the user want to interact with them. In this case, the robot (i.e. the wheelchair) should find the best way to join the group. In this paper, we propose a risk-based navigation method which include risk of collision but also risk of disturbance. Results exhibit new emerging behavior showing how the robot takes into account social conventions in its navigation strategy.

Index Terms—Proxemics, Human aware navigation, risk assessment.

I. INTRODUCTION

Robots enter more and more into human environments. As areas of mobile service robotics and robotic assistance of humans are becoming more common in everyday life, humans need to share the physical space with robots and robots need to take into account the presence of humans. To be accepted, robots must behave in a socially acceptable way. Their trajectories must be safe but also predictable. Their behavior should follow social conventions, respecting proximity constraints, avoiding people interacting or joining a group engaged in conversation without disturbing.

People maintain a set of social conventions related to space when they are interacting for example in a conversation [1]. The sociology literature often refer to the concept of personal space proposed by Hall [2] which characterizes the space around a human being in terms of comfort to social activity. Concerning interactions between people, an o-space is described in the sociology literature. This space models casual conversations between people interacting [1]. The perception of the territorial boundaries established by group of humans and the respect of these bounds is an evidence of social behavior. Moreover transporting a human restrict us to respect social conventions when navigating, then if we want to develop social robots or wheelchair like robots, the notion of human-human interaction must be explicitly addressed. The ideas presented in this paper will be implemented in a wheelchair to transport people with reduce mobility (PRM) in airports (fig. 1). Assistance to mobility in airports is a big challenge, in [3] many scenarios of application could be found mainly to restore to the users autonomy and privacy,

for example, some PRMs reported that they felt abandoned in PRM areas in the departure lounge, and were worried that they might miss their flights. The interaction between the wheelchair and the user poses also many challenges due to the fact that passengers present varying disabilities (including visually impaired and deaf and hard of hearing), but a solution for this problem is not discussed in this paper.

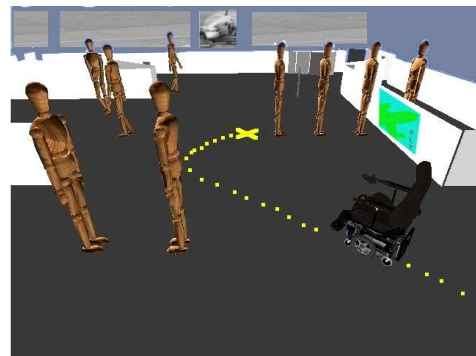


Fig. 1. Autonomous transport of people with reduced mobility in airports is a clear application of our strategy which takes in account human-human interaction. In this figure we can see a representation of the depart lounge of an airport, some social conventions could be observed, face to face interaction and staying in the queue. The wheelchair must navigate in the environment while respecting the cited conventions.

In this article, we propose a simple way to estimate the o-space in the case of two agents interacting based on their positions and orientations and propose an approach to take advantage of it in autonomous navigation of a wheelchair.

Section II proposes a state of the art of human aware navigation. Section III defines the concepts of spatial behavior and describes proxemics models used to take decisions in our navigation system. Section IV describes the algorithm of navigation called Risk-RTT and explains the extensions done. In section V the simulation of the navigation of an autonomous wheelchair in presence of humans interacting is presented. Section VI presents conclusions about the work and perspectives.

II. STATE OF ART

Considering the literature, we can observe the growing interest of the robotics community in research that includes

This work has been partially supported by CONACYT 250140/308006.

the behavior of humans and its impact in the development of tasks by the robot. In [4] it is argued that moving in easily understood and predictable ways will both improve people's trusting and comfort with the robot as well as will help to insure the safety of people moving near the robot.

In [5] the authors propose a method for a robot to join a group of people engaged in conversation. The results of the implementation and the experiments conducted with their platform show a human-like behavior as judged by humans. Some approaches [6]–[11] have been conducted to establish the rules that probably will govern the physical behavior of robots regarding interaction with humans. Closer to human aware navigation and management of physical space, we could mention [12] where a motion planner is presented which takes explicitly into account its human partners. They introduce criteria based both on the control of the distance between the robot and the human, and on the control of the position of the robot in the human's field of view.

In [13] an adaptive system for detecting whether a person seeks to interact with the robot based on the person's pose and position is introduced. In [14] a framework for representing social conventions as components of a constraint optimization problem is presented. A* path planner is implemented with constraints like shortest distance, personal space and pass on the right.

Work presented in [15] propose Spatial Behavior Cognition Model (SBCM), a framework to describe the spatial effects existing between people and people, and people and environment, SBCM is used to learn and predict behaviors of pedestrians and for helping a service robot to take navigation decisions.

In almost all the cited works the concept of personal space is present but the concept of o-space and f-formations have not been included explicitly. We think these last concepts can give us a clue to consider the interactions between the dynamic obstacles in the environment and to improve the autonomous navigation by a better understanding of humans management of space.

III. CONCEPTS OF SOCIAL BEHAVIOR

To understand the perceived behaviors in human-human interaction and the resulting management of space, we can support us on the works developed in the area of sociology to define some concepts as personal space, o-space and F-formations.

A. Personal Space

The term Proxemics was proposed for Hall [2] to describe the use of space between humans, he observed the existence of some rules not written that conducted the people to keep distances from others, and others respect this space, he proposed that space around a person in social interaction is classified as follows:

- the public zone > 3.6m,
- the social zone > 1.2m
- the personal zone > 0.45m
- the intimate zone < 0.45m

That definition is important because it represents a useful tool for a robot to understand the intentions of the humans. It is well known that these measures are not strict and that they change depending on age, culture and type of relationship but the categories proposed explain very well reactions like the uncomfortable sense of a stranger invading your intimate zone or the perception of somebody looking social interaction because he is entering to your social zone. In general people are more strict regarding their frontal space. In the rest of the article we use personal space as synonymous of personal zone plus intimate zone.

The model that we have implemented to represent personal space is defined in [16], it consist in blending two Gaussian functions both of them centered in the position of the person. The first one represents the personal space situated in front of human and for this reason it is wider than the last one representing the back space. The figure 2 shows an example of personal space for two people walking, the measures are projected in the plane of floor, the values obtained from the gaussian are higher in the center than in the borders.

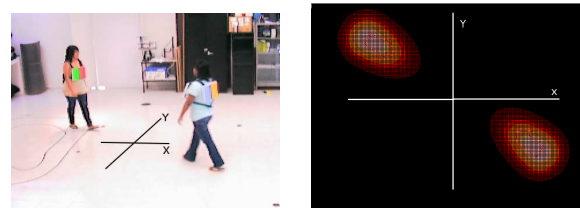


Fig. 2. Estimated personal space for two people that walk projected in the floor.

B. F-formations

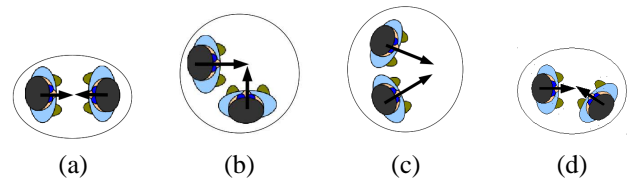


Fig. 3. Examples of F-formations: (a) Vis-a-vis, (b) L-Shape, (c) C-Shape, (d) V-Shape.

In [17] Kendon proposed that people interacting in groups follow some spatial patterns of arrangement. When people are executing some activity they claim an amount of space related to that activity, this space is respected by other people and Kendon referred it as individual's *transactional segment*. This transactional segment can vary depending on body size, posture, position and orientation during the activity. Moreover the groups can establish a joint or shared transactional segment and only the participants have permitted access to it, they protect it and others tend to respect it. The *o-space* is that shared transactional segment reserved for the main activity. This space is surrounded by a narrower one, called the *p-space*, which provides for the placement

of the participant's bodies and also personal things. An *F-formation* system is the spatial-orientation arrangement that people create, share and maintain around their o-space. To become a member of a formation of this sort, you have to be in the p-space.

C. Model of o-space in F-formations

As there is not an exact physical definition of o-space we will describe in this section how we can estimate its location. When more than two people are in conversation they exhibit an F-formation with circular shape then the o-space could be taken as a circle whose center coincides with that of the inner space. In the case of two people some F-formations have been identified as the most frequent [1]. In our model, the o-space will be dependent from the particular F-formation identified: Vis-a-vis, L-Shape, C-Shape or V-Shape (fig. 3). The definition found in the reference mentioned before permits to get a geometric representation for each F-formation, based in the position and orientation of the body of participants.

Given the positions of pedestrians $H_1 = (x_1, y_1)$ and $H_2 = (x_2, y_2)$ in the plane of the floor and their respective orientations ϕ_1 and ϕ_2 around the normal to that plane, we calculate D_H as the euclidean distance between H_1 and H_2 . We calculate also a point V_i as the intersection of the vectors beginning in H_1 and H_2 in the direction of ϕ_1 and ϕ_2 , respectively. Let H_{12} be the mean point between H_1 and H_2 . Let C be the mean point between V_i and H_{12} . Calculate D_i as the distance between V_i and H_{12} .

The o-space could be represented by a two-dimensional Gaussian function Γ_c of covariance matrix S and centered in C , then for each point Q around the center we have:

$$\Gamma_{C,S}(Q) = e^{-\frac{1}{2}(Q-C)^t S^{-1}(Q-C)} \quad (1)$$

where S is a diagonal covariance matrix defined as:

$$S = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}. \quad (2)$$

To get the shape of the o-space in function of the F-formations, the values chosen for the parameters are $\sigma_x = D_H/4$ and $\sigma_y = D_i/2$. In the particular case of the Vis-a-Vis formation $\sigma_y = 0.6$. The orientation of the Gaussian is in the direction of the segment $\overrightarrow{H_{12}C}$, this coincides with the location of the point of interest of humans as exhibited by the orientation of their bodies.

The p-space is considered as the area between the border of the o-space and the same border enlarged by the average size of the humans in conversation. For effects of implementation o-space is discretized using a grid and taking the result of evaluating the center of each cell as the value for the cell. All the elements defined can be seen in fig. 4 for the case of an L-Shape F-formation.

IV. THE ALGORITHM RISK-RRT

As starting point for navigation we chose the strategy proposed in [18]. This algorithm was thought to operate in

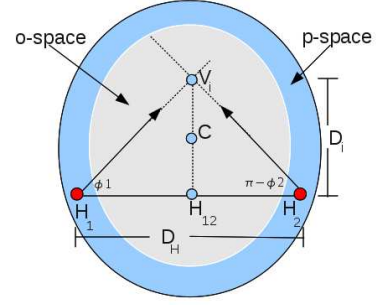


Fig. 4. Elements of the model o-space for L-Shape F-formation.

dynamic, uncertain environment, it supposes that the moving pedestrians detected in the environment follow typical motion patterns that are represented by Gaussian processes which have been learned by an off-board platform before navigation and to be known by the robot. The planning algorithm is based on an extension of the Rapidly-exploring Random Tree algorithm [19], where the likelihood of the obstacles future trajectory and the probability of collision is explicitly taken into account. The tree is grown in a random fashion but a bias is included to direct the search to the goal. Best trajectory (path in the tree) is chosen using as heuristic the “probability of success” and distance to the goal of its nodes. We extended the Risk-RRT by including the knowledge of personal space of pedestrians and the possible interactions between them. The interaction we are focusing on is the conversation between two pedestrians. We penalize paths that passes in the personal space of pedestrians and in the o-space of interactions taking place in the environment by calculating a cost for each one, see eq. 13 and eq. 11. In this section, we present the partial motion planning algorithm Risk-RRT and the collision risk assessment modified in order to include our new constraints.

A. Environment model

At a given instant, the robot knowledge about the state of the world, as proposed by [18], is represented by: an estimation of the state of the robot, a set of Gaussian Processes which represent the typical patterns of the dynamic obstacles, a goal position, an occupancy grid which represents the structure of the static environment and a list of moving objects their estimated position, velocity and previous observations. To take in account the new constraints we include to the list:

- 1) A model of personal space $PS(o_m)$ attached to each dynamic obstacle o_m , according to section III-A
- 2) A list $LI = \{Z_i\}_{i=1..r}$ of interactions detected in the environment, each interaction Z_i has a model of o-space attached to it.

B. Probabilistic Risk of Collision [20] [21]

When searching for a safe path, the algorithm must determine how much is the risk of collision of taking an

action $u \in U$ when in configuration $q(t_1)$. This risk can be written as $P(\text{coll}(q(t_1), u) = 1)$, the probability of collision will be referred as P_c in the rest of the paper. The risk is computed on the basis of the probability of occupation of the surface A which is swept by the robot moving from $q(t_1)$ under control u in the interval to $[t_1, t_2]$:

$$q(t_2) = f(q(t_1), u, \tau) \quad (3)$$

$$A = \iint_{t_1}^{t_2} q(t) dt \quad (4)$$

where $f(\cdot)$ is the motion model of the robot and $\tau = t_2 - t_1$ is the time step. The risk of collision must incorporate both the static and the moving obstacles. Even when two humans in conversation don't exhibit a significant motion they must be treated as dynamic ones because they represent more risk than static obstacles. The space occupied by personal space and o-space can't be detected by sensors and for this reason these spaces are linked to the dynamic obstacles and the computation of their costs is reflected on the probability of collision of the robot with them. We make also the hypothesis that moving obstacles and static obstacles cannot overlap, and consequently that collision with a static obstacle and collision with each one of the moving obstacles are mutually exclusive events, which yields:

$$P_c = P_{cs} + (1 - P_{cs}) \cdot P_{cd} \quad (5)$$

$$P_{cd} = 1 - \prod_{m=1}^M [1 - P_{cd}(o_m)] \quad (6)$$

where P_{cs} is the probability of collision due to the static obstacles, $P_{cd}(o_m)$ is the probability of collision due to the dynamic obstacle o_m and P_{cd} is the probability of collision due to all the dynamic obstacles.

The static obstacles are represented in the occupancy grid which is assumed to be stationary. Given $\mathcal{M}(t_0)$ with $t_0 \leq t_1$ the most recent estimation of the static map and $\varsigma \subset \mathcal{M}(t_0)$ the subset of cells which is the minimal approximation of surface A , the risk of collision with a static obstacle is given by the max probability over the subset ς :

$$P_{cs} = \max_{\varsigma} (P(\text{Occ}(\text{Cell}_{x,y}) = 1)) \quad (7)$$

where $\text{Cell}_{x,y}$ is the cell of the occupancy grid in the (x, y) position. The risk of collision with a moving obstacle o_m is approximated by the probability that the area swept by the robot intercepts the one swept by the obstacle in the considered interval:

$$P_{cd}(o_m) = P(o_m(t) \cap A \neq \emptyset, \forall t \in [t_1, t_2]) \quad (8)$$

The prediction $o_m(t)$ is given by a weighted sum (mixture) of Gaussian Processes. A Gaussian Process is a generalization of the Gaussian probability distribution in function space, see [21] for a more detailed explanation and equations for Gaussian Processes. First, each Gaussian component k is considered separately, then all the Gaussian components are

summed:

$$P_{cd}(o_m, k) = \int_A G(o_m(t), \mu_k, \Sigma_k) \quad (9)$$

$$P_{cd}(o_m) = \sum_{k=1}^K w_{mk} P_{cd}(o_m, k) \quad (10)$$

where $P_{cd}(o_m, k)$ is the probability of collision with the obstacle m moving along pattern k ; $G(o_m(t), \mu_k, \Sigma_k)$ is the Gaussian Process representing pattern k , given the observation history of object o_m . The probability is marginalized over the set of possible patterns to yield $P_{cd}(o_m)$, where w_{mk} is the weight of the k component for object m . In order to choose an appropriate path, the Risk-RRT uses the risk of collision of a particular action to calculate the "probability of success" of each partial path [18].

1) *Adding social constraints*: In this section we explain how we include the social constraints to the model before presented, being this the main contribution of the paper. First we define PZ_i as the probability of disturbing by passing inside the o-space (sec. III-B) of interaction i , and we calculate it as:

$$PZ_i = \max_{\varsigma} (\Gamma_{C_i, S_i}(\text{Cell}_{x,y})) \quad (11)$$

To reflect the fact of disturbing an interaction we think of it as a collision with a dynamic obstacle and modify the equation 6 to get:

$$P_{cd} = 1 - \prod_{m=1}^M [1 - P_{cd}(o_m)] \prod_{i=1}^r [1 - PZ_i] \quad (12)$$

In the case of the personal space we define P_{ps} as the probability of disturbing by passing in the personal space of the human o_m . We can approximate P_{ps} as the probability that A , the area swept by the robot, intercepts the one represented by the personal space:

$$P_{ps}(o_m, k) = \int_A PS(o_m(t)) \quad (13)$$

Where $PS(o_m(t))$ is the model of personal space centered in $o_m(t)$ as described in III-A. Again, to take in account this last constraint we need to modify the original equation 10 to get:

$$P_{cd}(o_m) = \sum_{k=1}^K w_{mk} P_{cd}(o_m, k) P_{sp}(o_m, k) \quad (14)$$

After these extensions the "probability of success" calculated for every partial path is given by the probability of not encountering a collision along the path and not entering in a personal space or an o-space.

C. The goal-oriented navigation algorithm

The goal oriented navigation proposed is described in Algorithm 1. It combines three tasks: one dedicated to perception (of static and moving obstacles), a task for planning partial but safe trajectories and a task for navigating

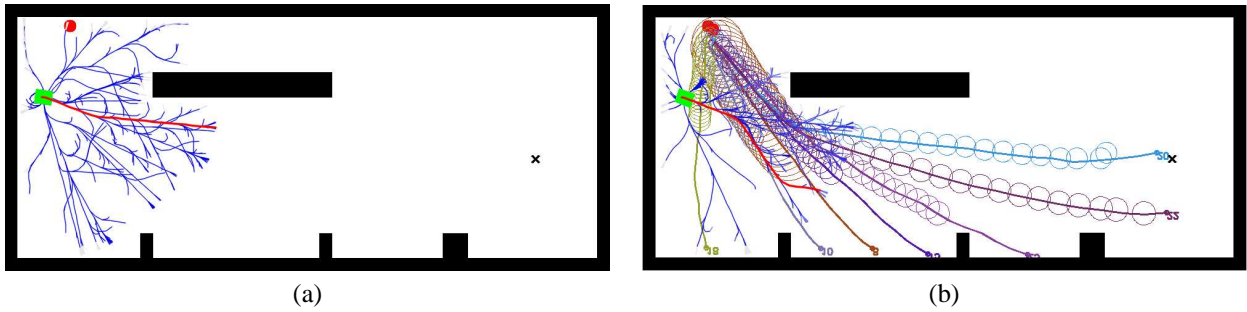


Fig. 5. Example of execution of Risk-RRT algorithm. In a) the robot navigation system has created a tree of possible paths to follow, robot is the green rectangle, the chosen path is in red. In (b) we can observe how the robot has adapted its trajectory trying to avoid a possible collision with pedestrian (in red) by considering the predictions of typical pedestrian trajectories.

safely along planned trajectories. The prediction done for forecasting the position of moving obstacles in the near future is based on learned Gaussian Processes [18].

Algorithm 1 Risk-RRT

```

1: procedure RISK-RRT
2:   trajectory = empty
3:   Tree = empty
4:   Goal = read()
5:   t = clock()
6:   while Goal not reached do
7:     if trajectory is empty then
8:       brake
9:     else
10:      move along trajectory for one step
11:    end if
12:    observe ( $X$ );
13:    delete unreachable trajectories( $T, X$ )
14:    observe( $Map, movingobstacles$ )
15:    t = clock()
16:    predict moving obstacles at time  $t, \dots, t + N\tau$ 
17:    if environment different then
18:      update trajectories( $T, Map, moving\ obstacles$ )
19:    end if
20:    while clock() <  $t + \tau$  do
21:      grow trajectories with depth <=  $N$  in  $T$ 
22:    end while
23:    trajectory = Choose best trajectory in  $T$ 
24:    t = clock()
25:  end while
26:  brake
27: end procedure

```

Risk-RRT takes explicitly into account the real-time constraint and limits the time available for planning to a fixed interval. After each planning cycle, the planned trajectory is generally just a partial trajectory. Execution and planning are done in parallel: while the robot moves a step along the planned partial path, the tree is updated (line 18 of Algorithm 1) with the information coming from the perception algorithm, the tree is grown and the new partial path is passed for execution when the time step is over. In the fig. 5 we

can observe an example of navigation employing Risk-RRT in the case of one pedestrian entering in the environment and robot going to its goal. At the beginning the robot has explored the environment and then decides to follow one trajectory, some steps ahead when it detects the presence of pedestrian, a prediction is realized based in the Gaussian processes and it must adjust its previous choice to avoid a collision with the human.

V. SIMULATION RESULTS

To test our models of interaction we have chosen a scenario that shows one conversation between two humans standing in a spacious area, this is because we want to decrease the effect of the structure of the environment in the management of space done by people. The simulation loads a map previously constructed by a SLAM function using a laser mounted on a wheelchair and creates an occupancy grid based on it. The pedestrians are placed in a Vis-a-Vis F-formation, that is, facing each other in their social zone (sec. III-A). The space between them is big enough to let the robot passing. Detecting conversation interactions is done, first by finding pedestrians that are closer than a maximum distance, then by check if their velocities are under a maximum velocity and finally taking in account the orientation of their bodies to match one of the F-formations defined in section III-B. The robot simulated is an autonomous wheelchair with two wheels, the model used is that of a differential robot system.

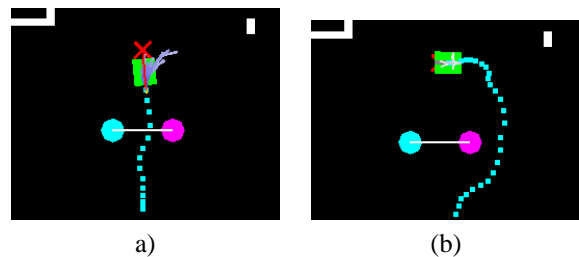


Fig. 6. Change in the behavior of the wheelchair (green). In (a) the navigation doesn't take in account the personal spaces nor the Vis-a-vis formation and chooses a path that interrupts the interaction of two humans (circles), the goal is the red cross, (b) the robot has more information and respects the social conventions of space.

The concepts exposed in section III have been implemented in a navigation algorithm taking as base our previously designed Risk-RRT approach [20], [18].

The first task was reaching the goal defined by the user, we choose an initial position for the robot and a goal location in such a way that the short distance between them passes in the middle of human positions. First, we run the algorithm original and we note that the chosen path has a tendency to interrupt the interaction, one example in fig. 6 (a), then we run the algorithm modified and we can see (fig. 6 (b)) that the tendency in the behavior of the wheelchair (in green) can be changed if we detect the interaction and reduce the probability of disturbing the conversation and the probability of disturbing by passing into the personal space of pedestrians.

As a second task, using the same scenario we let the wheelchair to explore the environment (fig. 7) and find a group in conversation to join it, this was done by choosing a random initial location for the wheelchair and random goals to reach, once that the wheelchair detect the first conversation, the new goal becomes the center of the o-space for the interaction detected. In this case we detect interactions only in a semicircular region centered in the wheelchair and oriented to the front of it. The wheelchair approaches to the group and because of the effect of interaction model it stops at p-space distance (sec. III-B), a behavior that coincides with that of a person approaching to a group and waiting for the reaction (acceptance) of the group.

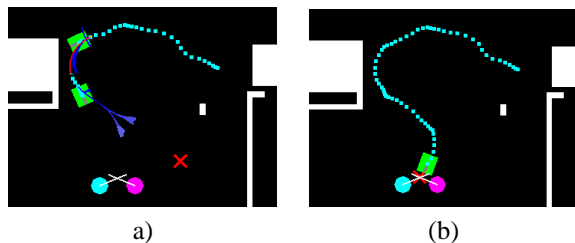


Fig. 7. The wheelchair (green) explores its environment (a), it detects a conversation, approach to humans and stops at p-space distance (b), a behavior that can be judged social

VI. CONCLUSIONS AND FUTURE WORK

The approach presented in this paper shows a way to take in account social conventions in navigation strategies providing the robot with the ability to respect the personal space and the o-space of people in its environment when moving safely towards a given goal. In the same way these models were useful to guide the robot for a “joining a group” task. The previous concepts have been implemented by extending a previously designed navigation algorithm, the Risk-RRT approach. We have shown in simulation that the behavior of a robot can be changed if we detect an interaction. Our current work aims of implementing our approach on a real autonomous system like a wheelchair and perform some experiments with real humans.

In a dynamic environment it is not enough detecting interactions because it could be too late to take a decision, we need

to predict when and where an interaction will take place. Our future work will be focused in adding a technique for better predicting the creation of an o-space in the path of the robot.

REFERENCES

- [1] M. Ciolek and A. Kendon, “Environment and the spatial arrangement of conversational encounters,” *Sociological Inquiry*, vol. 50, pp. 237–271, 1980.
- [2] E. T. Hall, *The hidden Dimension: Man’s Use of Space in Public and Private*. The Bodley Head Ltd, London, UK, 1966.
- [3] C. A. Authority. (2010) Accessible air travel workshop. Electronic Report. [Online]. Available: <http://www.caa.co.uk/>
- [4] R. Gockley, J. Forlizzi, and R. Simmons, “Natural person following behavior for social robots,” *HRI07*, 2007.
- [5] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. Christensen, “Navigation for human-robot interaction tasks,” *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1894–1900, 2004.
- [6] H. Huettnerrauch, K. S. Eklundh, A. Green, and E. A. Topp, “Investigating spatial relationships in human-robot interaction,” *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- [7] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, “How to approach humans. strategies for social robots to initiate interaction,” *HRI*, 2009.
- [8] M. L. Walters, K. Dautenhahn, R. te Boekhorst, K. L. Koay, D. S. Syrdal, and C. L. Nehaniv., “An empirical framework for human-robot proxemics,” *Proceedings New Frontiers in Human-Robot Interaction*, 2009.
- [9] L. Takayama and C. Pantofaru, “Influences on proxemic behaviors in human-robot interaction,” *IROS*, 2009.
- [10] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, “Design of an office-guide robot for social interaction studies,” *IEEE/RSJ*, 2006.
- [11] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu, “Human-centered robot navigation,towards a harmoniously human-robot coexisting environment,” *Robotics, IEEE Transactions on*, vol. 27, no. 1, pp. 99–112, 2011.
- [12] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, “A human aware mobile robot motion planner,” *IEEE Transactions on Robotics*, vol. 23, 2007.
- [13] S. T. Hansen, M. Svenstrup, H. J. Andersen, and T. Bak, “Adaptive human aware navigation based on motion pattern analysis,” *The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.
- [14] R. Kirby, R. Simmons, and J. Forlizzi, “Companion: A constraint-optimizing method for person acceptable navigation,” *The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.
- [15] S.-Y. Chung and H.-P. Huang, “A mobile robot that understands pedestrian spatial behaviors,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Taipei, Taiwan*, 2010, pp. 5861–5866.
- [16] H. Laga and T. Amaoka, “Modeling the spatial behavior of virtual agents in groups for non-verbal communication in virtual worlds,” *IUCS ’09*, 2009.
- [17] A. Kendon, “Spacing and orientation in co-present interaction,” in *Development of Multimodal Interfaces: Active Listening and Synchrony*, ser. Lecture Notes in Computer Science, A. Esposito, N. Campbell, C. Vogel, A. Hussain, and A. Nijholt, Eds. Springer Berlin / Heidelberg, 2010, vol. 5967, pp. 1–15.
- [18] C. Fulgenzi, A. Spalanzani, C. Laugier, and C. Tay, “Risk based motion planning and navigation in uncertain dynamic environment,” INRIA,” Research Report, 10 2010, submitted to IEEE, Transactions of Robotics. [Online]. Available: <http://hal.inria.fr/inria-00526601/en/>
- [19] S. LaValle and J. Kuffner, J.J., “Randomized kinodynamic planning,” *International Conference on Robotics and Automation*, vol. 1, pp. 473–479 vol.1, 1999.
- [20] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Probabilistic motion planning among moving obstacles following typical motion patterns,” in *IROS*, 2009, pp. 4027–4033.
- [21] T. Meng Keat Christopher, “Analysis of dynamic scenes: Application to driving assistance,” PhD in Computer Science, Institut National Polytechnique de Grenoble (INPG), 2009.



Session IV

2D and 3D Mapping & Localization

- **Keynote speaker: C. Stiller (Karlsruhe Institute of Technology)**
Title: 2D/3D mapping and localization
Co-Authors: A. Geiger, F. Moosmann
- **Title: A New Strategy for Feature Initialization in Visual SLAM**
Authors: G.Bresson, T. Feraud, R. Aufrere, P. Checchin and R. Chapuis
- **Title: Building Facade Detection, Segmentation, and Parameter Estimation for Mobile Robot Localization and Guidance**
Authors: J.A. Delmerico, P. David, J.J. Corso



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

Session IV

Keynote speaker: **C. Stiller (Karlsruhe Institute of Technology)**

2D/3D mapping and localization

Co-Authors: A. Geiger, F. Moosmann

Abstract : On-line environment modelling and mapping are of growing importance for autonomous robots and cognitive automobiles. Emerging from 2D or 2.5D flat world representations improved sensors and computing capabilities allow for 3D world representations and estimation of full 6 DOF robot motion. The choice of fast and expressive features enhances density while reducing noise. We present 3D dense maps acquired in real time from a lidar or stereo camera rig mounted on a traveling experimental vehicle without requiring any additional odometry or inertial sensors.

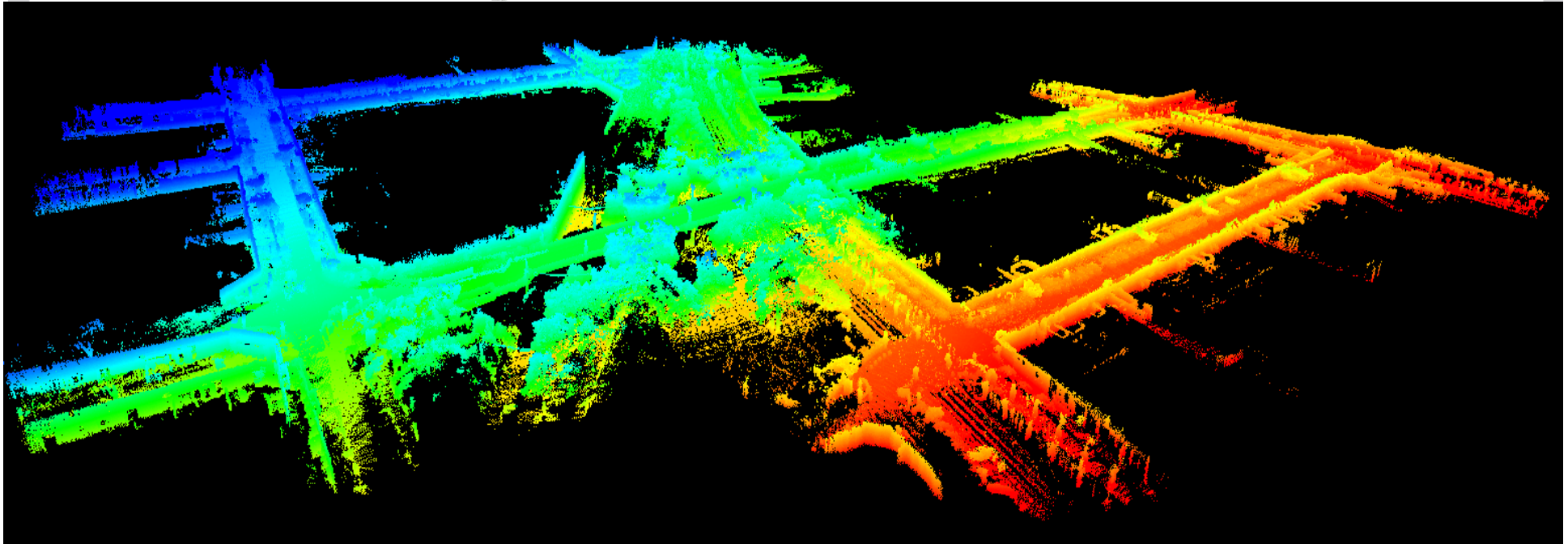
Biography: Christoph Stiller studied Electrical Engineering at Aachen, Germany, and Trondheim, Norway, and received the Diploma degree from Aachen University of Technology, Germany in 1988. In 1988 he became a Scientific Assistant at Aachen University of Technology.

After completion of his Dr.-Ing. degree (Ph.D., 1994), he worked at INRS-Telecommunications in Montreal, Canada as a post-doctoral Member of the Scientific Staff. In 1995 he joined the Corporate Research and Advanced Development of Robert Bosch GmbH, Hildesheim, Germany, where he was responsible for 'Computer Vision for Automotive Applications'. In 2001 he became chaired professor and director of the Institute for Metrology and Control Engineering at Karlsruhe Institute of Technology, Germany.

Dr. Stiller is President-Elect of the IEEE Intelligent Transportation Systems Society. He serves as Editor-in-Chief of the IEEE Intelligent Transportation Systems Magazine (2009-ongoing) and has served as Associate Editor for several IEEE Transactions and as General Chair for the IEEE Intelligent Vehicles Symposium 2011 in Germany.

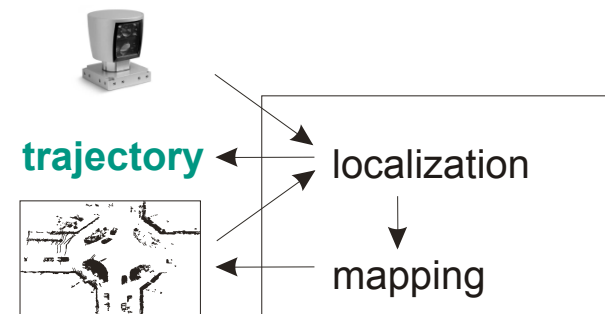


2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Simultaneous Localization and Mapping

- input: sequence of measurements
- output: trajectory and map



2D & 3D Mapping and Localization

*Christoph Stiller
Andreas Geiger
Frank Moosmann*

Institut für Mess-
und Regelungstechnik



Trends in SLAM

- 2D sensors,
flat world assumption
- sparse maps
- stop vehicle for scanning,
off-line computation
- highly accurate laser sensors
- additional ego-motion sensors
(odometer etc.)

3D world models

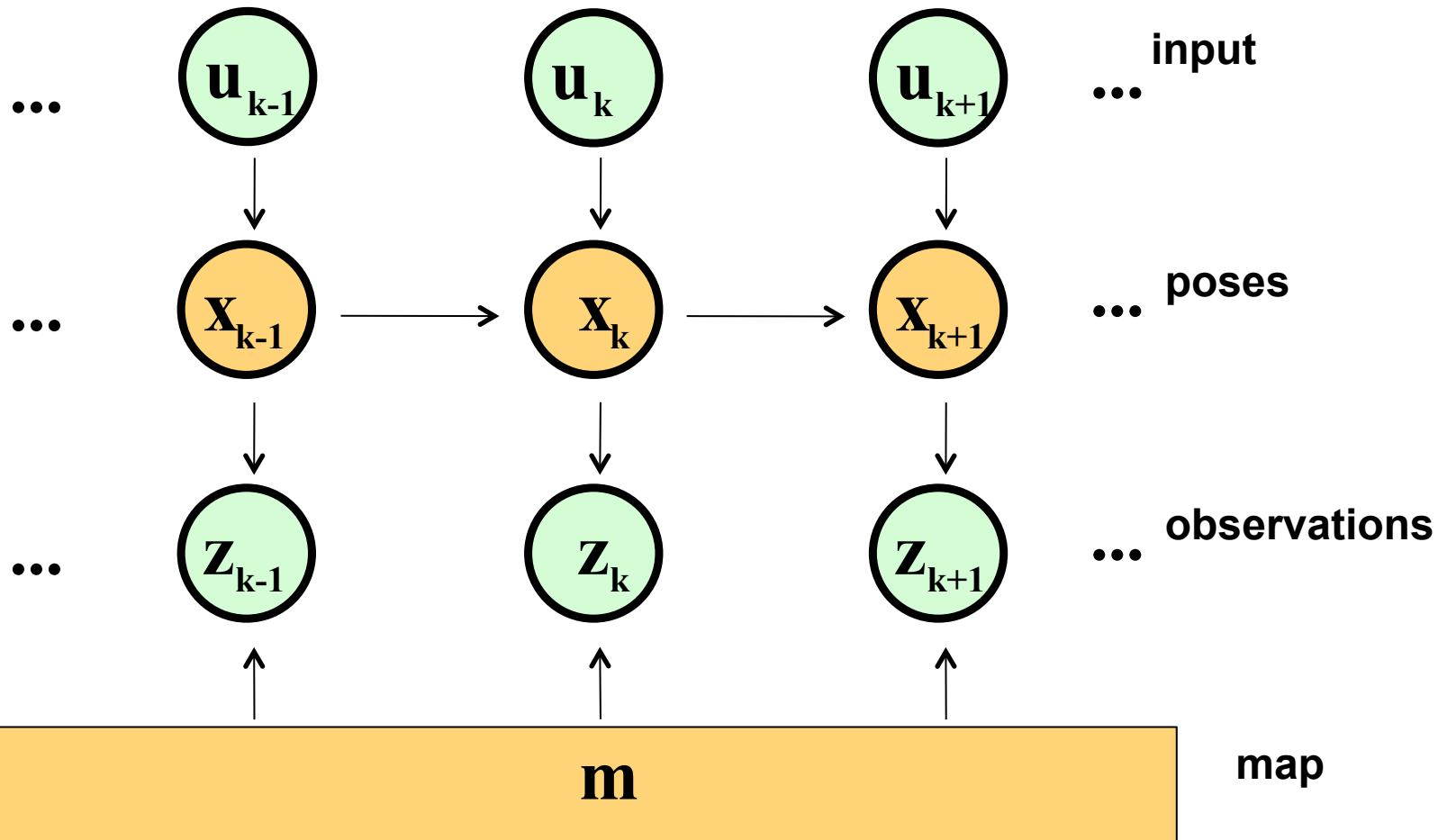
dense maps

on-the-fly scanning and
real-time computation

noise modelling,
visual SLAM

single sensor SLAM

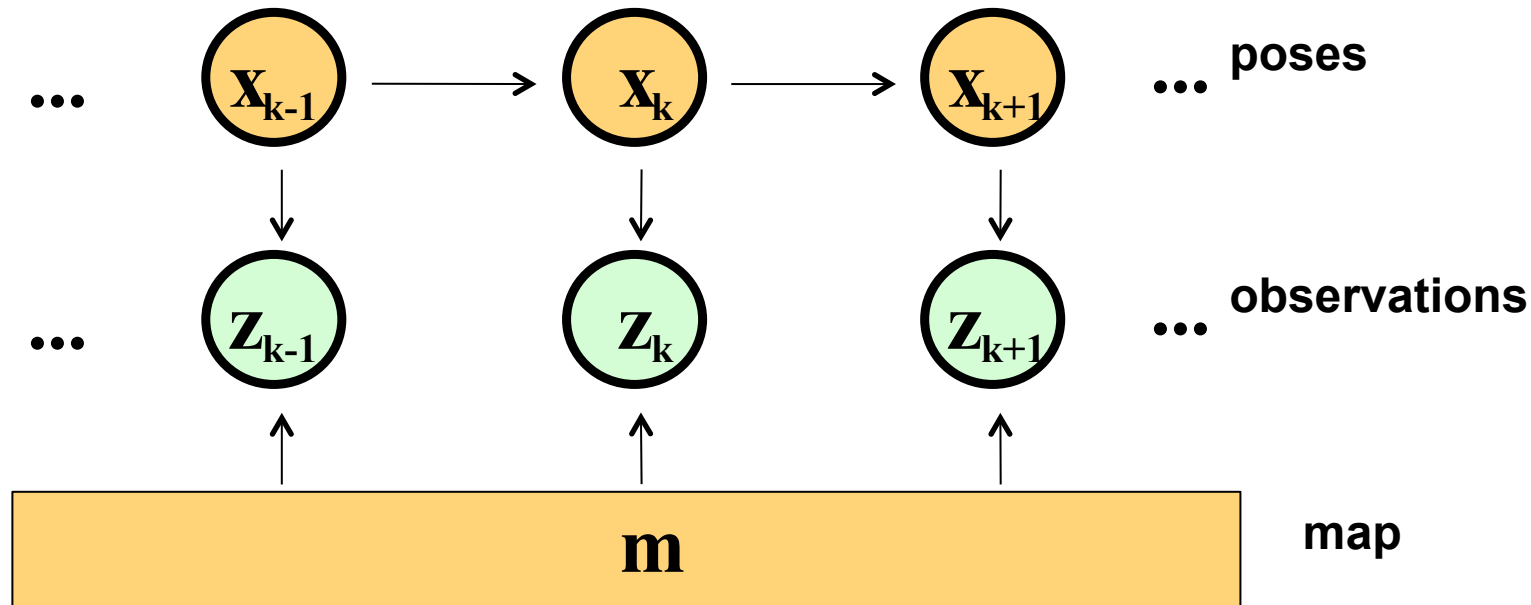
Graph SLAM



$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^k, \mathbf{u}^k) \quad \text{visual odometry (Online SLAM)}$$

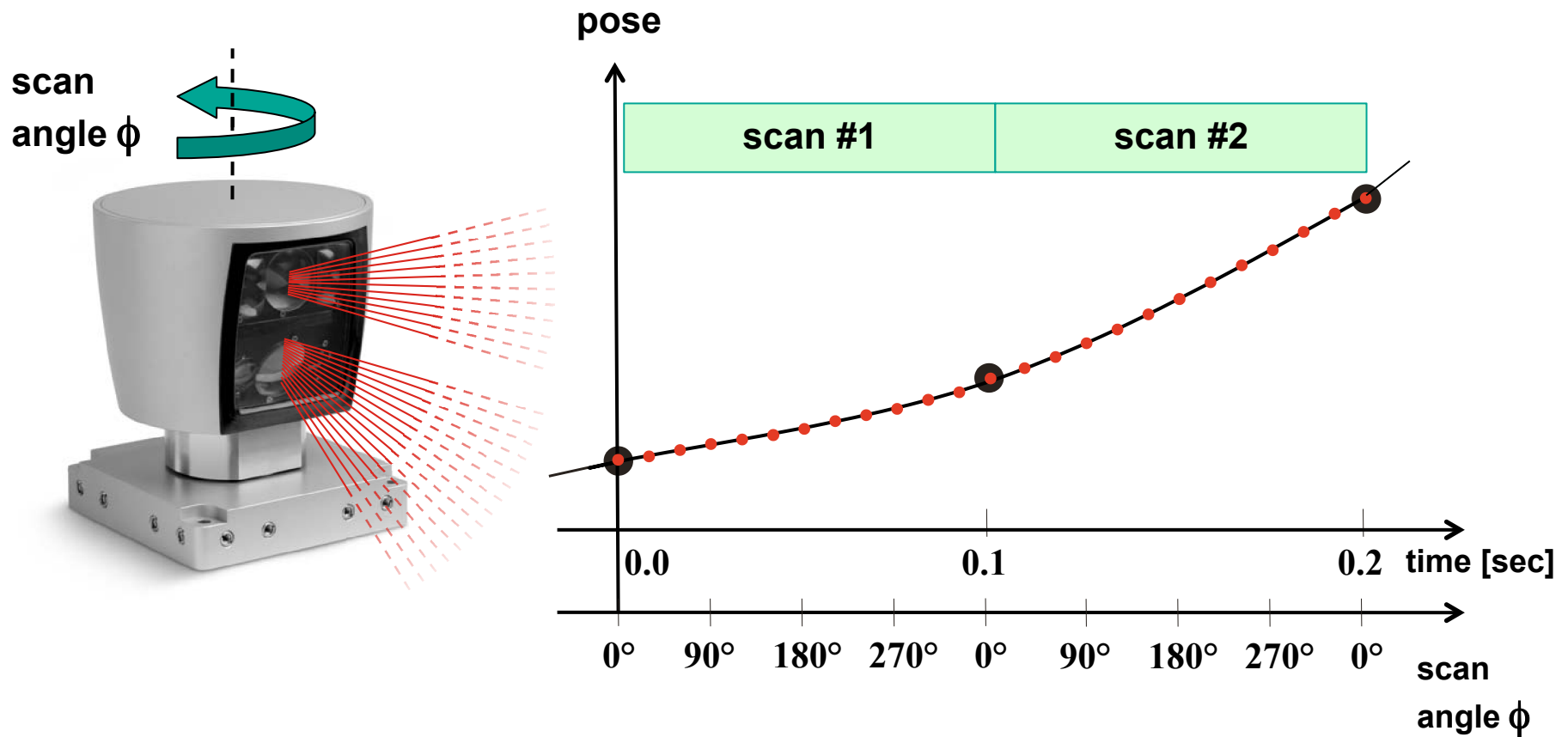
$$p(\mathbf{x}^k, \mathbf{m} | \mathbf{z}^k, \mathbf{u}^k) \quad \text{bundle adjustment (Full SLAM)}$$

Graph SLAM



$$\begin{aligned}
 p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^k) &= \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}, \mathbf{z}^{k-1}) p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^{k-1})}{p(\mathbf{z}_k | \mathbf{z}^{k-1})} \\
 &= \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}, \mathbf{z}^{k-1}) \int p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{z}^{k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{m}, \mathbf{z}^{k-1}) d\mathbf{x}_{k-1}}{p(\mathbf{z}_k | \mathbf{z}^{k-1})}
 \end{aligned}$$

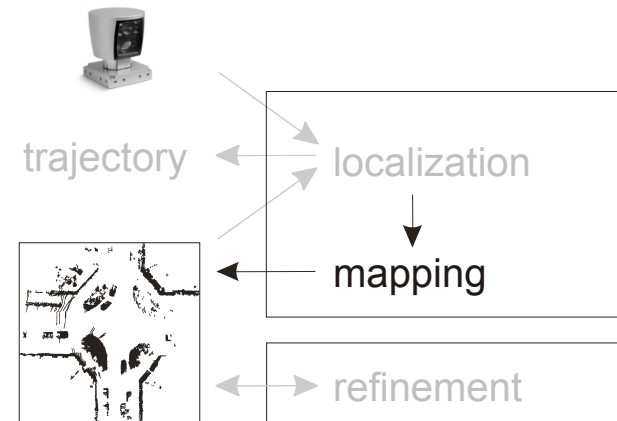
On-the-Fly Scanning



- continuous pose required
- estimate pose dynamics

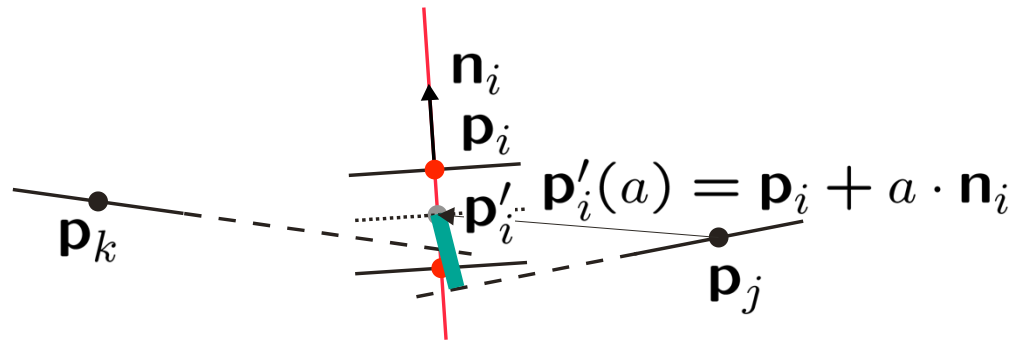
Features

- normal vectors
 - calculated as cross product from 4 neighbours
- flatness measure f_i
 - between 0 (non-flat) and 1 (flat)
 - probability if sampling theorem holds



Noise reduction

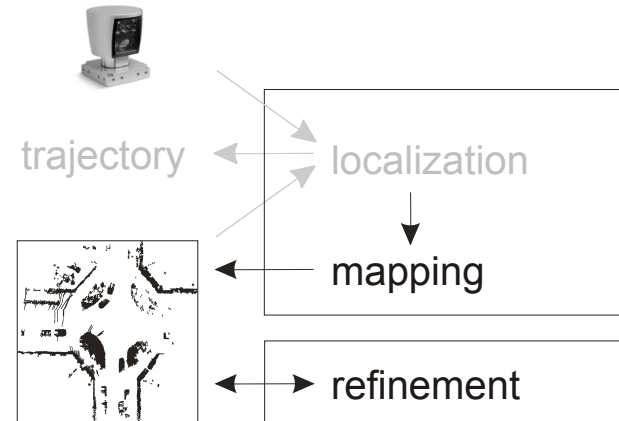
- shift points towards neighbouring planes
 - in „flat“ areas only



$$a = \arg \min_a \sum_j w_{ij} (\mathbf{n}_j^T (\mathbf{p}'_i(a) - \mathbf{p}_j))^2$$

$$w_{ij} = f_i f_j \mathbf{n}_i^T \mathbf{n}_j$$

- online: by using existing, already shifted surfaces of map
- offline: by using originally scanned surfaces

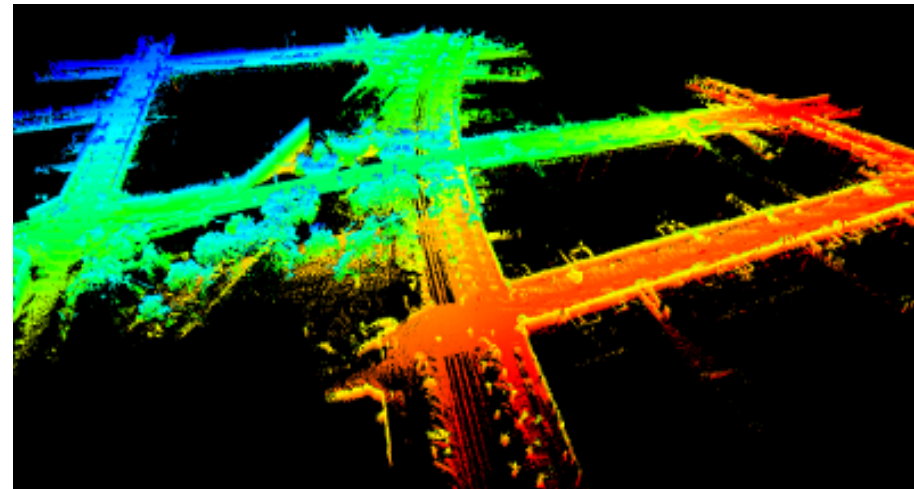
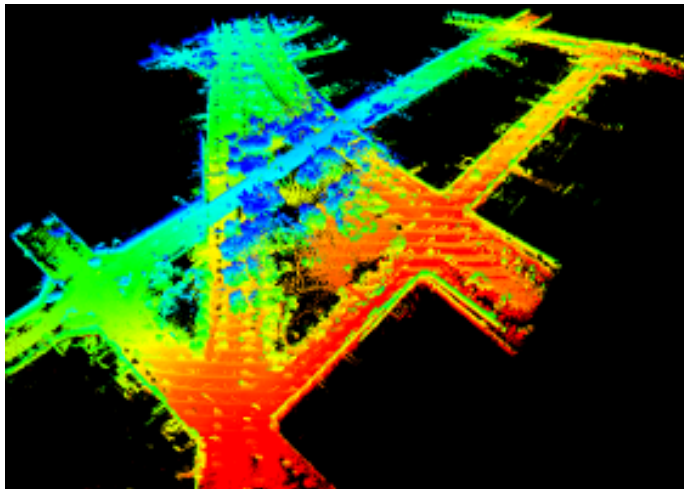


Results

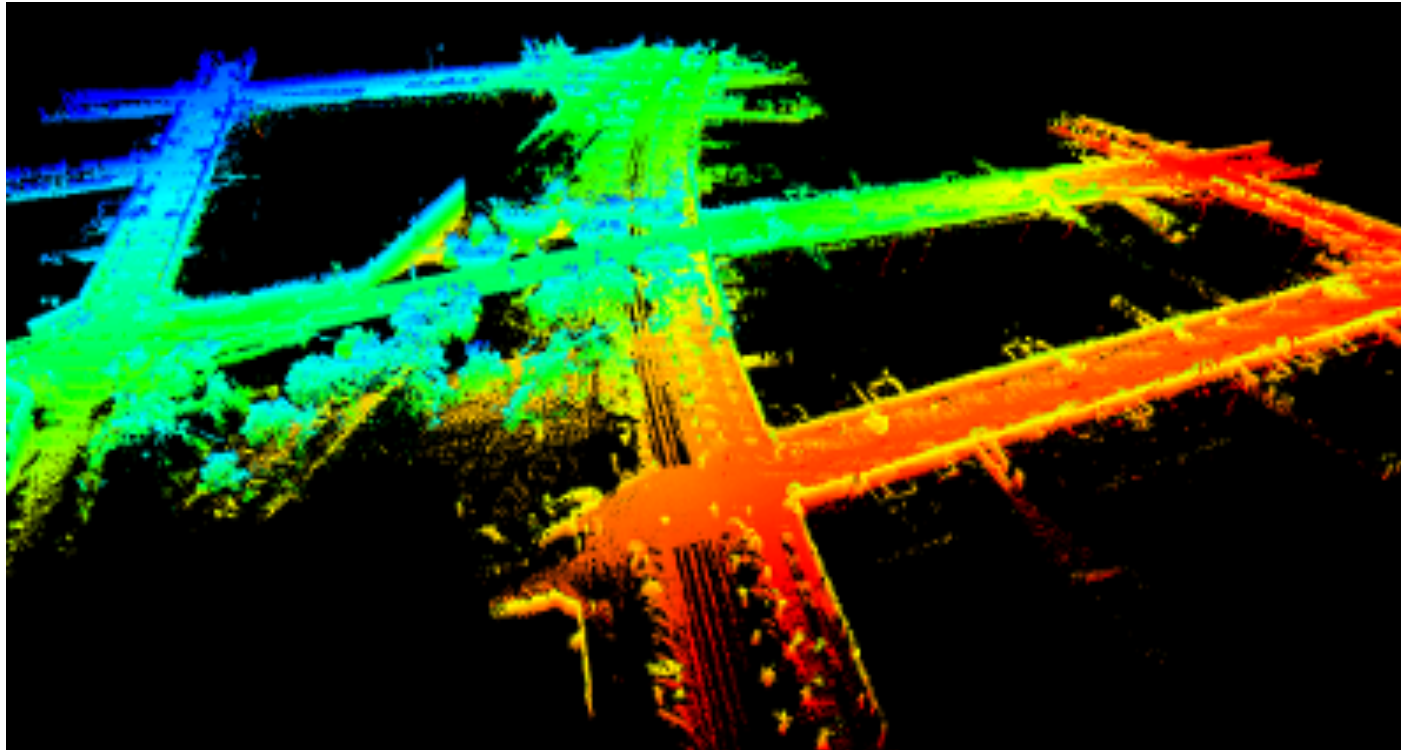
- evaluation of
 - map quality
 - localization precision

- new data set :

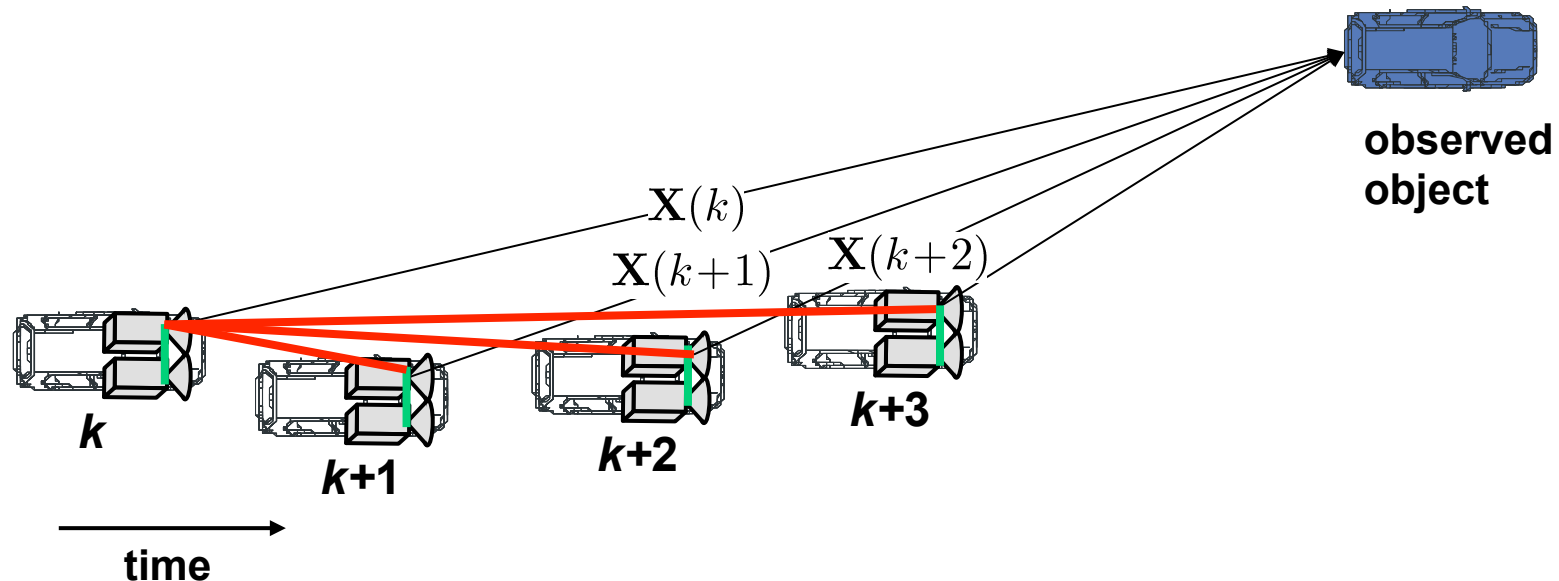
<http://www.mrt.kit.edu/z/publ/download/velodyneslam>



Results: generated map



3D SLAM Exploiting Multiple Visual Keys



stereo basis:

- Constant over time
- + favorable orientation
- range uncertainty $\sim \text{range}^2/\text{resolution}$

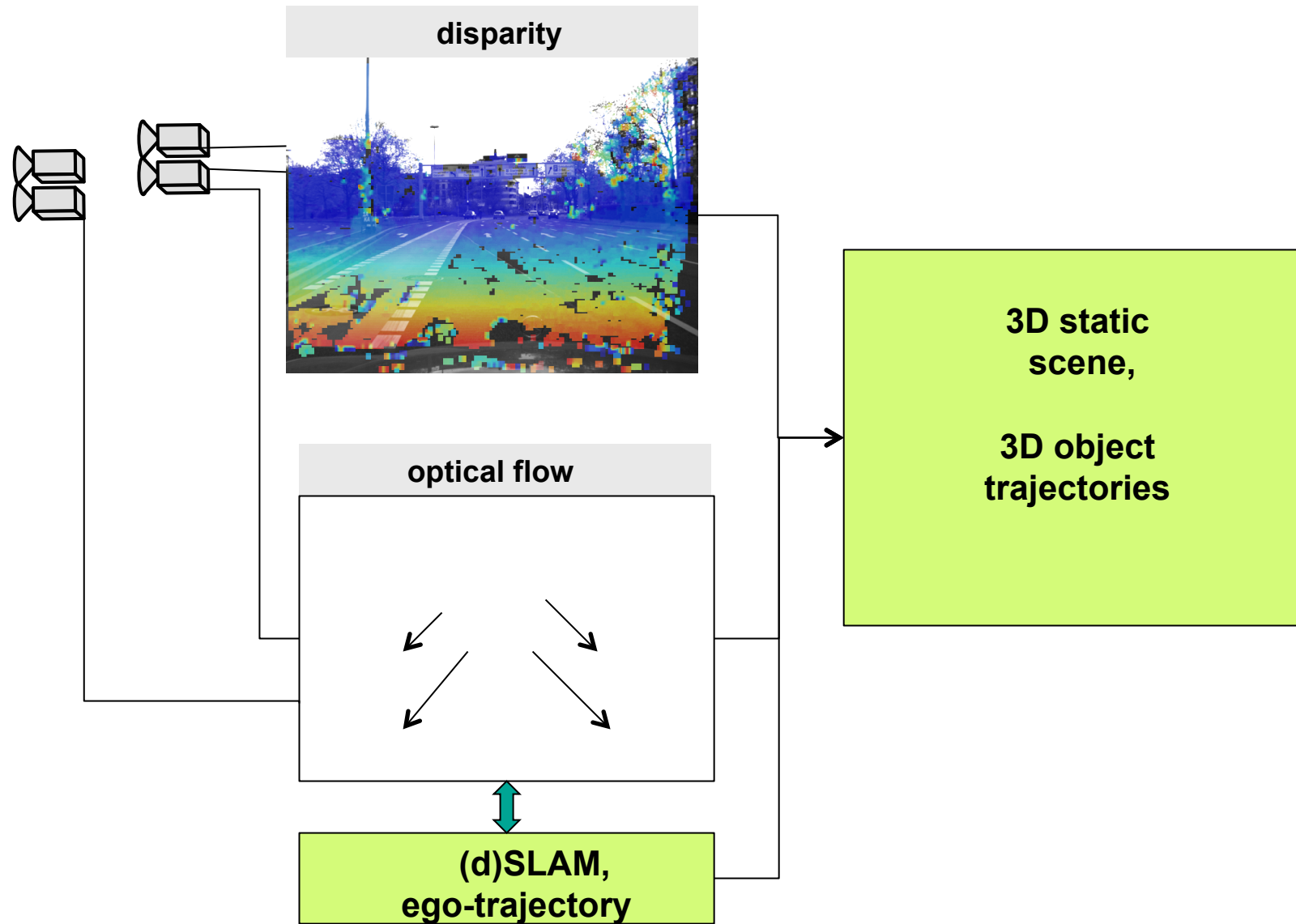
motion stereo basis:

- unfavorable orientation
- + increasing with time

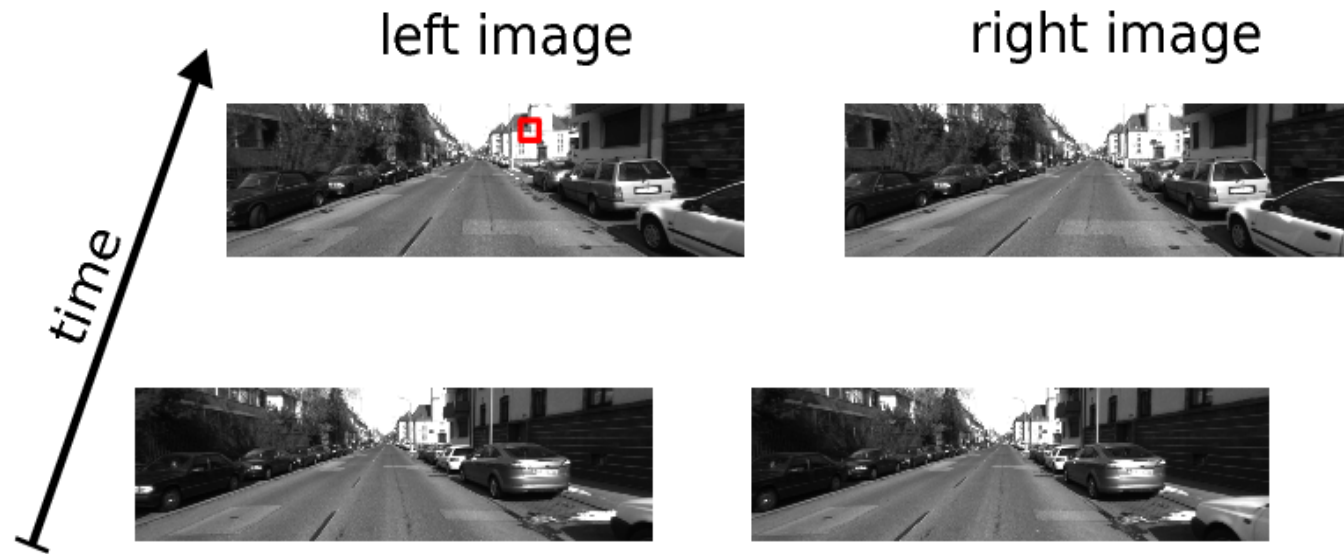
Simultaneous monoscopic and stereoscopic image sequence analysis:

- + Instantaneous object detection in near field
- + Range increasing with observation time

3D SLAM Exploiting Multiple Visual Keys



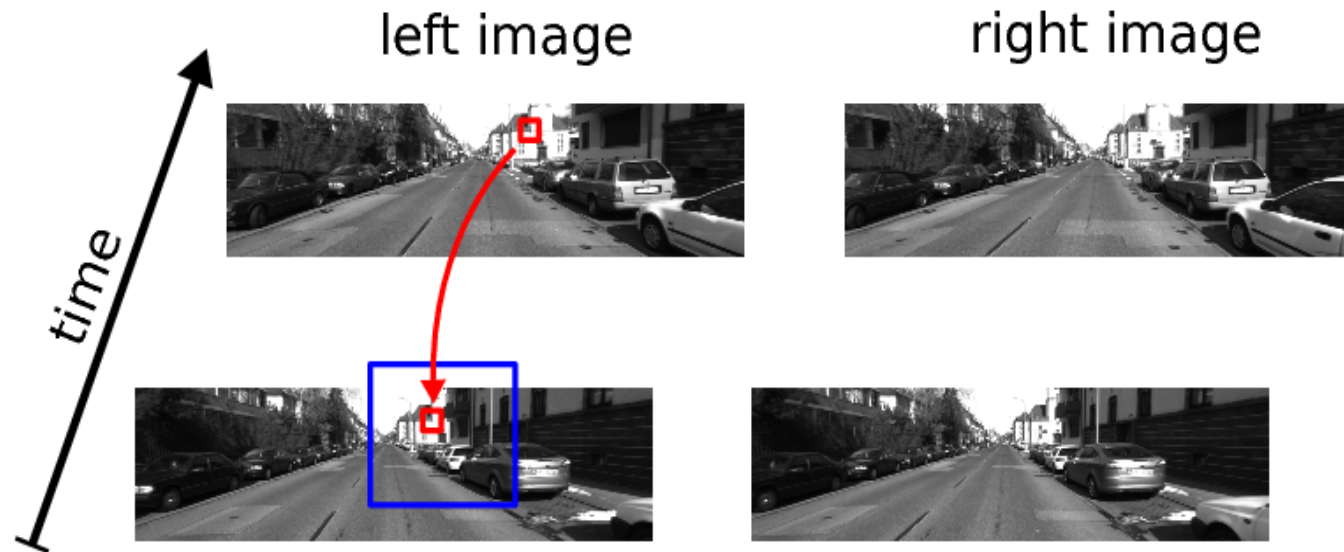
Correspondences



Feature matching:

- Detect interest points using non-maximum-suppression
- Match 4 images in a 'space-time' circle
- Use epipolar constraints for left ↔ right matching
- Accept if last feature coincides with first feature

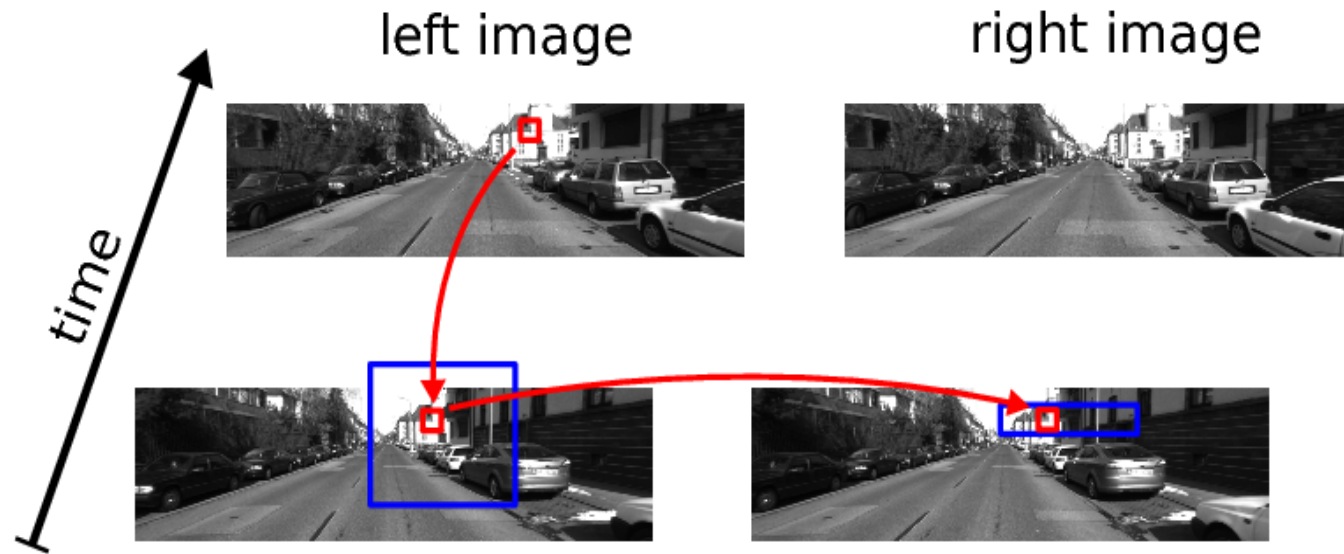
Correspondences



Feature matching:

- Detect interest points using non-maximum-suppression
- Match 4 images in a 'space-time' circle
- Use epipolar constraints for left ↔ right matching
- Accept if last feature coincides with first feature

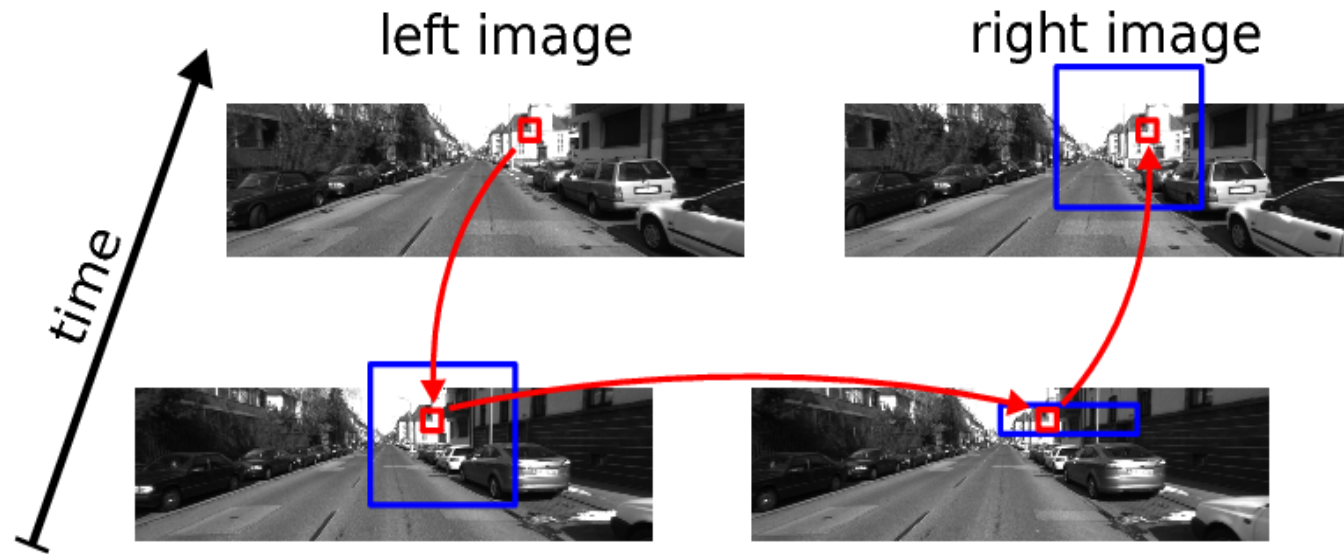
Correspondences



Feature matching:

- Detect interest points using non-maximum-suppression
- Match 4 images in a 'space-time' circle
- Use epipolar constraints for left \leftrightarrow right matching
- Accept if last feature coincides with first feature

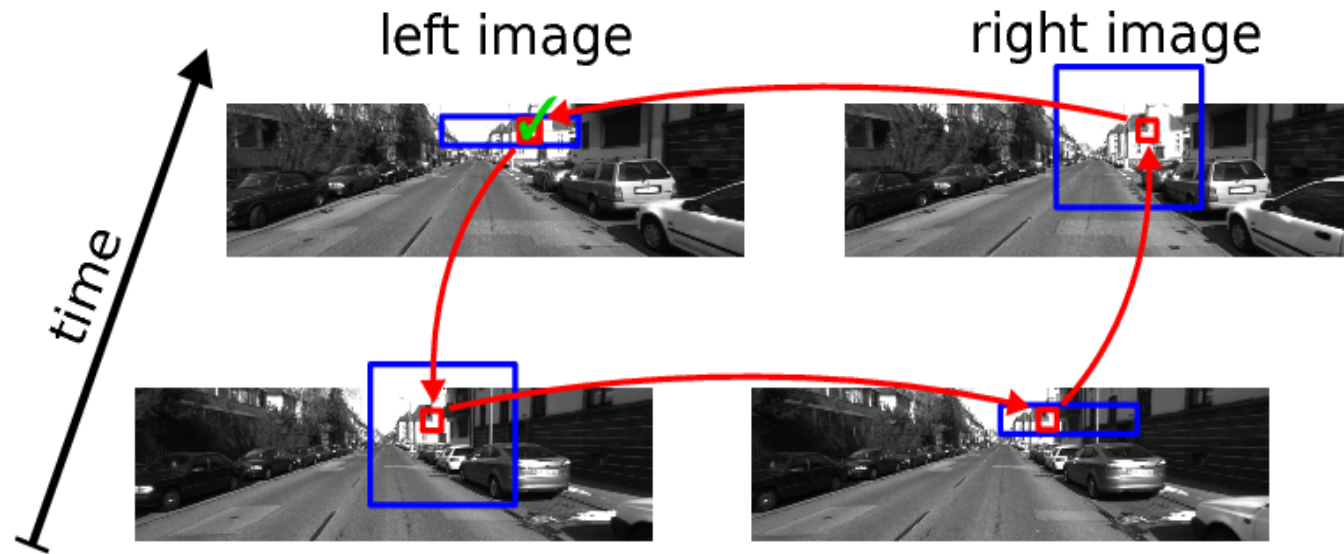
Correspondences



Feature matching:

- Detect interest points using non-maximum-suppression
- Match 4 images in a 'space-time' circle
- Use epipolar constraints for left \leftrightarrow right matching
- Accept if last feature coincides with first feature

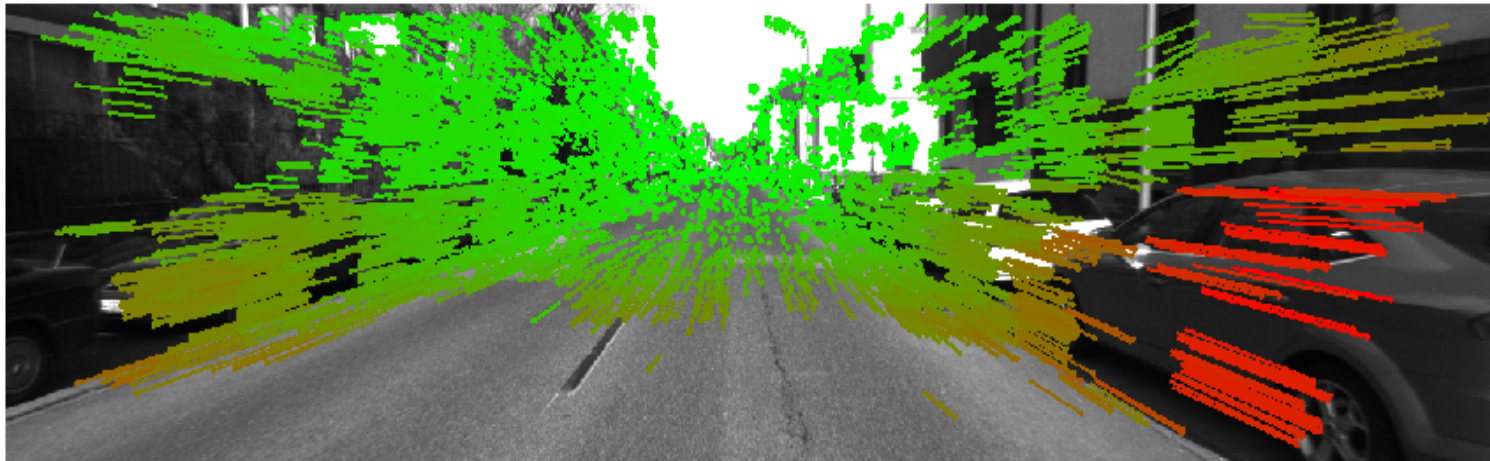
Correspondences



Feature matching:

- Detect interest points using non-maximum-suppression
- Match 4 images in a 'space-time' circle
- Use epipolar constraints for left \leftrightarrow right matching
- Accept if last feature coincides with first feature

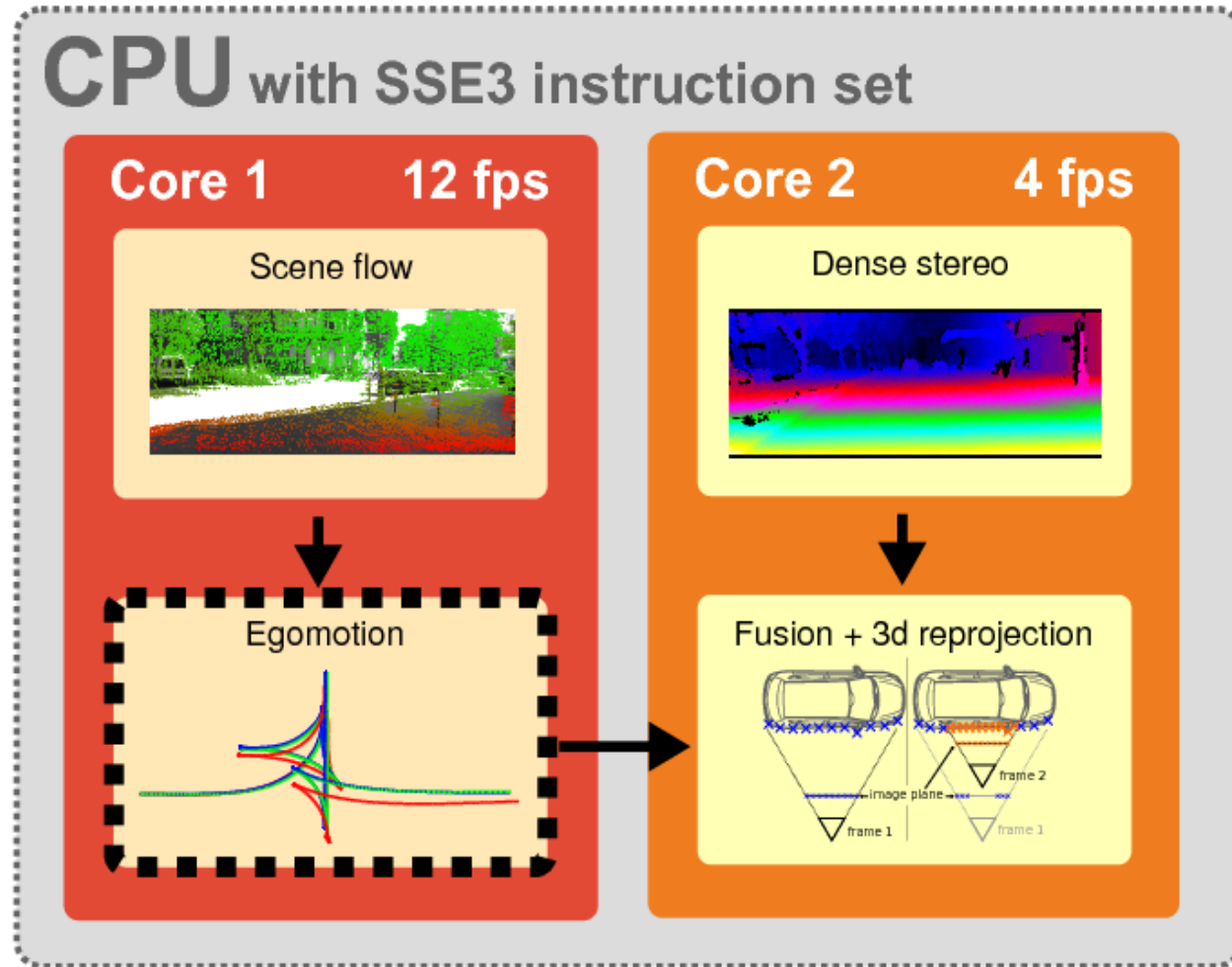
Features



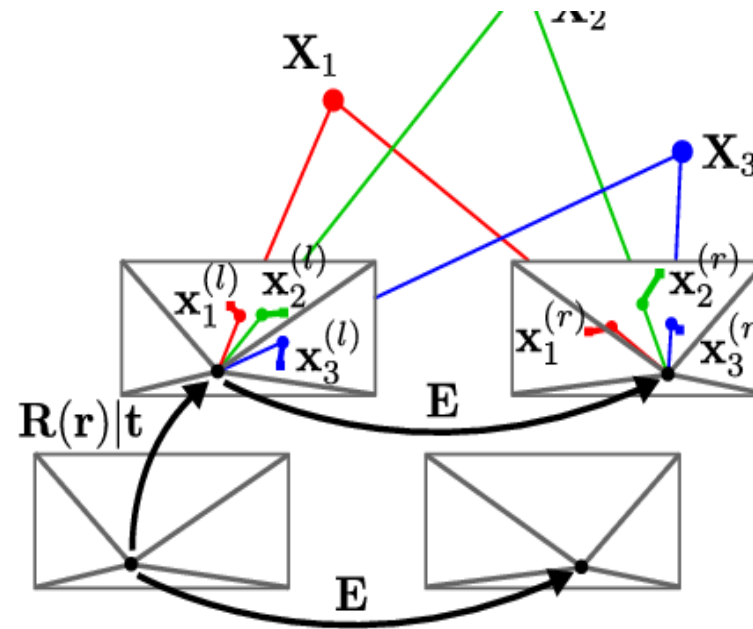
Fast feature matching:

- 1st: Match a sparse set of interest points within each class
- Build statistics over likely displacements within each bin
- Use this statistics for speeding up 2nd matching stage
- Rejection outliers (Delaunay triangulation)

Single CPU Implementation



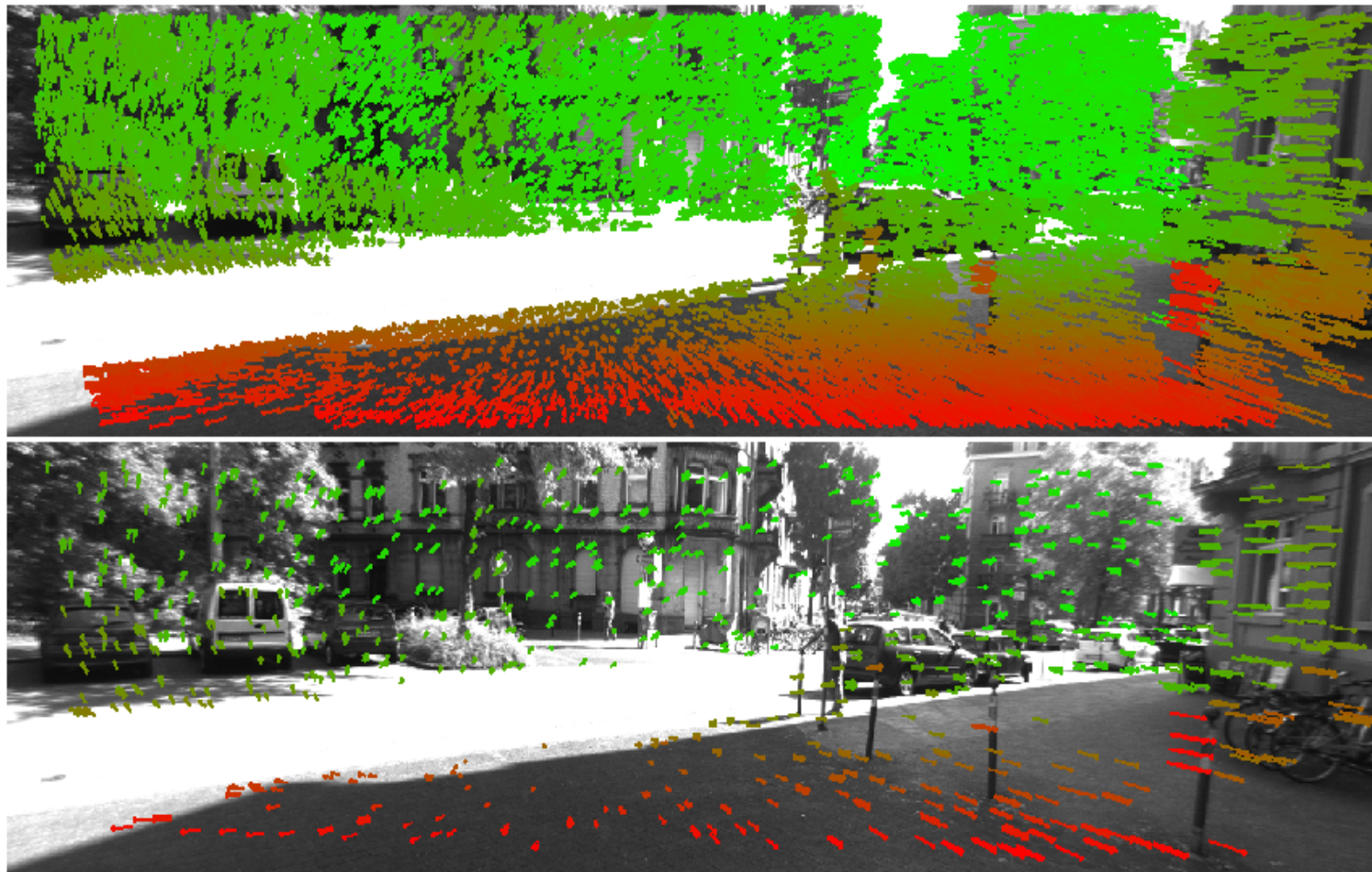
Ego-Motion



$$\min_{\mathbf{r}, \mathbf{t}} \sum_{i=1}^N \left\| \mathbf{x}_i^{(l)} - \pi^{(l)}(\mathbf{X}_i; \mathbf{r}, \mathbf{t}) \right\|^2 + \left\| \mathbf{x}_i^{(r)} - \pi^{(r)}(\mathbf{X}_i; \mathbf{r}, \mathbf{t}) \right\|^2$$

- Minimize reprojection errors (Gauss-Newton + RANSAC)
- Kalman Filter (constant acceleration model)

Results



Results

- video

Conclusions

- **3d map generation & ego-trajectory estimation**
 - **lidar or video camera**
 - **large scale**
 - **real time on single CPU**
 - **large scale**

- **challenges**
 - **local bundle adjustment**
 - **object trajectories**
 - **scene understanding**
 - **reliability, plausibility**

[Geiger, Ziegler, Stiller, IEEE Intelligent Vehicles Symposium 2011]

[Moosmann, Stiller, IEEE Intelligent Vehicles Symposium 2011]



Session IV

2D and 3D Mapping & Localization

- **Title: A New Strategy for Feature Initialization in Visual SLAM**
Authors: G.Bresson, T. Feraud, R. Aufrere, P. Checchin and R. Chapuis
- **Title: Building Facade Detection, Segmentation, and Parameter Estimation for Mobile Robot Localization and Guidance**
Authors: J.A. Delmerico, P. David, J.J. Corso



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

A New Strategy for Feature Initialization in Visual SLAM

Guillaume Bresson*, Thomas Féraud*, Romuald Aufrère*[†], Paul Checchin* and Roland Chapuis*

*LASMEA - UMR 6602 CNRS – [†]LIMOS - UMR 6158 CNRS

Blaise Pascal University

Aubière, France

first-name.last-name@univ-bpclermont.fr

Abstract—This paper presents a Visual EKF-SLAM process using an original and very efficient strategy for initializing landmarks. Usually, with Cartesian coordinates, new points are created along the line-of-sight with a large variance. However, this type of initialization is subject to significant linearization issues making landmarks diverge from their real position. The immediate consequence is a failure of the Visual SLAM process. We propose here a new strategy that avoids or drastically limits the linearization errors. The first part of this strategy takes place during the tracking process where a coherent window is needed in order to successfully follow a point and make it converge. The second part concerns the update step. Due to linearization errors, a landmark in front of the observer can be updated behind it. We compute a corrective of the Kalman gain in order to preserve the integrity. We applied this strategy to real data illustrating its efficiency.

I. INTRODUCTION

The Simultaneous Localization And Mapping (SLAM) has received a lot of attention over the last two decades [10]. Though solutions to the main problems have been found [9], they mostly rely on the use of range and bearing sensors [2]. However, sensors like LRFs (Laser Range Finders) are expensive and usually only allow 2D perception. Conversely, monocular SLAM is cheap and allows to estimate the 6 DoF of a camera pose (3D position and the 3 associated angles). Nevertheless, cameras are bearing-only sensors. Estimating the accurate depth of a landmark requires many observations with a sufficient parallax. When dealing with Visual SLAM, it is thus a difficult problem to correctly initialize and track landmarks until they converge to their real position.

There exist various ways to perform SLAM. Among them, the most popular are Extended Kalman Filters (EKFs) [20], particle filters [11][22], bundle adjustment based methods [14] and Unscented Kalman Filters (UKFs) [5]. All these approaches can be expensive in terms of memory and processing time. Nevertheless, we chose the EKF as submapping methods [4][12] can be utilized in order to limit this load.

When dealing with bearing-only SLAM, two types of initialization can be found in the literature: delayed and undelayed ones. Delayed initialization, like in [1] and [8], has the advantage to only use accurately defined landmarks. However, it means that a large amount of information is lost. For example, when the vehicle is turning, landmarks will pass out of the field-of-view without being initialized leading to a large error on the localization of the robot. Concerning undelayed initializations, the easiest way to perform it would

be to create a point on the line-of-sight of its first observation with a large variance. However, this leads to significant linearization issues when tracking and updating landmarks [3]. To avoid this problem, the Inverse Depth (ID) parametrization is usually chosen to represent landmarks [7][16]. Yet, this solution is an over-parametrization of the problem where 6 parameters are required to estimate a 3D pose. Consequently, this representation is computationally expensive (size of the state vector duplicated) and so not suitable for long distance SLAM. In practice, a strategy allowing to switch from ID parametrization to Cartesian coordinates is used to prevent the computational load from becoming too significant [6]. In [15] and [21], multiple hypothesis are utilized for a single point. They are integrated into the state vector and discarded when considered wrong. This solution is also computationally expensive and needs a special strategy when performing the update step of the Kalman filter due to the wrong hypothesis inserted in the state vector.

In this paper, we propose a solution to the problems induced by the linearization steps when using a simple Cartesian representation for landmarks in the EKF framework. The problem we address here is twofold:

- Getting a coherent tracking window (uncertainty projection into the image plane) from a landmark when the real point is far from the fictitious one
- Updating correctly the state vector when the landmark is initialized far from its real position

The next section will show why tracking and update fail when using the classical Jacobian computation. Section III will present a solution to the first problem stated (tracking window). Then, in Section IV, a correction of the Kalman gain is proposed so as to avoid the loss of the integrity. Next, Section V will present the experiments conducted and the results obtained illustrating that our approach is efficient. Finally, Section VI will conclude and give some insights about what we are planning to do next.

II. PROBLEM STATEMENT

Considering an observation in the image plane $\mathbf{z}_i = (u \ v)^T$, we search the 3D point corresponding in the camera frame $\mathbf{z}_c = (x \ y \ z)^T$. A representation of the frames used in this paper is given in Figure 1 (the mobile camera evolves along the \vec{x} axis).

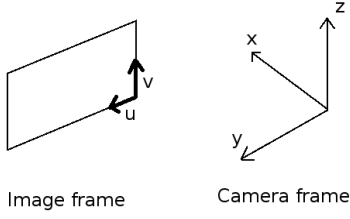


Fig. 1. Camera and image frames used in this paper.

The pinhole model provides the following relation:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{F} \cdot \mathbf{z}_c \quad (1)$$

with

$$\mathbf{F} = \begin{pmatrix} c_u & f_u & 0 \\ c_v & 0 & f_v \\ 1 & 0 & 0 \end{pmatrix}$$

where f_u and f_v are the focal distances in pixels and c_u and c_v represent the coordinates of the optical center in the image frame. We initialize the landmark with a fictive depth (x coordinate) corresponding to half the maximum distance possible for a point. This distance has to be chosen depending on the environment the camera is moving in. Thanks to this fictive depth, we can infer the pose of the fictitious 3D point:

$$\begin{cases} y = \frac{ux - c_u x}{f_u} \\ z = \frac{vx - c_v x}{f_v} \end{cases} \quad (2)$$

The covariance associated to this point also needs to be initialized. Let $g(\mathbf{z}_c, \mathbf{F})$ be the non-linear function linking the 3D point to the 2D one. Let \mathbf{G} be the Jacobian associated to g . This gives the following relation:

$$\mathbf{P}_i = \mathbf{G} \mathbf{P}_c \mathbf{G}^T \quad (3)$$

where \mathbf{P}_i and \mathbf{P}_c are respectively the covariance of the point newly initialized in the image and the camera frame:

$$\mathbf{P}_i = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{pmatrix} \quad \mathbf{P}_c = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 \end{pmatrix}$$

and \mathbf{G} is defined as follows:

$$\mathbf{G} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{pmatrix}$$

with

$$\frac{\partial u}{\partial x} = -\frac{f_u y}{x^2} \quad \frac{\partial u}{\partial y} = \frac{f_u}{x} \quad \frac{\partial u}{\partial z} = 0$$

$$\frac{\partial v}{\partial x} = -\frac{f_v z}{x^2} \quad \frac{\partial v}{\partial y} = 0 \quad \frac{\partial v}{\partial z} = \frac{f_v}{x}$$

The error made on a new point cannot be greater than the observation noise (defined as a diagonal matrix) as no updates (or other observations) have been made yet. To initialize properly the covariance of the 3D point, we rotate it on the \vec{x} axis of the camera frame. This leads to the following 3D point: $(\rho \ 0 \ 0)^T$ where $\rho = \sqrt{x^2 + y^2 + z^2}$ and consequently the following diagonal covariance:

$$\mathbf{P}'_c = \begin{pmatrix} (\rho - d_{min})^2 & 0 & 0 \\ 0 & \frac{\sigma_u^2 \rho^2}{f_u^2} & 0 \\ 0 & 0 & \frac{\sigma_v^2 \rho^2}{f_v^2} \end{pmatrix} \quad (4)$$

where d_{min} is the minimal distance from which the camera can see. \mathbf{P}'_c is then put on the line-of-sight of the observation thanks to the appropriate rotation. Finally, the estimated pose of the camera allows to put the point and its covariance in the world frame. The uncertainty associated to the position of the camera is taken into account during this step.

The point fully initialized is then tracked in order to make it converge towards its real position. Let $\hat{\mathbf{x}}$ be the state vector (including the camera and landmark poses). Let $h(\hat{\mathbf{x}}_{k|k-1})$ be the non-linear function (observation function) utilized to predict the measurement from the predicted state $\hat{\mathbf{x}}_{k|k-1}$ at time k and \mathbf{H}_k be the Jacobian associated to h . Let $\mathbf{H}_{c,k}$ be the Jacobian associated to the covariance of the mobile sensor $\mathbf{P}_{c,k|k-1}$ and $\mathbf{H}_{l_w,k}$ be the Jacobian associated to the covariance of the landmark l in the world frame. The prediction of the covariance of a landmark l in the image frame can be computed as follows:

$$\mathbf{P}_{l_i,k|k-1} = \mathbf{H}_{c,k} \mathbf{P}_{c,k|k-1} \mathbf{H}_{c,k}^T + \mathbf{H}_{l_w,k} \mathbf{P}_{l_w,k|k-1} \mathbf{H}_{l_w,k}^T \quad (5)$$

However, in (5), we linearize around the fictitious point which can be far from the real pose causing the projection of the ellipsoid corresponding to the covariance matrix to be wrong. The tracking issues resulting from this are illustrated in Figure 2 where no updates are performed (prediction only).

Section III will present a method to compute a bounding box that properly fits the projection of the ellipsoid into the image.

Another problem arises when performing the update step of the EKF. The equations of the update are as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \quad (6)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \quad (7)$$

where \mathbf{z}_k is the observation at time k and \mathbf{K}_k is the Kalman gain whose equation is:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (8)$$

where \mathbf{R}_k is the observation noise. The problem is similar to the one exposed previously. Indeed, we linearize around the

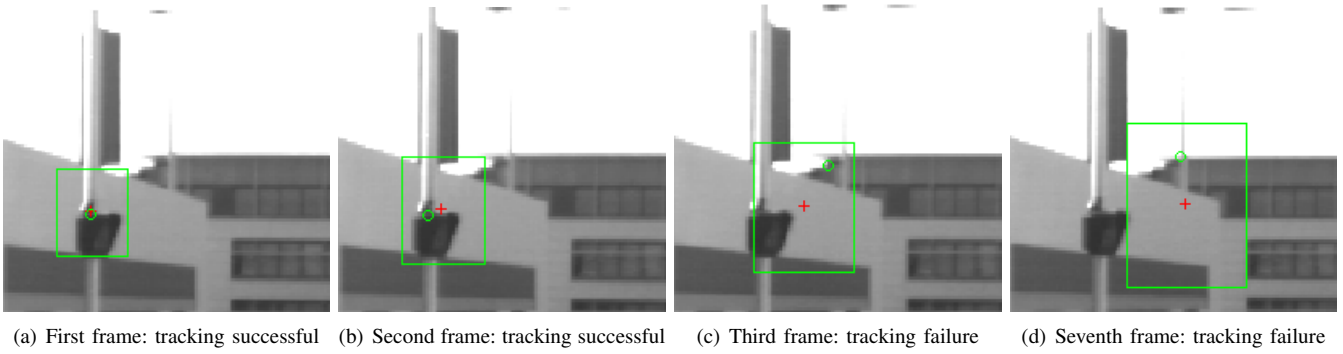


Fig. 2. Tracking failure due to significant linearization errors. The green rectangle is the predicted window where the landmark is supposed to be. The red cross is the estimate of the landmark position. The green circle is the observation (patch matched with ZNCC).

fictitious point leading to a wrong estimate. As a consequence, a landmark can be updated at a position behind the vehicle which is impossible. This problem is illustrated in Figure 3.

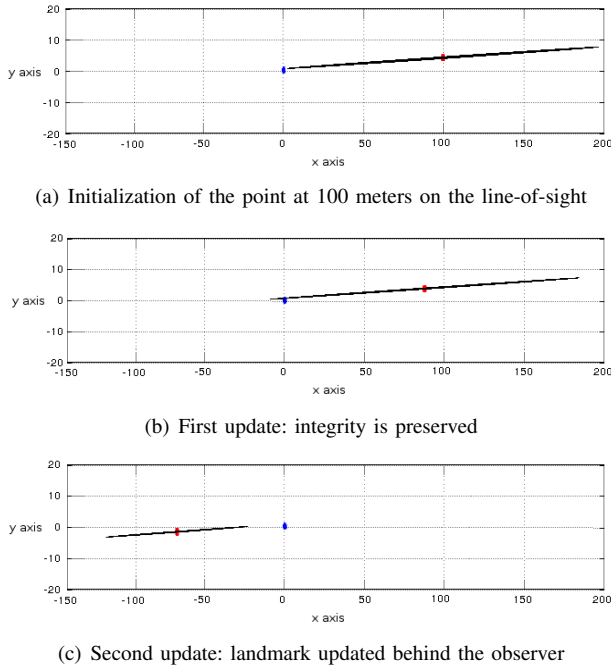


Fig. 3. Top view of a point updated behind the observer. The blue circle is the position of the vehicle. The red square is the landmark. The black ellipse is the uncertainty associated to the landmark.

This problem has also been stated and treated for ID parametrization in [17] but the proposed solution cannot be applied with Cartesian landmarks. In Section IV, we will present a solution working with a simple Cartesian pose that will correct the update step and prevent the landmarks from being updated at a wrong position.

III. BOUNDING BOX FOR TRACKING

We propose here a method to find the bounding box corresponding to the projection of the ellipsoid associated to a landmark uncertainty in the image frame thanks to its tangent planes. Other methods exist to find a proper projection. In

[18], Laplace's method is used to correctly fit a Gaussian at the global maximum of the observation function. However, a minimization step is required to find a correct solution. The approach proposed here does not rely on a minimization step. The idea is to find the planes tangent to the ellipsoid that give the maximum size of the corresponding ellipse in the image. Four tangent planes are needed in order to have a correct bounding box. Let \mathbf{P}_c be the covariance matrix of a landmark in the camera frame:

$$\mathbf{P}_c = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$$

and

$$\begin{aligned} \mathbf{P}_c^{-1} &= \frac{1}{\det \mathbf{P}_c} \begin{pmatrix} df - e^2 & ce - bf & be - cd \\ ce - bf & af - c^2 & cb - ae \\ be - cd & cb - ae & ad - b^2 \end{pmatrix} \\ &= \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \end{aligned}$$

Then, a relationship verified by the points on the surface of the ellipsoid \mathcal{E} generated by \mathbf{P}_c is the Mahalanobis distance:

$$\begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \quad (9)$$

where $(x_0 \ y_0 \ z_0)^T$ is the center of \mathcal{E} . The planes tangent to \mathcal{E} are also orthogonal to the surface normal. Moreover, the normal at a point on the surface is given by the gradient:

$$\begin{aligned} \vec{n} &= \nabla \mathcal{E}(x, y, z) = \left(\frac{\partial \mathcal{E}}{\partial x} \quad \frac{\partial \mathcal{E}}{\partial y} \quad \frac{\partial \mathcal{E}}{\partial z} \right)^T \\ &= \begin{pmatrix} 2(x - x_0)A + 2(y - y_0)B + 2(z - z_0)C \\ 2(x - x_0)B + 2(y - y_0)D + 2(z - z_0)E \\ 2(x - x_0)C + 2(y - y_0)E + 2(z - z_0)F \end{pmatrix} \end{aligned} \quad (10)$$

To be part of a plane p , orthogonal to the normal, a point must verify the following relation:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \in p \Leftrightarrow \begin{pmatrix} x_p - x \\ y_p - y \\ z_p - z \end{pmatrix}^T \cdot \vec{n} = 0 \quad (11)$$

As we are looking for the points located on the planes passing by the origin (camera), we can simplify equation (11):

$$(x \ y \ z) \cdot \vec{n} = 0 \quad (12)$$

The other constraint we can add is that the tangent planes intersect the image plane to form vertical or horizontal lines. This means that vertical planes will contain the \vec{z} axis and horizontal ones will contain the \vec{y} axis. Consequently, for the points belonging to the vertical planes:

$$\vec{z} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \quad (13)$$

and for those belonging to the horizontal planes:

$$\vec{y} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \quad (14)$$

These constraints give us the following system to solve for the vertical planes:

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x \ y \ z) \cdot \vec{n} = 0 \\ (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \end{cases} \quad (15)$$

and the following for the horizontal planes:

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T \mathbf{P}_c^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x \ y \ z) \cdot \vec{n} = 0 \\ (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \end{cases} \quad (16)$$

By solving the system (15), we find two points on the ellipsoid matching the different constraints. We then project them into the image thanks to the intrinsic parameters. This gives us the left and right limits of the bounding box. We also find two points by resolving (16) which gives us the top and bottom limits of the bounding box. Thanks to this information, we can successfully track the landmarks in the image plane. Figure 4 gives, with the same set of images used in Figure 2, the tracking results with the bounding box method presented here. The landmark is tracked successfully with no update steps (prediction only) illustrating the effectiveness of our approach.

IV. UPDATE CORRECTION

When dealing with a point newly initialized, the Jacobian associated to the observation function h can be erroneous as it is linearized around a fictitious point which can be far from the real landmark pose (cf. Fig. 3).

In order to avoid this problem, we introduce here a corrective of the Kalman gain [13] whose goal is to limit the impact of the state update on the point when it is needed. Indeed, as long as the update does not put the landmark before the observation, the Kalman gain is used without any correction. Otherwise, the idea is to have the following relation verified after the update:

$$\mathbf{z}_k = h(\hat{\mathbf{x}}_{k|k}) \quad (17)$$

Indeed, as the transformation from one space (image) to another of higher dimension (camera) is not well-conditioned, it is necessary to find a factor that makes (17) true. Let Ω_k be the Kalman gain once the corrective factor r has been applied. It is thus defined as follows:

$$\Omega_k = r \times \mathbf{K}_k \quad (18)$$

In the case of our Visual SLAM, for a landmark $\mathbf{x}_l = (x_l \ y_l \ z_l)^T$ defined in the world frame, the observation function can be written:

$$\begin{cases} u_{est} = \frac{\mathbf{F}_1 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})} \\ v_{est} = \frac{\mathbf{F}_2 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})} \end{cases} \quad (19)$$

where u_{est} and v_{est} correspond to the estimated position of the landmark in the image, \mathbf{F}_i is the i^{th} line of the intrinsic parameters matrix, \mathbf{R}_{cw} is the rotation matrix passing points from the camera frame to the world frame and \mathbf{t}_{cw} is the translation associated to the rotation. With $\mathbf{z}_k = (u_{obs} \ v_{obs})^T$ and by taking into account (17) and (18), we can write:

$$\begin{cases} u_{obs} = \frac{\mathbf{F}_1 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})} \\ v_{obs} = \frac{\mathbf{F}_2 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})}{\mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l + \Omega \Delta - \mathbf{t}_{cw})} \end{cases} \quad (20)$$

with the innovation $\Delta = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$ and Ω the corrected Kalman gain associated to \mathbf{x}_l . This leads to the two following corrections:

$$r_u = \frac{(u_{obs} - u_{est})De}{(\mathbf{F}_1 - u_{obs}\mathbf{F}_3)\mathbf{R}_{cw}^T \mathbf{K} \Delta} \quad (21)$$

$$r_v = \frac{(v_{obs} - v_{est})De}{(\mathbf{F}_2 - v_{obs}\mathbf{F}_3)\mathbf{R}_{cw}^T \mathbf{K} \Delta} \quad (22)$$

where $De = \mathbf{F}_3 \mathbf{R}_{cw}^T (\mathbf{x}_l - \mathbf{t}_{cw})$. As these scalars are correctors they cannot exceed 1. Keeping a corrective factor above 1 will cause the EKF to be overconfident leading to a wrong estimate. We will keep the lowest value from (21) and (22) to avoid this problem. If the chosen corrector is greater than 1, then no correction is applied.

This corrective factor will prevent the landmarks from being updated behind the observer. Linearization errors are drastically reduced and the integrity is maintained. More details

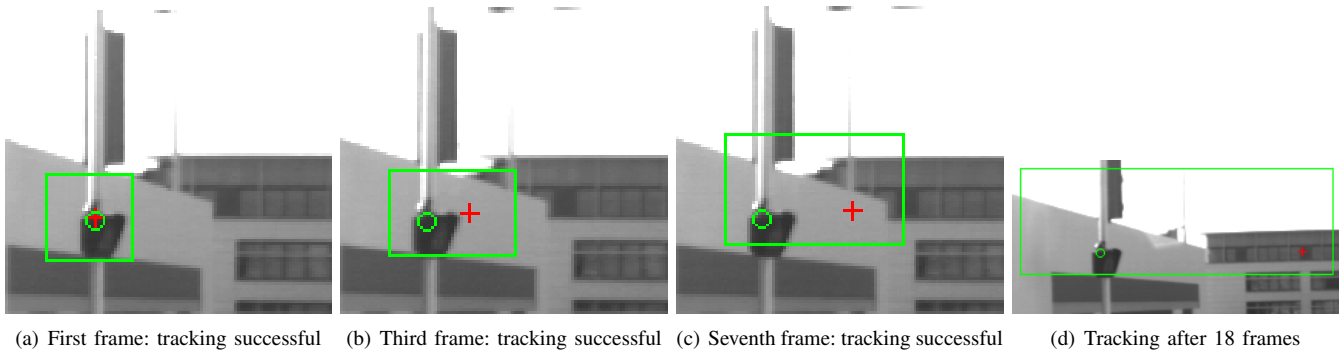
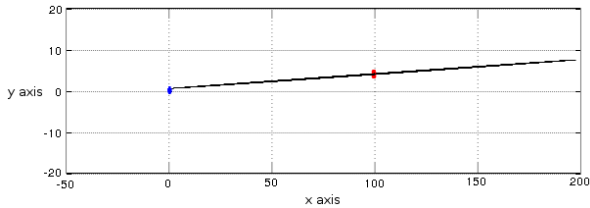
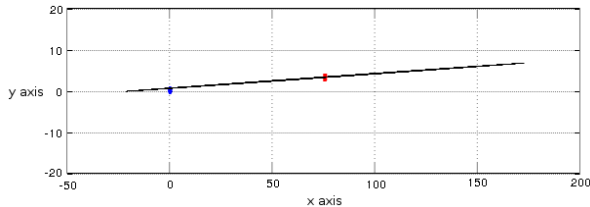


Fig. 4. Tracking with the bounding box method presented here. The green rectangle is the predicted window where the landmark is supposed to be. The red cross is the estimate of the landmark position. The green circle is the observation (patch matched with ZNCC).

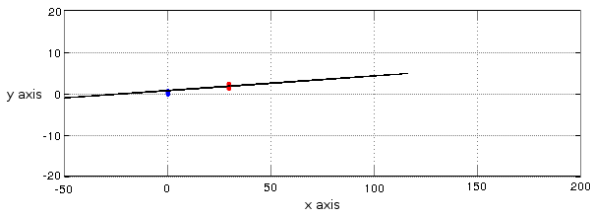
about this correction can be found in [13]. Figure 5 shows the same update case as Figure 3 with the corrective factor applied to the Kalman gain. The updated position of the landmark is coherent and converging towards its real one.



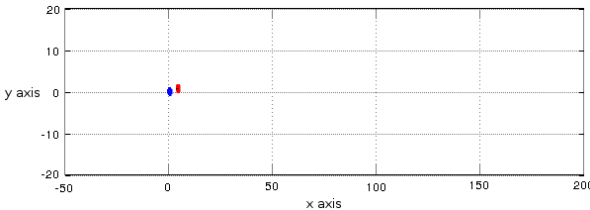
(a) Initialization of the point at 100 meters on the line-of-sight



(b) First update: integrity is preserved



(c) Second update: integrity still preserved



(d) After 10 updates, the landmark has converged

Fig. 5. Top view of a properly updated point thanks to the Kalman gain correction. The blue circle is the position of the vehicle. The red square is the landmark. The black ellipse is the uncertainty associated to the landmark.

V. EXPERIMENTS

In order to evaluate the efficiency of our monocular SLAM, a 30-meter trajectory has been performed. It was accomplished with an electrical vehicle (Cycab) in an small realistic urban experimental platform called PAVIN. This platform recreates a human environment with paved roads, crosswalks, traffic lights, building façades... Our vehicle was equipped with a single camera and proprioceptive sensors giving odometric information and the steering angle. A Differential GPS was also available but was only used as the ground truth for comparison purposes. The camera was mounted on top of the vehicle (2 meters) and acquired 15 images per second. The vehicle was traveling at approximately 1 meter per second. Landmarks were initialized by extracting areas of 15×15 pixels around Shi-Tomasi features [19]. Data association was performed using Normalized Cross Correlation between the previously extracted patch and the computed bounding box.

The trajectory was used to compare two algorithms: a naive implementation of a Visual EKF-SLAM with a simple initialization and a similar version with the strategy presented in this paper. This trajectory is composed of a long straight line to show that landmarks are tracked and updated correctly over a long distance and of a bend to demonstrate that the localization remains consistent with landmarks passing out of view quickly. Figure 6 shows the results of the two algorithms with at least 10 landmarks available per image.

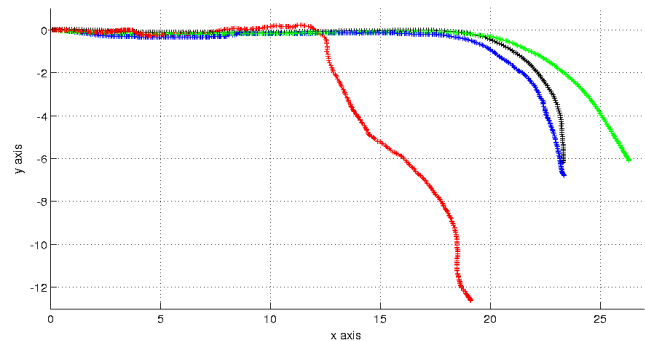


Fig. 6. Localization of the vehicle. **Black:** ground truth (DGPS). **Green:** prediction only (based on proprioceptive sensors). **Red:** naive EKF-SLAM. **Blue:** our strategy applied to the EKF-SLAM.

As expected, the naive implementation diverges completely whereas our approach remains consistent during the whole trajectory. Nevertheless, the restrictions on the size of the bounding box and the slow velocity of the vehicle have allowed the naive SLAM to be correct during the beginning of the trajectory. Thanks to our strategy, the bend has been followed quite closely even though the proprioceptive information was not accurate. These results can be improved with a smarter data association and more trusted landmarks especially when proprioceptive sensors are not reliable. The number of landmarks initialized is a good indicator of the quality of a monocular SLAM as new landmarks are only created when old ones cannot be matched (mostly due to linearization errors). Table I indicates the number of landmarks that have been initialized and that have converged during the whole trajectory.

	Naive SLAM	Our SLAM
Number of landmarks initialized	650	182
Number of landmarks conserved	79	61

TABLE I
NUMBER OF INITIALIZATIONS.

We can see that far less landmarks are initialized with our method. Indeed, tracking is possible as long as a landmark is visible and the update step is controlled by the corrective factor applied to the Kalman gain. As a consequence, landmarks converge within a few frames and are still being predicted correctly. However, an important number of landmarks are eliminated during the process. For most of them, they have disappeared of the field-of-view before convergence. For the others, a better data association could avoid this problem.

VI. CONCLUSION

We have presented a Visual EKF-SLAM that is able to use landmarks defined in Cartesian coordinates. The significant linearization errors occurring during the tracking and update steps are avoided or reduced thanks to the strategy exposed in this paper.

Our approach consists of a way to easily compute a bounding box that allows to track a landmark and make it converge towards its real position. This method, based on the computation of tangent planes, permits to find a bounding box which is coherent with the 3D uncertainty of a landmark.

The second part of this strategy permits to drastically reduce linearization errors during the update step. It is done thanks to the computation of a corrective factor able to reduce the impact of the update on a landmark. It allows to preserve the integrity when updating a landmark position.

We have evaluated our system with real data showing that our approach is efficient. We have compared our results with a naive implementation of a monocular EKF-SLAM and demonstrated that our strategy allows a more accurate localization while requiring less landmark initializations.

For future work, we plan to decentralize our monocular EKF-SLAM over several vehicles in order to have a better localization accuracy.

REFERENCES

- [1] T. Bailey. Constrained Initialisation for Bearing-Only SLAM. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1966–1971, 2003.
- [2] T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.
- [3] K. E. Bekris, M. Glick, and L. E. Kavraki. Evaluation of Algorithms for Bearing-Only SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1937–1943, 2006.
- [4] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas Framework for Scalable Mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1899–1906, 2003.
- [5] D. Checklov, M. Pupilli, W. Mayol-Cuevas, and A. Calway. Real-Time and Robust Monocular SLAM using Predictive Multi-Resolution Descriptors. *Advances in Visual Computing*, 4292:276–285, 2006.
- [6] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse Depth to Depth Conversion for Monocular SLAM. In *IEEE International Conference on Robotics and Automation*, pages 2778–2783, 2007.
- [7] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.
- [8] A. J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE International Conference on Computer Vision*, pages 1403–1410, 2003.
- [9] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [10] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(2):99–110, 2006.
- [11] E. Eade and T. Drummond. Scalable Monocular SLAM. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 469–476, 2006.
- [12] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.
- [13] T. Féraud, R. Chapuis, R. Aufrère, and P. Checchin. Improving Results of Non-Linear Observation Function Using a Kalman Filter Correction. In *International Conference on Information Fusion*, 2011. To appear.
- [14] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, 2007.
- [15] N. M. Kwok and G. Dissanayake. An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 736–741, 2004.
- [16] J. Montiel, J. Civera, and A. J. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Robotics: Science and Systems*, 2006.
- [17] M. P. Parsley and S. J. Julier. Avoiding Negative Depth in Inverse Depth Bearing-Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2066–2071, 2008.
- [18] A. Petrovskaya and S. Thrun. Model Based Vehicle Detection and Tracking for Autonomous Urban Driving. *Autonomous Robots*, 26(2):123–139, 2009.
- [19] J. Shi and C. Tomasi. Good Features to Track. In *Computer Vision and Pattern Recognition*, pages 593–600, 1994.
- [20] R. Smith, M. Self, and P. Cheeseman. A Stochastic Map for Uncertain Spatial Relationships. In *4th International Symposium on Robotics Research*, pages 467–474, 1988.
- [21] J. Solà, A. Monin, M. Devy, and T. Lemaire. Undelayed Initialization in Bearing Only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2499–2504, 2005.
- [22] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association. *Journal of Machine Learning Research*, 4(3):380–407, 2004.

Building Facade Detection, Segmentation, and Parameter Estimation for Mobile Robot Localization and Guidance

Jeffrey A. Delmerico
SUNY at Buffalo
jad12@buffalo.edu

Philip David
Army Research Laboratory
Adelphi, Maryland
phild@arl.army.mil

Jason J. Corso
SUNY at Buffalo
jcorso@buffalo.edu

Abstract—Building facade detection is an important problem in computer vision, with applications in mobile robotics and semantic scene understanding. In particular, mobile platform localization and guidance in urban environments can be enabled with an accurate segmentation of the various building facades in a scene. Toward that end, we present a system for segmenting and labeling an input image that for each pixel, seeks to answer the question “Is this pixel part of a building facade, and if so, which one?” The proposed method determines a set of candidate planes by sampling and clustering points from the image with RANSAC, using local normal estimates derived from PCA to inform the planar model. The corresponding disparity map and a discriminative classification provide prior information for a two-layer Markov Random Field model. This MRF problem is solved via Graph Cuts to obtain a labeling of building facade pixels at the mid-level, and a segmentation of those pixels into particular planes at the high-level. The results indicate a strong improvement in the accuracy of the binary building detection problem over the discriminative classifier alone, and the planar surface estimates provide a good approximation to the ground truth planes.

I. INTRODUCTION

Accurate scene labeling can enable applications that rely on the semantic information in an image to make high level decisions. Our goal of labeling building facades is motivated by the problem of global localization of mobile robots in GPS-denied areas. This problem arises in urban locations, so the approach currently being developed by our group depends on detection of buildings within the field of view of the cameras on a mobile platform. In particular, with known facade orientations and an overhead view of the region from which building footprints can be extracted, we are working toward accurate global localization (as normally provided by GPS) from only semantic information. Within this problem, accurate detection and labeling is critical for the high level localization and guidance tasks. We restrict our approach to identifying only planar building facades, and require image input from a stereo source. Since most buildings have planar facades, and many mobile robotic platforms are equipped with stereo cameras, neither of these assumptions is particularly restrictive. The application of this work to localization of mobile platforms is forthcoming, but our methods are intended for extension to that problem.

In this paper, we propose a method for building facade detection (binary labeling) in stereo images that further segments the individual facades and estimates the parameters

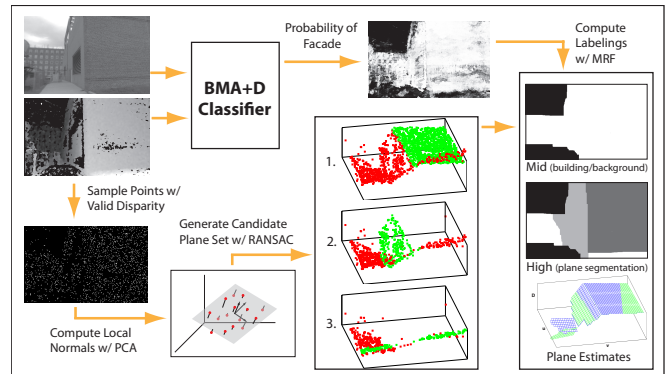


Fig. 1. Workflow of the proposed method. We use our BMA+D classifier to compute a probability map for binary classification, generate a set of candidate planes with parameter estimates using a RANSAC model which incorporates local PCA normal approximations, and then we use a two-layer MRF to compute labelings for the binary classification at the mid-level and for facade segmentation at the high-level.

of their 3D models. Our approach proceeds in three main steps: discriminative modeling, candidate plane detection through PCA and RANSAC, and energy minimization of MRF potentials. A diagram of the workflow for candidate plane detection and high-level labeling is provided in Fig. 1. Each step of this process is explained in Section II.

Our work leverages stereo information from the beginning. Our discriminative model is generated from an extension of the Boosting on Multilevel Aggregates (BMA) method [1] that includes stereo features [2] on the disparity map, which we call BMA+D. Boosting on Multilevel Aggregates uses hierarchical aggregate regions coarsened from the image based on pixel affinities, as well as a variety of high-level features that can be computed from them, to learn a model within an AdaBoost [3] two- or multi-class discriminative modeling framework. The multilevel aggregates exploit the propensity of these coarsened regions to adhere to object boundaries, which in addition to the expanded feature set, offer less polluted statistics than patch-based features, which may violate those boundaries. Since many mobile robot platforms are equipped with stereo cameras, and can thus compute a disparity map for their field of view, our approach of using statistical features of the disparity map is a natural extension of the BMA approach given our intended platform. Since buildings tend to have planar surfaces on their exteriors, we use the stereo features to exploit the property

that planes can be represented as linear functions in disparity space and thus have constant spatial gradients [4]. We use the discriminative classification probability as a prior for inference of facade labeling.

In order to associate each building pixel with a particular facade, we must have a set of candidate planes from which to infer the best fit. We generate these planes by sampling the image and performing Principal Component Analysis (PCA) on each local neighborhood to approximate the local surface normal at the sampled points. We then cluster those points by iteratively using Random Sample Consensus (RANSAC) [5] to find subsets which fit the same plane model and have similar local normal orientations. From these clusters of points, we are able to estimate the parameters of the primary planes in the image.

We then incorporate both of these sources of information into a Bayesian inference framework using a two-layer Markov Random Field (MRF). We represent the mid-level MRF as an Ising model, a layer of binary hidden variables representing the answer to the question “Is this pixel part of a building facade?” This layer uses the discriminative classification probability as a prior, and effectively smooths the discriminative classification into coherent regions. The high-level representation is a Potts model, where each hidden variable represents the labeling of the associated pixel with one of the candidate planes, or with no plane if it is not part of a building. For each pixel, we consider its image coordinates and disparity value, and evaluate the fitness of each candidate plane to that pixel, and incorporate it into the energy of labeling that pixel as a part of that plane. A more in-depth discussion of these methods can be found in Section II-B.1.

We use the Graph Cuts energy minimization method [6] to compute minimum energy labelings for both levels of our MRF model.

A. Related Work

Building facade detection and segmentation have been and continue to be well-studied problems. Many recent papers in the literature have focused on segmentation of building facades for use in 3D model reconstruction, especially in the context architectural modeling or geo-spatial mapping applications such as Google Earth. Korah and Rasmussen use texture and other *a priori* knowledge to segment building facades, among other facade-related tasks [7]. Wendel et al. use intensity profiles to find repetitive structures in coherent regions of the image in order to segment and separate different facades [8]. Hernández and Marcotegui employ horizontal and vertical color gradients, again leveraging repetitive structures, to segment individual facades from blocks of contiguous buildings in an urban environment [9].

Several other methods utilize vanishing points for planar surface detection. David identifies vanishing points in a monocular image by grouping line segments with RANSAC and then determining plane support points by the intersection of the segments which point toward orthogonal vanishing point ultimately clustering them to extract the planes of the

facade [10]. Bauer et al. implement a system for building facade detection using vanishing point analysis in conjunction with 3D point clouds obtained by corresponding a sweep of images with known orientations [11]. Lee et al. use a line clustering-based approach, which incorporates aerial imagery, vanishing points, and other projective geometry cues to extract building facade textures from ground-level images, again toward 3D architectural model reconstruction [12].

Our work draws on the contributions of Wang et al., whose facade detection method using PCA and RANSAC with LiDAR data inspired our approach with stereo images [13]. Perhaps the approach most similar in spirit to ours is that of Gallup et al. [14], who also use an iterative method for generating candidate plane models using RANSAC, and also solve the labeling problem using graph cuts [6]. However, their approach relies on multiview stereo data and leverages photoconsistency constraints in their MRF model, whereas we perform segmentation with only single stereo images. In addition, on a fundamental level their method involves finding many planes that fit locally, and stitching them together, whereas we aim to extract our planar models from the global data set, without an explicit restriction on locality. We present quantitative results on the accuracy of our planar modeling as well.

Although many of these results are directed toward 3D model reconstruction, some other work has been focused toward our intended application of vision-based navigation, namely [10], [15], [16]. Additionally, our work is focused on retrieval of the estimated plane parameters, as implemented in the planar surface model of [4], and not on 3D model reconstruction.

II. METHODS

Please refer to Fig. 1 for a diagrammatic representation of how the following methods interface.

A. BMA+D Classifier

We implement the standard Boosting on Multilevel Aggregates algorithm described in [1], but with extensions for working with disparity maps and their associated features. These additions include accommodations for working with invalid data in the disparity map: areas of the scene outside the useful range of the stereo camera, and dropouts where the disparity can not be computed within the camera’s range due to occlusion or insufficient similarity between the images for a match at that point. Although in principle any classifier could be used for this step, so long as it could produce a probability map for binary classification in identifying building pixels, we developed the BMA+Disparity Classifier as a way to incorporate problem-specific knowledge into the boosting framework.

B. MRF Model and Facade Parameter Estimation

1) *Plane Parameters*: Throughout this discussion, we assume that we have stereo images which may or may not

be calibrated. Since we do not aim for full 3D reconstruction, the camera's calibration parameters can be unknown but constant. Thus, we can determine the surface normal parameters up to a constant that describes the camera parameters; and since that constant will be the same across all candidate planes, we can use the computed surface normals to differentiate between planes.

A plane in 3D space can be represented by the equation $ax + by + cz = d$ and for non-zero depth, z , this can be rewritten as:

$$a\frac{x}{z} + b\frac{y}{z} + c = \frac{d}{z} \quad (1)$$

We can map this expression to image coordinates by the identities $u = f \cdot \frac{x}{z}$ and $v = f \cdot \frac{y}{z}$, where f is the focal length of the camera. We can also incorporate the relationship of the stereo disparity value at camera coordinate (u, v) to the depth, z , using the identity $D(u, v) = \frac{fB}{z}$, where D is the disparity and B is the baseline of the stereo camera. Our plane equation reduces to:

$$a\frac{u}{f} + b\frac{v}{f} + c = \frac{d \cdot D(u, v)}{fB} \quad (2)$$

$$\left(\frac{aB}{d}\right)u + \left(\frac{bB}{d}\right)v + \left(\frac{cfB}{d}\right) = D(u, v) \quad (3)$$

Although $\mathbf{n} = (a, b, c)^T$ is the surface normal in world coordinates, for our purposes we can seek to determine the following uncalibrated plane parameters $\mathbf{n}' = (a', b', c')$, where:

$$a' = \frac{aB}{d}, b' = \frac{bB}{d}, c' = \frac{cfB}{d} \quad (4)$$

$$\mathbf{n}' \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = a'u + b'v + c' = D(u, v) \quad (5)$$

This new set of plane parameters relates the image coordinates and their corresponding disparity values by incorporating the constant but unknown camera parameters.

2) *Candidate Plane Detection*: We perform the second phase of our approach by iteratively using RANSAC to extract a set of points which fit a plane model in addition to having a local normal estimate which is consistent with the model. The extracted plane models become the set of candidate planes for our high-level MRF labeling. Each pixel in the image will be labeled by the MRF as belonging to one of these candidate planes or else assigned a null label.

a) *Local Normal Estimation*: Based on our assumption of rectilinear building facades, we can use Principal Component Analysis to determine a local normal to a point in disparity space as in [17]. We first construct the covariance matrix of the neighborhood around the point in question. To do this, we consider all points $p_i = (u_i, v_i, -D(u_i, v_i))$ with valid disparity in a 5×5 window centered on this point. Note that stereo cameras that compute the disparity map with onboard processing in real-time often do not produce dense disparity maps, so the neighborhood may be sparse. Other neighborhood sizes could be used, but we found that a 5×5 window provided good estimates while remaining

local. We compute the centroid, $\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i$, of the points $\{p_i\}_{i=1 \dots N}$ in the neighborhood, and calculate the 3×3 covariance matrix with:

$$W = \frac{1}{N} \sum_{i=1}^N (p_i - \bar{p}) \otimes (p_i - \bar{p}) \quad (6)$$

where \otimes is the outer product. We then compute the eigenvalues of W , and the eigenvectors corresponding to the largest two eigenvalues indicate the directions of the primary directions on the local planar estimate. The eigenvector corresponding to the smallest eigenvalue thus indicates the direction of the local surface normal, $\mathbf{n}_{(u,v)}$.

b) *RANSAC Plane Fitting*: We take a sample, S , of image points with valid disparity, and compute the local planar surface normal estimates by the aforementioned method. We then seek to fit a model to some subset of S of the form:

$$\alpha v + \beta u + \epsilon(-D(u, v)) + \theta = 0 \quad (7)$$

where $\tilde{\mathbf{n}} = \frac{1}{\epsilon}(\alpha, \beta, \theta)$ is the surface normal from Eq. (5). Since RANSAC finds the largest inlier set, P_{in} , that it can among S , we will fit the most well-supported plane first [5]. We then remove the inliers, $S' = S \setminus P_{in}$, and repeat this process iteratively, finding progressively less well-supported planes, until a fixed percentage of the original S has been clustered into one of the extracted planes. In our experiments, we used a sample of 2000 points from the image, and concluded the plane extraction once 80% of the points had been clustered, or when RANSAC failed to find a consensus set among the remaining points. We assume Gaussian noise on the inlier set for our RANSAC plane model, and throughout we use a standard deviation of $\sigma_\eta = 5$.

Although we use RANSAC to fit a standard plane model, we use a modified error term in order to incorporate the information in the local normal estimates. Here, since our local normal estimate required the use of a three dimensional coordinate system $(u, v, -D(u, v))$, and produces a normal of that form, we must use a slightly different normal formulation of $\mathbf{n}_m = (\alpha, \beta, \epsilon)$. The standard measure of error for a plane model is the distance of a point from the plane: $E_m = |\alpha v + \beta u + \epsilon(-D(u, v)) + \theta|$, assuming $\mathbf{n}_m = (\alpha, \beta, \epsilon)$ is a unit vector. We compute another measure of error, E_{norm} , the dot product of the model plane normal \mathbf{n}_m and the local normal estimate $\mathbf{n}_{(u,v)}$, which is the cosine of the dihedral angle between the two planes defined by those normals. If we take its magnitude, this metric varies from 0 to 1, with 1 representing normals which are perfectly aligned, and 0 representing a dihedral angle of 90° . Since the range of E depends on the properties of the image (resolution, disparity range), we combine these two metrics as follows:

$$E = (2 - E_{norm})E_m = (2 - |\langle \mathbf{n}_m, \mathbf{n}_{(u,v)} \rangle|)E_m \quad (8)$$

such that the dihedral angle scales the error term from E_m to $2E_m$, depending on the consistency of the model and local normals.

3) *MRF Model*: We model our problem in an energy minimization framework as a pair of coupled Markov Random Fields. Our mid-level representation seeks to infer the correct configuration of labels for the question “Is this pixel part of a building facade?” Based on this labeling, the high-level representation seeks to associate those pixels which have been positively assigned as building facade pixels to a particular candidate plane. Our motivation for this design stems from the fact that these are related but distinct questions, and they are informed by different approaches to modeling buildings. The mid-level MRF represents an appearance-based model, while the high-level MRF represents a generative model for the planar facades.

a) *Mid-level Representation*: We want our energy function for the mid-level model to capture the confidence (probability) of our discriminative classification, and we want there to be a penalty whenever a pixel with a high confidence is mislabeled, but a smaller penalty for pixels with lower confidence in their a priori classification. We will use an Ising model to represent our mid-level MRF, where our labels x_s for $s \in \lambda$, our image lattice, come from the set $\{-1, 1\}$. We define a new variable b_s to represent a mapping of the $X_s \in \{-1, 1\}$ label to the set $\{0, 1\}$ by the transformation $b_s = \frac{X_s + 1}{2}$. For a particular configuration of labels l , we define our mid-level energy function as:

$$E(l) = \sum_{s \in \lambda} [(1 - b_s)p(s) + b_s(1 - p(s))] - \gamma \sum_{s \sim t} x_s x_t \quad (9)$$

where $p(s)$ is the discriminative classification probability at s and γ is a constant weighting the unary and binary terms. The b_s quantity in the unary term essentially switches between a penalty of $p(s)$ if the label at s is set to -1 , and a penalty of $1 - p(s)$ if the label at s is set to 1 . Thus for $p(s) = 1$, labeling $x_s = -1$ will incur an energy penalty of 1 , but labeling $x_s = 1$ will incur no penalty. Similarly for $p(s) = 0$, labeling $x_s = -1$ will incur no penalty, but labeling it 1 will incur a penalty of 1 . A probability of 0.5 will incur an equal penalty with either labeling. Our smoothness term is from the standard Ising model. In our experiments, we used a γ value of 10 .

b) *High-level Representation*: In designing our energy function for the high-level MRF, we want to penalize points which are labeled as being on a plane, but which do not fit the corresponding plane equation well. Our label set for labels $y_s, s \in \lambda$, is $\{0, \dots, m\}$, with m equal to the number of candidate planes identified in the plane detection step. It corresponds to the set of candidate planes indexed from 1 to m , as well as the label 0 , which corresponds to “not on a plane”. We define a set of equations $E_p(s)$ for $p \in \{0, \dots, m\}$ such that

$$E_p(s) = |a'_p u + b'_p v + c'_p - D(s)| \quad (10)$$

where the surface normal $\mathbf{n}'_p = (a'_p, b'_p, c'_p)$ corresponds to the plane with label p , and $D(s)$ is the disparity value at s . We normalize this energy function by dividing by the maximum disparity value, in order to scale the maximum energy penalty down to be on the order of 1 . For consistency

in our notation, we define $E_0(s)$ to be the energy penalty for a label of 0 at s , corresponding to the “not on a plane” classification. We set $E_0(s) = b_s$, such that a labeling of -1 in the mid-level representation results in $b_s = 0$, so there is no penalty for labeling s as “not on a plane”. Similarly, when $x_s = 1, b_s = 1$, so there is a penalty of 1 to label any of the non-planar pixels as a plane.

To construct our overall energy function for the high-level MRF, we incorporate the exponential of the set of planar energy functions E_p with a delta function, so the energy cost is only for the plane corresponding to the label y_s . Since we cannot compute E_p without a valid disparity value, we use an indicator variable $\chi_D \in \{0, 1\}$ to switch to a constant energy penalty for all planes and the no-plane option, in order to rely strictly on the smoothness term for that pixel’s label. For the smoothness term, we use a Potts model, weighted like the mid-level representation with a constant γ . In our experiments, though, this value of γ was 1 . Thus the high-level energy function we are seeking to minimize is:

$$E(l) = \sum_{s \in \lambda} \sum_{p=0}^m \delta_{y_s=p} \cdot \exp(\chi_D E_p(s)) + \gamma \sum_{s \sim t} \delta_{y_s=y_t} \quad (11)$$

III. EXPERIMENTAL RESULTS

We have performed quantitative experiments using our method on a new dataset that consists of 141 grayscale images from the left camera of a stereo imager¹ each with a corresponding 16 -bit disparity map. All images have 500×312 resolution and human-annotated ground truth for both binary classification and facade segmentation. There are a total of 251 facades represented in the dataset, and for each one, we have computed a gold-standard plane model from its ground truth facade segmentation. We are not aware of another publicly available, human-annotated, quantitative stereo building facade dataset, and we believe this can become a benchmark for the community. We performed 6 -fold cross-validation with our BMA+D classifier and computed the facade segmentations and plane estimates based on the corresponding trained models.

A. Facade Detection

The mid-level MRF results exhibit improvement in accuracy over BMA+D alone; Table I shows a quantitative comparison of these two methods. With the Bayesian inference of the MRF, we achieve a classification accuracy of almost 80% for each class, and an improvement in overall accuracy of 3% over BMA+D.

B. Facade Segmentation and Parameter Estimation

We computed the facade segmentations and the plane parameters for each of the labeled planes in all of the images from the dataset; some examples are shown in Figure 2. For each of the manually labeled ground truth planes in the dataset, we computed ground truth parameters by sampling the labeled region and using RANSAC to determine the

¹TyZx DeepSea V2 camera with 14 cm baseline and 62° horizontal field of view.

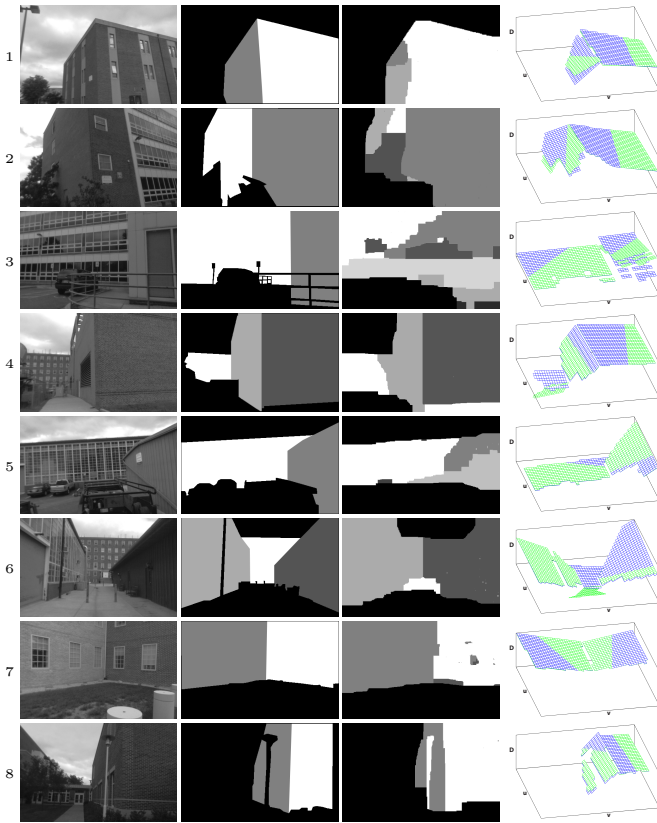


Fig. 2. Segmentations and planar facade estimates on multi-facade images. For each example, they are (L to R) the original image, ground truth segmentation, high-level MRF labeling, and 3D plane projection. In the plane projection plots, the perspective of the original image is looking down on the 3D volume from the positive D-axis. The ground-truth planes are in blue, and the estimated planes are in green (view in color).

plane parameters. Out of 251 total facades in the set, 40 of them were misclassified as background by the mid-level labeling. The other 211 facades were labeled with at least one candidate plane in the high-level labeling for a detection rate of 84%.

As noted above, some of the ground truth facades are not detected by the mid-level MRF, but multiple segmented planes per ground truth facade are also common. In order to assess the accuracy of our plane parameter estimation, we compute a weighted error measure as the mean pixel-wise angular error between the labeled plane and the ground truth facade, averaged over all pixel in the dataset where the ground truth and high-level labeling are both non-null. Our angular error metric is the dihedral angle between the estimated plane and the ground truth plane (with normal vectors \mathbf{n}_e and \mathbf{n}_g , respectively): $\phi = \arccos(\mathbf{n}_e \cdot \mathbf{n}_g)$. The average angular error for any such pixel over the entire dataset is 24.07° .

A histogram showing the number of pixels labeled with a plane model having angular error in each bin (see Fig. 3) indicates that the peak of the distribution of errors is the range of $0 - 10^\circ$. Similarly, the examples shown in Figure 2 indicate that some facades are modeled very accurately, while others have high angular error. This discrepancy motivates our further analysis, which we discuss in the next section.

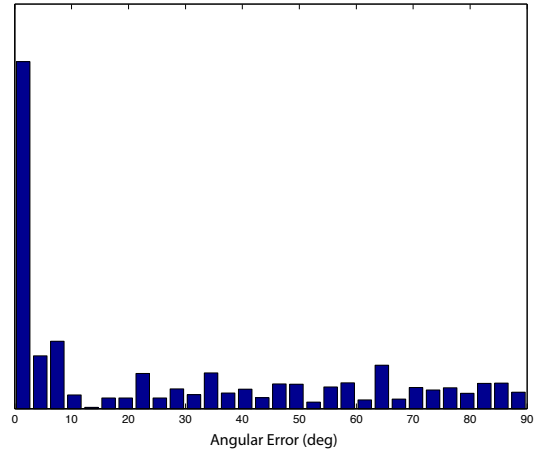


Fig. 3. Histogram representing the number of pixels labeled with a plane model having corresponding angular error.

C. Analysis

Our method often segments a detected facade into multiple plane labels, which makes 1-to-1 comparison difficult. In order to overcome this challenge, and to examine the error distribution of Fig. 3 further, we consider two methods for comparing the segmentations to the ground truth. First, for each ground truth facade, we compare to it the plane whose label occupies the largest portion of that facade’s area in our segmentation. We have noticed that there is often one (or more) accurate plane estimate on each ground truth facade, but it may only cover a minority of the ground truth facade. For example, in the second row of Figure 2, the facade on the left in the ground truth is best modeled by the plane corresponding to the white label in the estimate, but the majority of that facade is labeled with less accurate planes. In order to measure the accuracy of our method in estimating at least some portion of each ground truth facade, our second method of comparison chooses the most accurate plane estimate out of the set of labels that cover each facade’s region. In both cases, we compute the average angular error between the chosen segmented plane (largest or best) and the ground truth facade, weighted by the size of the segment, as well as the average percentage of the ground truth facade covered by the chosen label. These results are collected in Table II. Analysis of the error per segment for both methods indicates that most of the high-error segmentations occur with small areas: the vast majority of facades larger than 10 % of the frame have less than 10 degree error. This implies that the errors are generally small ($< 10^\circ$) for the major facades in the image, and it may be possible to restrict or post-process the labeling to eliminate the minor and erroneous plane labels, although that is beyond the scope of this paper.

The quality of the disparity map is likely to be at least somewhat responsible for this phenomenon, as the usable range of most stereo cameras is limited. For example, the camera used to capture our dataset can only resolve features up to 45 cm at a distance of 15 m. Thus, even moderately distant facades are likely to be significantly more prone to

TABLE I
ACCURACY AND MISCLASSIFICATION RATES FOR THE MID-LEVEL MRF LABELING AND THE BMA+D CLASSIFIER.

		BG	Building
BMA+D MRF	BG	75.33 79.98	24.67 20.01
BMA+D MRF	Building	23.51 21.15	76.49 78.85
BMA+D MRF	F-scores	0.7421 0.7773	

TABLE II
ACCURACY FOR OUR TWO METHODS OF COMPARISON TO GROUND TRUTH: LARGEST SEGMENT AND MOST ACCURATE SEGMENT

Method	Avg. Err.	Avg. Size (% of GT area)
Largest	21.973	66.57
Best	13.765	53.00

large errors in their estimates; they will be both small in the frame and less likely to find an accurate consensus set in RANSAC due to the uncertainty in their disparity values. Similarly, for a facade with many invalid disparity values, it may not be sampled adequately, and the points it does have may erroneously be included as part of an inlier set that does not actually lie on the facade. Perhaps on account of this phenomenon, we have observed that many of the high-error segmentations are rotated primarily about a horizontal axis, but are much more accurate in their rotation about a vertical axis. Under the assumption that facades tend to be vertical planes, in the future we intend to explore the possibility of incorporating a verticality constraint into the RANSAC plane model to restrict the candidate plane set to only vertical plane models.

Without the context of the ground truth facade segmentation, it would not be possible to choose the largest or best label as we do in this analysis, but it is encouraging that on average we're able to achieve $< 15^\circ$ error over a majority of each facade. This result will motivate some of our future work in developing ways to better disambiguate the labels in order to decrease those average errors and increase the area of the most accurate labels.

IV. CONCLUSIONS

We have presented a system for automatic facade detection, segmentation, and parameter estimation in the domain of stereo-equipped mobile platforms. We use a discriminative model that leverages both appearance and disparity features for improved classification accuracy. From the disparity map, we generate a set of candidate planes using RANSAC with a planar model that also incorporates local PCA estimates of plane normals. We combine these in a two-layer Markov Random Field model which allows for inference on the binary (building/background) labeling at the mid-level, and for segmentation of the identified building pixels into individual planar surfaces corresponding to the candidate plane models determined by RANSAC.

The combination of the BMA+D discriminative model and the mid-level MRF are able to achieve a classification accuracy of approximately 80%. We were able to identify 84% of the building facades in our dataset, with an average angular error of 24° from the ground truth. However, the distribution of errors peaks in frequency below 10° , indicating that a large percentage of the labels provide very accurate estimates for the ground truth, although some of the labels produced by our method have very high error. Further analysis shows that these high-error labelings most often occur on small segmented regions. Thus our method produces accurate plane estimates for at least the major facades in the image.

REFERENCES

- [1] J. Corso, "Discriminative modeling by boosting on multilevel aggregates," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [2] J. Delmerico, J. Corso, and P. David, "Boosting with Stereo Features for Building Facade Detection on Mobile Platforms," in *e-Proceedings of Western New York Image Processing Workshop*, 2010.
- [3] Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [4] J. Corso, D. Burschka, and G. Hager, "Direct plane tracking in stereo images for mobile navigation," in *IEEE International Conference on Robotics and Automation*, 2003.
- [5] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2002.
- [7] T. Korah and C. Rasmussen, "Analysis of building textures for reconstructing partially occluded facades," *Computer Vision–ECCV 2008*, pp. 359–372, 2008.
- [8] A. Wendel, M. Donoser, and H. Bischof, "Unsupervised Facade Segmentation using Repetitive Patterns," *Pattern Recognition*, pp. 51–60, 2010.
- [9] J. Hernández and B. Marcotegui, "Morphological segmentation of building façade images," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2010, pp. 4029–4032.
- [10] P. David, "Detecting Planar Surfaces in Outdoor Urban Environments," ARMY Research Lab, Adelphi, MD. Computational and Information Sciences Directorate, Tech. Rep., 2008.
- [11] J. Bauer, K. Karner, K. Schindler, A. Klaus, and C. Zach, "Segmentation of building models from dense 3D point-clouds," in *Proc. 27th Workshop of the Austrian Association for Pattern Recognition*. Citeseer, 2003, pp. 253–258.
- [12] S. Lee, S. Jung, and R. Nevatia, "Automatic integration of facade textures into 3D building models with a projective geometry based line clustering," in *Computer Graphics Forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 511–519.
- [13] R. Wang, J. Bach, and F. Ferrie, "Window detection from mobile LiDAR data," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 58–65.
- [14] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1418–1425.
- [15] J. Kosecka and W. Zhang, "Extraction, matching, and pose recovery based on dominant rectangular structures," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 274–293, 2005.
- [16] W. Zhang and J. Kosecka, "Image Based Localization in Urban Environments," in *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*. IEEE Computer Society, 2006, pp. 33–40.
- [17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points." *Computer Graphics (ACM)*, vol. 26, no. 2, pp. 71–78, 1992.



Session V

Mobile robot modeling and control

- **Keynote speaker: R. Lenain (Cemagref, France)**
Title: Generic algorithm for high accurate trajectory control in different conditions
Co-authors: B. Thuilot, C. Cariou, P. Martinet
- **Title: A control strategy taking advantage of inter-vehicle communication for platooning navigation in urban environment**
Authors: P. Avanzini, B. Thuilot, P. Martinet
- **Title: Semiautonomous Longitudinal Collision Avoidance Using a Probabilistic Decision Threshold**
Authors: J. Johnson, Y. Zhang, K. Hauser



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Session V

Keynote speaker: **R. Lenain (Cemagref, France)**

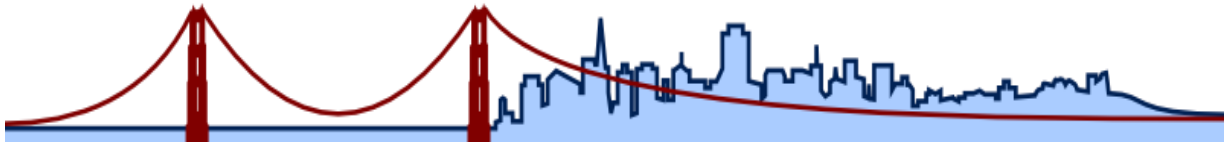
Generic algorithm for high accurate trajectory control in different conditions

Co-authors: B. Thuilot, C. Cariou, P. Martinet

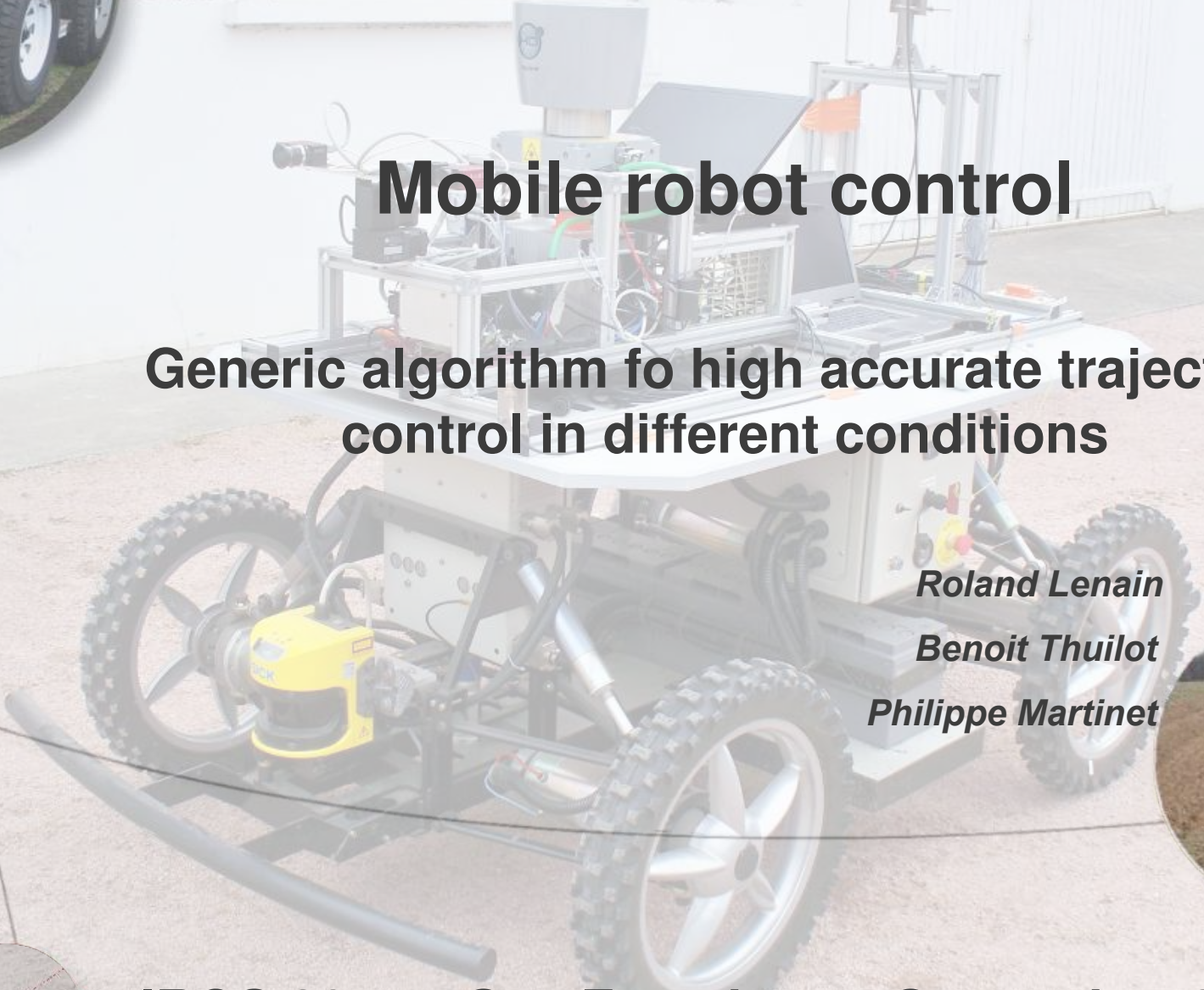
Abstract: From public transportation to agriculture, many fields of application may benefit from automation in the area of mobile robotics. As a result, research in that topic is subject of more and more investigation in order to propose new systems, from driver assistance (e.g automatic parking...) up to fully autonomous vehicles (such as autonomous robots acting in hazardous environment). In order to be fully effective, these innovations have to be accurate, safe, and able to act in various conditions. Many open problems then need to be addressed in order to propose such innovations in several part. If perception and navigation issues constitute important key points, the problem of motion control remains an important point since the control law to be embedded have to face a variability of conditions impacting directly their behaviour. These conditions rely on constant parameter, pending on the considered robot or vehicle (mechanical properties, actuators, specifications, ...), but also depends on the variable interaction with the environment (grip conditions, terrain geometry, reachable velocity, ...). As a result in order to propose an efficient and accurate motion whatever the conditions variability, control laws have to account of the different dynamics encountered. This talk investigates the motion control of mobile robot in different conditions through the example of path tracking. It proposes several strategies to preserve the motion accuracy and safety whatever the encountered conditions. A correlation between the reachable velocity and the terrain complexity is proposed to extract the different effects which have to be accounted and related control objective. Based on this classification several modelling and control strategies are illustrated to face the considered phenomena. Starting from classical kinematic controller for simple path tracking task at low speed on flat terrain with good grip conditions, the talk investigates a rising complexity of situation. Adaptive control based on advanced kinematic model is proposed to face low grip conditions. This adaptive control is then associated with predictive control in order to preserve accuracy when increasing the velocity. Limitation of this controller with respect to the increasing speed and safety is pointed out and a new observer mixing kinematic and dynamic representation is proposed. This model permits also to account for 3D motion and permit to investigate the risk of instability rising at high speed. A control law acting on velocity in order to limit the rollover risk is then derived. This notion is then extended in a predictive way to adress the topic of obstacle avoidance and traversability for mobile robots. Finally, the notion of predictive control on velocity is extended to preserve the integrity of mobile robot, i.e, to preserve the stability, the controlability, and the accuracy of motion control. The capabilities of the different algorithm are investigated on actual experiments, using different kind of robots and vehicle, moving on different kind of ground.

Biography: Roland LENAIN is a research fellow in Cemagref on the topic of off-road robotics. His research interests include the modeling and the control of mobile robots, submitted to uncertain effects. This covers the motion control as well as safety aspects (rollover, instability). These works are applied in various areas: agriculture and environment, driver assistance or military aspects.

Currently in Cemagref since 2006, Roland Lenain is in Charge of several National Project in the topic of mobile robot and vehicle control in hazardous context (FAST, ActiSurTT, ...). He achieved a Post Doctoral Position in Lund University (dpt Automatic Control). He received is Ph.D. degree in Robotics from the Blaise Pascal University in 2005. He was graduated from IFMA (French Institute for Advanced Mechanics) in 2002. He received is Ph.D. degree in Robotics from the Blaise Pascal University in 2005.



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

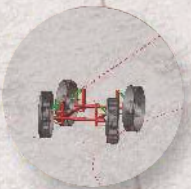


Mobile robot control

Generic algorithm for high accurate trajectory control in different conditions

*Roland Lenain
Benoit Thuilot
Philippe Martinet*

IROS 2011 - San Francisco - September 2011



Introduction

Growing interest for autonomous vehicles for various application

- Transportation system
- Exploration
- Military application
- Work in different kind of environment

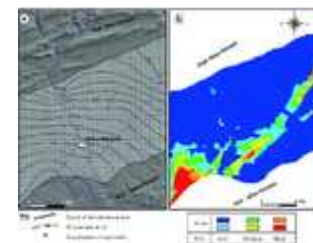
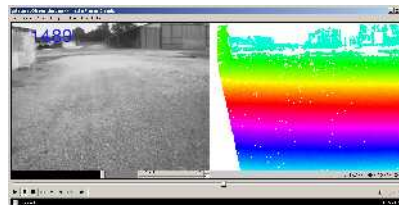


- Civil engineering
- Forestry
- Agriculture



Different contexts requiring different behaviors and configurations

- Sensor – perception system
- Navigation strategy
- **Control law**



Introduction

A variety of environment impacting robot dynamics

- Environments and terrain properties

- Structured/unstructured environment
- Nature of soil - grip conditions
- Terrain geometry (regularity, slope, ...)



- Robot design and capabilities

- Velocity
- General robot design (0/2/4WS, 2/4WD, ...)
- Design parameters (length, mass, inertia, CoG position, ...)
- Actuator properties and capabilities



- Phenomena to be accounted?

- Sliding influence
- Inertial effects
- Dynamic stability
- Robot controllability
- Traversability and obstacles



Introduction

Robot dynamics to be accounted with respect to the context

Terrain complexity

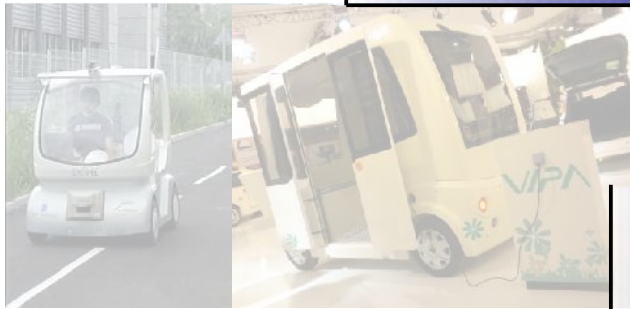


Velocity

Introduction

Robot dynamics to be accounted with respect to the context

Terrain complexity



Dynamic stability

Sliding influence

Robot controllability

Traversability and obstacles

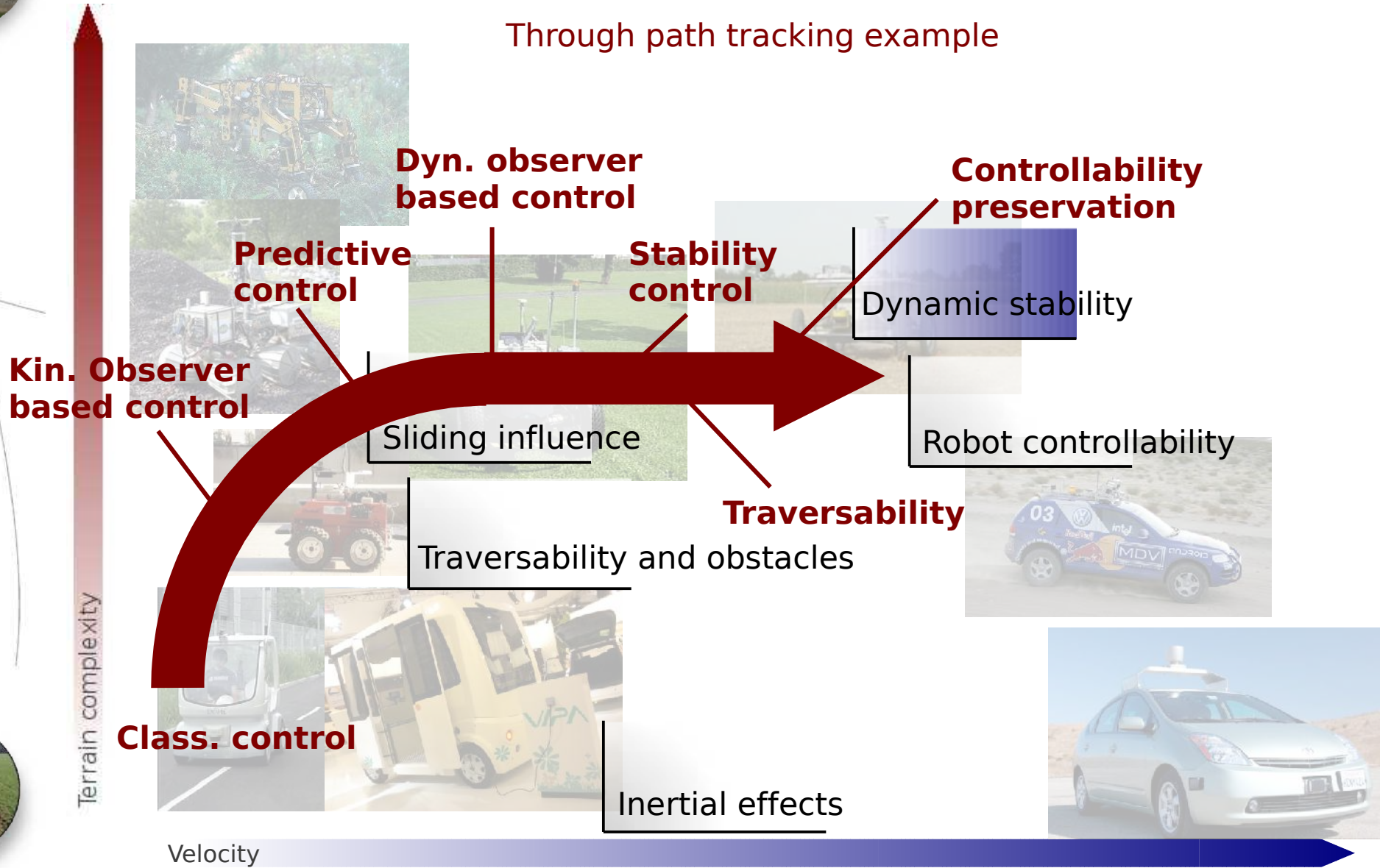
Inertial effects

Velocity

Introduction

Robot dynamics to be accounted with respect to the context

Through path tracking example



▶ Classical path tracking control



Classical kinematic model

- Based on rolling without sliding assumption
- Can be turned into linear form

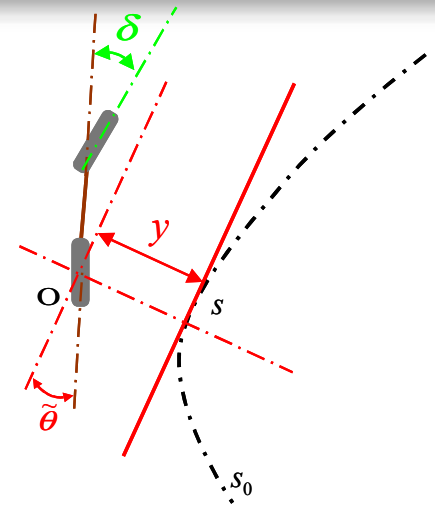
$$\begin{cases} \dot{s} = v \frac{\cos(\tilde{\theta} + \delta_R)}{1 - c(s)y} \\ \dot{y} = v \sin(\tilde{\theta} + \delta_R) \\ \dot{\tilde{\theta}} = v \left[\cos(\delta_R) \frac{\tan(\delta_F) - \tan(\delta_R)}{L} - \frac{c(s) \cos(\tilde{\theta} + \delta_R)}{1 - c(s)y} \right] \end{cases}$$

Control law design

- Built in linear state space
- Objective is to ensure convergence of lateral and angular error to 0
- Reverse transformation leads to non linear control expression

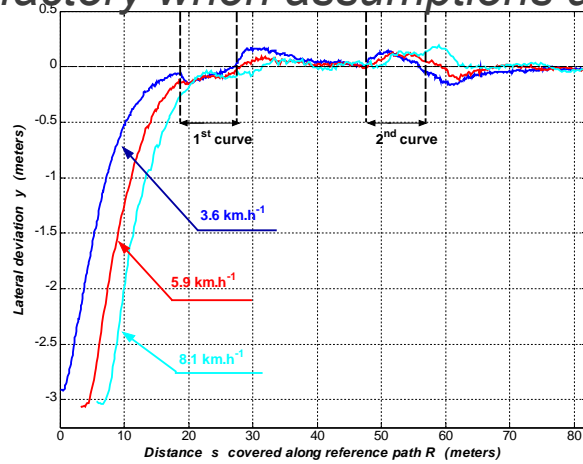
$$\delta_{RSG}(y, \tilde{\theta}) = \arctan\left(L \left[\frac{\cos^3 \tilde{\theta}}{(1 - c(s)y)^2} \left(\frac{dc(s)}{ds} y \tan \tilde{\theta} - K_d (1 - c(s)y) \tan \tilde{\theta} - K_p y + c(s)(1 - c(s)y) \tan^2 \tilde{\theta} \right) + \frac{c(s) \cos \tilde{\theta}}{1 - c(s)y} \right]\right)$$

- Under the assumption that velocity is a known variable
- Result theoretically independent from velocity
- Control gains tune a settling distance



▶ Classical path tracking control

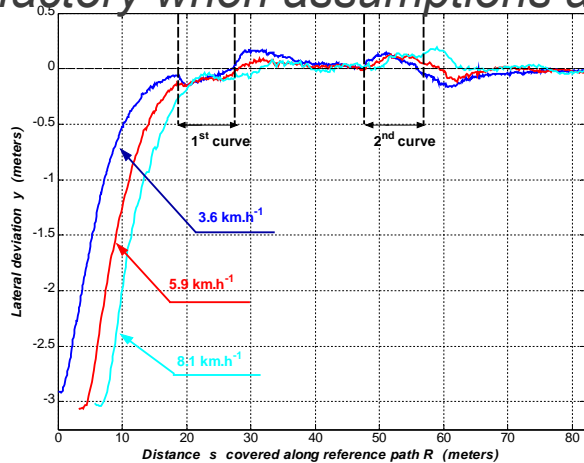
Satisfactory when assumptions are valid



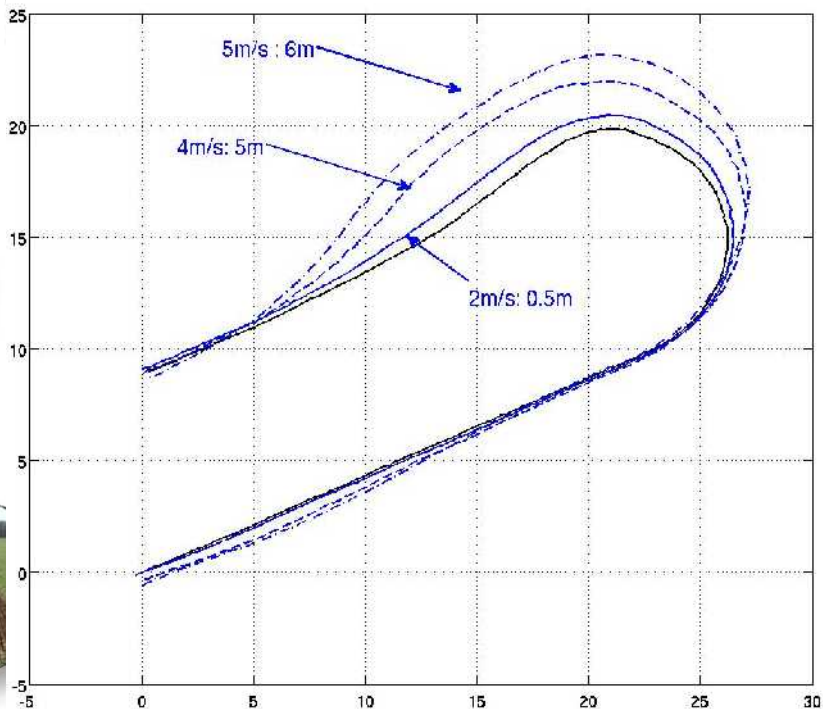
Classical path tracking control



Satisfactory when assumptions are valid



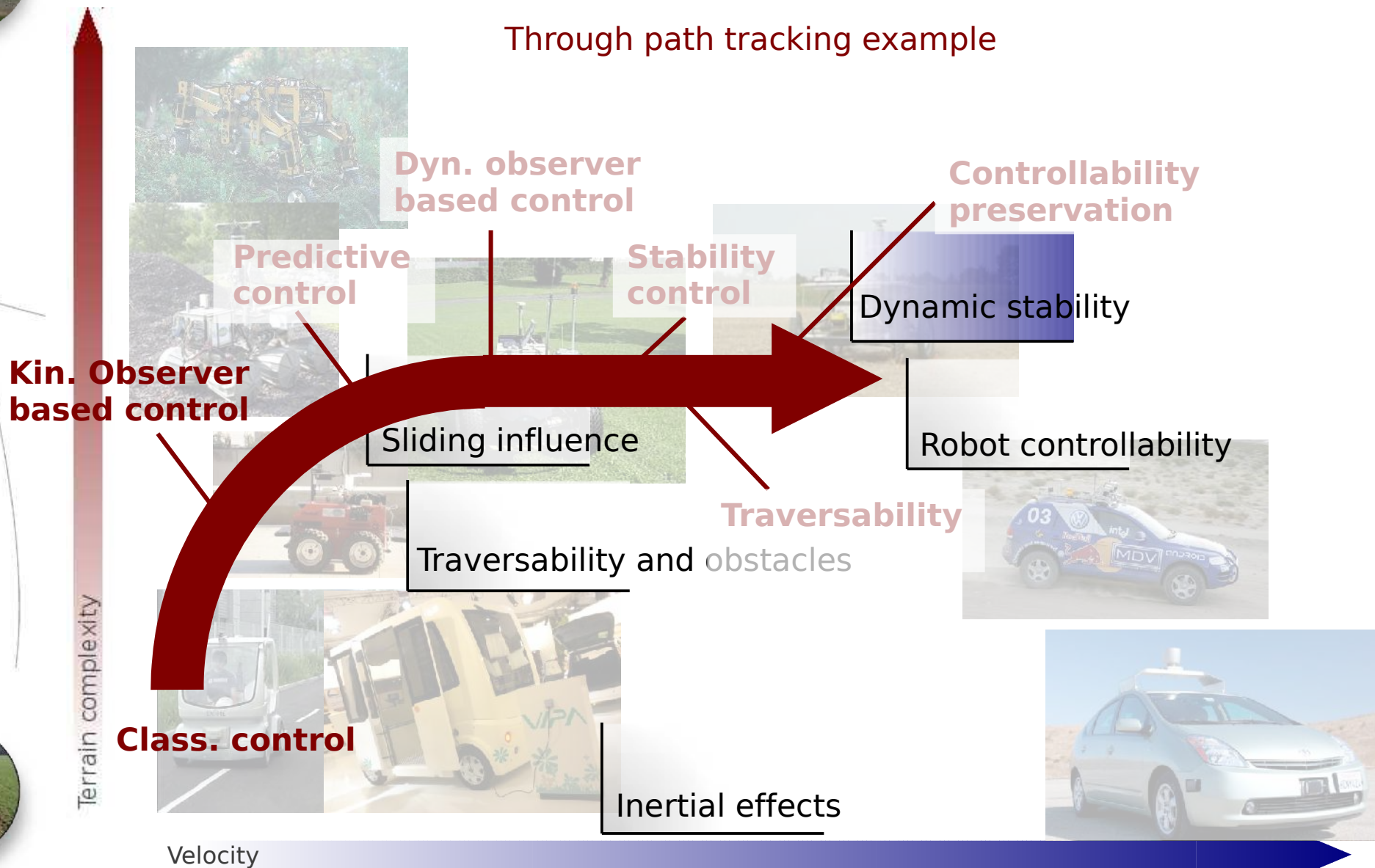
Unsuitable when running on natural ground [depends on speed!]



▶ Sliding integration at low speed

Robot dynamics to be accounted with respect to the context

Through path tracking example



Sliding integration at low speed

Extended kinematical approach

- Introduction of sideslip angles in kinematics [Iros04]

β^F Front sideslip angles

β^R Rear sideslip angles

→ Tire-Based Kinematic Model (TBKM)

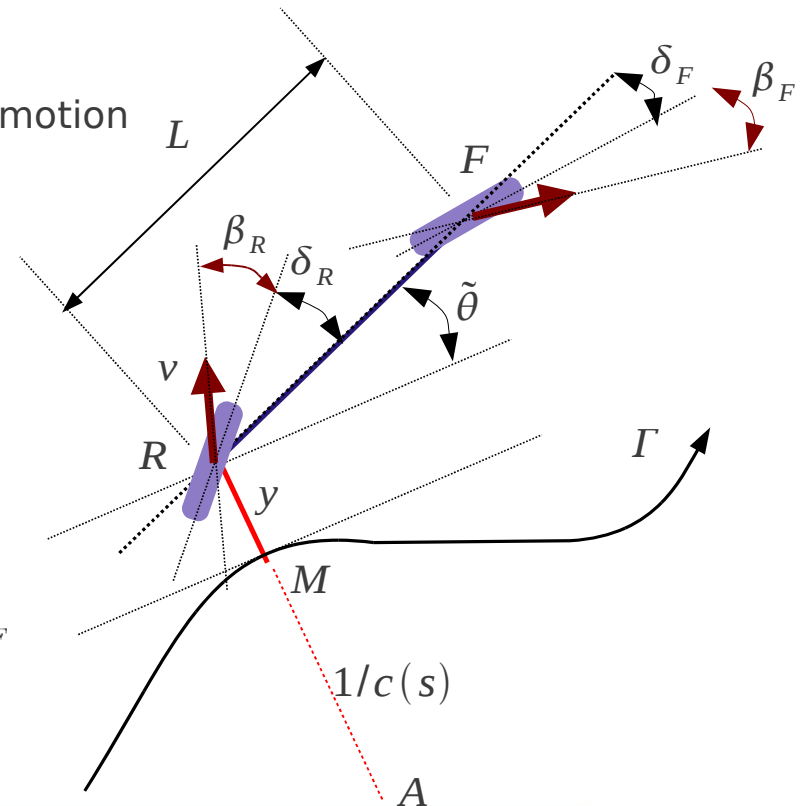
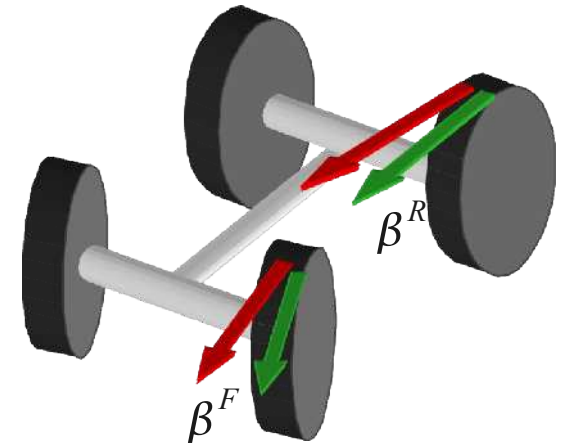
- Grip conditions viewed as part of model
- Representative of sliding effect on vehicle motion
- Preserve the “non-holonomic” notion

$$\begin{cases} \dot{s} = V_r \frac{\cos(\tilde{\theta} + \delta_R - \beta_R)}{1 - c(s)y} \\ \dot{y} = V_r \sin(\tilde{\theta} + \delta_R - \beta_R) \\ \dot{\tilde{\theta}} = V_r [\cos(\delta_R - \beta_R)\lambda_1 - \lambda_2] \end{cases}$$

- Provided sideslip angles control is similar

$$\delta_F = \arctan \left\{ \tan(\delta_R - \beta_R) + \frac{L}{\cos(\delta_R - \beta_R)} \left(\frac{c \cos \tilde{\theta}_2}{\alpha} + \frac{A \cos^3 \tilde{\theta}_2}{\alpha^2} \right) \right\} + \beta_F$$

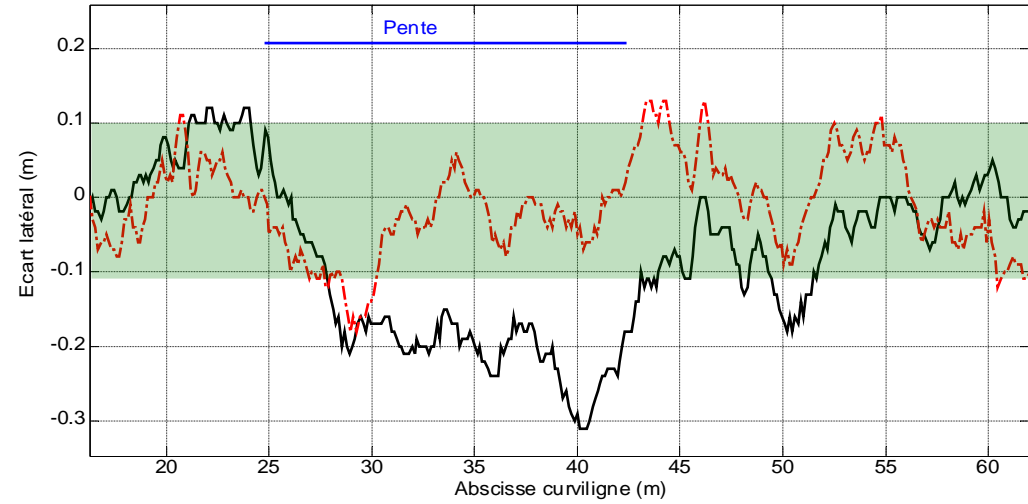
$$\begin{cases} \tilde{\theta}_2 = \tilde{\theta} + \delta_R - \beta_R \\ \alpha = 1 - cy \\ A = -\frac{K_d^2 y}{4} - K_d \alpha \tan \tilde{\theta}_2 + c \alpha \tan^2 \tilde{\theta}_2 \end{cases}$$



▶ Sliding integration at low speed

Validation on straight-line following on slope

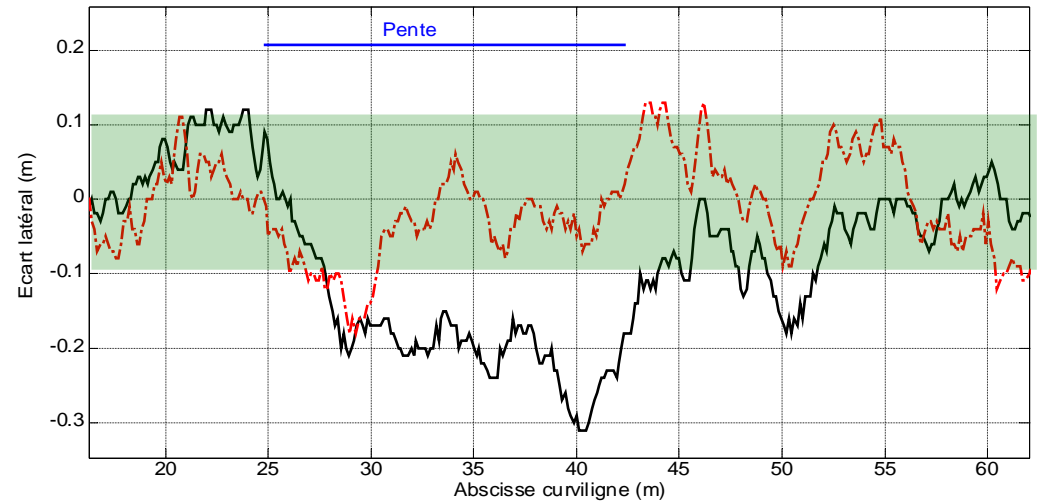
- Validation on tractor (5km/H)



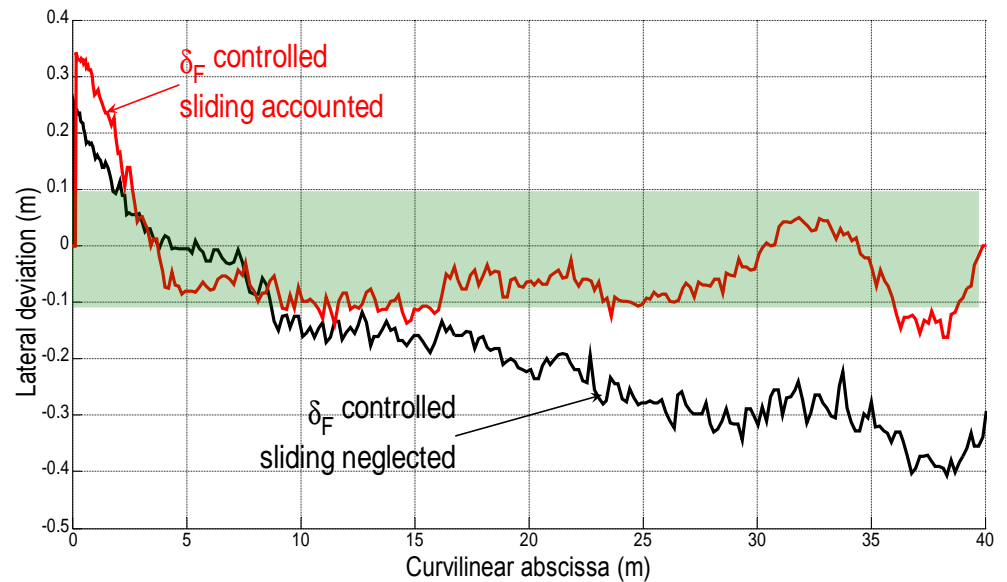
▶ Sliding integration at low speed

Validation on straight-line following on slope

- Validation on tractor (5km/H)



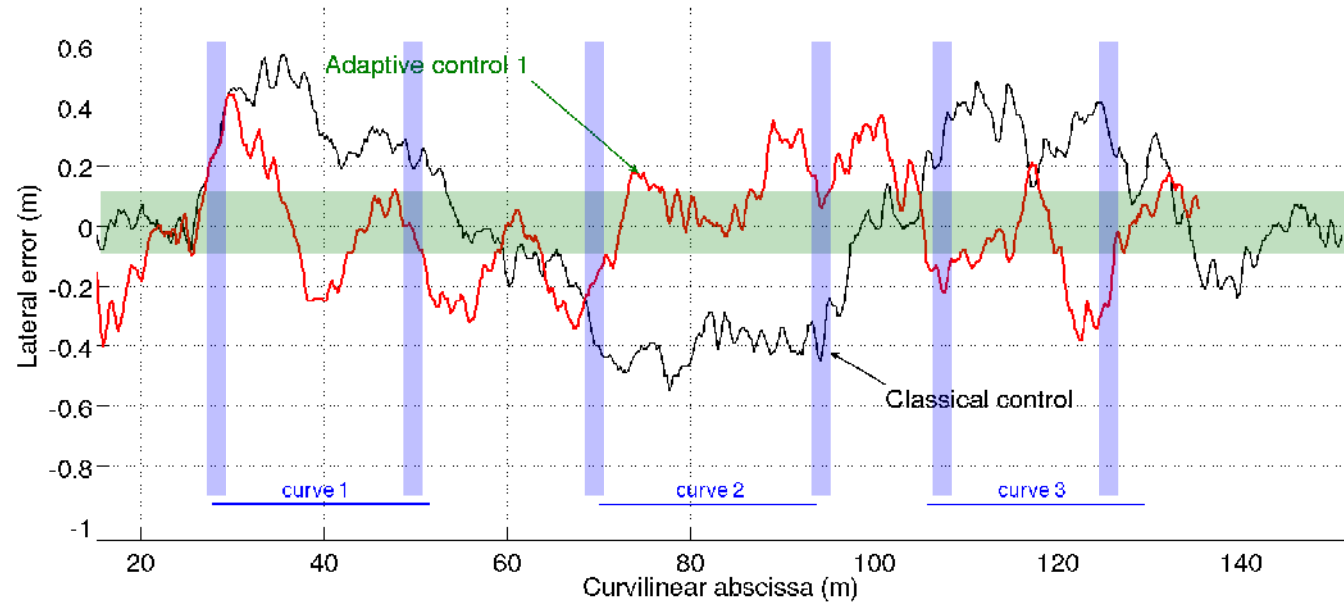
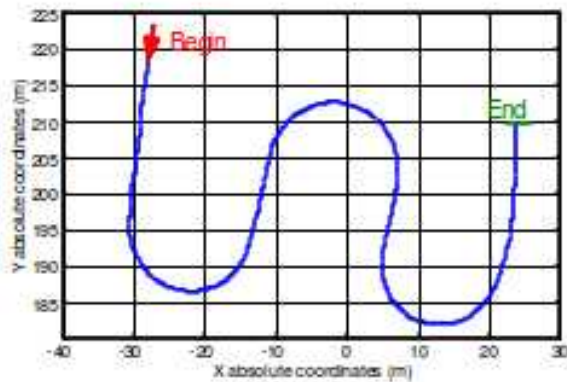
- Validation on Arroco (5km/H)



▶ Sliding integration at low speed



- Tracking at 5km/H



- Satisfactory in steady state (constant curve)

- Constant slope
- Constant curve

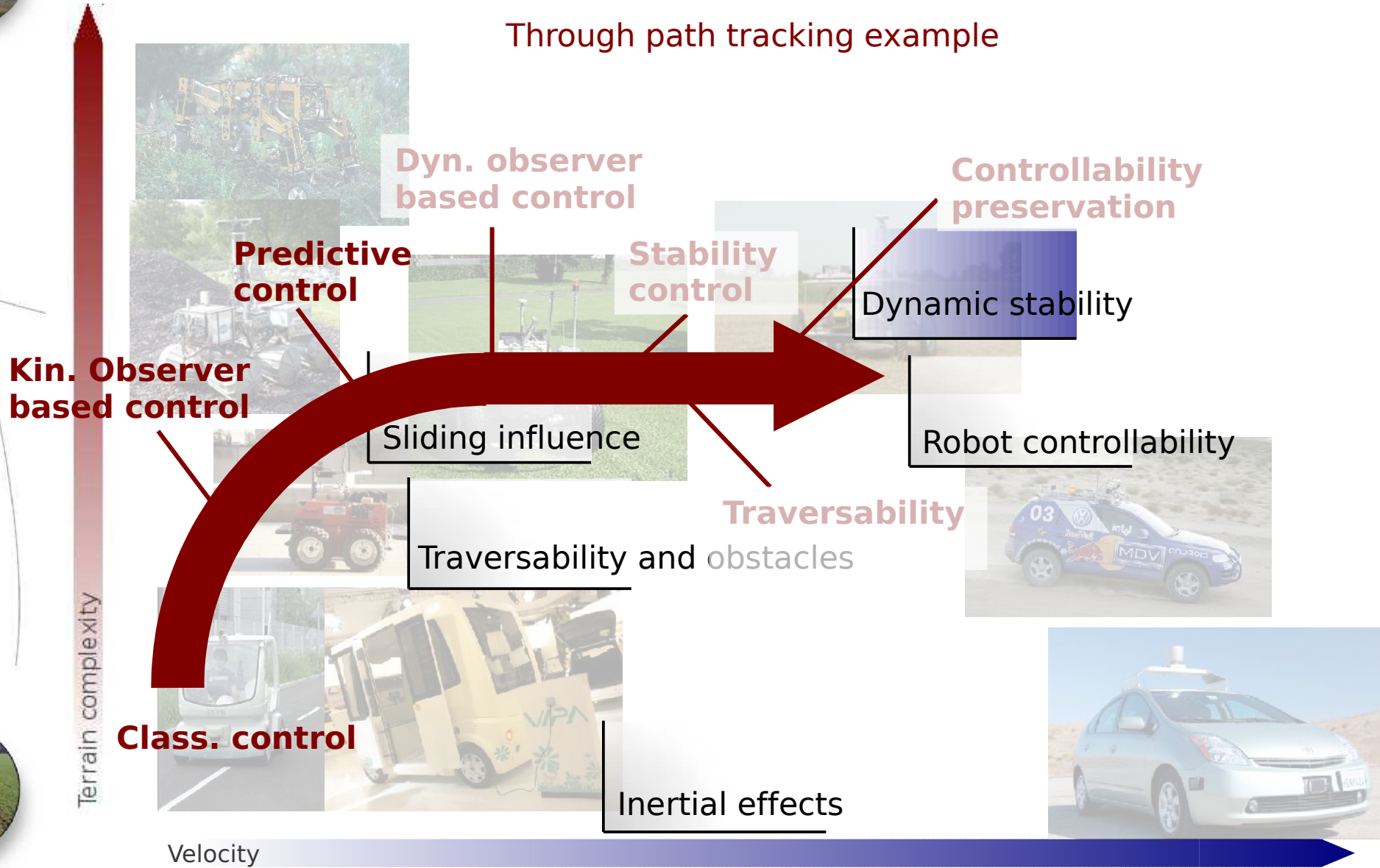
- Non negligible overshoots during transition

- Low level delay
- Inertial effects

Anticipation of delays (actuator/inertial)

Robot dynamics to be accounted with respect to the context

Through path tracking example



Anticipation of delays (actuator/inertial)

Anticipation of transition in ref path curvature [AutonRobots2006, Icara2008]

- Decomposition of control law expression

→ Terms attached to deviations sliding compensation

$$\delta_{Deviation} = \arctan\left(\frac{v}{1+uv+u^2}\right) \quad v = \frac{L}{\cos\beta^R} A \frac{\cos^3\theta_2}{\alpha^2} + \tan\beta^R$$

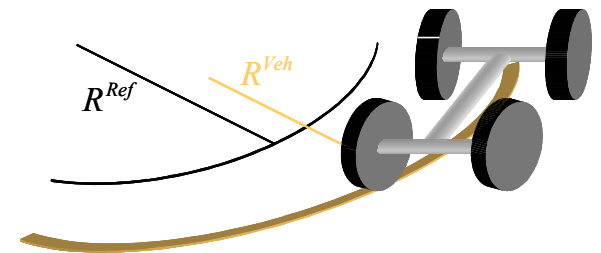
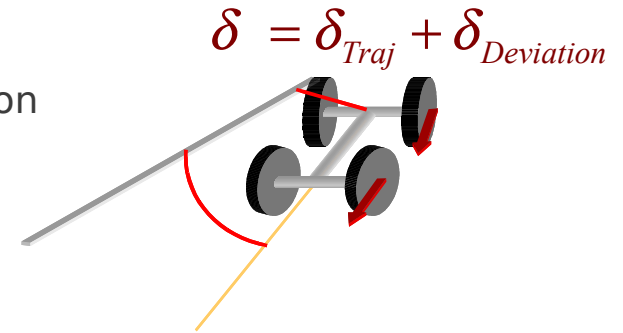
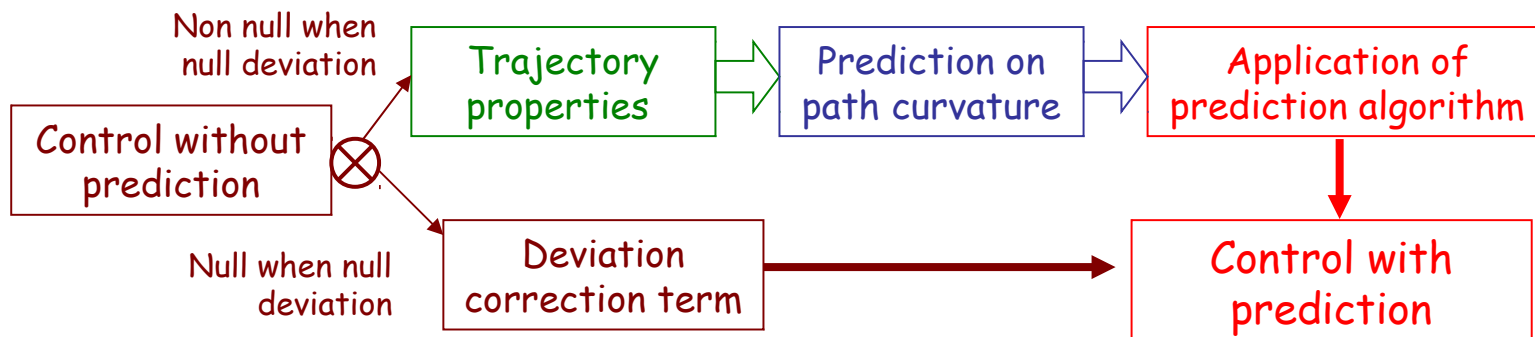
→ **Has to be reactive (no prediction available)**

→ Terms attached path curvature following

$$\delta_{Traj} = \arctan(u) \quad u = \frac{L}{\cos\beta^R} c(s) \frac{\cos\theta_2}{\alpha}$$

→ **Can be predicted (knowledge of the reference path)**

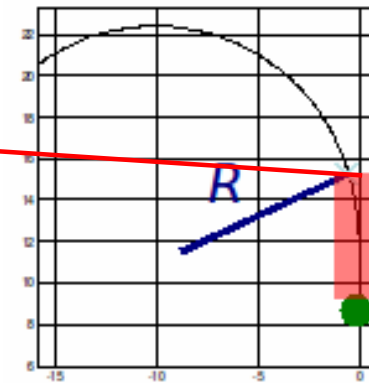
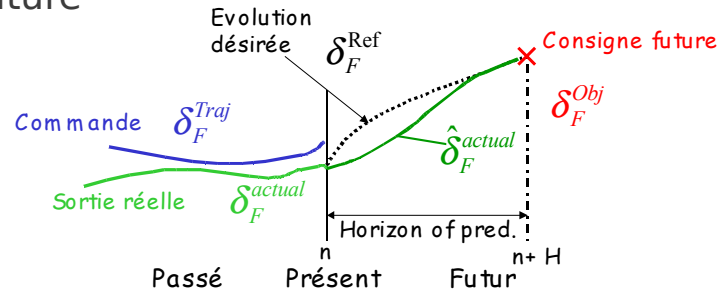
- Predictive and adaptive control algorithm



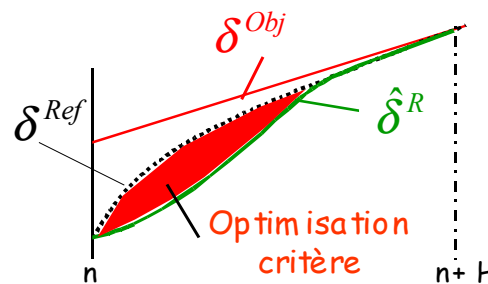
Anticipation of delays (actuator/inertial)

- Predictive term calculation:

- Extraction of future set point attached to path curvature



- Definition of desire shape to reached the set point
- Computation of minimizing control sequence
- Based on low level model



- Adaptive and predictive control law

$$\delta = \delta_{Traj}^{Pred} + \delta_{Deviation}$$

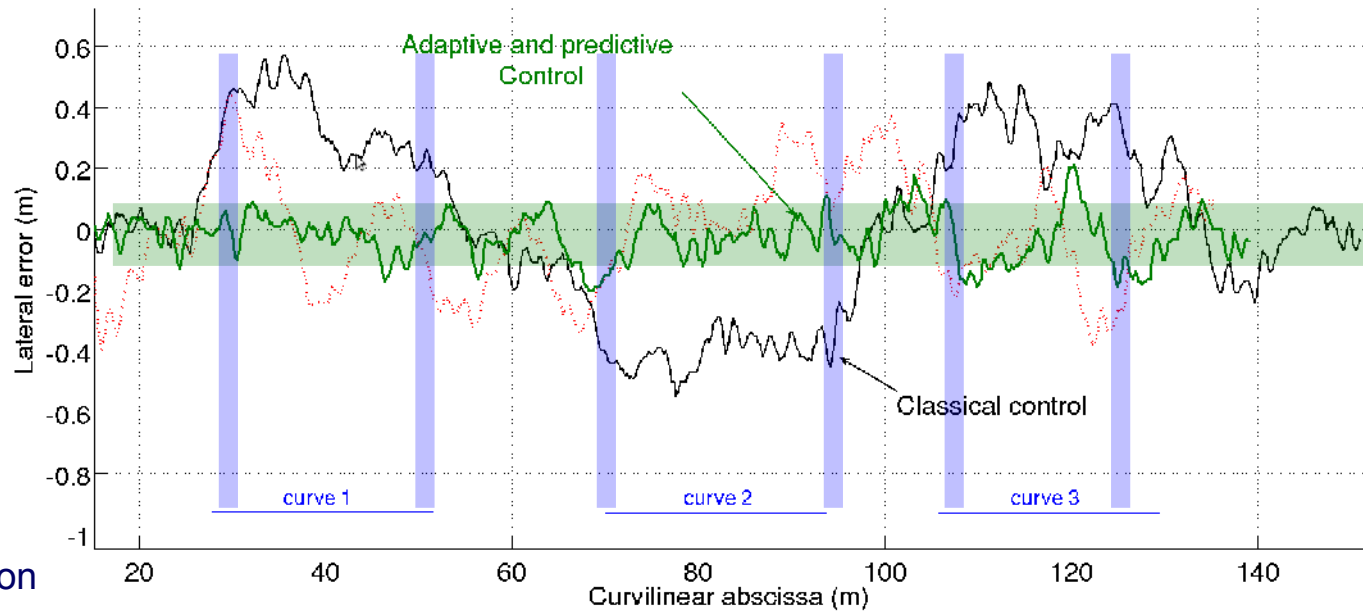
Anticipation of delays (actuator/inertial)



- Tracking at 5km/H



- Satisfactory whatever the variation

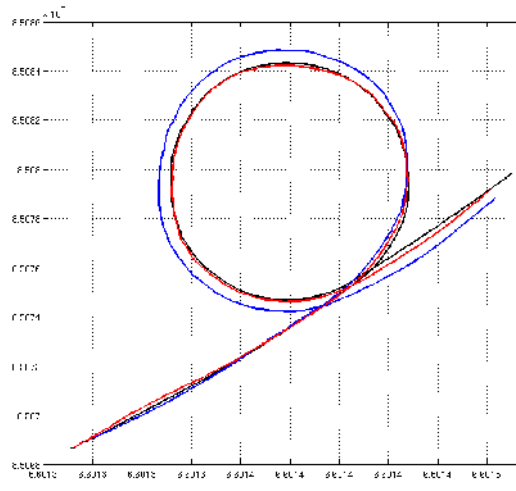


Anticipation of delays (actuator/inertial)

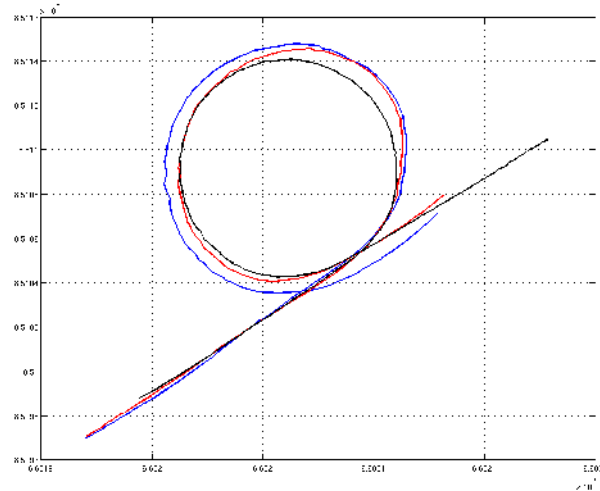


Reactivity of sliding estimation based on extended kinematic is limited

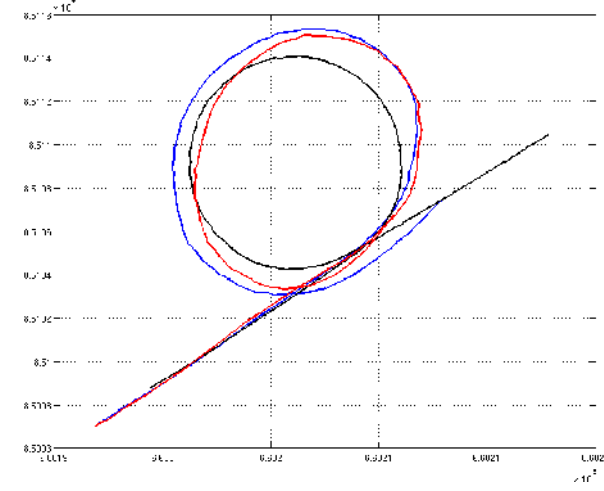
• 2m/s : 0.1m



• 3m/s : 0.5m



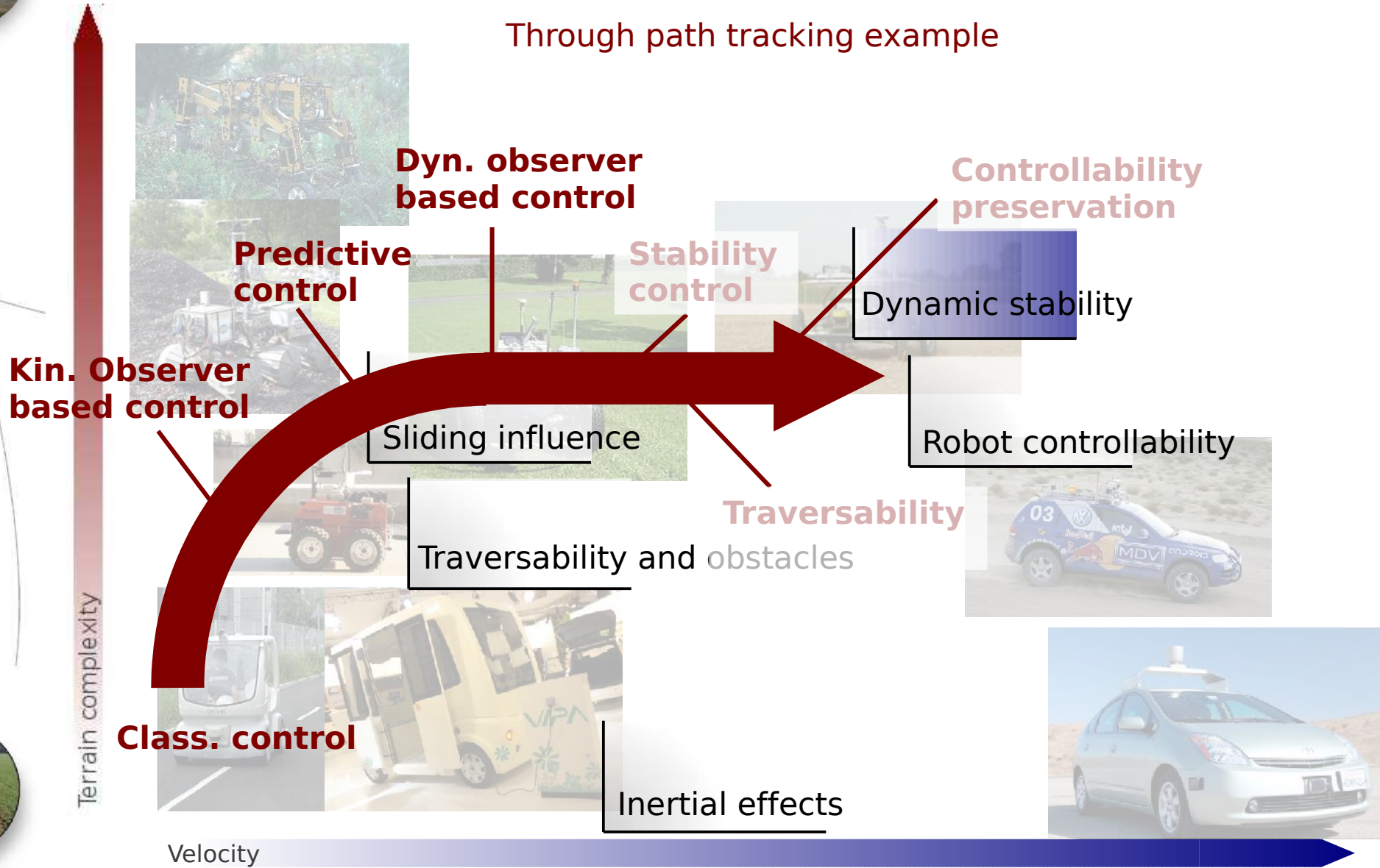
• 4m/s : 1m



► Dynamic observer based motion control

Robot dynamics to be accounted with respect to the context

Through path tracking example

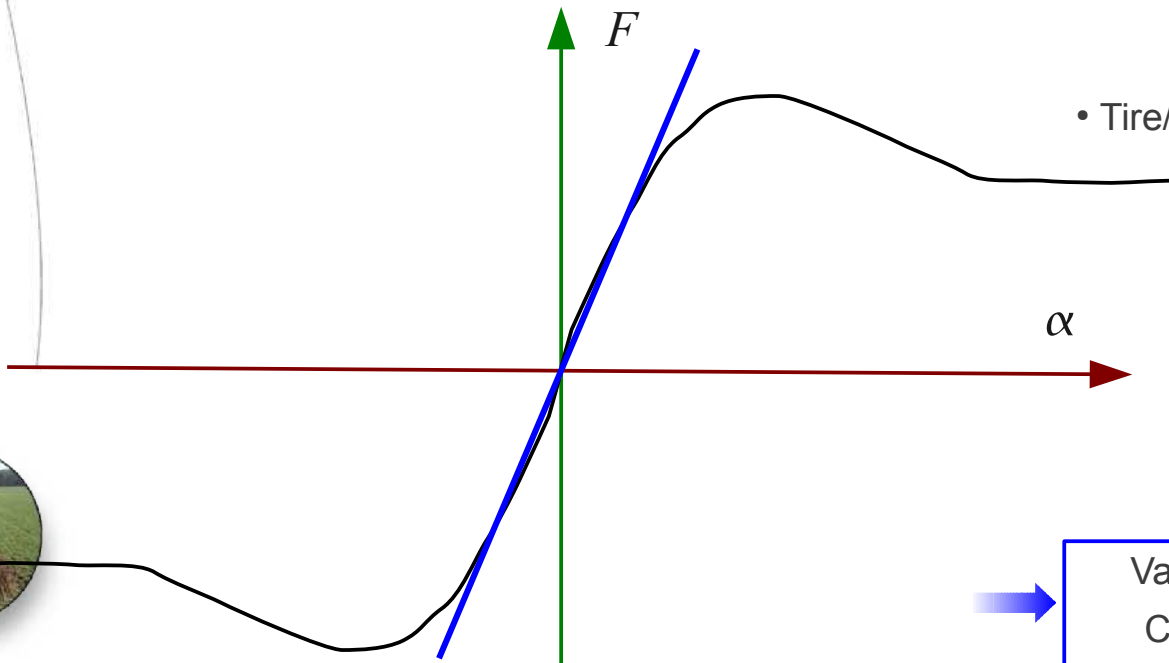
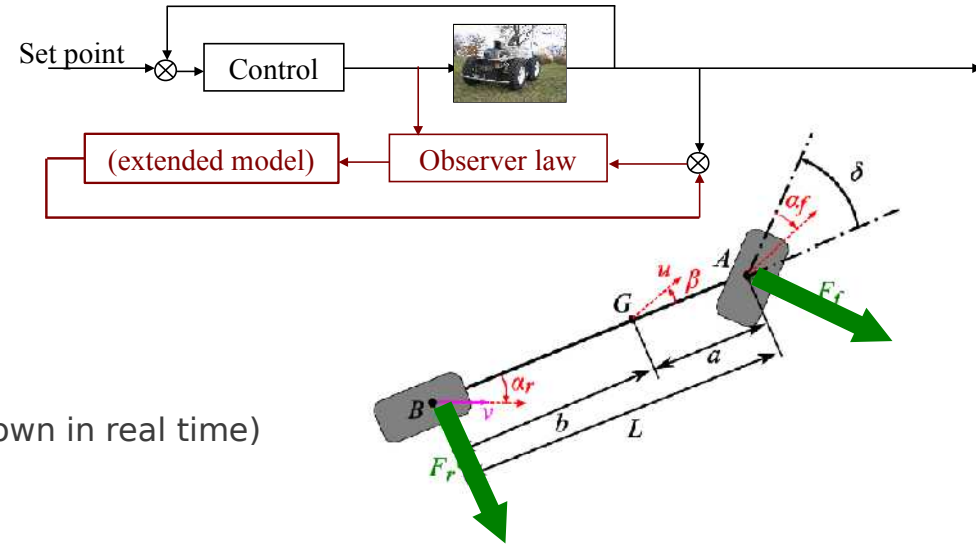


Dynamic observer based motion control



New observer design

- Extended kinematic observer
 - Accurate
 - Slow reactive (dynamic neglected)
- Dynamic model based observer
 - More reactive
 - Not accurate (grip conditions to be known in real time)



• Tire/soil model depends on 14 param.

- Ground properties
- Vertical load
- Tire properties

Not Tractable for control purpose

Variable Linear Description
Cornering stiffnesses

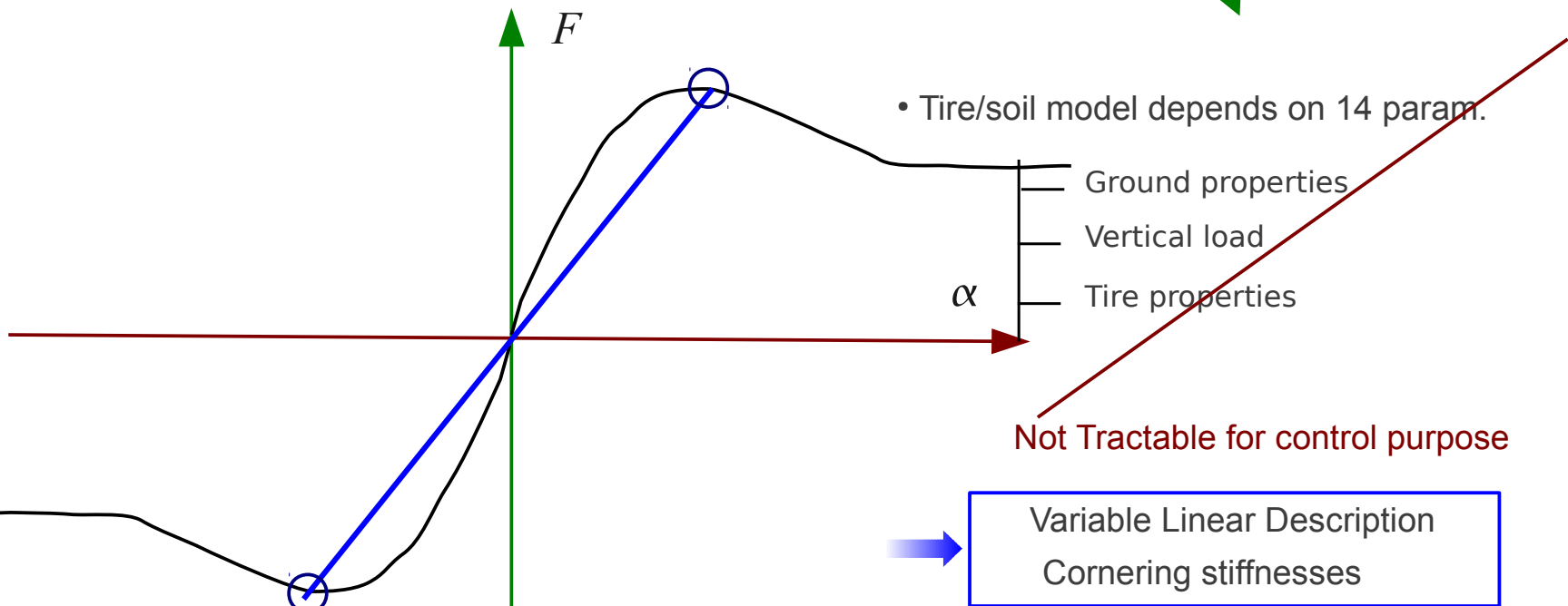
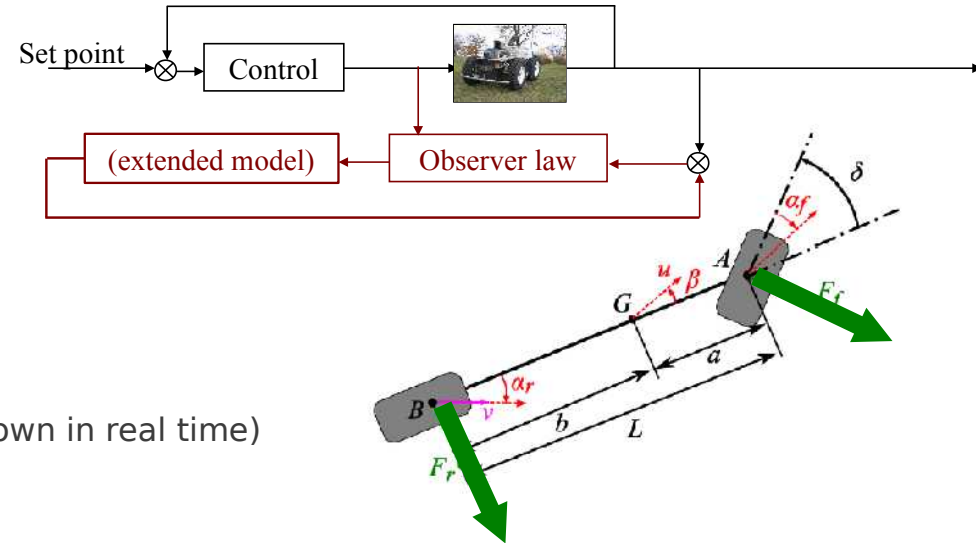
Dynamic observer based motion control



Principle

- Extended kinematic observer
 - Accurate
 - Slow reactive (dynamic neglected)

- Dynamic model based observer
 - More reactive
 - Not accurate (grip conditions to be known in real time)

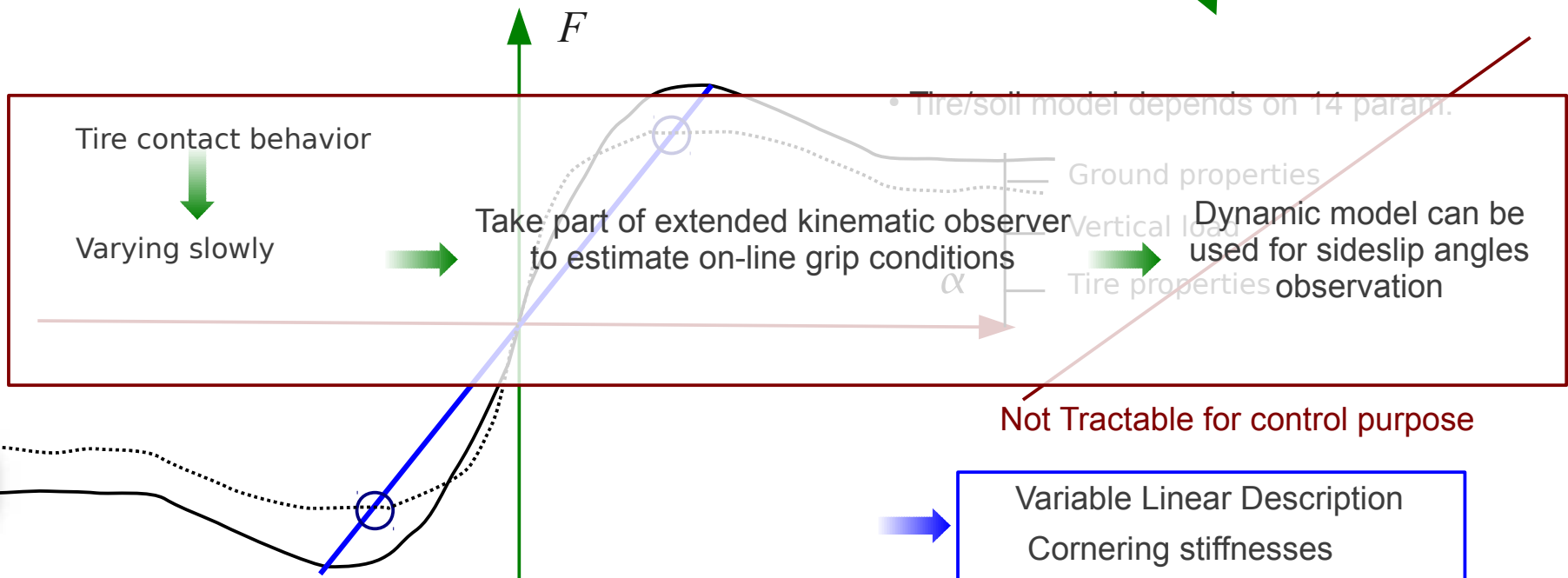
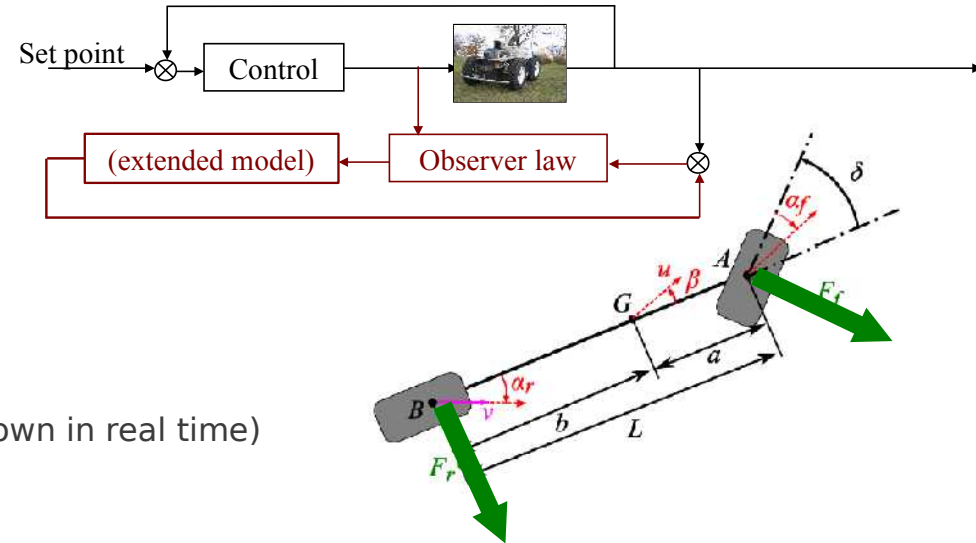


Dynamic observer based motion control



Principle

- Extended kinematic observer
 - Accurate
 - Slow reactive (dynamic neglected)
- Dynamic model based observer
 - More reactive
 - Not accurate (grip conditions to be known in real time)



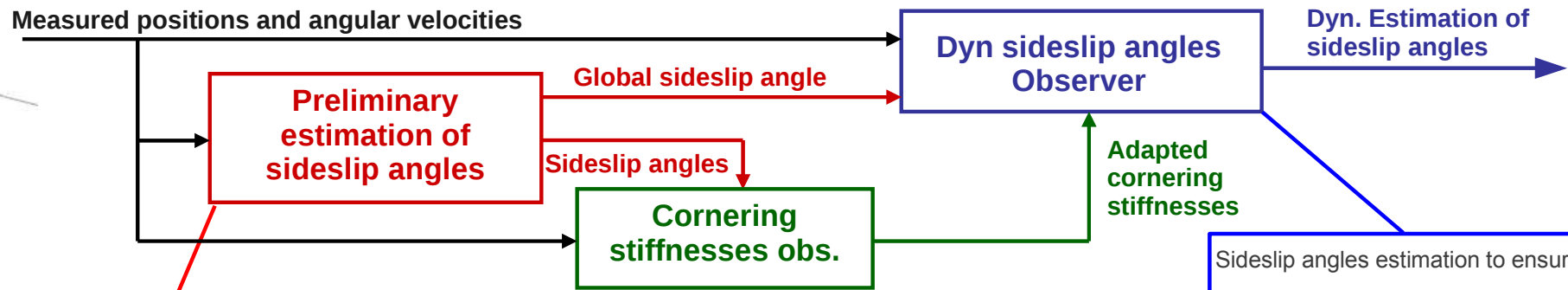
Dynamic observer based motion control



Proposed algorithm: Mixed-model observers

- Decomposed in several steps

- S1: Preliminary sideslip angles observer.
- S2: Cornering stiffnesses adaptation
- S3: Sideslip angles observation based on dynamic model



Sideslip angles calculation to ensure:

$$(\hat{y}, \hat{\theta}) \rightarrow (\bar{y}, \bar{\theta})$$

outputs: $\left. \begin{matrix} \beta^F \\ \beta^R \end{matrix} \right\} \bar{\beta}$

Cornering stiffnesses estim.

To ensure:

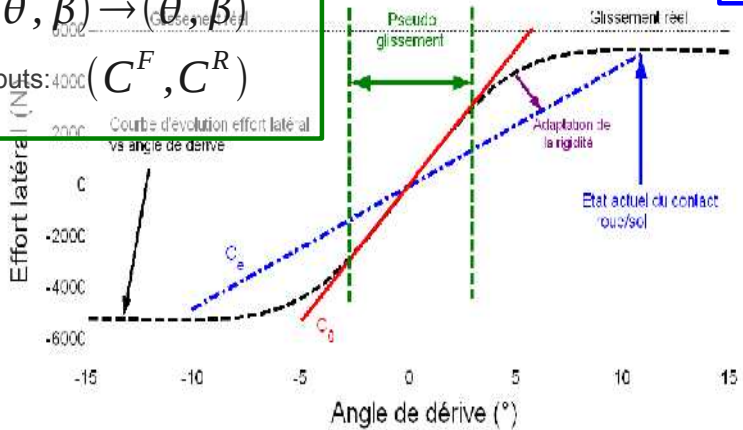
$$(\hat{\theta}, \hat{\beta}) \rightarrow (\bar{\theta}, \bar{\beta})$$

outputs: (C^F, C^R)

Sideslip angles estimation to ensure:

$$(\hat{\theta}_2, \hat{\beta}_2) \rightarrow (\bar{\theta}, \bar{\beta})$$

outputs: $(\beta_{Dyn}^F, \beta_{Dyn}^R)$



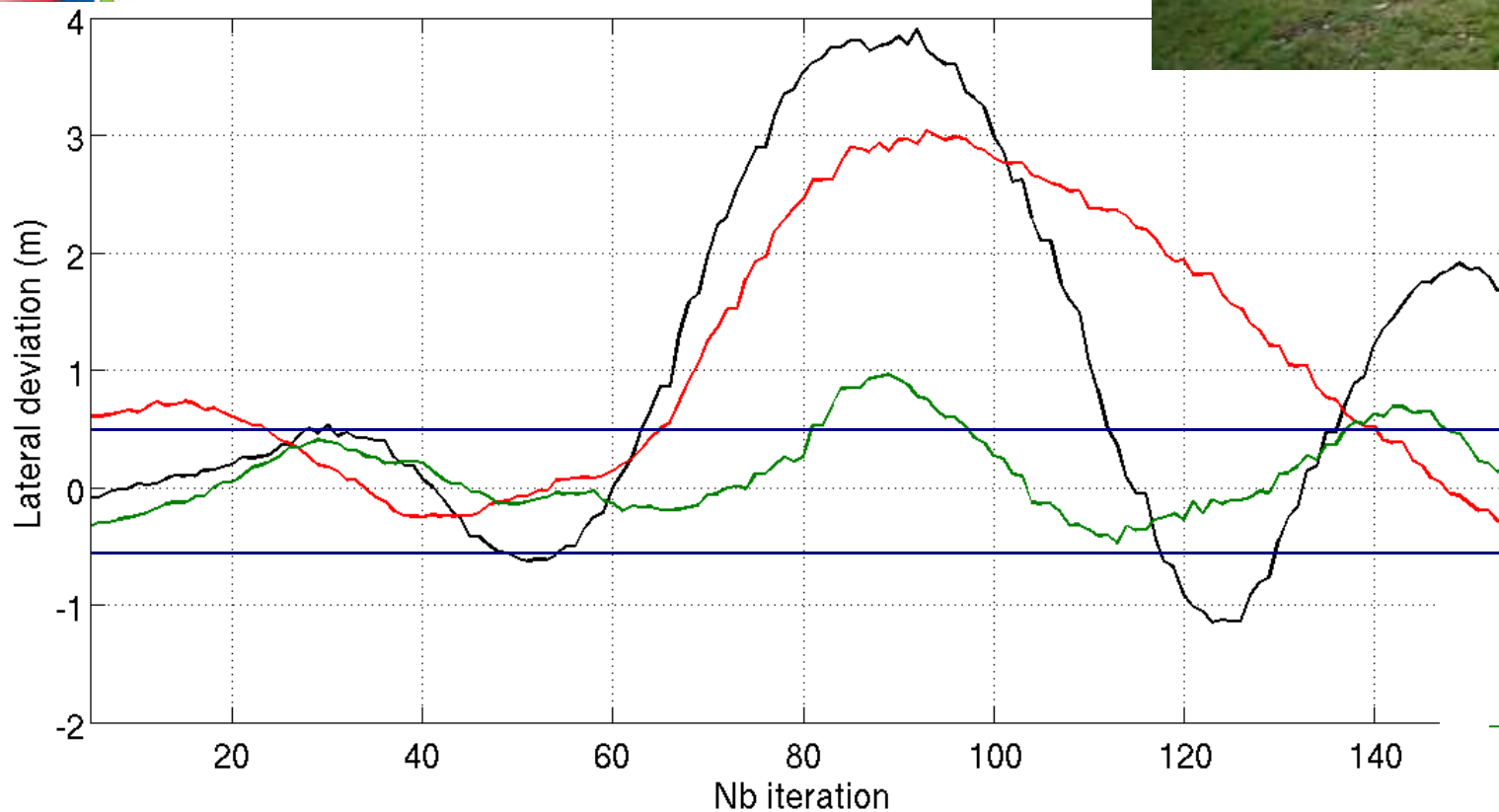
► Dynamic observer based motion control



Path tracking results

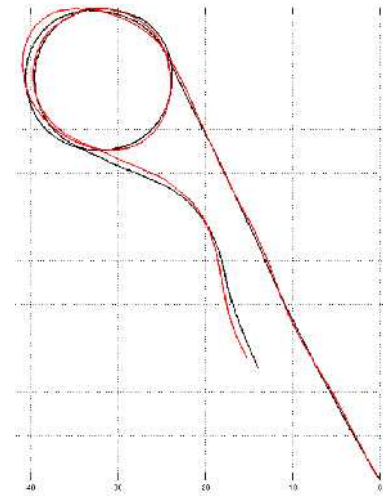
- Experimental background

- grass terrain
- **Velocity of 6m/s**
- Half-turn path tracking

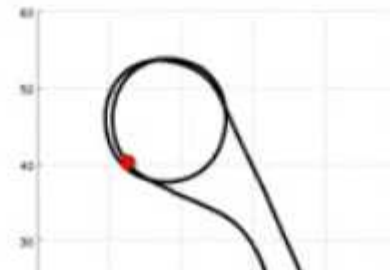


— Control with proposed observer

► Dynamic observer based motion control



Velocity of 6m/s



Max dev: 0.9m
(max st angle)



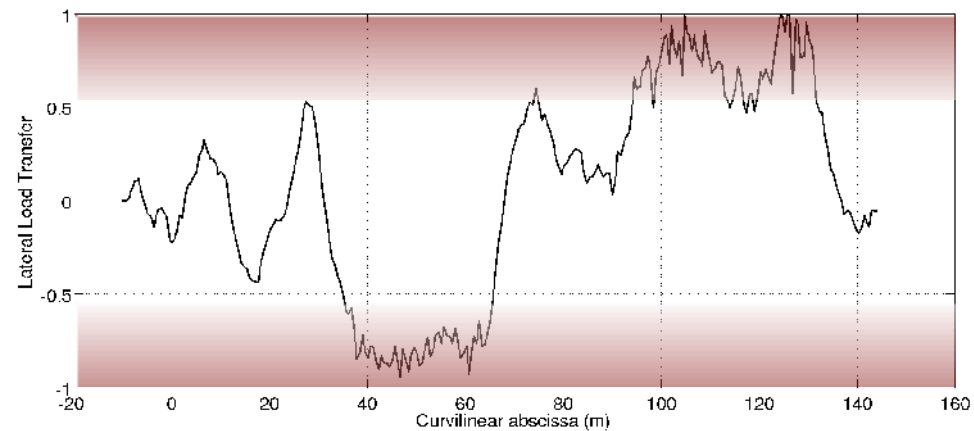
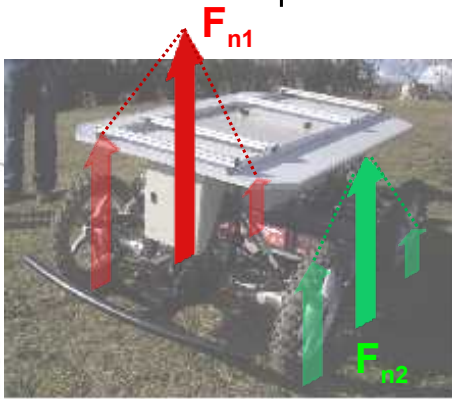
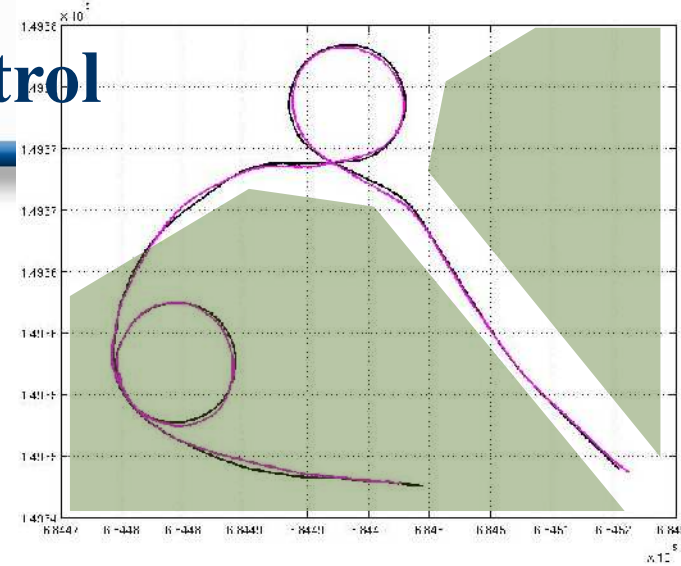
Dynamic observer based motion control

Good accuracy, but stability is at the limit

- Motion control without considering stability/achievability

— Lateral load transfer considered and computed

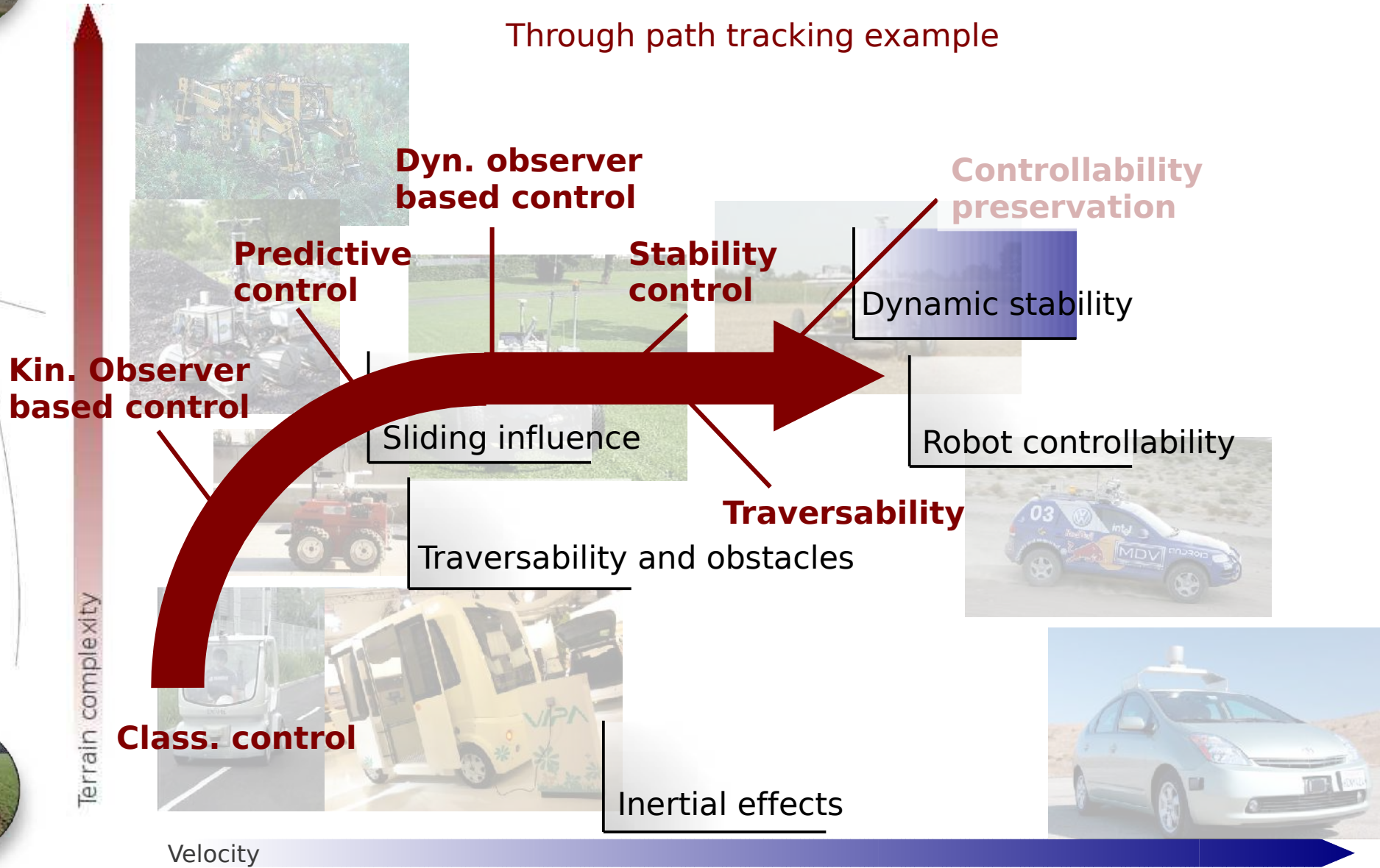
— Reaches high values (2 wheels lift-off)



► Dynamic observer based motion control

Robot dynamics to be accounted with respect to the context

Through path tracking example

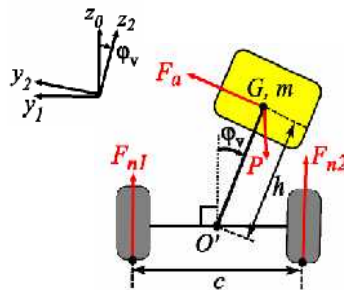
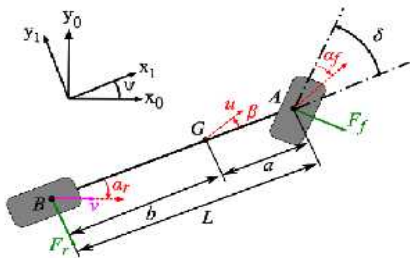
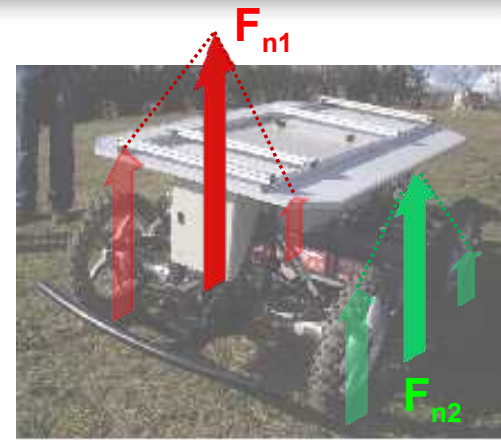


Dynamic stability preservation

Moderation of robot speed is investigated

- Predictive control is applied to compute maximal velocity

- An limit for Lateral Load Transfer is chosen
- Velocity leading to this LLT threshold is computed

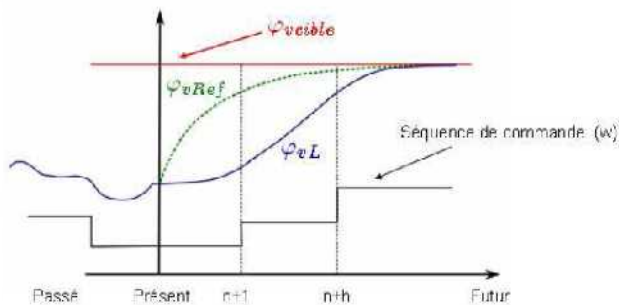


$$\phi_v = \frac{1}{hc \cos(\beta)} [h\phi_v^2 \sin(\phi_v) + h\psi^2 \sin(\phi_v) + u\psi \cos(\beta) + \dot{u} \sin(\beta) + u\beta \cos(\beta) - \left(\frac{k_r \phi_v + b_r \dot{\phi}_v}{mh} \right) \cos(\phi_v)] \quad (7)$$

$$F_{n1} + F_{n2} = m [-h\ddot{\phi}_v \sin(\phi_v) - h\dot{\phi}_v^2 \cos(\phi_v) + g \left(\frac{k_r \phi_v + b_r \dot{\phi}_v}{mh} \right) \sin(\phi_v)] \quad (8)$$

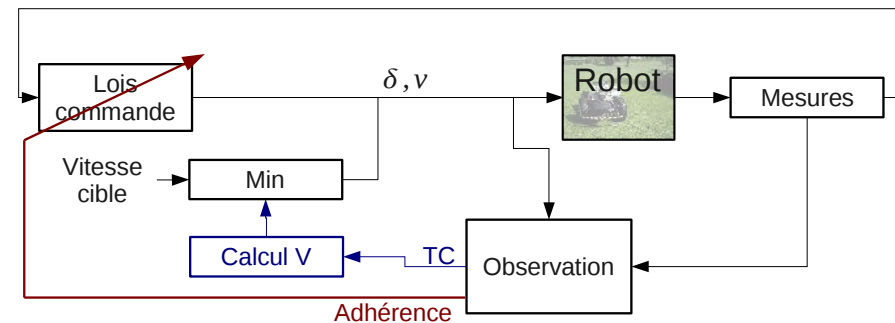
Objective :

$$F_{n1} - F_{n2} = \frac{2}{c} \left[\frac{mh}{I_x \phi_v - I_y} \sin(\phi_v) \right] \psi^2 \cos(\phi_v) \sin(\phi_v) - h \sin(\phi_v) (F_{n1} + F_{n2}) \quad (9)$$



Applied control :

$$v = \min(v_{max}, v_{consigne})$$

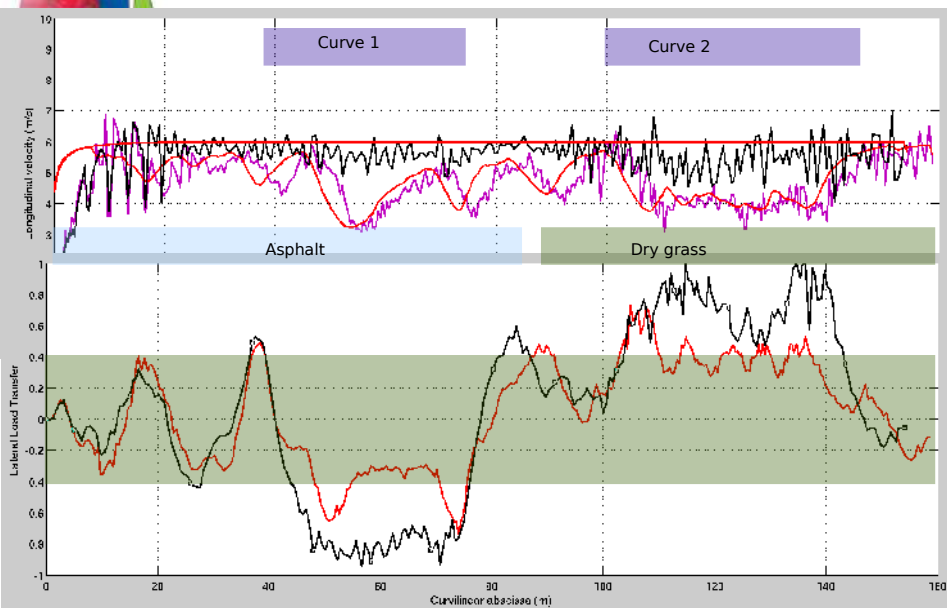
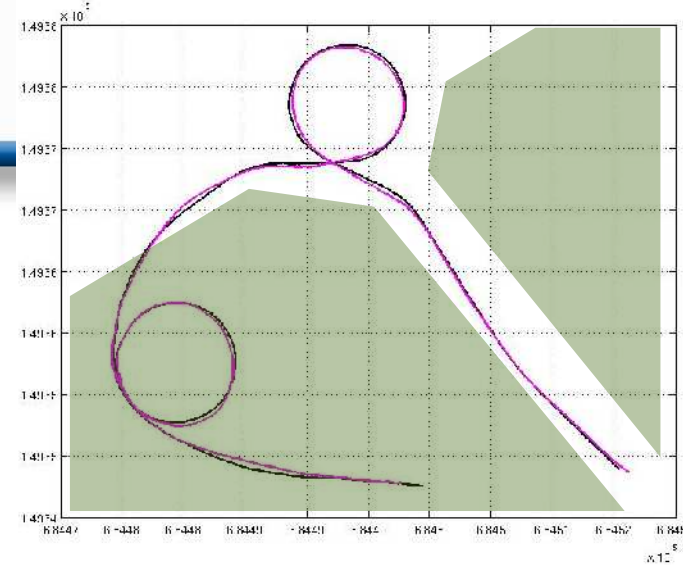
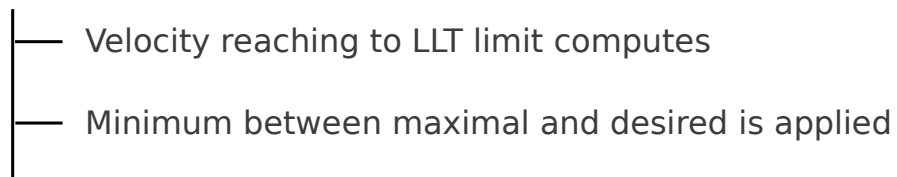


Dynamic stability preservation



Maximal velocity considered to limit LLT

- Robot slow down when a risk is anticipated



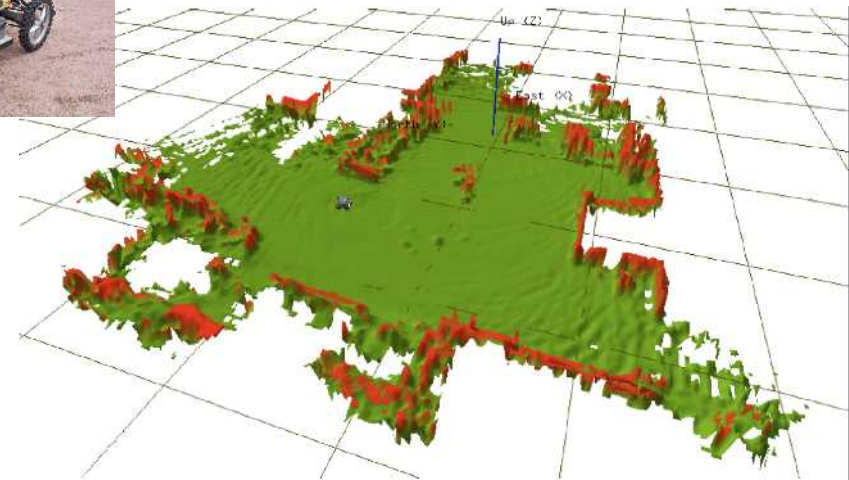
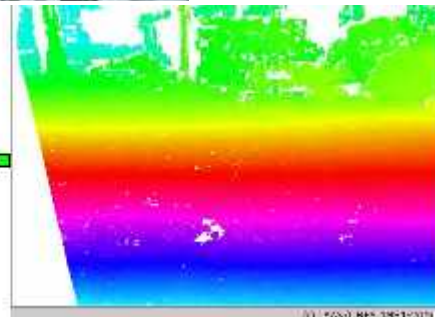
► Extension to traversability – obstacle avoidance

Considering that a MNT reconstruction is available

Stereo-vision

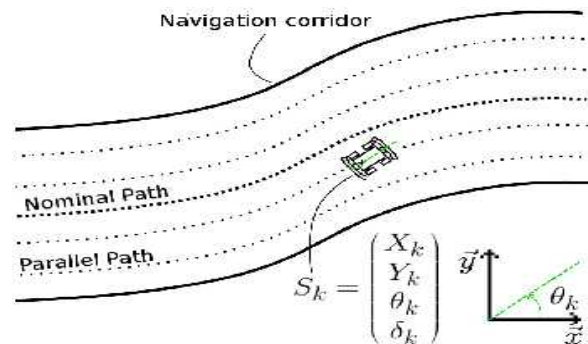


Velodyne

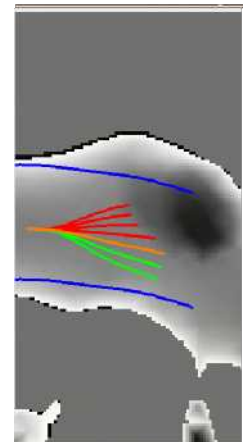


Stability is considering in front of the robot

- Generation of a fixed number of lateral off-set along trajectory



- Definition of trajectories to reach these offset

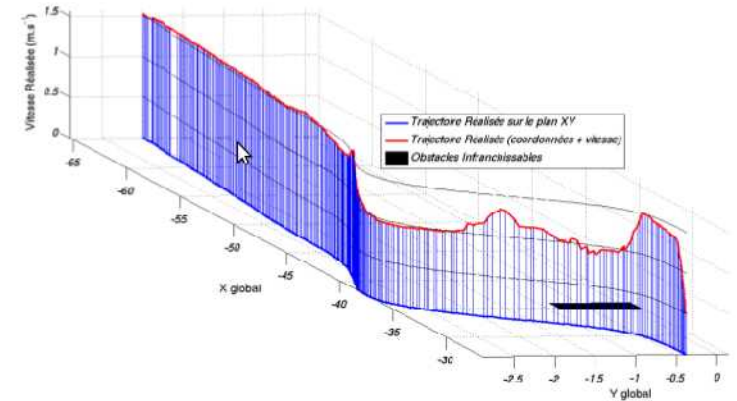


► Extension to traversability – obstacle avoidance



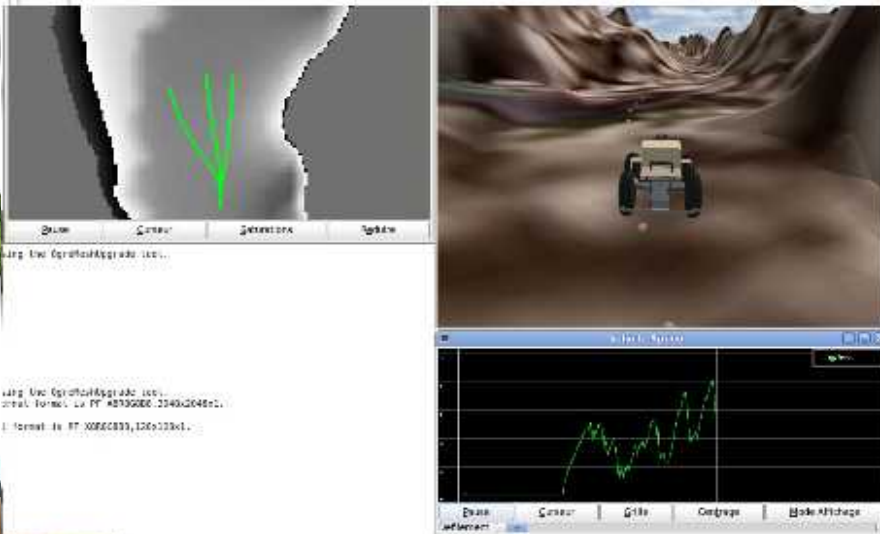
Stability is considering in front of the robot

- Evaluation of maximal speed along generated paths
- Selection of « optimal path »
 - Deviation with respect to the reference path
 - Deviation with respect to the desired speed



Results

- Simulation (MNT at high speed)
- Experimentation (limited speed)

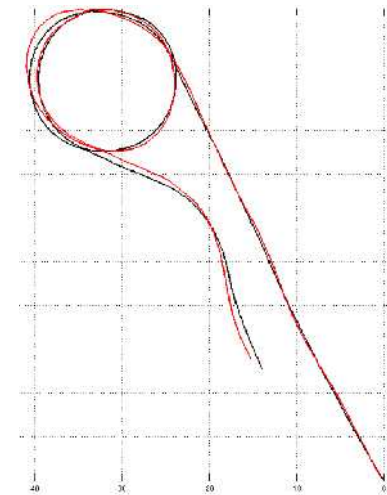


► Dynamic stability preservation



Rollover risk is not the unique sense of stability

- Motion control does not check
 - Reference path achievability (computed or manually recorded at low speed)
 - Robot controllability pending on environment (e.g grip conditions/speed)
 - Transient deviation possibly above desired max error
- Example of controllability loss



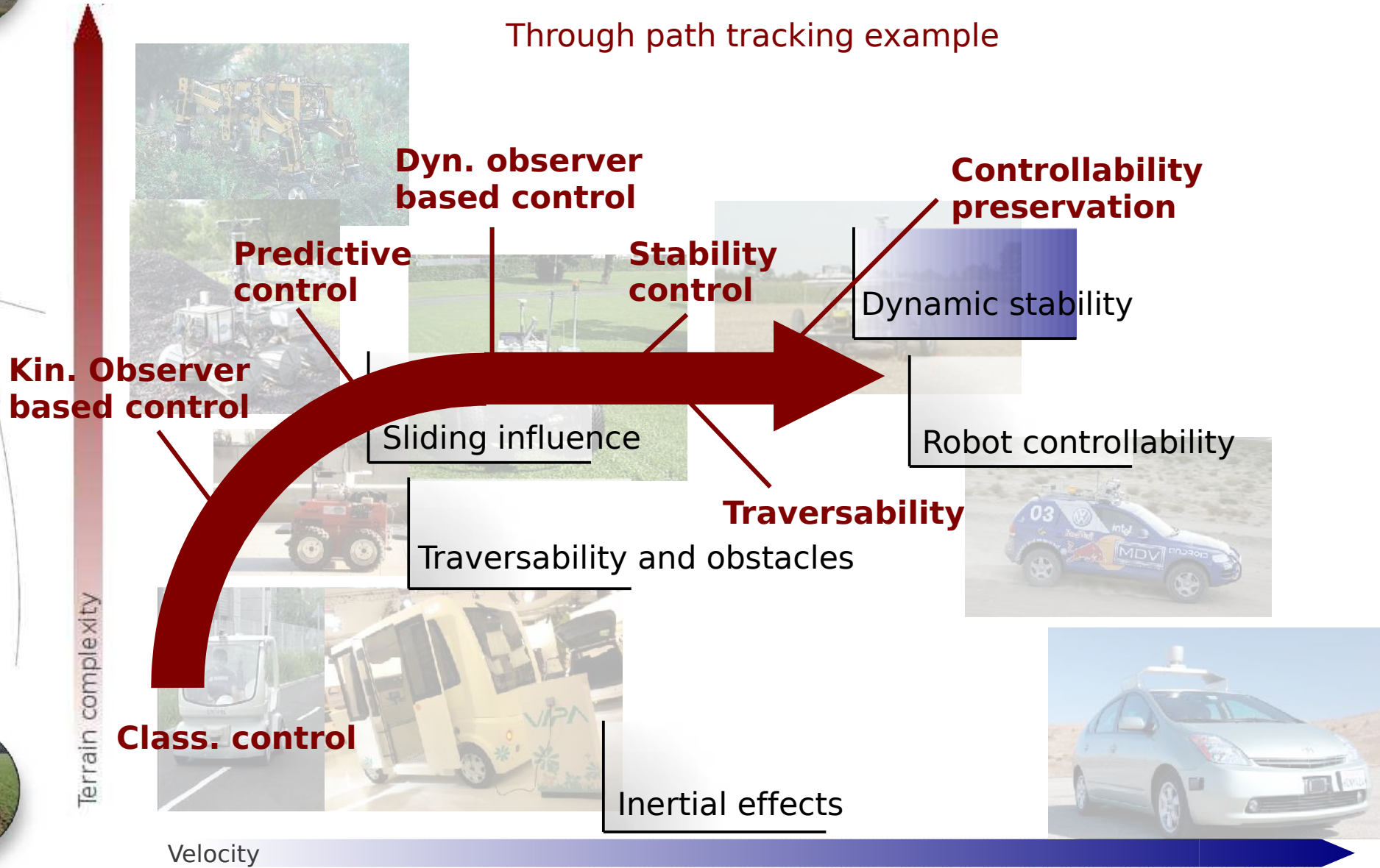
Achievable at 6m/s
[1m because St
angle]

Stab loss at 7m/s

▶ Controllability preservation...

Robot dynamics to be accounted with respect to the context

Through path tracking example

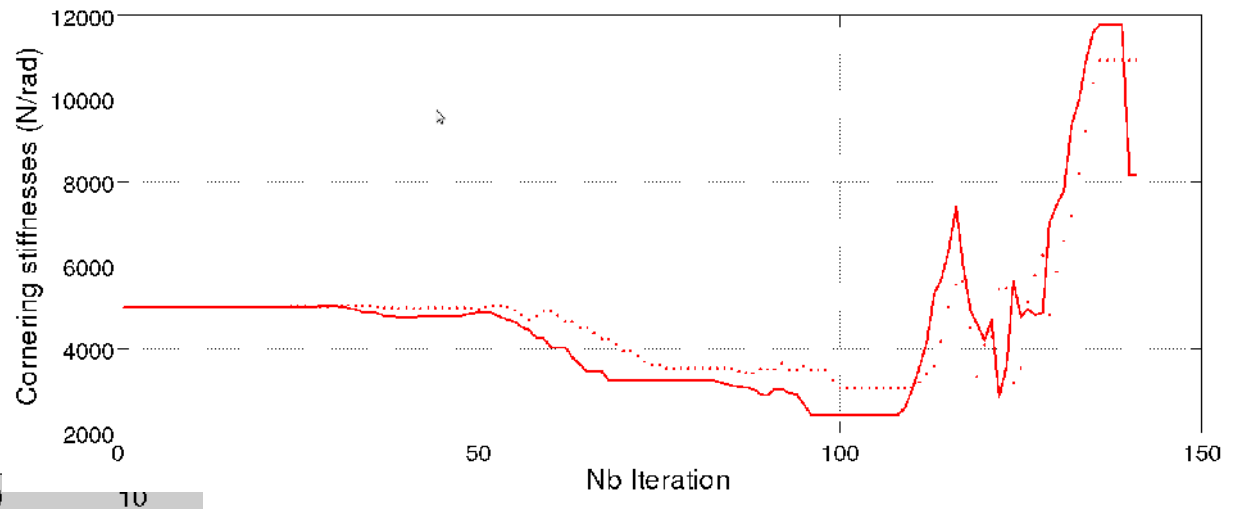
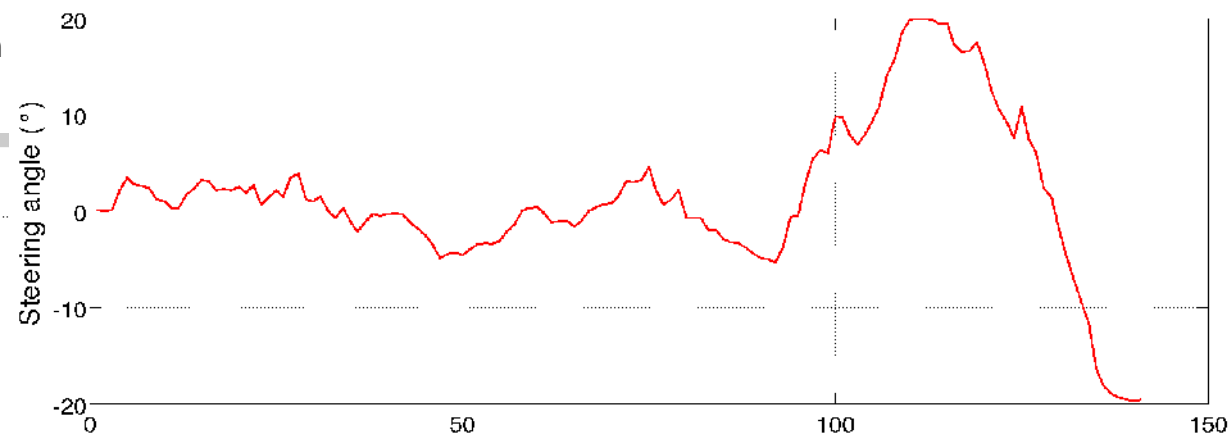
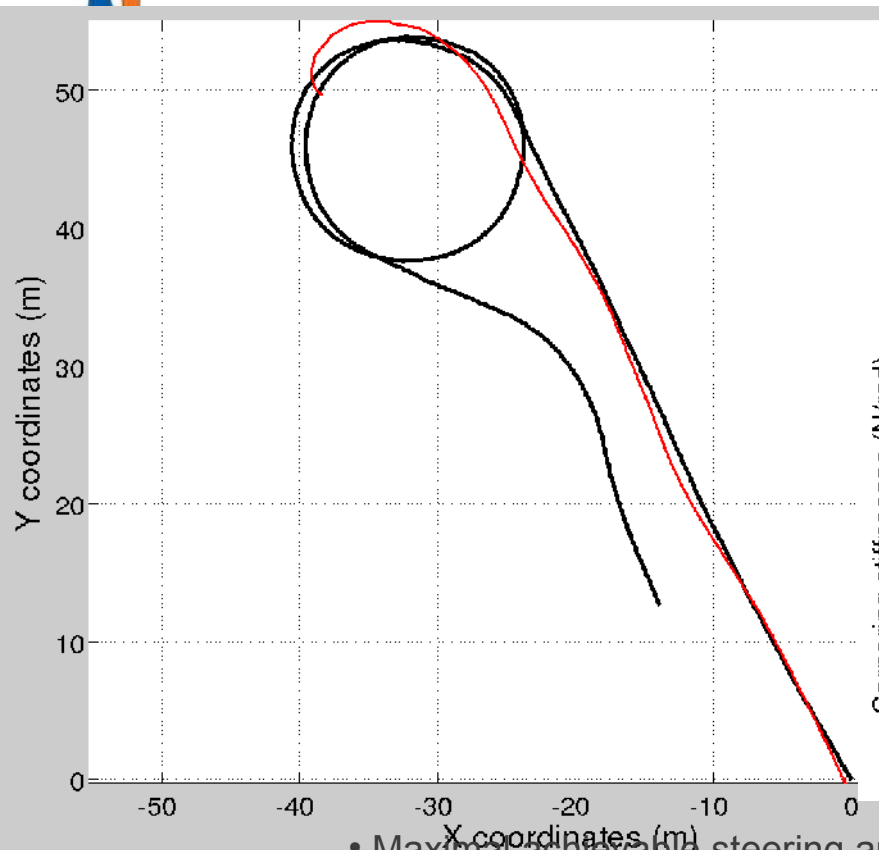


Controllability preservation...



Desired path even recorded is no more achievable at high speed

- Required high steering angle
 - The fastest, the lowest grip conditions
 - Mobile robot inertia



- Maximal achievable steering angle reached

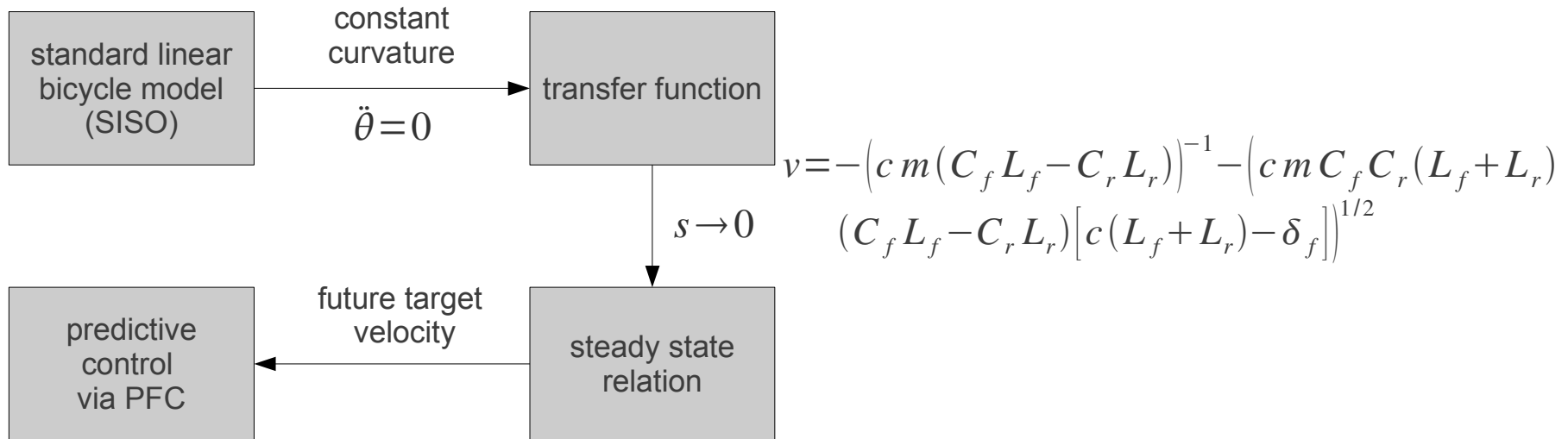
→ Steering angle limit viewed as an achievability criterion

Controllability preservation...

Objective of the controllability preservation

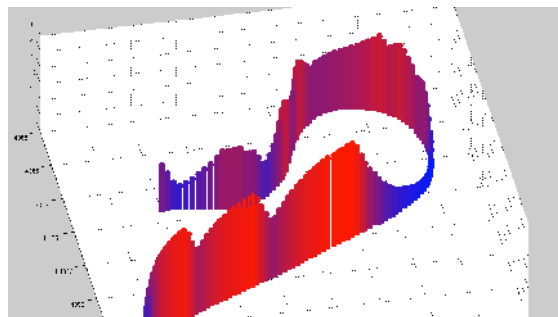
- Derive the maximal velocity
- Leading to a steering angle limit

Computation of the relationship between steering angle and velocity



- Future desired path curvatures

- Current grip conditions

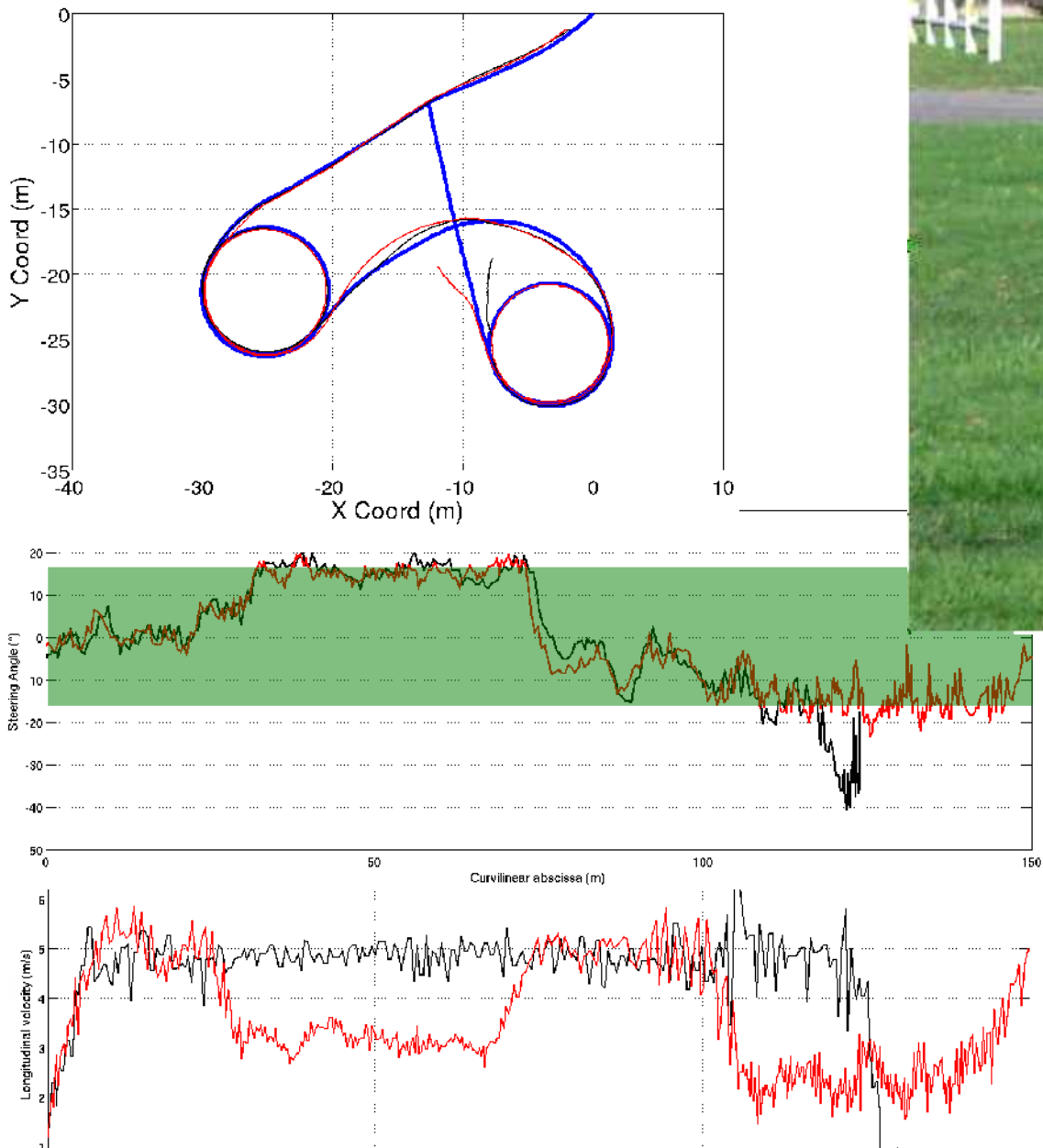


Computation of velocity profile

Controllability preservation...



Saturation set to 17°



- Without speed limitation
- Applying speed limitation

Summary

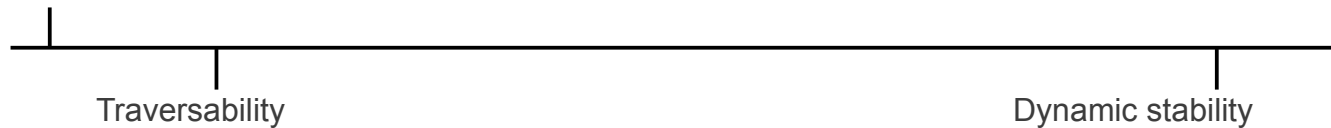


Off-road path tracking algorithm

- Generic approach to meet different kind of robots
- Preserving a high level of accuracy

- Grip conditions
- Terrain geometry
- Delay and dynamical effects
- Robot speed

- Preserving its integrity



Terrain geometry reconstruction
Obstacle avoidance



Longitudinal and lateral Rollover
Spin around situations



Summary



Several control law for advanced path tracking

- Adaptive and predictive control
- High accuracy despite grip conditions
- On several and widely different robot

Limited number of measures

- Only position and orientation is needed
 - Adaptive and predictive control
 - 4WS and trailer control
- Yaw rate may be required
 - Half turn (has speed cross 0)
 - High speed (to feed dynamic equations)

Current developments

- Perception system (avoid RTK-GPS)
- Account for robot limitations
- Ensure the robot performances

Arroco (RobucarTT)

- 4 electrical WD
- 4 WS
- 700kg
- Up to 3.7m/s (13km/H)



Automated farm tractor

- 2/4 WD
- 2 WS
- 7000kg
- Up to 11 m/s (40km/H)



RobuFAST

- 4 electrical WD
- 4 WS
- 400kg
- Up to 8m/s (30km/H)



Mti-Xsens IMU



RTK-GPS





2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011



Session V

Mobile robot modeling and control

- **Title: A control strategy taking advantage of inter-vehicle communication for platooning navigation in urban environment**
Authors: P. Avanzini, B. Thuilot, P. Martinet
- **Title: Semiautonomous Longitudinal Collision Avoidance Using a Probabilistic Decision Threshold**
Authors: J. Johnson, Y. Zhang, K. Hauser



2011 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Francisco, California, USA, September 30th, 2011

A control strategy taking advantage of inter-vehicle communication for platooning navigation in urban environment

P. Avanzini^{1,3}, B. Thuilot^{1,3} and P. Martinet^{2,3}

¹ Clermont Université, Université Blaise Pascal, LASMEA, BP 10448, 63000 Clermont-Ferrand, France

² Clermont Université, IFMA, LASMEA, BP 10448, 63000 Clermont-Ferrand, France

³ CNRS, UMR 6602, LASMEA, 63177 Aubière, France

Pierre.AVANZINI@lasmea.univ-bpclermont.fr

Abstract—This paper deals with platooning navigation in the context of innovative solutions for urban transportation systems. More precisely, a sustainable approach centered on automated electric vehicles in free-access is considered. To tackle the major problem of congestions in dense areas, cooperative navigation according to a platoon formation is investigated. With the aim to ensure the formation stability, i.e. longitudinal disturbances within the platoon do not grow when progressing down the chain, a global decentralized platoon control strategy is here proposed. It is supported by inter-vehicle communications and relies on nonlinear control techniques. A wide range of experiments, carried out with up to four urban vehicles, demonstrates the capabilities of the proposed approach: two localization devices have been tested (RTK-GPS and monocular vision) along with two guidance modes (the path to be followed is either predefined or inferred on-line from the motion of the manually driven first vehicle).

Index Terms—mobile robots, automatic guided vehicles, platooning, nonlinear control, path following.

I. INTRODUCTION

Urban mobility is currently being developed under a new conceptual framework induced by the significant traffic increase in metropolitan areas and growing sustainable considerations. Reducing congestion appears to be a critical goal which can be achieved by adopting a balanced and diversified mobility approach. As a consequence, the use of autonomous electric vehicles in free-access is a promising and environment-friendly alternative, especially when the public demand is properly structured, e.g. commutations within inner-cities or large industrial estates. The large flexibility that can be obtained with such a transport system (commutation at any time and along any route) is definitely its main attractive feature and should meet user expectations.

One functionality of special interest that can enhance this transport system is automated platooning, i.e. several autonomous vehicles following the trajectory of a first one, with pre-specified inter-distances. Such a functionality, on the one hand allows to easily adapt the transport offer to the actual need (via platoon length), and on the other hand eases maintenance operations, since only one person (driving the first vehicle) can then move several vehicles at a time (e.g. to bring vehicles back to some station). Moreover, since cooperative navigation can ensure more coherent motions, an increase in traffic as well as an enhancement in safety can be expected. Platooning is therefore considered in this paper.

Since a main objective is to manage the traffic flow, the performances of the whole formation must be guaranteed and therefore the concept of string stability [21] has to be considered when designing the control strategy. Namely, the stability of a platoon formation requires that the effects of disturbances are reducing when propagating from the leading vehicle to the follower ones, thus ensuring that unacceptable oscillations within the platoon can not be induced by sensor noises and/or actuator delay.



Fig. 1. Experimental vehicles: a Cycab leading three Cycab/RobuCab

The paper is organized as follows: the control architecture for vehicle platooning is first discussed in Section II. A global decentralized control strategy is then sketched in Section III and the integration of navigation functionalities is presented in Section IV. Finally, in Section V, experiments involving up to four electric vehicles demonstrate the capabilities of the proposed approach with different set-up and guidance modes.

II. CONTROL ARCHITECTURE DISCUSSION

As introduced in [20], string stability mainly depends on the information used for vehicle control. The different approaches proposed in the literature can then be classified into two categories: local and global strategies. The most common approaches rely on *local strategies*, i.e. each vehicle is controlled exclusively from the information it can acquire, relative only to the neighboring vehicles. The well-known *leader-follower approach* belongs to this category and considers only the preceding vehicle. Unfortunately, it has been proven [17] that an infinite string can not be stabilized by a linear controller even when the vehicle model is simplified as a double integrator. String stability can be achieved by considering either a non-identical approach [11]

or a variable spacing policy [8], [12]. In the first case, a communication is required between adjacent vehicles to adapt the control gains. However the gains are increasing with the vehicle index, so that from a practical point of view platoon length is limited. In the second case, the spacing policy depends on vehicle velocity and consequently affects the platoon tightness at the expense of the traffic flow. A related alternative, relying on a cyclic topology, ensures string stability by controlling the lead vehicle with respect to the last one [14]. Nevertheless, such a scheme lacks of flexibility and demands for the platoon to be supervised. As a result, more complex topologies have been investigated, using information from several nearby vehicles to achieve string stability. For instance, since spring-damper systems present intrinsic robustness to model error and measurement signal noise, such an analogy has been introduced, regarding vehicles as a serial chain of mass particles [23], [24], [7]. A control law is then derived from the combined front and rear information relying on the analogy with virtual mechanical forces. Unfortunately, in that case, string stability is only satisfied in realistic conditions for finite strings of autonomous vehicles (see [23]).

Such a problem can be overcome by considering *global strategies*, i.e. each vehicle is now controlled from data shared between all the vehicles. This category requires a communication network and a balanced management of information flows is essential. For instance, if a centralized architecture is adopted, all interactions between subsystems can be taken into account. Platooning can then be formulated as an optimization problem [13] and/or can use the formalism of generalized coordinates [6], where the formation is characterized by its geometry and its position with respect to some reference point. In both cases, the string stability is ensured, since the control input for each vehicle depends on the spacing errors of the entire formation. However, this poses a burdensome data handling problem, especially when the vehicle string is long. To circumvent this technological limitation, our research is focused on distributed control approaches. The emphasis is put on ensuring a stable formation while minimizing the communication cost between agents. From this point of view, a stable guidance approach is proposed in [19], [18], where the velocity and accelerations information of the lead vehicle are transmitted to all the following vehicles. Nevertheless, in practical situations, collision risks between adjacent vehicles can occur when using such a minimal communication scheme. The control of more general formations is considered in [9], [10], relying on virtual rigid structures whose dynamics is dictated with respect to some reference vehicle. Nevertheless, these techniques aim at imposing some pre-specified geometric pattern, and not that each vehicle accurately reproduces the trajectory of the first one. Instead, in this paper, a trajectory-based strategy is proposed, as illustrated in Fig. 2. The strategy relies on nonlinear control techniques: lateral and longitudinal controls are decoupled, so that lateral guidance of each vehicle with respect to the same reference path can be achieved independently from longitudinal control, designed

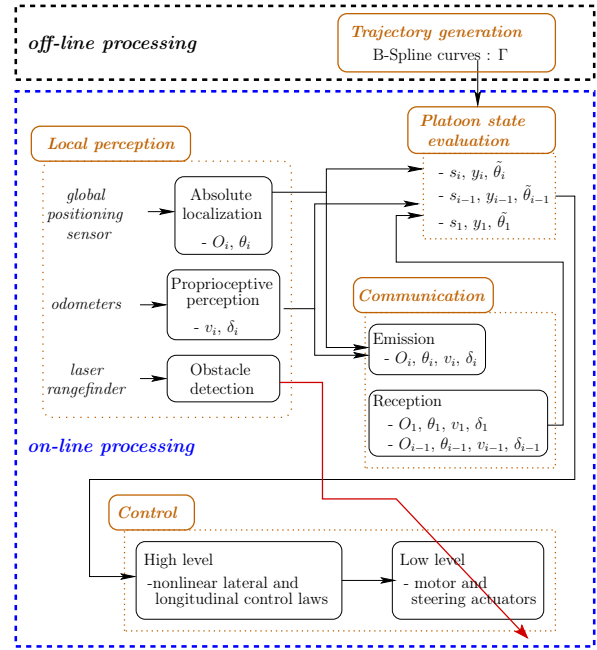


Fig. 2. Platoon architecture

to maintain a pre-specified curvilinear vehicle inter-distance. In order to ensure string stability as well as passengers security and comfort, the platoon behavior is imposed using data from the immediate front vehicle and from the leader one.

III. GLOBAL DECENTRALIZED CONTROL STRATEGY

A. Modeling assumptions

The scope of this study concerns small-sized vehicles acting in urban areas. As a result, several simplifying assumptions can be made. It is assumed that vehicles are rigid bodies, symmetrical with respect to their main axis and moving at quite low speed. It is thus possible to neglect the dynamic components (suspension . . .) and the tire deflection. In addition, dedicated navigation areas are asphalt roads and the contact between the wheels and the ground can be assumed without slipping. Given these assumptions, the control laws can be designed from a kinematic model and the most suitable one is the bicycle model (also called Ackermann model): the vehicle is then schematized by a rear driving wheel and a front steering wheel. To represent the system, the following notation, illustrated in Fig. 3 and 4, are introduced:

Notation:

- Γ is the common reference path for any vehicle (specified in advance or to be inferred from the trajectory of the first one), defined in an absolute frame $[A, X_A, Y_A]$.
- O_i is the center of the i^{th} vehicle rear axle.
- M_i is the closest point to O_i on Γ .

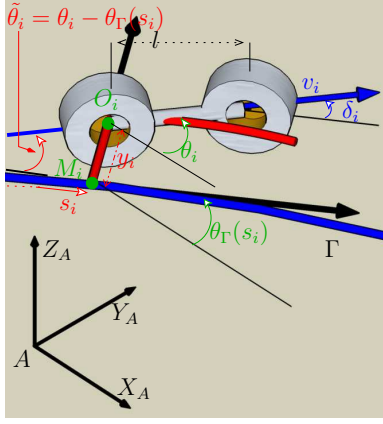


Fig. 3. Vehicle model

- s_i is the arc-length coordinate of M_i along Γ and $k(s_i)$ is the curvature of path Γ at this point.
- $\theta_\Gamma(s_i)$ is the orientation of the tangent to Γ at M_i with respect to $[A, X_A, Y_A]$.
- θ_i is the heading of i^{th} vehicle with respect to $[A, X_A, Y_A]$.
- $\tilde{\theta}_i = \theta_i - \theta_\Gamma(s_i)$ is the angular deviation of the i^{th} vehicle with respect to Γ .
- y_i is the lateral deviation of the i^{th} vehicle with respect to Γ .
- δ_i is the i^{th} vehicle front wheel steering angle.
- l is the vehicle wheelbase.
- v_i is the i^{th} vehicle linear velocity at point O_i .

Finally, Fig. 4 introduces the quantity $d_{i,j} = s_i - s_j$ that denotes the longitudinal distance between vehicles i and j in the platoon, evaluated as an arc-length distance along path Γ .

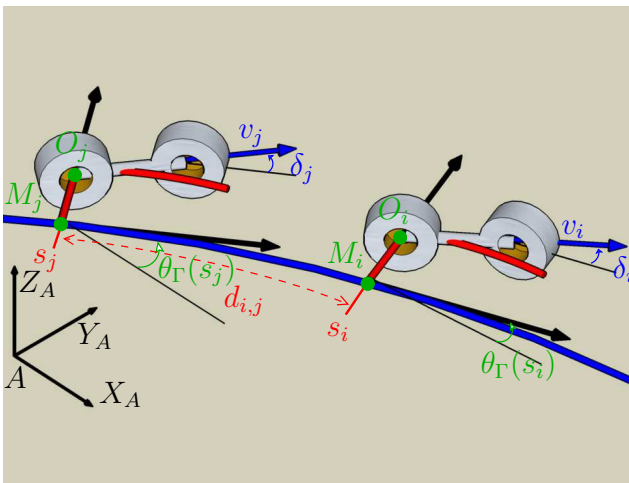


Fig. 4. Longitudinal distance $d_{i,j}$

State Space Model Derivation: The state of the i^{th} vehicle can be described in the Frenet frame along the reference trajectory Γ by the triplet $[s_i, y_i, \tilde{\theta}_i]$ and the state space equation of the bicycle model can be written as follows (see [5]):

$$\begin{cases} \dot{s}_i = \frac{v_i \cdot \cos \tilde{\theta}_i}{1 - y_i \cdot k(s_i)} \\ \dot{y}_i = v_i \cdot \sin \tilde{\theta}_i \\ \dot{\tilde{\theta}}_i = \frac{v_i \cdot \tan \delta_i}{l} - \frac{v_i \cdot k(s_i) \cdot \cos \tilde{\theta}_i}{1 - y_i \cdot k(s_i)} \end{cases} \quad (1)$$

The model (1) is clearly singular if $y_i = \frac{1}{k(s_i)}$, i.e. if O_i is superposed with the path Γ curvature center at abscissa s_i . However, this singularity is never encountered in practical situations, firstly because the curvature along the reference trajectory Γ is generally quite small, and secondly because the vehicle is expected to remain close to Γ .

B. Chained form of the state space model

Vehicle model (1) is nonlinear. However, it has been established in [16] that kinematic models of non-holonomic mobile robots can be converted via invertible state and control transformations into so-called *chained forms*, more convenient to address vehicle control. In the case of model (1), the state and control transformations are respectively given by (2) and (3):

$$\begin{aligned} \Phi([s_i \ y_i \ \tilde{\theta}_i]) &= [a_{1i} \ a_{2i} \ a_{3i}] \\ &\triangleq [s_i \ y_i \ (1 - y_i \cdot k(s_i)) \cdot \tan \tilde{\theta}_i] \end{aligned} \quad (2)$$

$$(m_{1i}, m_{2i}) = \Xi(v_i, \delta_i) \quad (3)$$

with:

$$m_{1i} \triangleq v_i \frac{\cos \tilde{\theta}_i}{1 - y_i \cdot k(s_i)} \quad (4)$$

$$m_{2i} \triangleq \frac{d}{dt}((1 - y_i \cdot k(s_i)) \cdot \tan \tilde{\theta}_i) \quad (5)$$

Transformations Φ and Ξ are invertible under the conditions $y \neq \frac{1}{k(s_i)}$ (model singularity discussed above), $v_i \neq 0$ and also $\tilde{\theta}_i \neq \frac{\pi}{2}[\pi]$, unexpected configurations if the platoon has been properly initialized. Substituting (2), (4) and (5) into (1), the nonlinear bicycle model can be rewritten without approximation as the following chained form:

$$\begin{cases} \dot{a}_{1i} = m_{1i} \\ \dot{a}_{2i} = a_{3i} m_{1i} \\ \dot{a}_{3i} = m_{2i} \end{cases} \quad (6)$$

Model (6) depends on two control variables: m_{1i} is consistent with the vehicle curvilinear velocity \dot{s}_i along Γ and m_{2i} is related to its angular velocity $\omega_i = \frac{v_i \cdot \tan \delta_i}{l}$. Control laws are now designed, relying on model (6). It is shown that vehicle guidance along a path can be achieved with performances independent of the longitudinal velocity, so that lateral and longitudinal control can be decoupled (described respectively in Sections III-C and III-D).

C. Lateral control

To this aim, chained system (6) is rewritten by differentiating with respect to the curvilinear abscissa rather than to the time. Denoting $a'_{ji} = \frac{da_{ji}}{da_{1i}} = \frac{da_{ji}}{ds_i}$, the chained form (6) is then:

$$\begin{cases} a'_{1i} = 1 \\ a'_{2i} = a_{3i} \\ a'_{3i} = \frac{m_{2i}}{m_{1i}} = m_{3i} \end{cases} \quad (7)$$

Path following consists in regulating to zero lateral and angular deviations. According to (2), it is equivalent to impose the convergence of variables a_{2i} and a_{3i} to zero. Given the structure of system (7) (a double integrator), a proportional derivative controller for variable m_{3i} is then chosen:

$$m_{3i} = -K_d a_{3i} - K_p a_{2i}, \quad (K_p, K_d) \in \mathcal{R}^{+*} \times \mathcal{R}^{+*} \quad (8)$$

where K_d and K_p are positive scalars specifying the convergence rate. By reporting (8) into (7), this controller indeed leads to the following differential equation:

$$a''_{2i} + K_d a'_{2i} + K_p a_{2i} = 0. \quad (9)$$

As equation (9) is differentiated with respect to the variable $a_{1i} = s_i$, a settling distance is specified by gains K_p and K_d instead of a settling time. Therefore, for a given initial positioning error of the i^{th} vehicle, its lateral behavior is imposed regardless of the vehicle linear velocity v_i , assumed non-zero, even if that velocity is time-varying [22]. By inverting chained transformations (2-4-5), the nonlinear control law is expressed as follows:

$$\delta_i(y_i, \tilde{\theta}_i) = \arctan \left(l \left[\frac{\cos^3 \tilde{\theta}_i}{(1-k(s_i) \cdot y_i)^2} \left(\frac{dk(s_i)}{ds_i} \cdot y_i \cdot \tan \tilde{\theta}_i - K_d(1-k(s_i) \cdot y_i) \tan \tilde{\theta}_i - K_p y_i + k(s_i)(1-k(s_i) \cdot y_i) \tan^2 \tilde{\theta}_i \right) + \frac{k(s_i) \cdot \cos \tilde{\theta}_i}{1-k(s_i) \cdot y_i} \right] \right) \quad (10)$$

It is well defined under the three conditions above mentioned ($v_i \neq 0$, $y_i \neq \frac{1}{k(s_i)}$ and $\tilde{\theta}_i \neq \frac{\pi}{2}[\pi]$).

D. Longitudinal Control

A local longitudinal error and a global one are here considered. First, the local error between the i^{th} and the $i+1^{\text{th}}$ vehicles is defined as:

$$e_{i+1}^i = d_{i,i+1} - d^* = s_i - s_{i+1} - d^* \quad (11)$$

Regulating e_{i+1}^i to zero would impose a constant curvilinear distance d^* between any pair of vehicles. Collision risks would then be explicitly addressed. However, as discussed in Section II, the main disadvantage of this local strategy is that the string stability can not be ensured. To overcome this problem, a global error e_{i+1}^1 is also considered:

$$e_{i+1}^1 = d_{1,i+1} - i \cdot d^* = s_1 - s_{i+1} - i \cdot d^* \quad (12)$$

Regulating e_{i+1}^1 to zero would impose a constant curvilinear distance with respect to a common absolute reference, chosen here as the abscissa of the platoon leader. Nevertheless, for obvious safety reasons, longitudinal control law cannot completely ignore the local error e_{i+1}^i because of the collision

risk: for instance, if the i^{th} vehicle stops or slows down, it will abnormally be approached by the $i+1^{\text{th}}$ one since this latter continues to maintain a constant gap with the lead vehicle.

In view of these remarks, a new hybrid error variable c_{i+1} is built from (11) and (12):

$$c_{i+1} = \sigma_{i+1}(z_{i+1}) \cdot e_{i+1}^1 + (1 - \sigma_{i+1}(z_{i+1})) \cdot e_{i+1}^i \quad (13)$$

$$\text{with } z_{i+1} = e_{i+1}^i + \frac{d^* - d_s}{2}, \quad (14)$$

and d_s denotes a security distance, defined as the minimal curvilinear distance that always must be observed between two vehicles. The function σ_{i+1} , defined in the interval $[0, 1]$, is defined as:

$$\sigma_{i+1}(z_{i+1}) = 0.5 \left(\frac{1 - e^{-az_{i+1}}}{1 + e^{-az_{i+1}}} + 1 \right) = \frac{1}{1 + e^{-az_{i+1}}}, \quad a > 0 \quad (15)$$

The behavior of function σ_{i+1} is illustrated in Fig. 5 with $a = 2.5$, $d^* = 5m$ and $d_s = 2, 3$ and $4m$. As evidenced by the ‘‘S’’ shape of the curve, function σ_{i+1} is used to give the predominance to either the global error or to the local one:

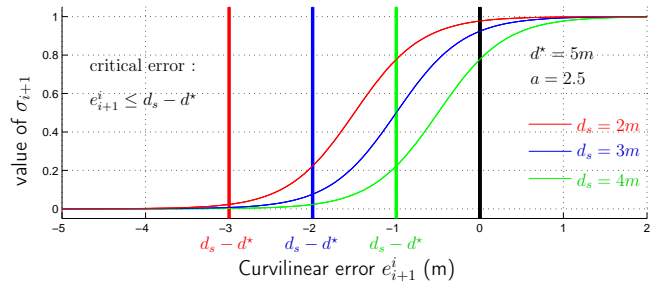


Fig. 5. Function σ_{i+1}

- when there is no imminent collision risk, i.e. $e_{i+1}^i \gg d_s$, the global approach relying on a common reference can be safely used ($\sigma_{i+1} = 1$ and therefore $c_{i+1} = e_{i+1}^1$).
- on the contrary, when the collision risk is important, i.e. e_{i+1}^i becomes less than $d_s - d^*$, the local approach must prevail over the global one ($\sigma_{i+1} = 0$ and therefore $c_{i+1} = e_{i+1}^i$).

The scheme of the longitudinal control is depicted in Fig. 6.

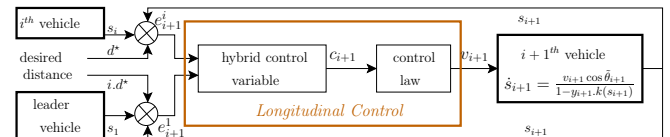


Fig. 6. Longitudinal control with a global strategy

Control law design: Differentiating the hybrid control variable c_{i+1} leads to:

$$\dot{c}_{i+1} = \sigma_{i+1} \dot{e}_{i+1}^1 + (1 - \sigma_{i+1}) \dot{e}_{i+1}^i + \dot{\sigma}_{i+1} e_{i+1}^1 - \dot{\sigma}_{i+1} e_{i+1}^i \quad (16)$$

In order to simplify the equations, let us denote:

$$A(z_{i+1}) = \frac{ae^{-az_{i+1}}}{(1 + e^{-az_{i+1}})^2} \quad (17)$$

Therefore $\dot{\sigma}_{i+1}$ can be written as:

$$\dot{\sigma}_{i+1} = A(z_{i+1})\dot{e}_{i+1}^i \quad (18)$$

Just as in lateral control design, exact linearization techniques can also be used: nonlinear equation (16) can be converted without approximation into the linear one:

$$\dot{c}_{i+1} = m_4(i+1) \quad (19)$$

by introducing the virtual control variable $m_4(i+1)$ related to v_{i+1} according to (just inject the expressions of \dot{e}_{i+1}^1 , \dot{e}_{i+1}^i and $\dot{\sigma}_{i+1}$ in (16):

$$v_{i+1} = \frac{1-y_{i+1}.k(s_{i+1})}{\cos \tilde{\theta}_{i+1}[1+A(z_{i+1})e_i^1]} \left(\sigma_i \frac{v_1 \cos \tilde{\theta}_1}{1-y_1.k(s_1)} + \right. \quad (20)$$

$$\left. [1 - \sigma_{i+1} + A(z_{i+1})e_i^1] \frac{v_i \cos \tilde{\theta}_i}{1-y_i.k(s_i)} - m_4(i+1) \right)$$

Convergence of c_{i+1} to zero can then be ensured by choosing a proportional controller for the variable $m_4(i+1)$:

$$m_4(i+1) = -K.c_{i+1}, \text{ avec } K \in \mathcal{R}^{+*} \quad (21)$$

The actual nonlinear longitudinal control law is finally obtained by reporting (21) into (20). It presents one singularity, namely $1 + A(z_{i+1})e_i^1 = 0$. However, this corresponds to a very special configuration of the first, the i^{th} and the $(i+1)^{th}$ vehicles, which is not expected to be encountered in practical situations. Moreover, if this configuration was met, v_{i+1} would increase to reach very large values, that would then be corrected by monitoring (not presented here).

IV. NAVIGATION FUNCTIONALITIES

The potentialities of this control strategy have first been demonstrated with the experimental vehicles shown in Fig.1, when their absolute localization is supplied by accurate RTK GPS receivers: completely automated platooning, with respect to some given reference trajectory specified beforehand, has been investigated in [4]. In order to increase the flexibility of this transportation system and enable the guidance in realistic urban conditions, new functionalities have been developed in a second step.

Manual guidance mode

A manual conveying functionality has been proposed in [1]. The lead vehicle is no longer in an autonomous mode, but manually driven and defining on-line the trajectory to be followed by the other vehicles. The architecture of the lead vehicle is then modified as shown in Fig. 7. Such a manual guidance mode is very attractive, for public transportation as well as for maintenance operations, since vehicles can then instantaneously be driven along any route, without requiring a previously recorded reference trajectory. The challenge consists in creating on-line a C^2 reference path (as required in control laws) as close as possible to the trajectory of the leader, although its raw localization data are noisy. Uniform B-Spline curves, extended on-line according

to an iterative optimization process, have been proposed to represent the reference path. A sliding window, only containing the latest localization data of the lead vehicle, enables to bound the computation time and adjust only the extremity of the reference trajectory, without impacting the part of this trajectory that the other vehicles are already tracking. Optimization parameters, namely the size of the sliding window, the number of control points to be adjusted and the degree of the B-Spline curves, are specified according to the vehicle velocity (see [1]).

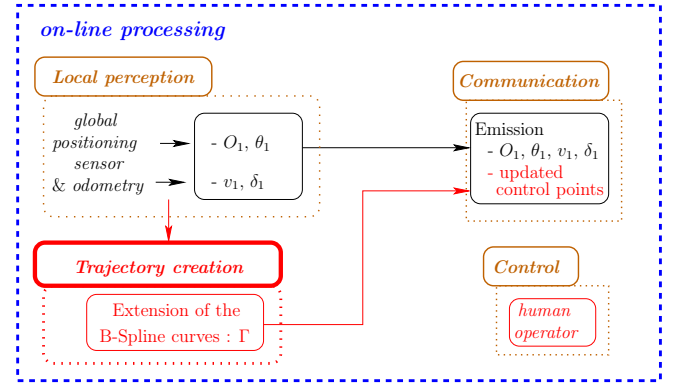


Fig. 7. Architecture of the lead vehicle

Localization by monocular vision

RTK GPS receivers are not realistic sensors in the context of urban transportation systems: they are not reliable since the satellite signals can be masked by tall buildings. Cameras appear more appropriate, since the buildings offer a rich environment from an image processing point of view (and in addition, they are definitely cheaper).

Vehicle localization relying on monocular vision has been investigated in [15]. The difficulty lies in the fact that absolute localization is expressed in a virtual vision world, slightly distorted with respect to the actual metric one. Such deformations, that mainly occur in the curved parts (i.e. when the points of interest used to localize the vehicle are changing), alter noticeably the estimation of inter-vehicle distances, and therefore impair longitudinal control performances. When completely automated platooning is addressed, two strategies have been proposed to estimate on-line local scale factors between the two worlds, along the specified reference trajectory (see Fig. 10). These local scale factors can then be used to correct raw localization data, enabling accurate distance evaluation.

Two vehicles are involved in the first approach [3], whose architecture is depicted in Fig. 8 for the second vehicle: its laser rangefinder is used to evaluate the direct distance to the leading vehicle, and an iterative optimization relying on these telemetric data is run to obtain the local scale factors. The second approach [2] is easier to implement since it relies only on the odometric data of the leading vehicle, as shown in Fig. 9: the local scale factors are then derived according to a nonlinear observer.

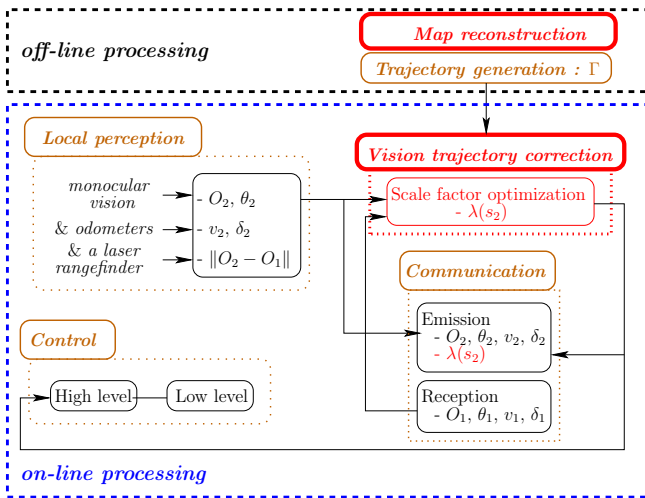


Fig. 8. Architecture of the second vehicle: first approach

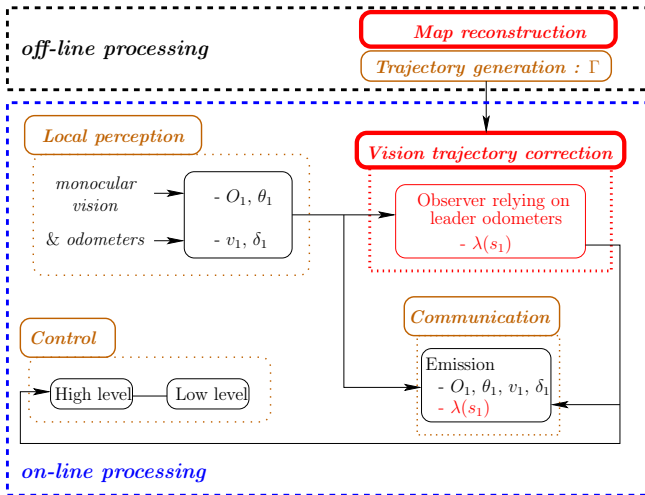


Fig. 9. Architecture of the lead vehicle: second approach

The relevance of these two strategies is illustrated in Fig. 10: it can be noticed that both scale factors sets computed on-line are very close to the actual ones evaluated off-line from RTK-GPS measurements. The scale factor peak, visible around $52m$, is due to the sudden appearance and disappearance of a tree in the field of perception of the camera.

V. EXPERIMENTAL RESULTS

In order to investigate the capabilities of the proposed control laws, several experiments have been carried out in Clermont-Ferrand on “PAVIN Site”, an open platform devoted to urban transportation system evaluation.

A. Experimental set-up

The experimental vehicles are shown in Fig. 1. They are electric vehicles, powered by lead-acid batteries providing 2 hours autonomy. Two (*resp. four*) passengers can travel

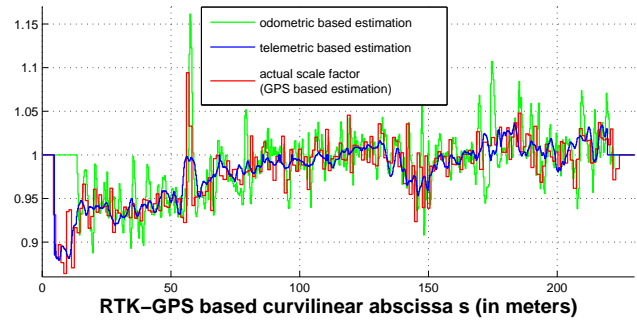


Fig. 10. On-line scale factor estimation

aboard the Cycab (*resp. the RobuCab*). Their small dimensions (length $1.90m$, width $1.20m$) and their maximum speed ($5m.s^{-1}$) are appropriate to urban environments. Vision-based localization and platoon control laws are implemented in C++ language on Pentium based computers using RTAI-Linux OS. Laser rangefinders provide telemetric data at a $60Hz$ sampling frequency, with a standard deviation within $2cm$. The cameras supply visual data at a sampling frequency between 8 and $15Hz$, according to the luminosity. Each vehicle is also equipped with an RTK-GPS receiver, running at a $10Hz$ sampling frequency, either used for vehicle control (manual guidance mode) or exclusively devoted to performance analysis (vision-based localization). Finally, inter-vehicle communication is ensured via WiFi technology. Since the data of each vehicle are transmitted as soon as the localization step is completed, the communication frequency is similar to the frequency of the localization device.

B. Experimental results

The experiments reported below consist in platoon control with a constant leader velocity $v_1 = 1m.s^{-1}$. Several scenarios have been investigated.

1) *Manual guidance mode*: Three vehicles follow the path generated on-line by a manually driven one along a $240m$ -long path. The lateral deviation of of the three followers remains mainly within $\pm 10cm$ from the leader trajectory and does not exceed $14cm$, see Fig. 11. Lateral guidance is therefore as satisfactory as in previous work, when vehicles were guided with respect to a pre-specified reference trajectory (see [4]).

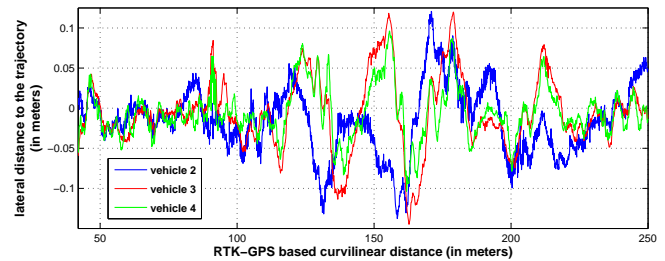


Fig. 11. Vehicle lateral deviations: manual guidance mode

The accuracy of the longitudinal control law is investigated in Fig. 12. Once the platoon is in nominal mode (i.e. all vehicles have reached their desired inter-distances), the behavior is identical to what was observed in previous work [4], namely a 10cm accuracy: the on-line reference trajectory generation does not disturb the longitudinal performances.

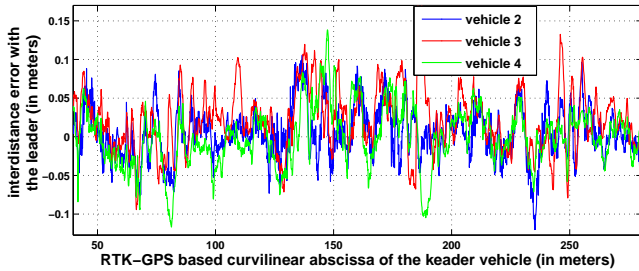


Fig. 12. Vehicle inter-distance errors: manual guidance mode

2) *Localization by monocular vision*: Three vehicles navigate in an autonomous mode along a 230m-long predefined reference trajectory. Local scale factors are computed on-line either from the odometric data of the leading vehicle or from the telemetric data of the first follower (as discussed in section IV) and then retransmitted to the other vehicles in the platoon.

Platoon control performances are evaluated in Fig. 13 (longitudinal error of the second and third vehicles are shown respectively in the top and bottom graphs). When inter-distance error is directly deduced from raw localization vision data, longitudinal control is largely erroneous, see the black curve in Fig. 13. These large errors, namely 40cm for the second vehicle and 70cm for the third one, show clearly the necessity for local corrections. The relevance of the two proposed strategies is demonstrated in Fig. 13: whatever the vehicle, the longitudinal errors satisfactorily remain within 10cm without exceeding 16cm and 21cm, respectively with odometric and telemetric-based corrections.

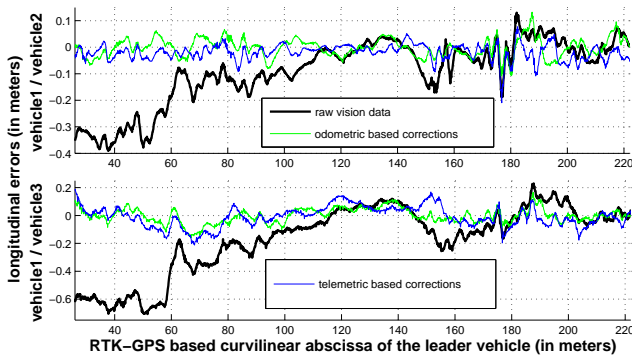


Fig. 13. Vehicle inter-distance errors: localization by vision

VI. CONCLUSION

In this paper, vehicle platooning in urban environment has been investigated. First, a global decentralized control strategy, taking advantage of inter-vehicle communications, has been proposed, in order to ensure the stability of the entire formation and therefore avoid error accumulation inherent to local control approaches. Moreover, nonlinear control techniques have been considered, in order to take explicitly into account the nonlinearities in vehicle models, so that the same high accuracy can be expected in any situation (for instance, whatever the reference trajectory curvature).

Flexibility and adaptation to urban areas have been increased by introducing two functionalities: first it is possible to manually drive the platoon, for instance to ease maintenance operations. Secondly, instead of using GPS receivers for vehicle localization, more realistic vision sensors are considered: vehicle localization can be inferred from monocular vision, complemented with local scale factors estimated on-line either from telemetric or odometric data.

Full scale experiments, carried out with up to four vehicles, have finally demonstrated the efficiency of the proposed approach. Reported control performances are satisfactory whatever the navigation mode and localization device.

REFERENCES

- [1] P. Avanzini, B. Thuilot, T. Dallej, P. Martinet, and J. P. Dérutin. On-line reference trajectory generation for manually conveying a platoon of automatic urban vehicles. In *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, pages 1867–1872, St. Louis (USA), 2009.
- [2] P. Avanzini, B. Thuilot, and P. Martinet. Accurate platoon control of urban vehicles, based solely on monocular vision. In *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, pages 6077–6082, Taipei (Taiwan), 2010.
- [3] P. Avanzini, B. Thuilot, and P. Martinet. Urban vehicle platoon using monocular vision: Scale factor estimation. In *IEEE Intern. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 2010.
- [4] J. Bom, B. Thuilot, F. Marmoiton, and P. Martinet. A global strategy for urban vehicles platooning relying on nonlinear decoupling laws. In *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, pages 1995–2000, Edmonton (Canada), 2005.
- [5] Jonathan Bom. *Étude et mise en oeuvre d'un convoi de véhicules urbains avec accrochage immatériel*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand (France), july 2006.
- [6] R. Caicedo, J. Valasek, and J.L. Junkins. Preliminary results of one-dimensional vehicle formation control using structural analogy. In *American Control Conference*, 2003.
- [7] J.M. Contet, F. Gechter, P. Gruer, and A. Koukam. Bending virtual spring-damper: a solution to improve local platoon control. In *Intern. Conf. on Computational Science (ICCS)*, pages 601–610, Baton Rouge (USA), 2009.
- [8] P. Daviet and M. Parent. Longitudinal and lateral servoing of vehicles in a platoon. In *IEEE Symp. on Intelligent Vehicles (IV)*, pages 41–46, Tokyo (Japan), 1996.
- [9] T. Dierks and S. Jagannathan. Control of nonholonomic mobile robot formations: backstepping kinematics into dynamics. In *IEEE Intern. Conf. on Control Applications (CCA)*, pages 94–99, Singapore, 2007.
- [10] A. Gil-Pinto, P. Fraisse, and R. Zapata. Decentralized strategy for car-like robot formations. In *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, pages 4176–4181, San Diego (USA), 2007.
- [11] M.E. Khatir and E.J. Davison. Decentralized control of a large platoon of vehicles using non-identical controllers. In *American Control Conference (ACC)*, pages 2769–2776, Boston (USA), 2004.
- [12] S. Klinge and R.H. Middleton. Time headway requirements for string stability of homogeneous linear unidirectionally connected systems. In *IEEE Conf. on Decision and Control (CDC)*, pages 1992–1997, Shanghai (China), 2009.

- [13] W. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, 11(3):355–361, 1966.
- [14] J.A. Rogge and D. Aeyels. Vehicle platoons through ring coupling. *IEEE Transactions on Automatic Control*, 53(6):1370–1377, 2008.
- [15] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier, and F. Marmoiton. Outdoor autonomous navigation using monocular vision. In *IEEE Intern. Conf. on Intelligent Robots and Systems (IROS)*, pages 3395–3400, Edmonton (Canada), 2005.
- [16] C. Samson. Control of chained systems: application to path following and time-varying point stabilization of mobile robots. *IEEE Trans. on Automatic Control*, 40(1):64–77, 1995.
- [17] P. Seiler, A. Pant, and K. Hedrick. Disturbance propagation in vehicle strings. *IEEE Transactions on automatic control*, 49(10):1835–1841, 2004.
- [18] S. Sheikholeslam and C. A. Desoer. Longitudinal control of a platoon of vehicles. i, ii and iii. Institute of transportation studies, research reports, Institute of Transportation Studies, UC Berkeley, 1990.
- [19] S.E. Shladover. Operation of automated guideway transit vehicles in dynamically reconfigured trains and platoons. Technical report, U.S. Department of Transportation, Urban Mass Transportation Administration, Washington, D.C., july 1979.
- [20] D. Swaroop. *String Stability of Interconnected Systems: An application to platooning in AHS*. PhD thesis, University of California at Berkeley, December 1994.
- [21] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.
- [22] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet. Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor. In *IFAC Symp. on Intelligent Autonomous Vehicles (IAV)*, Lisbon (Portugal), 2004.
- [23] D. Yanakiev and I. Kanellakopoulos. A simplified framework for string stability analysis in AHS. In *IFAC World Congress*, pages 177–182, San Francisco (USA), 1996.
- [24] S.Y. Yi and K.T. Chong. Impedance control for a vehicle platoon system. *Mechatronics*, 15(5):627–638, 2005.

Semiautonomous Longitudinal Collision Avoidance Using a Probabilistic Decision Threshold

Jeffrey Johnson Yajia Zhang Kris Hauser

School of Informatics and Computing, Indiana University at Bloomington

{jj56, zhangyaj, hauserk}@indiana.edu

Abstract—Automated emergency maneuvering systems can avoid or reduce the severity of collisions by taking control of a vehicle away from the driver during high-risk situations. The choice of when to switch to emergency control is challenging in the presence of uncertain information (imperfect sensors, road conditions, uncertain object behavior, etc.) and many dynamic obstacles. This paper considers longitudinal collision avoidance problems for a vehicle traveling along a known path. A probabilistic decision threshold framework is presented in which the user’s control is overridden if the probability that it would lead the system into an unsafe state exceeds some threshold. We apply the technique to collision imminent braking for obstacles traveling along the vehicle’s path, and present preliminary results extending the technique to the scenario of obstacles crossing the vehicle’s path.

I. INTRODUCTION

Over 6.3 million automobile crashes occurred in the U.S. in 2007, including 1.8 million injury crashes and 37,435 fatalities at a cost of hundreds of billions of dollars [9]. Although the numbers of injuries and fatalities per traveled mile have decreased significantly due to advances in passive safety equipment (seat belts, air bags, stability control, etc) over the last four decades, these numbers have leveled off over the last two decades. Semiautonomous active safety systems, which override driver control of the vehicle in emergency situations, are one promising approach to achieving significant further improvements in safety.

These systems must distinguish between emergency scenarios, assess collateral impact of collision-mitigating or collision-avoidance strategies, and understand the behavior of drivers, including the drivers of other cars, in response to the car’s actions. Also, they must carefully balance keeping the driver feeling “in control” and taking necessary deviations from the driver’s actions (or inaction) for increased safety. It is also likely that too much automation could increase the risk of inattentive or careless driving and also increase liability for auto manufacturers; hence a *minimal necessary interference* principle should be applied to the design of semiautonomous safety systems at least for the foreseeable future.

This paper applies this principle to the prevention of longitudinal collision, specifically, rear end collisions during single-lane driving and transverse collisions during intersection crossing. Given a known path of the vehicle but unknown speed and driver input, the problem is applying longitudinal controls (accelerating and braking) only when necessary to avoid collision with moving or static objects

in the environment. Uncertainty is a major challenge in the driving environment due to noisy distance readings; unknown behavior of the object; errors in speedometer readings due to tire wear and environmental factors; and unknown stopping time due to brake wear and road surface characteristics. In the context of rear-end collision avoidance we apply a probabilistic decision thresholding technique that activates control when the risk of collision exceeds some threshold.

Assuming that the vehicle tracks a state distribution using an Extended Kalman Filter, we used Monte-Carlo simulations to evaluate the technique’s performance on a suite of scenarios including dry and wet pavement, static and braking obstacles, and false positives and negatives in object detection. We aggregated four performance metrics — collision velocity, completion time relative to an ideal driver, stop distance relative to the obstacle, and jerkiness — across 10 scenarios into a risk index, which quantifies the overall severity of collisions, and an interference index, which quantifies the overall disturbance to the human driver. Plotting these indices demonstrates a clear tradeoff between increased interference and increased safety as the activation threshold is varied.

We also describe first steps toward solving the intersection crossing problem in which the car accelerates or decelerates in order to avoid collision with cars in opposing lanes. We introduce a conservative reachable set computation in the vehicle’s position/velocity/time space for an arbitrary number of lanes of traffic in a fully observable environment. Using this computation we intend to address both accelerating and braking under uncertainty in a tractable manner using the same probabilistic decision threshold technique.

II. RELATED WORK

Collision-imminent braking systems in some existing Volvo and Mercedes-Benz models use a variety of sensors to detect collision-imminent scenarios and apply brakes to reduce the severity of a crash. We are interested in extrapolating collision-*imminent* braking to its inevitable conclusion: collision-*prevention* braking.

Autonomous driving has recently become a tremendously active field of research, inspired by major successes such as the DARPA Grand Challenge [12]. A Google system that has logged over 140,000 autonomous miles, including 1000 miles without intervention from the human driver [10]. Despite these major advances, there is still a major gap between

these systems and human drivers. Although human drivers occasionally err, they are extremely reliable in general: a fatality crash occurs approximately once in 100 million miles driven [9]. Even if an autonomous vehicle relies on the human driver input once every 10,000 miles, the driver must be attentive 99.99% of the time for the system to perform as well as a human alone!

By contrast, active safety systems take control of the vehicle only when emergency happens or potential accident is foreseen to mitigate or avoid the consequences of an accident. There are several strategies for choosing when to override driver control. A longitudinal collision-mitigation braking strategy was described by Hillenbrand et al (2006) that gradually applies stronger braking as the collision boundary is approached, which smoothes the control output and copes some uncertainty [5]. Anderson et al (2009) present a 2D hazard avoidance scheme based on model predictive control which allows varying levels of autonomy based on risk assessed by control magnitudes [2]. Our approach introduces the additional considerations of uncertainty which provides a more natural definition of risk. Karlsson et al (2004) introduced a statistical decision rule that applies the brake if the probability of impact is greater than some threshold α [7]. This approach is advantageous in the presence of uncertainty. Similar thresholding techniques were applied to autonomous driving in environments mapped using 2D range finders [1]. We generalize this approach in this research to both acceleration and braking using a probabilistic minimum necessary interference criterion that treats safety as a hard probabilistic constraint and driver interference as a soft constraint. Furthermore, we consider unknown road surface characteristics and obstacle behaviors in our uncertainty model. By incorporating this uncertainty into decision-making, fewer collisions result on wet pavement, but the vehicle behaves more conservatively on dry pavement.

The path-time decomposition was introduced by Kant and Zucker (1986) who examined the problem of dynamic obstacle avoidance along a given path under velocity constraints[6]. Liu and Arimoto (1992) introduced an algorithm for the more general problem of shortest path planning among polygonal and curved obstacles[8]. We extend this approach to include path velocity and acceleration bounds in order to compute reachable sets in unprotected lefthand turns. A straightforward method for integrating uncertainty in obstacle velocity and behavior is considered as well.

III. SAFETY-CONSTRAINED MINIMAL INTERFERENCE PRINCIPLE

The vehicle's policy π is given the user's desired control u_t^d and sensor input z_t . Using z_t it infers a distribution $P(x_t)$ over hypothetical car-obstacle system states. Although our model can be generalized to two-dimensional motion with steering and velocity control, here we will only consider the longitudinal control problem in which the vehicle travels along a one dimensional space of a known single-lane road, which may be curved or straight. The output of the system

is a continuous control $u \in [-1, 1]$, where $u = 0$ indicates no control, $u = -1$ indicates maximum braking, and $u = 1$ indicates maximum acceleration.

We define the *safety-constrained minimum interference* control u_t^* at probability α as the result of the following optimization:

$$\begin{aligned} u_t^* &= \arg \min_{u \in [-1, 1]} |u - u_t^d| \\ \text{s.t. } & P(\text{safe}|u_t = u) \geq \alpha \end{aligned} \quad (1)$$

where we define $P(\text{safe}|u_t = u)$ as the probability that the system remains safe given the choice of u at the current time step and the safest sequence of controls thereafter. If no such u meets the α threshold, we set

$$u_t^* = \arg \max P(\text{safe}|u_t = u). \quad (2)$$

This framework has several advantages in that the user's control will be replicated exactly if it is sufficiently safe, and safety and driver interference can be tuned using a single parameter α . The major challenge in this framework is evaluating the $P(\text{safe}|u_t = u)$ because it essentially requires solving a stochastic optimal control problem with nonlinear noise terms and constraints. To address this we make the approximation that the probability can be approximated by integrating over the optimal hypotheses evaluated under $P(x_t)$ assuming that the underlying state hypothesis is true. In other words,

$$P(\text{safe}|u_t = u) \approx \int_x S(x, u) P(x_t = x) dx \quad (3)$$

where $S(x, u)$ evaluates whether the system can remain safe under known state x and initial control u . Below we will present two concrete implementations whereby $S(x, u)$ can be analytically evaluated, which makes the evaluation of (3) tractable.

IV. COLLISION IMMINENT BRAKING

A. Assumptions and System Structure

First we assume the vehicle is moving in the same direction as the obstacle and the obstacle does not move in reverse. We assume that the vehicle is equipped with a speedometer and a range finder (e.g., radar or lidar). The behavior of an obstacle is considered as a black box, and the vehicle needs to infer whether an obstacle is still, accelerating, or braking through the information received through its sensors. In order to synthesize this information, we suppose the car runs an Extended Kalman Filter (EKF) to track a probability distribution over the system state [4]. It then incorporates this distribution into a probabilistic decision rule.

B. Stochastic Dynamics Model

The state of the car can be described using car's position p_c , velocity v_c and the maximum deceleration a_{cmax} that the car currently can apply. Additionally, when an obstacle is present, the obstacle position p_o , velocity v_o , and acceleration a_o are modeled as part of the system state. The vehicle receives a noisy speedometer v_s and range reading d . Decisions

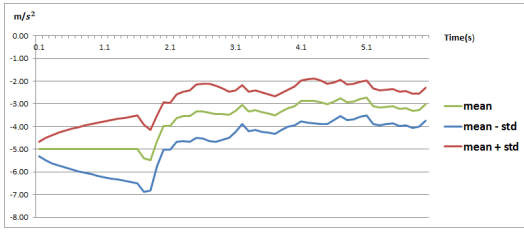


Fig. 1: The vehicle starts with an estimate of $-5 m/s^2$ maximum deceleration. As it starts braking on a wet pavement at $t = 1.5 s$, the Kalman filter adjusts the estimate toward the correct value of $-3 m/s^2$.

TABLE I: Dynamics and Observation Models

Road Surface \hat{a}_c	Maximum applicable acceleration is a random walk $\hat{a}_c \sim \mathcal{N}(0, \Delta t)$
Actuation Errors \hat{v}_c	Proportional to control and maximum deceleration, $\hat{v}_c = ua_c(1 + \epsilon_u)$, with $\epsilon_u \sim \mathcal{N}(0, 0.01^2)$
Object Behavior \hat{a}_o	Random noise with 99.99% within [$-5.0 m/s^2, 5.0 m/s^2$], $\hat{a}_o \sim \mathcal{N}(0, 1.25^2)$
Speedometer v_s	Multiplicative noise on actual velocity, $v_s = v_c(1 + \epsilon_s)$, $\epsilon_s \sim \mathcal{N}(0, 0.025^2)$ (99.99% within 10% of current velocity)
Range Reading d	Combined linear and multiplicative noise $d = n_{dL} + (p_o - p_c)(1 + n_{dM})$, with $n_{dL}, n_{dM} \sim \mathcal{N}(0, 0.0125^2)$ (99.99% within by 5 cm + 5% of true distance)

are made at a time step of Δt (0.1 s in our implementation). The stochastic dynamics and sensor noise models are listed in Table I.

C. Extended Kalman Filter

The vehicle is assumed to employ an extended Kalman filter (EKF) in order to estimate the state from the stochastic dynamics and observations. An EKF is a version of the Kalman filter that addresses nonlinear systems by linearizing about the estimated mean and covariance [13]. The dynamics at time step t can be written in the following form:

$$x_{t+1} = f(x_t, u_t) + w_t \quad (4)$$

$$z_t = h(x_t) + v_t \quad (5)$$

$$w_t \sim \mathcal{N}(0, Q_t) \quad (6)$$

$$v_t \sim \mathcal{N}(0, R_t) \quad (7)$$

Here, x_t denotes the state $(p_c, v_c, a_{c_{max}}, p_o, v_o, a_o)$, u_t denotes the braking control input, z_t is the observation (d, v_s) at time step k . w_t is the process error term with Q_t as its covariance matrix. v_t is the measurement noise term with R_t as its covariance matrix.

At each step, the EKF maintains a state estimate \hat{x}_t and covariance matrix P_t . Upon reading the observation z_t from the vehicle's sensors, the EKF performs a Kalman update using the system linearized about \hat{x}_t to obtain a new state estimate \hat{x}_{t+1} and covariance P_{t+1} . (Figure 1)

Because obstacles may appear and disappear from the range sensor reading, the obstacle state and distance measurements are included in the EKF update only when an

obstacle is detected. When an obstacle appears for the first time, its position estimate is initialized to the raw range sensor estimate $\mathcal{N}(d, (0.0125d)^2)$. Its velocity is initialized to a broad distribution $\mathcal{N}(\hat{v}_c/2, (\hat{v}_c/2)^2)$, and its acceleration is initialized to $\mathcal{N}(0, (2.5 m/s^2)^2)$.

Although the EKF is known to suffer from problems in highly nonlinear systems, in our case the system is close to linear and the EKF seems to provide sufficiently accurate performance. Nevertheless our decision-making algorithms still apply to more general state estimators, like particle filters.

V. BRAKING POLICIES AND EVALUATION

A braking policy $\pi(\hat{x}, P)$ produces a braking output u given the state estimate \hat{x} from the Kalman filter and its covariance matrix P . Given perfect state information, the optimal policy is essentially trivial (known-state policy). But in the presence of uncertainty, optimality is not easy to define. We design a probabilistic approach to deal with uncertainty and produce human-like braking behavior.

A. Known-State Policy

The basic optimal braking policy $\pi_D(x)$ in the presence of complete state certainty is a bang-bang control given as follows.

Algorithm 1 Bang-Bang Policy

```

 $p'_c \leftarrow p_c + v_c^2 / (-2a_{c_{max}}) + C$ 
 $t' \leftarrow v_c / (-a_{c_{max}})$ 
if  $v_o + a_o t' \geq 0$  then
   $p'_o \leftarrow p_o + v_o t' + 1/2 a_o t'^2$ 
else
   $p'_o \leftarrow p_o + v_o^2 / (-2a_o)$ 
end if
if  $p'_c > p'_o$  then
  return  $u = -1$ 
else
  return  $u = 0$ 
end if

```

Here, C is a constant that is used for indicating some safety margin, which we set to 1 m. p'_c is the estimated stopping position of the car if it initiates maximum braking. p'_o defines the estimate position of the obstacle when the car stops, and it will either stop after the car does (first conditional branch) or before (second branch). If $p'_c > p'_o$, there will be a collision between car and obstacle, otherwise, no collision.

Using this policy we can implement (1) in a straightforward manner. First note that stronger braking is always guaranteed to be safer, which helps with the minimization. Then the process boils down to finding the weakest braking control that keeps the system safe with probability α . To approximate the integral in (3), we sample N candidate states $x^{(i)} \sim \mathcal{N}(\hat{x}, \hat{P})$, ($i \in [1, N]$) and compute the weakest (highest value) control $u^{(i)} = \pi_D(x^{(i)})$. We let $\pi(\hat{x}, \hat{P})$ take on the value of the control output at the $100(1-\alpha)$ percentile.

B. Smoothing in Post-Processing

The control policies discussed above are Markovian in that they produce the control output merely based on the state and covariance estimate at the current time step. Jerkiness might occur if control outputs are quite different in consecutive time steps. Therefore, We implement a discounting method that produces a weighted sum of $(1 - D)$ of the smoothed control in its previous time step and D of the current π output. The constant D is known as the *discount factor*, and when D is low the output is highly smoothed, and when $D = 1$ the output is identical to the unsmoothed signal. Through experiments we found that the setting $D = 0.5$ strikes a good balance control smoothness and responsiveness.

C. Policy Evaluation

A good braking policy should be able to achieve high safety and low unnecessary control interference. In order to evaluate the policy we define the a risk index (RI) and interference index (II) which are functions of the following metrics:

- 1) *Collision velocity* (CV): the relative velocity if the car hits the obstacle, 0.0 otherwise.
- 2) *Discontinuity Time* (DT): measures the amount of jerk experienced by passengers. We integrate the time over which the acceleration at two time subsequent time steps is greater than some threshold, which we set to 4 m/s^2 .
- 3) *Excess Time* (ET): measures the amount of time consumed by excessive braking in a scenario. A scenario is considered completed if the car stops, collides with an obstacle, or passes some marker. ET is computed by measuring the policy completion time and subtracting the completion time for an optimal collision-free policy with perfect state information.
- 4) *Stopping Distance* (SD): measures the distance to the obstacle after the vehicle stops, or 0.0 if the scenario is completed in any other manner.

RI and II are computed as follows:

$$RI = (CV_{avg}/CV_{safe})^2$$

$$II = c_1DT + c_2ET + c_3SD$$

CV_{safe} is a constant that denotes a relatively safe velocity at which a collision is unlikely to lead to serious injury (set to 5 m/s in our implementation). RI is made proportional to the kinetic energy of the collision. Policies with $RI < 1$ are relatively safe. c_1 , c_2 , and c_3 are proportionality constants that are set to 2 s^{-1} , 1 s^{-1} , and $1/2 \text{ m}^{-1}$ respectively based on some amount of tuning. The goal of a braking policy is achieving low RI and II . It is important to distinguish between the interference objective used in (1), which is an instantaneous criterion used in the vehicle's internal decision mechanism, and II, which is a global measure of emergent system performance.

We designed five test scenarios S_1, \dots, S_5 simulating actual environments that a collision imminent braking system would face in practice. These scenarios are illustrated in



Fig. 2: *Fixed Obstacle* scenario. The car moves toward a fixed obstacle 100 m ahead with initial velocity 20 m/s .



Fig. 3: *Emergency Braking Obstacle* Scenario. The car moves toward the other car at position 50 m ahead which is decelerating with 5 m/s^2 . The initial velocity for both cars is 20 m/s .

Figures 2–6. We also investigated two variants of all five scenarios, in which the maximum deceleration $a_{c_{max}}$ is held constant at two different values.

- 1) *Dry Pavement* (DP): $a_{c_{max}} = -5 \text{ m/s}^2$.
- 2) *Wet Pavement* (WP): $a_{c_{max}} = -3 \text{ m/s}^2$.

Note that the behavior of the obstacle and simulation constants are *not* known in advance by the car, and it must rather infer them through sensor readings.

We tested the α -thresholding policies with $\alpha \in [10, 100]$ with 10 as interval and 95 to 99 with 1 as interval. Both smoothed and unsmoothed variants are tested. Each policy is tested 10 times in each of the 10 scenarios in the set $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\} \times \{DP, WP\}$ described above using a Monte-Carlo simulation. Figure 7 depicts the results along with the ideal control at the origin. The smoothed 100-threshold is closest to ideal.

VI. INTERSECTION CROSSING

Intersection crossing requires consideration of both braking as well as acceleration in order to avoid crossing too slowly. It also requires considering the behavior of multiple obstacles which makes optimal decision boundaries more complex even in the known-state case. Here we present an analytical conservative computation of the safety of a given state $S(x)$ in the presence of multiple-lane intersection crossings. We are currently performing experiments applying this technique under uncertainty using the construction of (1).

A. Constraints in the Path-Time Space

The state is defined as the time t , arc-length parameterized position p , and tangential velocity v along a known path, which can be straight as in street crossings or curved as in unprotected lefthand turns. Acceleration and deceleration is assumed bounded. The car begins at state (p_t, v_t) and ends a maneuver at final position p_T with a range of admissible final velocities $[\underline{v}_T, \overline{v}_T]$.

The planning problem is to find a trajectory between (p_t, v_t) and the goal region that respects acceleration and deceleration bounds and also avoids obstacles. The obstacle avoidance constraint for obstacle O_i , $i = 1, \dots, n$ can be

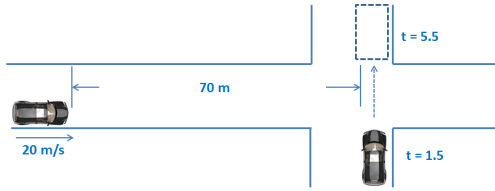


Fig. 4: *Transient Object* scenario. At first the car is at position 0 and moving with no obstacles detected. After 1.5 seconds, the car will detect an obstacle ahead at position 70. The obstacle will appear in the radar for 4 seconds and then disappear.

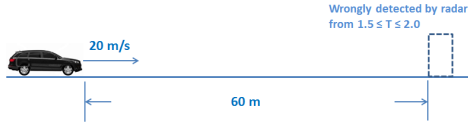


Fig. 5: *False Positive* scenario. The car starts at 20m/s with no obstacles detected. After 1.5 seconds, the car will wrongly detect an obstacle at position 60. This false positive period lasts for 0.5 seconds.

geometrically constructed as a forbidden region PO in the path-time space (p, t) by examining the set of (p, t) points that would cause the vehicle to overlap O_i [3]. This requires that the length and width of O_i are known and that future behavior of O_i is known. We currently assume that each O_i travels along a given path with confidence intervals of velocity and acceleration, which is assumed constant.

For each obstacle O we construct conservative rectangular forbidden regions $[p_O, \bar{p}_O] \times [t_O, \bar{t}_O]$ in the (p, t) plane. The forbidden interval $[t_O, \bar{t}_O]$ is constructed by examining the minimum and maximum extents of the vehicle along O 's path, and examining the time that O occupies these extents plus a minimum time-to-collision margin [11]. The interval $[p_O, \bar{p}_O]$ is based upon the vehicle dimensions plus certain comfort margins on either side.

B. Analytical Planning with Piecewise Constant Accelerations

Given this construction of forbidden regions, we extend the path-time planning approach of [6] to consider acceleration constraints. This requires planning in the path-velocity-time space, and it is useful to note that any optimal trajectory will either connect directly to the goal state, or pass tangentially along either the upper-left or lower-right corner of one or more obstacles [8]. Based on this observation we consider searching among the set of piecewise constant acceleration controls with discontinuities at lower-right and upper-left obstacle corners.

The search explores a graph G where each node is associated with set in the (p, v, t) space. Typically these sets are specified as an interval of reachable velocities R at a point in the (p, t) plane, except the initial node is specified as a point (p_t, v_t, t) , while the goal region has specified position, unspecified time, and a range of admissible velocities. Each

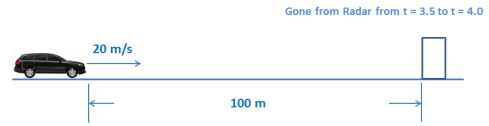


Fig. 6: *False Negative* scenario. The car starts at 20m/s, and a fixed obstacle is 100m away. After 3.5s, the car loses track of the obstacle for 0.5s seconds and then finds the obstacle again.

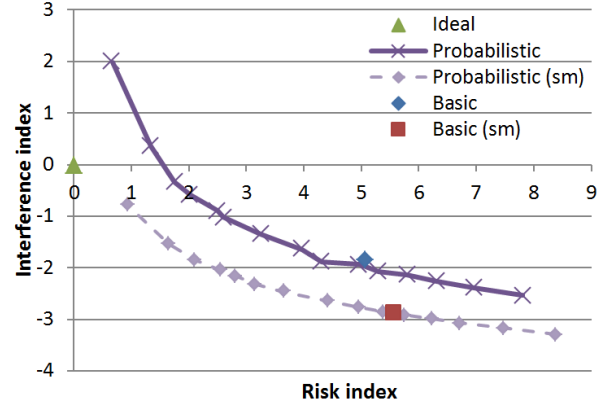


Fig. 7: Performance of smoothed and unsmoothed variant of α -thresholding policies with $\alpha \in [10, 100]$ with 10 as interval. Risk index and interference index exhibit an inverse relationship as α increases. The triangle dot (origin) represent the ideal control.

expansion step checks whether the node can be connected with a dynamically feasible, obstacle free trajectory to a vertex (p', t') of an obstacle region (either the upper left or lower right vertex) or to the goal region. If so, each connected component of the set of reachable velocities is instantiated as a new node.

The planner makes heavy use of the NextReachableSet subroutine to propagate the set of reachable velocities, excluding obstacle avoidance constraints, from one node to another. A variant is used to connect nodes to the goal region.

Algorithm 2 Compute the dynamically reachable set of velocities R_{i+1} after a change in position $\Delta p, \Delta t$ from an initial point with velocities in interval R_i .

```

NextReachableSet( $\Delta p, \Delta t, R_i$ )
   $A = [\frac{\Delta p - (1/2)a_{max}\Delta t^2}{\Delta t}, \frac{\Delta p - (1/2)a_{min}\Delta t^2}{\Delta t}] \cap R_i$ 
   $B = [2\frac{\Delta p}{\Delta t} - upper(A), 2\frac{\Delta p}{\Delta t} - lower(A)]$ 
  return  $R_{i+1} = A \cap [0, inf)$ 

```

The recursive depth first search is given in Algorithm 3. The algorithm first checks for a connection to the goal region. If successful, the range of reachable velocities at the goal is returned. The recursion can then be terminated, or continued to collect all trajectories that reach the goal. Next, it examines all subsequent trajectories from the node through obstacle vertices.

Of the dynamically feasible trajectories computed by Nex-

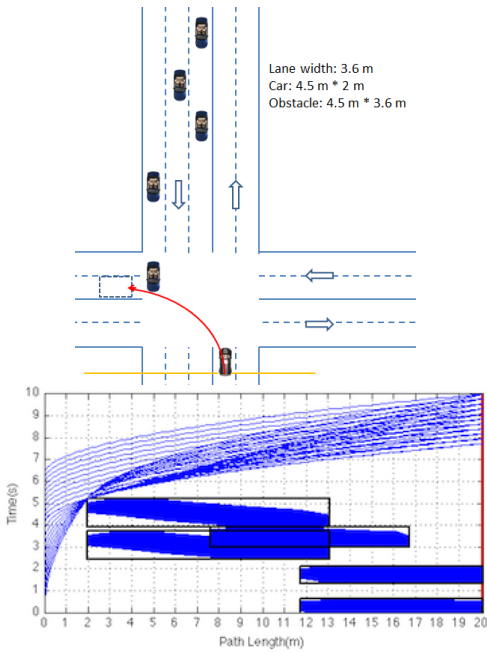


Fig. 8: **Above:** A lefthand turn scenario. The red line is the car's path with reference point at the middle of its rear axis. The orange line is considered a reference line. The distances of obstacles to the line are 13 m, 28 m, 44 m, 37 m, 52 m. All obstacles are moving at 10 m/s. **Below:** Feasible trajectories that arrive within 10 s. Rectangles are path-time obstacles.

tReachableSet , it calls the $\text{CollisionCheckRegion}$ subroutine to partition the range of reachable velocities into a set of disjoint intervals \mathbf{R} . (It is possible for small (p, t) obstacles to partition the trajectories into multiple components). This test is performed by examining where the (p, t) trajectories overlap each obstacle as the acceleration is swept between a_{min} and a_{max} .

Algorithm 3 Perform a recursive depth first search for a feasible path from any state (p, t, v) where $v \in R_i$.

ExpandNode(p, t, R_i)

If $\exists v \in R_i$ s.t. (p, t, v) can be connected to the goal, output the path leading to the goal.

for all (p', t') corners of O_1, \dots, O_n **do**
 $\hat{R} \leftarrow \text{NextReachableSet}(p' - p, t' - t, R_i)$
 $\mathbf{R} \leftarrow \text{CollisionCheckRegion}(p, t', R_i, \hat{R})$

for all $R \in \mathbf{R}$ **do**
 $V \leftarrow V \cup N$, with $N = (p', t', R)$
 ExpandNode(p', t', R)

end for
end for

Since $p' > p$ and $t' > t$, the recursion necessarily terminates in at most n steps. Further, it can be shown that the graph contains no more than $2n+2$ vertices, and hence a naive implementation of the search runs in $O(n^3)$ time and $O(n^2)$ space. The assumption that discontinuities in control only occur at corner points simplifies the problem, but may

lead the algorithm to failure in a cluttered space that requires carefully chosen acceleration and braking. We believe that these situations rarely arise in practice, and plan to address them in future work. We are currently working to integrate our probabilistic decision thresholds with this algorithm.

VII. CONCLUSION

We present a generic framework for conducting semi-autonomous collision avoidance and two concrete implementations that can be used under that framework. First, a probabilistic based collision-avoidance braking strategy in terms of their behavior in the presence of uncertainty in vehicle dynamics, sensor noise, and unpredictable obstacle behavior. A number of Monte-Carlo simulations demonstrate that this probabilistic braking strategy can achieve good performance in different testing scenarios. Second, a technique for determining safe trajectories for unprotected lefthand turns, which was shown to be time-optimal under the given constraints.

ACKNOWLEDGEMENT

This work is partially supported by the Indiana University Collaborative Research Grant fund of the Office of the Vice President for Research.

REFERENCES

- [1] M. Althoff, O. Stursberg, and M. Buss. Model-based probabilistic collision detection in autonomous driving. In *Intelligent Transportation Systems*, 2009.
- [2] S. Anderson, S. Peters, K. Iagnemma, and T. Pilutti. A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2009.
- [3] T. Fraichard. Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach, 1993.
- [4] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. In *Transactions on Signal Processing*, 2002.
- [5] J. Hillenbrand, A. Spieker, and K. Kroschel. Efficient decision making for a multi-level collision mitigation system. In *Intelligent Vehicles Symposium*, 2006.
- [6] K. Kant and S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *Int. J. Rob. Res.*, 5:72–89, September 1986.
- [7] R. Karlsson, J. Jansson, and F. Gustafsson. Model-based statistical tracking and decision making for collision avoidance application. In *American Control Conference*, 2004.
- [8] Y.-H. Liu and S. Arimoto. Path planning using a tangent graph for mobile robots among polygonal and curved obstacles. *Int. J. Rob. Res.*, 11:376–382, August 1992.
- [9] National Highway and Traffic Safety Administration. Fatality analysis reporting system general estimates system: 2008 data summary. national highway and traffic safety administration report no.: Dot hs 811 171.
- [10] New York Times. Google cars drive themselves, in traffic, October 9 2010.
- [11] B. Sultan and M. McDonald. Assessing the safety benefit of automatic collision avoidance systems. In *International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2003.
- [12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. S. C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge. *J. Robot. Syst.*, 23(9):661–692, 2006.
- [13] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, 2006.