The SmartTer for ELROB 2006 – a Vehicle for Fully Autonomous Navigation and Mapping in Outdoor Environments

Pierre Lamon[1], Sascha Kolski[1], Rudolph Triebel[2], Roland Siegwart[1], Wolfram Burgard[2]

http://www.smart-team.ch/

[1] Autonomous Systems Lab, Ecole polytechnique Fédérale de Lausanne, Switzerland
firstname.name@epfl.ch

[2]Autonomous Intelligent Systems, Albert-Ludwigs-University of Freiburg, Germany

Abstract

The goal of our participation ELROB 2006 is to show fully autonomous navigation and 3D mapping in outdoor setting. Our robots is based on a standard Smart car that has been equipped with five distance laser sensors, three cameras, a differential GPS and an Inertial Measurement Unit (IMU) and four computers. The car's systems states are directly accessed through the can bus. *Localization* and *Navigation* is realized by fusing all available sensory information in a probabilistic way. This enables us for high precision localization and dynamic local and global path planning. Using the 3D point clouds extracted from the rotating lasers and the omnidirectional image the smartTer is able to consistently register the *3D Maps* and *Analyze the Scene* for regions of interest.

1 Team presentation

The core of the SmartTeam is composed of around a dozen of researchers from EPFL's Autonomous Systems Lab (ASL) and ALU-FR's Autonomous Intelligent Systems Lab (AIS). Our team is technically and financially supported by RUAG Land Systems, Singleton Technology, Bluebotics, Carnegie Mellon University and the two European Projects SPARC and BACS. It is structured in four main areas concerned with Systems Design, Navigation, 3D Mapping and Scene Analysis.

Team Leader:				
	Roland	Siegwart	ASL-EPFL / ETHZ	(team leader, director ASL)
	Wolfram	Burgard	AIS-ALU-FR	(co-team leader, director AIS)
Technical Team Leader:				
	Pierre	Lamon	ASL-EPFL / ETHZ	(technical project leader)
Core Team:				
	Sascha	Kolski	ASL-EPFL	(Navigation / Systems Design)
	Frederic	Pont	ASL-EPFL	(Computing / Systems Design)
	Cyrill	Stachniss	AIS-ALU-FR	(Scene Analysis /3D Mapping)
	Rudolph	Triebel	AIS-ALU-FR	(3D Mapping / Scene Analysis)
	Luciano	Spinello	ASL-EPFL	(3D Mapping / Scene Analysis)
	Kristijan	Macek	ASL-EPFL	(Navigation)
	Felix	Mayer	ASL-EPFL	(Systems Design / Navigation)
	François	Pomerleau	ASL-EPFL	(Systems Design / Navigation)
	Patrick	Pfaff	AIS-ALU-FR	(3D Mapping)

Other Key Members:

Tarek	Baaboura	ASL-EPFL	(Systems Design)
Davide	Scaramuzza	ASL-EPFL	(3D Mapping / Scene Analysis)
Marcelo	Becker	ASL-EPFL	(Systems Design / Navigation)
Dave	Ferguson	RI - CMU	(Navigation)
Stefan	Gaechter	ASL-EPFL	(Scene Analysis)
Giorgio	Grisetti	AIS-ALU-FR	(Scene Analysis /3 D Mapping)
Bjoern	Jensen	Singleton Technology	(3D Mapping)
Beat	Ott	RUAG Land Systems	(Systems Design)
Hanspeter	Kaufmann	RUAG Land Systems	(Systems Design)
Viet Tuan	Nguyen	ASL-EPFL	(Scene Analysis)
Christian	Plagemann	AIS-ALU-FR	(Scene Analysis /3D Mapping)
Ferdi	Sessiz	RUAG Land Systems	(Systems Design)
Jan	Weingarten	ASL-EPFL	(3D Mapping)
Oscar	Martinez	AIS-ALU-FR	(3D Mapping)
Alex	Rottmann	AIS-ALU-FR	(3D Mapping)
Gregoire	Terrien	BlueBotics	(Systems Design)
Gernot	Spiegelberg	SPARC	(Systems Design)
Armin	Sulzmann	SPARC	(Systems Design)

ASL-EPFL, Autonomous Systems Lab, Ecole Polytechnique Fédérale de Lausanne, Switzerland

AIS-ALU-FR, Autonomous Intelligent Systems, Albert-Ludwigs-University of Freiburg, Germany

ASL-ETH Zurich, Autonomous Systems Lab, Swiss Federal Institute of Technology Zurich, Switzerland

RI - CMU (Navigation)**,** Robotics Institute, Carnegie Mellon University, USA

RUAG Land Systems (Vehicle Design, Testing), Thun, Switzerland

Singleton Technology Sarl (3D Mapping), Lausanne, Switzerland

Bluebotics (Support), Lausanne, Switzerland

SPARC – European Project (Vehicle, Navigation), Leader: DaimlerChrysler AG, Germany) - http://www.sparc-eu.net

BACS – European Project (Localization, Navigation, Scene Analysis)**,** Leader: ETH Zurich, - http://www.bacs.ethz.ch/

2 Vehicle Description

Our vehicle, called SmartTer (Smart all Terrain), is based on a standard Smart car that has been enhanced for fully autonomous driving in somewhat flat outdoor environments. The model is a smart fortwo coupé passion of year 2005 equipped with a 45 kW engine. This model has been chosen because it is compact, light, has power steering, auto gear shift, and a CAN bus that is easily accessible. All these features facilitate the process of converting the vehicle for autonomous driving.

1.1 Vehicle modifications

The next sections describe the main modifications that have been performed on the car.

Tires

Tires with better grip and larger diameters have been mounted. This allows for slightly higher ground clearance and much better traction in rough terrain.

Fig 1: Side and front view of SmartTer

Power generator

A 24V power generator has been mounted in order to power all the electronic devices and additional actuators. The generator is driven by a belt and pulley that is directly connected to the engine output axis. Two batteries placed in the trunk act as an energy buffer. They have a total capacity of 48Ah and are continuously recharged when the engine is running.

Power steering

Originally the power steering system minimizes the torque that has to be supplied by the driver by applying an additional torque M_{add} on the direction column. This task is fulfilled by the Driving Assistance unit (DA). The gain of the controller is set based on the car's current velocity, which is broadcasted on the Vehicle's CAN bus.

A specific electronic board has been designed in order to use the power steering motor for "drive by wire" (see Fig 2). To fulfil this goal, the vehicle's CAN-bus is disconnected from the DA and routed to a computer so that the steering angle is accessible to our controller. An additional CAN bus is connected to the Driving Assistance unit in order to supply it with the minimal required set of messages i.e. engine and car velocity messages. The same bus is used to command a CAN to analog module which fakes the torque voltage needed by the Driving Assistance unit. Finally the steering angle is controlled with a PID controller which minimizes the steering angle error. A switch enables the selection of manual or controlled mode.

Break

A system of cable and pulleys is used to activate the break pedal. The motor that pulls the cable is placed under the driver seat and is commanded by the CAN computer.

Gas pedal

A specific electronic system has been designed to enable the use of a computer to set the gas command. The voltage, originally provided by the potentiometer in the gas pedal, is simply

generated by a CAN to analog device. This device receives commands from the CAN computer. A switch enables to select the normal or controlled mode

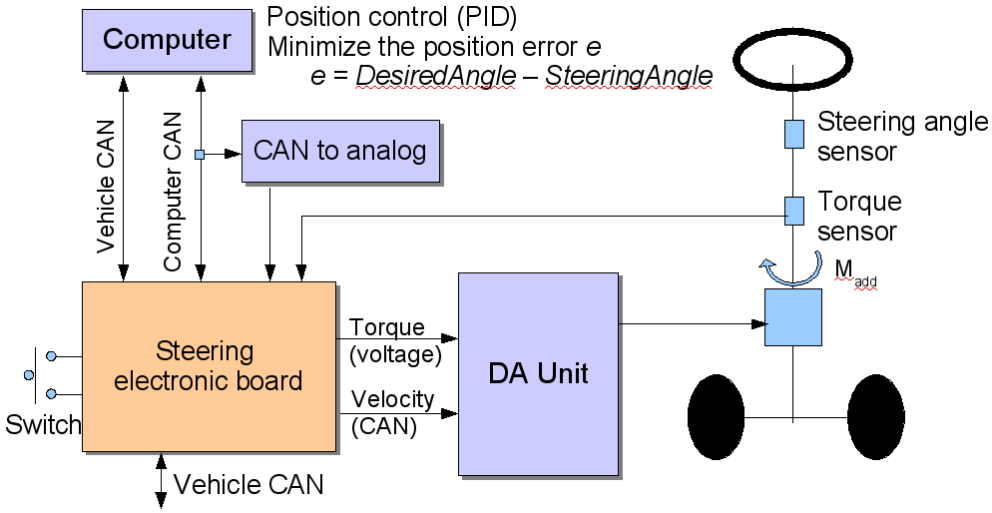

Fig 2: Steer by wire system

1.2 Sensing

The car is equipped with the following sensors:

- ***three navigation SICK*** laser scanner sensors (Sick LMS291-S05, outdoor version, rain proof): They are used mainly for obstacle avoidance and local navigation. One Sick is placed at the lower front and two on the roof, looking slightly to the sides. This configuration enables a large field of view. They are depicted in Fig. 3 and 4.
- ***rotating scanner (3D SICK)***: This is a custom made sensor that is mounted on the roof (see Fig 4). It consists of two Sick LMS291-S05 rotating around a vertical axis. Each second, a full 3D scan of the environment around the vehicle is acquired. This data is mainly used to compute a consistent 3D digital terrain model of the environment.
- ***omnidirectional camera*** (Sony XCD-SX910CR, focal length 12mm, with parabolic mirror): The setup is mounted in front of the rotating scanner. It enables the acquisition of panoramic images that are used to supply texture information for the 3D digital terrain maps. The images are also exploited to detect artificial object in the scene.
- ***monocular camera*** (Sony XCD-SX910CR, focal length 4.2mm): This camera is placed in the car, behind the wind shield. Like the omnidirectional camera, it is used to detect artificial object in the scene.
- ***life video streaming*** (Sony camera): For life video streaming from the car, with a transmission range up to two km.
- ***inertial measurement unit*** (Crossbow NAV420, waterproof): This unit contains 3 accelerometers, 3 gyroscopes, a 3D magnetic field sensor and a GPS receiver. The

embedded digital signal processor of the unit combines these information to provide the filtered attitude of the vehicle i.e. roll, pitch and heading to true north.

- ***differential GPS system*** (Omnistar Furgo 8300HP, rain proof antenna): This system provides an accurate position estimate together with its standard deviation when DGPS satellites are visible from the car. When no correction is available standard GPS is provided.
- ***internal car state sensors***: Accessible through the Vehicle CAN bus
    - *vehicle flags*: engine on, door closed, break pedal pressed etc.
    - *engine data*: engine rpm, instantaneous torque, gear shift, temperature etc.
    - *odometry*: global vehicle speed, individual wheel speeds, ABS activated
    - *gas pedal value*
    - *steering wheel angle*

Fig 3: Navigation SICK (left and right)

Fig 4: Omnidirectional camera and rotating sick

3 Autonomous Operation and Mission Concept

The SmartTer is intended to establish a consistent 3D digital terrain map (using the 3D laser and omnicam images) and detect artificial objects in the scene while autonomously driving along a specified route. The main tasks for accomplishing this mission are: ***Localization, Planning and Motion Control***, ***3D Mapping*** and ***Scene analysis***.

1.3 Processing

The processing architecture consists of four compact PCI computer racks communicating trough a gigabit Ethernet link. All the racks have the same core architecture i.e. Pentium M @ 2GHz, 1.5 Gbyte of RAM, Gigabit and Fast Ethernet, 2 serial ports and a 30 Gbyte hard disk. Each rack is dedicated to specific tasks and acquires measurement from different sensors as it is depicted in Fig 5.

The *Vehicle* rack is endowed with a firewire (IEEE 1394) board and a CAN interface that are used to respectively communicate with a digital camera and the vehicle CAN buses. On the other hand, the inertial measurement unit and the DGPS are connected to the rack through

standard serial ports. The main tasks of the Vehicle rack are to keep track of the vehicle position and to control its motion (steering, breaking, velocity control).

The *Navigation* rack acquires data from three laser range sensors using high speed RS422 serial ports. The main task of this computer is to plan a safe path to the goal using the sensor measurements and to provide the motion commands to the Vehicle rack.

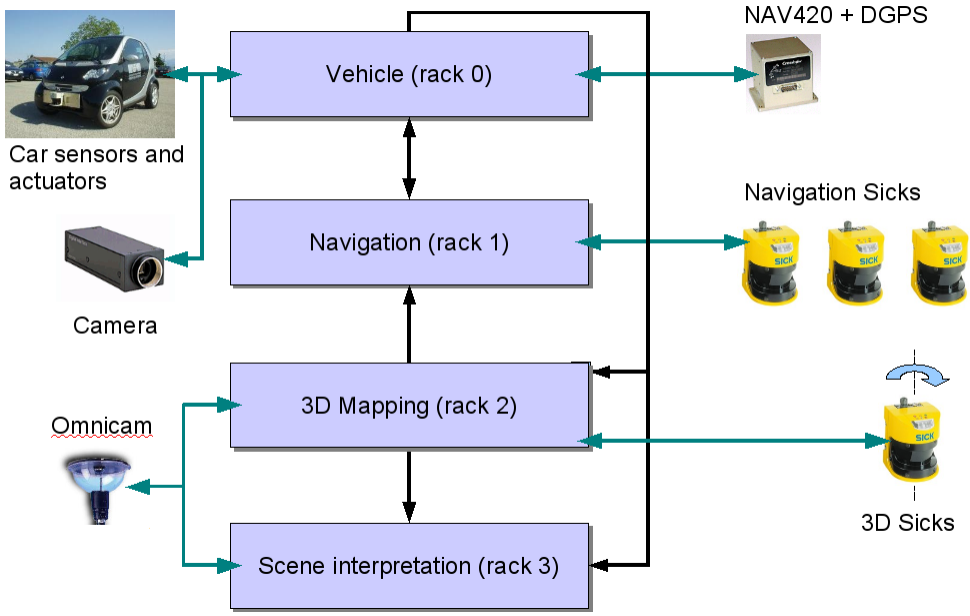

Fig 5: Processing and sensing architecture

A 3D map of the traversed environment is updated on the *3D Mapping* rack using the measurements acquired by the rotating SICK sensor. For more realistic rendering, the texture information acquired by the omnicam is mapped to the 3D model as it is depicted in figures 6 and 7.

Finally, the scene analysis is performed on the *Scene interpretation* rack. The artificial objects are extracted from the textured 3D map and their representation is stored in memory as the vehicle moves along the path.

1.4 Software architecture

GenoM [Genom] and Carmen [Carmen] are the two main robotic software architectures used in the framework of this project. The different functional modules that run on different computers exchange data using the Inter Process Communication [IPC] and the high bandwidth Gigabit Ethernet. The architecture has been designed in such a way it minimizes the amount of data to be transmitted. The modular architecture together with the IPC based communication enables the system to be flexible and reconfigurable.

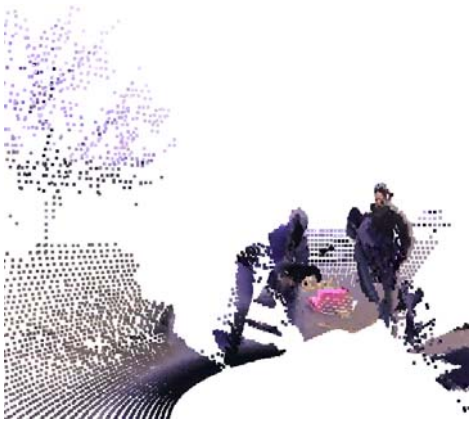

Fig 6: 3D scan with texture

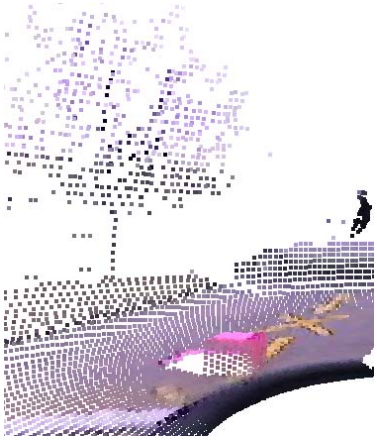

Fig 7: A pink box and street markings

1.5 Localization

The proposed localization algorithm is based on the information form of the Kalman filter i.e. the Information filter. This filter has the property of summing information contributions from different sources in the update stage. This characteristic is very interesting when many sensors are involved, which is the case in our system.

To accurately localize the vehicle, three different sensors are used: GPS (Omnistar 8300HP), inertial measurement unit (NAV420) and vehicle state sensors (wheel encoders). The combination of their measurements allows the estimation of the vehicle's 6 degrees of freedom i.e. the 3 coordinates and the attitude [Lamon 05] [Dissanayake 01] [Sukkarieh 99]. Here is a list of conditions that affects the quality of the measurements:

- the vehicle goes on a covered area, underground or in a tunnel: the GPS signal is not available
- the vehicle drives next to large structures: GPS reflexion can occur, this is the worst case because GPS is available but provides wrong measurements
- when traversing rough terrain: the wheel encoders provides wrong measurements when wheels are slipping
- the vehicle drives next to iron structures: the earth magnetic field is locally distorted and the compass provides erroneous measurements

Thus, each sensor has its own advantages and drawbacks depending on the situation and sensor fusion is crucial to compute robust position estimates.

In Fig 8, the raw GPS data is plotted to illustrate the difficulty to handle noisy and sometimes erroneous data. One can see that the DGPS signal is not always available along the path. The outages occur mainly when the vehicle drives close to the buildings or underground. On the other hand, the standard GPS is available more often because it does not rely on the visibility of geostationary satellites providing the DGPS correction. Indeed, these communication satellites are low above the horizon at our latitude and are often occluded. However, standard GPS is much more noisy and less accurate than DGPS. The filtered trajectory combining the vehicle model, the inertial measurement unit, the odometry and the GPS is depicted in Fig 9. The plot in Fig 10 depicts the evolution of the standard deviation of the position over time. The uncertainty increases when GPS is not available because the system relies only on dead

reckoning. On the other hand, when GPS is available the position is reset and the uncertainty decreases.

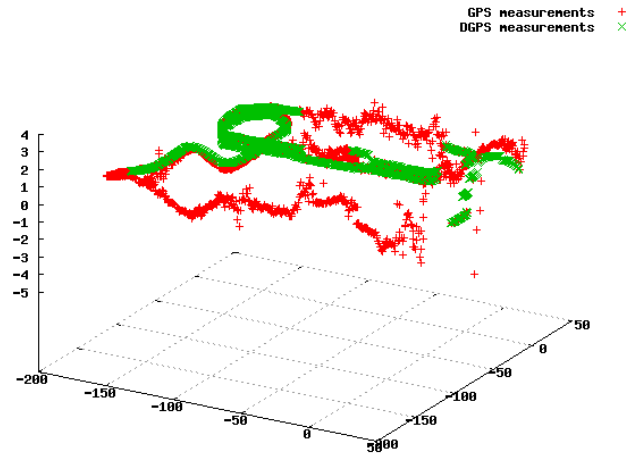

Fig 8: Raw GPS measurements

Fig 9: Overlay of the filtered trajectory on the ortho photo of the EPFL campus.

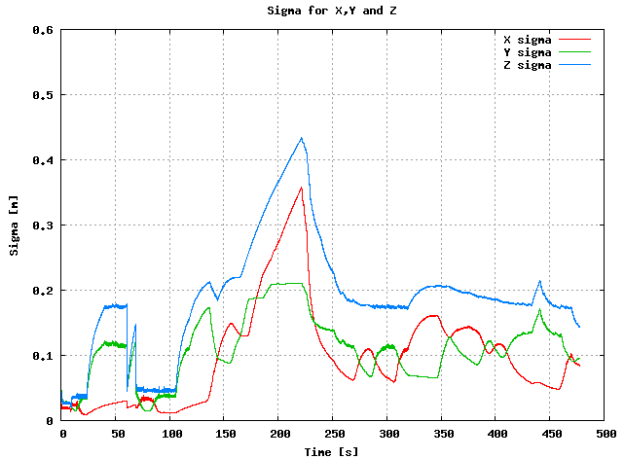

Fig 10: Standard deviation of the filtered position

1.6 Planning and Motion Control

Our approach to autonomous navigation combines a global planner with a local planning capability.

Local planning

We use the information from our laser range finders to construct a local grid-based cost map specifying the nearby obstacles and difficult areas to traverse for the vehicle. Each cell in the grid is a square of width 20 cm. Cells containing obstacles are assigned an infinite cost, representing untraversable areas, with cells corresponding to less difficult terrain assigned less-expensive cost values. We perform a configuration space expansion on this map, which has the effect of 'growing' the obstacles and other expensive areas out by the width of the vehicle. This allows us to treat the vehicle as a single point during planning. Given this local

map and the current vehicle position and orientation within this map, we can then project potential vehicle actions onto the map and check the cost of these actions. We use a discrete set of variable-length arcs for our vehicle actions, corresponding to different steering angles and vehicle speeds [Kolski 06] [Bares 89]. Each of these arcs represents an action that is feasible from the current vehicle position, orientation, and velocity. We then choose the best of these arcs according to their costs and perhaps also some general objective, such as the amount of distance the arc takes us in our desired direction of travel. This arc can then be directly executed by the vehicle.

Global planning

Our global planner is based on the Field D* algorithm, which has been incorporated into several fielded robotic systems [Singh 00] [Ferguson 05]. This algorithm provides very low-cost paths through grid-based representations of an environment. These paths do not take into account the heading restrictions of the vehicle and instead approximate the least-cost path to the goal for a vehicle that can turn in place. Because Field D* does not encode the mobility constraints of the vehicle, it cannot be used alone for accurate trajectory planning for the vehicle. Consequently, we combine it with our local, vehicle-specific arc-based planner to provide feasible paths. Our combined system maintains a global map of the environment containing all the observed obstacles and high-cost areas. Then, every planning cycle, the vehicle projects out its set of available arcs into this map and computes the cost of each arc based on its distance and the cost of the cells it travels through. This gives the cost of traversing the arc itself. To this value we then add the cost of a global path from the end of the arc to the goal. This cost is provided by our global Field D* planner. Then, the arc with the minimum combined cost is selected and executed by the vehicle.

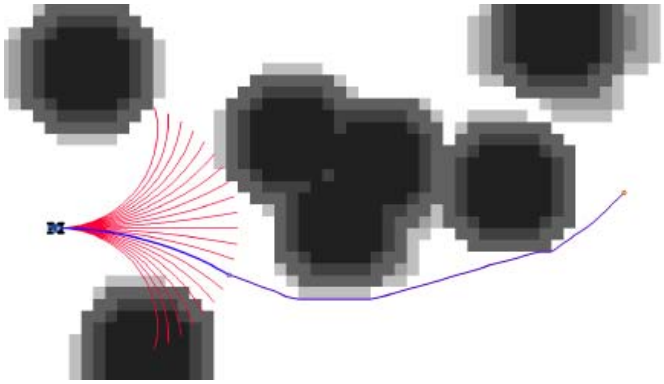

Fig 11: Traversability map with optimal arc selection

Fig 11 shows an illustrative example of this combined approach. The set of available arcs are shown in red/gray, with the best arc shown in blue/black. Here, the best arc was selected based on a combination of the cost of the arc itself and the cost of a global path from the end of the arc to the goal (the goal is shown as a filled circle at the right of the figure). The global path from the end of the best arc to the goal is also shown in blue/black. In this example, a purely local planner would have selected the straight arc leading directly to the right, as this brings it closest to the goal in terms of straight line distance. However, such an arc could cause it to get stuck behind the clump of obstacles in the middle of the map.

Longitudinal velocity controller

The design of a velocity controller for a vehicle driving in rough terrain is not trivial. Indeed, the controller has to deal with large non-linearities due to time varying parameters such as

gear shifting, slopes, soil parameters, delays and engine performance. Given these non-linearities and unknown parameters, it turned out that a Fuzzy Logic (FL) controller performs better and more reliability than a standard PID controller.

1.7 3D Mapping

The 3D Mapping unit creates consistent global 3D point clouds using the inputs of the rotating SICKs and the localization module. In parallel, a local traversability map is created on-line that is used for local planning.

We use multi-level probabilistic surface map (MLS maps), which can be regarded as an extension to elevation maps [Bares 89] [Herbert 89] [Singh 96] [Parra 99] [Olson 00] [Lacroix 02]. This allows to compactly represent multiple surfaces in the environment as well as vertical structures. It enables a mobile robot to correctly deal with multiple traversable surfaces in the environment as they, for example, occur in the context of bridges. Our approach also represents an extension of the work by Pfaff et al. [Pfaff 05]. Whereas this approach allows dealing with vertical and overhanging objects in elevation maps, it lacks the ability to represent multiple surfaces.

A multi-level surface map (MLS map) consists of a 2D grid of variable size where each cell c_{ii}, $i, j \in Z$, in the grid stores a list of surface patches $P_{ii}^1, \ldots, P_{ii}^k$. A surface patch in this context is represented as the mean μ_{ii}^k and variance σ_{ii}^k of the measured heights at the position of the cell c_{ii} in the map. Each surface patch in a cell reflects the possibility of traversing the 3D environment at the height given by the mean μ_{ii}^k, while the uncertainty of this height is represented by the variance σ_{ii}^k. We assume that the error in the height underlies a Gaussian distribution, therefore we will use the terms surface patch and Gaussian in a cell interchangeably.

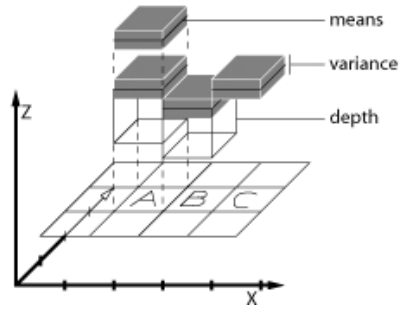

Fig 12: Example of different cells in an MLS Map. Cells can have many surface patches (cell A), represented by the mean and the variance of the measured height. Each surface patch can have a depth, like the patch in cell B. Flat objects are represented by patches with depth 0, as shown by the patch in cell C.

In addition to the mean and variance of a surface patch, we also store a depth value d for each patch. This depth value reflects the fact that a surface patch can be on top of a vertical object like a building, bridge or ramp. In these cases, the depth is defined by the difference of the height h_{ii}^k of the surface patch and the height $h_{ii}'^k$ of the lowest measurement that is considered to belong to the vertical object. For flat objects like the floor, the depth is 0. Figures 13 and 14 depict two examples of the map cells in an MLS map.

In order to register single MLS maps into one global map we first extract features from the local maps and then apply map matching based on the iterative closest point algorithm (ICP) to these features (vertical objects, flat surface, rough terrain). To further improve the maps, we apply a loop-closing technique based on a constraint network whenever a loop is detected. At the current moment, loops are detected manually. The resulting MLS maps are especially useful to distinguish between traversable and non-traversable regions of the environment. In

the figures, green areas represent traversable map cells, whereas red and blue are non-traversable.

For further treatment of the environment information (e.g. scene analyses) the 3D point clouds are combined with the omnicam images to a colored point-cloud map.

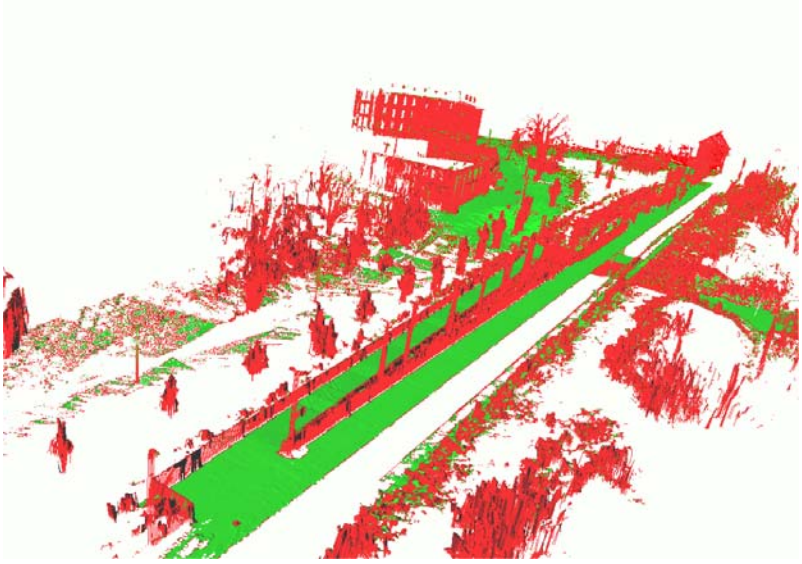

Fig 13: Resulting MLS map with a cell size of 10cm x 10cm. The area was scanned by a outdoor robot and spans approximately 299 by 147 meters. During the data acquisition, where the robot collected 172 scans consisting of 45,139,000 data points, the robot traversed a loop with a length of 560m.

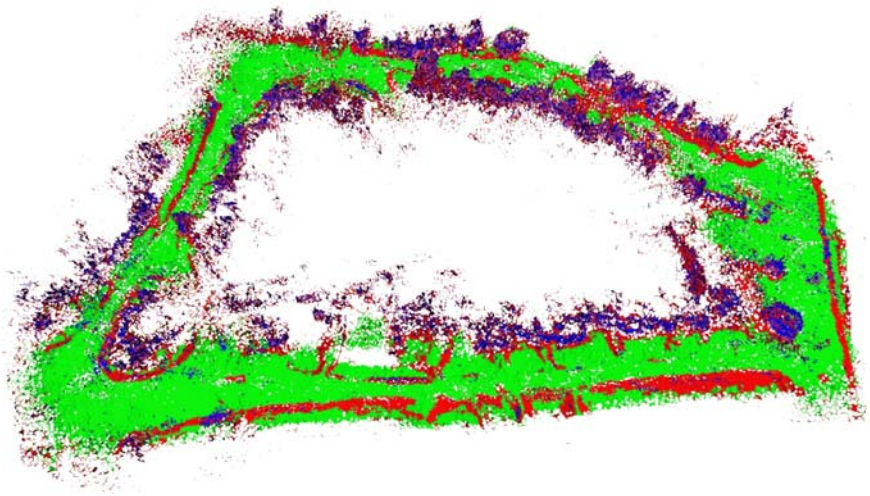

Fig 14: MLS map of an outdoor terrain. The data was collected with the smart car and a rotating laser device mounted on top of the car roof. The grid size in this example is 50cm.

1.8 Scene Analyses

The colored 3D point-cloud map is further used for scene analyses. This module is using the inherent properties of artificial objects in order to extract it from the environment map. The

question arising is what an artificial object looks like. Our definition considers an artificial object to be composed of a sufficiently smooth surface and a sufficiently extended area, which has distinctive color with respect to the surrounding world. The artificial object detection task is accomplished without human supervision or any trained object database through three steps, in a quasi real-time. In the first step (called laser layer), linked smooth surfaces with sufficient area extension are extracted from the 3D map (Fig. 15 and 16). In the second step (called camera layer), a complex image color segmentation is performed in order to identify useful color blobs (Fig. 17 and 18). Finally, in the third step, the laser and camera layers are fused together in a probabilistic and unsupervised way, where every instance has a given confidence of being an artificial object or not using a particular defined untrained neural network. In this way, it is possible to manage the uncertainty of false positives or the partial and noisy information coming from the two sensors. The results are shown in Fig. 19.

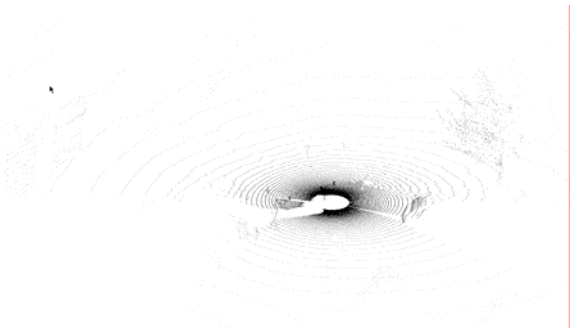

Fig. 15: Raw 3D scene

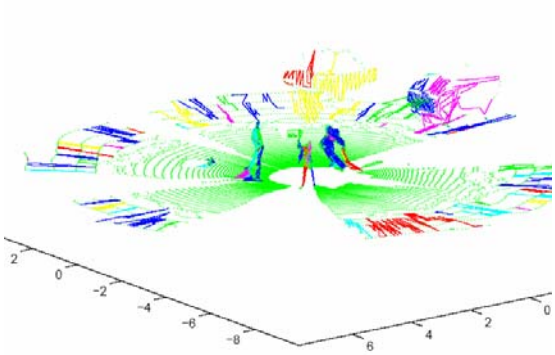

Fig. 16: 3D scene with smooth surface detected

Fig. 17: The omnidirectional picture of the road, with isolated region of interest

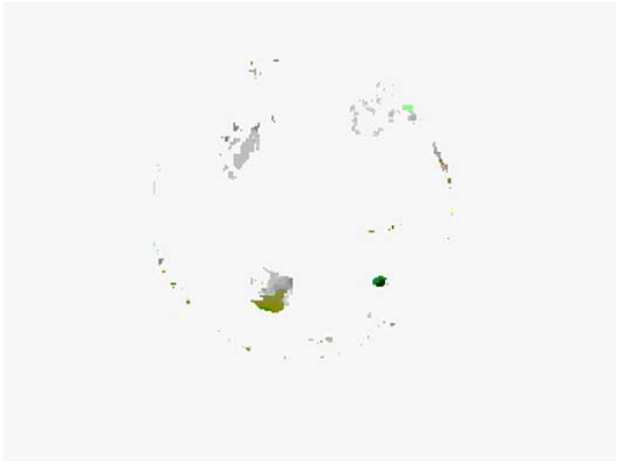

Fig. 18: Segmented omnidirectional picture

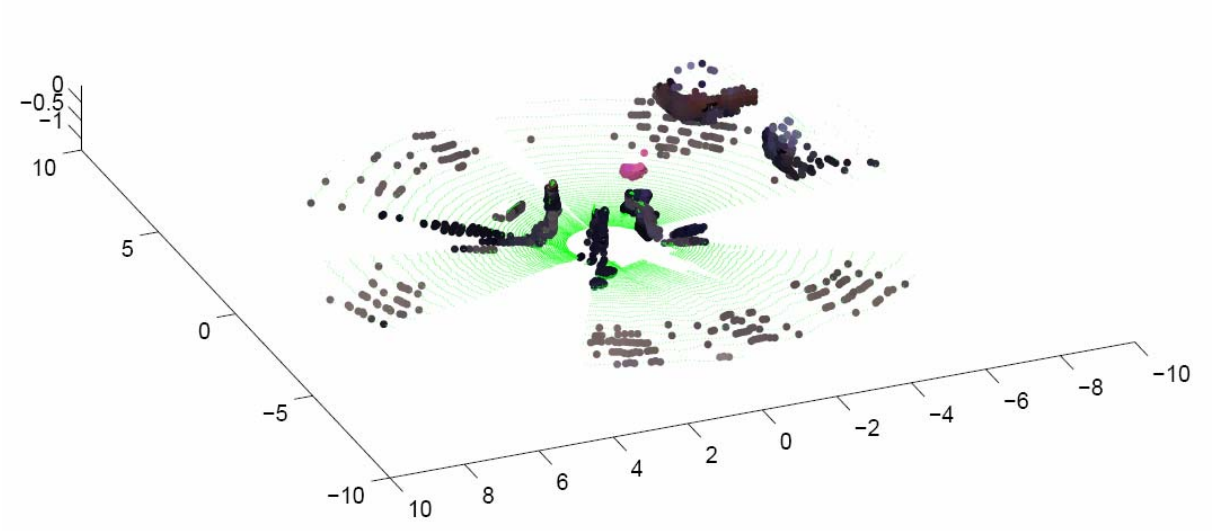

Fig. 19: The result of artificial object detection. The figure shows a 3D scan with the detected objects

References

[Bares 89] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, andW. R. L. Whittaker. Ambler: An autonomous rover for planetary exploration. IEEE Computer Society Press, 22(6):18.22, 1989.

[Dissanayake 01] G. Dissanayake, S. Sukkarieh, E. Nebot, H. Durrant-Whyte: "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications", IEEE Transactions on Robotics and Automation, 2001.

[Ferguson 05] D. Ferguson and A. Stentz, "Field D*: An Interpolation-based Path Planner and Replanner," in Proceedings of the International Symposium on Robotics Research (ISRR), 2005.

[Herbert 89] M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pages 997.1002, 1989.

[Kelly 95] A. Kelly, "An intelligent predictive control approach to the high speed cross country autonomous navigation problem", Ph.D. dissertation, Carnegie Mellon University, 1995.

[Kolski 06] S. Kolski, D. Ferguson, M. Bellino and R. Siegwart. Autonomous Driving in Structured and Unstructured Environments. IEEE Intelligent Vehicles Symposium, 2006.

[Lacroix 02] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury and; M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. International Journal of Robotics Research, 21(10-11):917.942, 2002.

[Lamon 05] P. Lamon and R. Siegwart (2005) 3D Position Tracking in Challenging Terrain. Proceedings of the Field and Service Robotics FSR 2005, August, 2005.

[Olson 00] C.F. Olson. Probabilistic self-localization for mobile robots. IEEE Transactions on Robotics and Automation, 16(1):55.66, 2000.

[Parra 99]C. Parra, R. Murrieta-Cid, M. Devy, and M. Briot. 3-d modelling and robot localization from visual and range data in natural scenes. In 1st International Conference on Computer Vision Systems (ICVS), number 1542 in LNCS, pages 450–468, 1999.

[Pfaff 05] Pfaff P. and Burgard W. An effcient extension of elevation maps for outdoor terrain mapping. In Proc. of the International Conference on Field and Service Robotics (FSR), pages 165.176, 2005.

[Philippsen 05] R. Philippsen and R. Siegwart (2005) An Interpolated Dynamic Navigation Function. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)

[Philippsen 06] R. Philippsen, B. Jensen. and R. Siegwart, (2006) Toward Online Probabilistic Path Replanning. Workshop on Autonomous Sensor-Based Motion for Complex Robots in Complex Environments.

[Sukkarieh 99] S. Sukkarieh, E.M. Nebot and H.F. Durrant-Whyte: “A high integrity IMU/GPS navigation loop for autonomous land vehicle applications”, IEEE Transactions on Robotics and Automation, Jun 1999.

[Singh 96] S. Singh and A. Kelly. Robot planning in the space of feasible actions: Two examples. In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 1996.

[Singh 00] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, K. Schwehr, “Recent Progress in Local and Global Traversability for Planetary Rovers”, IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000.

[Weingarten 05] J. Weingarten and R. Siegwart, (2005) EKF-based 3D SLAM for Structured Environment Reconstruction. In Proceedings of IROS, Edmonton, Canada, August 2-6, 2005.

[smart-team] http://www.smart-team.ch

[Genom] http://softs.laas.fr/openrobots/tools/genom.php

[Carmen] http://www.cs.cmu.edu/~carmen/links.html

[IPC] http://www.cs.cmu.edu/afs/cs/project/TCA/www/ipc/index.html