

Lighting-Invariant Adaptive Route Following Using ICP

Philipp Krüsi

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
philipp.kruesi@mavt.ethz.ch

Bastian Bücheler

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
bbuecheler@me.com

François Pomerleau

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
francois.pomerleau@mavt.ethz.ch

Ulrich Schwesinger

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
ulrich.schwesinger@mavt.ethz.ch

Roland Siegwart

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
rsiegwart@ethz.ch

Paul Furgale

Autonomous Systems Lab
ETH Zurich
8092 Zurich, Switzerland
paul.furgale@mavt.ethz.ch

Abstract

Topological/metric route following, also called teach and repeat (T&R), enables long-range autonomous navigation even without globally consistent localization. In the teach pass, the robot is driven manually and builds up a topological/metric map of the environment: a graph of metric submaps connected by relative transformations. For repeating the route autonomously, the map only needs to be locally consistent; errors on the global level due to localization drift are irrelevant. This renders T&R ideal for applications where a global positioning system may not be available, such as navigation through street canyons or forests in search and rescue, reconnaissance in underground structures, surveillance, or planetary exploration. We present a T&R system based on iterative closest point matching (ICP) using data from a spinning three-dimensional (3D) laser scanner. Our algorithm is highly accurate, robust to dynamic scenes and extreme changes in the environment, and independent of ambient lighting. It enables autonomous navigation along a taught path in both structured and unstructured environments, including highly 3D terrain. Furthermore, our system is able to detect obstacles and avoid them by adapting its path using a local motion planner. It enables autonomous route following in non-static environments, which is not possible with classical T&R systems. We demonstrate our algorithm's performance in two long-range driving experiments, one in a highly dynamic urban environment, the other in unstructured, rough, 3D terrain. In these experiments our robot autonomously drove a distance of over 22 kilometers in both day and night. We analyze the localization accuracy of our system and show that it is highly precise. Moreover, we compare our ICP-based method to a state-of-the-art stereo-vision-based technique and show that our approach has a greatly increased robustness to path deviations and is less dependent on environmental conditions.

1 Introduction

Reliable long-range autonomous navigation in difficult terrain and under harsh and variable environmental conditions is a requirement for many promising applications of mobile robots, such as search and rescue, outdoor surveillance, or NBC reconnaissance¹. To operate these challenging scenarios, a robot must be able to build its own map of the environment and to localize itself precisely and reliably, even in absence of a Global Positioning System (GPS), at day and night, and irrespective of changes and dynamic objects in the environment. Moreover, it has to reliably detect static and dynamic obstacles on its way, and to plan safe trajectories for avoiding collisions while remaining as close as possible to the desired route.

Topological/metric route following using cameras or lasers—sometimes called T&R—has been shown to enable long-range autonomy, especially in unstructured, 3D terrain where typical two-dimensional (2D) localization and mapping techniques fail. Previously published algorithms have relied on appearance-based sparse-feature pipelines for mapping and localization. These techniques require the sensor resolution to be high across the full field of view (FOV). This was one of the main lessons learned when building the appearance-based lidar system in (McManus et al., 2013b); when the resolution is too low, sparse feature pipelines like SURF (Bay et al., 2008) simply fail. To account for this, McManus et al. used laser intensity images of 480×360 pixels. As a consequence, appearance-based route-following systems that need to operate in real time often operate on a restricted FOV. This results in poor robustness to deviations from the initial path, as the area in which the system can successfully localize against the map is restricted to a narrow band where the previously mapped area is seen by the sensor. This can be problematic in dynamic environments, where the robot may have to leave the nominal path for avoiding obstacles.



Figure 1: Our robot ARTOR (Autonomous Rough Terrain Outdoor Robot) successfully drove autonomously over 22 kilometers in different environments using a spinning Velodyne HDL-32E to build and localize within a topological/metric map representation. This picture was taken during autonomous driving on a snowy night in Zurich as part of the long-range route following experiments described in Section 7.4.

In this paper, we develop and evaluate a system for autonomous navigation along a previously driven route for search and rescue or outdoor surveillance applications. Tasks suitable for T&R related to search and rescue include automatic return after a manually operated survey, muling between two locations for transportation of goods or people, or continuous monitoring of critical areas. Motivated by the shortcomings of existing T&R systems in terms of dependency on ambient light (vision-based approaches) and robustness to path deviations due to the restricted FOV (systems using a directed sensor), we propose to use an omnidirectional, active sensor, namely a spinning 3D lidar. Our sensor, a Velodyne HDL-32E, produces range and intensity

¹Reconnaissance for nuclear, biological and chemical hazards

images at 870×32 pixels, which is clearly too small for sparse appearance-based localization and mapping². Instead of using the laser intensity data, we develop a mapping and localization framework based on 3D range measurements. We show how iterative closest point (ICP) may be used to build all the components necessary for T&R, such as relative motion estimation, map building, and localization. Compared to state-of-the-art camera-based systems (Zhang and Kleeman, 2009; Furgale and Barfoot, 2010; Churchill and Newman, 2012; Cherubini and Chaumette, 2013), our algorithm is more robust to changes in lighting and environmental conditions, and it has a much higher robustness to path deviations. While many state-of-the-art laser-based systems are constrained to road-based navigation in 2D (Levinson and Thrun, 2010; Baldwin and Newman, 2012b,a), we demonstrate that our algorithm also performs in highly 3D terrain. Similar route-following performance in 3D has been demonstrated by the appearance-based laser approach of McManus et al. (2013b). However, their system can only handle very small deviations from the initial path (± 1 m according to the authors), and the maps produced by our system are much more accurate, which can be of great benefit when higher-level autonomy systems are built on top (e.g. terrain assessment and path planning), or when maps must be shared with human operators.

Further, we develop a reliable obstacle detection algorithm that uses the 3D lidar data, and integrate a system-compliant local motion planner. These components enable the robot to smoothly drive around obstacles appearing on the route, deviating from the original path and exploiting the increased robustness to path deviations of our localization system. The result is an *adaptive* route following framework that allows for reliable and completely autonomous navigation even in non-static environments.

To demonstrate these claims, we conducted extensive field tests both in a dynamic urban environment and in unstructured, rough, 3D terrain. We assessed the accuracy as well as the robustness to path deviations of our approach in both settings using precise ground truth. We further investigated the robustness of our system in two long-range route following experiments, likewise in both urban and unstructured, non-planar environments. We show that our robot was able to drive autonomously over more than 22 km, in both day and night, in sunshine, rain, and snowfall. The long-range field tests in the urban environment prove that our system is able to reliably detect obstacles and locally adapt its path accordingly, enabling fully autonomous driving in dynamic environments. Furthermore, we conducted evaluation against a state-of-the-art T&R approach that uses stereo vision (Furgale and Barfoot, 2010). We show that our algorithm provides similar localization accuracy in the nominal scenario (sufficient ambient light, only small deviations from the initial path, low driving speed), but is more accurate at higher driving speeds, has a greatly increased robustness to path deviations, and a larger operating range in terms of environmental conditions.

The remainder of this article is organized as follows. Section 2 surveys related work. In Section 3 we introduce the general concept of T&R, and present an overview of our system. Section 4 explains the details of the ICP framework and the point cloud filters that are used to build the T&R pipeline described in Section 5. In Section 6 we introduce our obstacle avoidance scheme, which consists of an obstacle detection module and a local motion planner. Section 7 describes the field experiments, evaluating the accuracy, the robustness to path deviations, and the reliability of our approach. In Section 8 we discuss the comparison to stereo-vision-based T&R, the impact of noise and dynamic environments on ICP, and remaining challenges. We summarize and conclude in Section 9.

2 Related work

Many scalable route following algorithms have been published recently based either on cameras or lasers. They have been shown to be useful in a diverse number of situations where a robot must revisit places over and over such as planetary exploration (Stenning et al., 2012), or road vehicle localization (Stewart and Newman, 2012; Churchill and Newman, 2012; Baldwin and Newman, 2012b,a). At the core of all of these algorithms

²McManus et al. (2013a) conducted experiments with data from a Velodyne HDL-64E lidar, and concluded that the vertical resolution (64 laser beams in their case) is not sufficient for stable keypoint extraction. We assume that the same applies to our lidar, as its resolution is even lower (only 32 laser beams)

is the concept of topological/metric localization and mapping where accurate local metric environmental representations are topologically connected. We review the most relevant related work here. Throughout the review, we focus on the requirements dictated by such demanding tasks as search and rescue or outdoor surveillance: robustness to extreme lighting and environmental conditions, ability to handle deviations from the initial path, and suitability for operation in dynamic and changing scenes.

Camera-based approaches were originally developed for wheeled robots navigating on planar surfaces (Matsumoto et al., 1996; Zhang and Kleeman, 2009), until the work of Furgale and Barfoot (2010) showed how such a system could be used over long routes (up to 3 km routes and 32 km driven autonomously in total) in highly 3D environments. However, these approaches are known to be brittle in the face of lighting or environmental changes. Recent work has attempted to address these shortcomings to create a system suitable for lifelong localization and mapping by storing multiple experiences of the same place (Churchill and Newman, 2012). But, this approach uses a stereo camera and hence it still requires ambient lighting and has poor robustness to path deviations based on the camera’s FOV. Researchers at the French Institute for Research in Computer Science and Control (INRIA) have developed a camera-based system that allows a wheeled robot to follow an ordered set of key images along a route (Diosi et al., 2007; Šegvić et al., 2009). Their latest work, (Cherubini and Chaumette, 2013), added a planar laser range finder to detect and avoid obstacles not present when the route was learned. This work demonstrates the complexity that arises when route following and avoiding obstacles with a narrow FOV sensor. As their algorithm uses a single perspective camera and a visual homing control law, they absolutely must maintain a view of the key images even when avoiding obstacles. To this end, they mount the camera on a pan unit, control the pan angle during maneuvers, and reduce the robot speed to avoid getting lost. There have been some visual route following algorithms based on omnidirectional cameras (e.g. (Goedemé et al., 2007), (Zhang and Kleeman, 2009) and (Meilland et al., 2010)), but there is no evaluation of how they work in unstructured, 3D terrain and, as they use cameras, they are still sensitive to lighting changes.

A different class of approaches addresses navigation along existing structures in the environment, such as paths and trails (Rasmussen et al., 2012) or roads (Chang et al., 2012), rather than along previously driven routes. The latter is based on monocular vision and image segmentation to first detect the road and subsequently estimate its shape. Rasmussen et al. (2012) present a system using omnidirectional cameras and a tiltable laser scanner for detecting and navigating on small trails. Combining appearance and structural cues for path detection decreases the sensitivity to lighting conditions, yet cannot eliminate the dependency on ambient light.

Laser-based approaches are better suited to search and rescue tasks as they don’t require ambient light and exhibit robustness to environmental changes, while still providing sufficiently rich information for route modeling and localization. Push broom lasers (Baldwin and Newman, 2012b,a) and spinning lasers (Levinson and Thrun, 2010) have been shown to enable localization within a previously surveyed area. However, these approaches require offline processing after the initial survey phase. Moreover, the former relies on a survey conducted with different sensors than used for online localization, and the latter uses a GPS/IMU system in addition to the laser measurements. Our approach employs only one single sensor, a 3D lidar, and it does not require any offline processing. The system is immediately ready for autonomous navigation after a single manually controlled drive. It is thus well suited for search and rescue tasks, where time typically is a critical factor. The closest work to ours is the appearance-based lidar algorithm of (McManus et al., 2013b). They clearly show the ability of lidar to enable lighting-invariant navigation. However, their use of the sparse feature pipeline and the slow scanning rate reduce the algorithm’s robustness to path deviations and degrade its metric accuracy³. According to the article, they can only handle deviations of ± 1 m from the path, and the limited metric accuracy of their visual odometry (VO) prevents successful operation over distances of more than 3 m without localization against the map. A solution to the problem of metric accuracy has later been shown in (Anderson and Barfoot, 2013). In terms of map-based localization, however, the lack of robustness to path deviations persists, as it is inherent to the use of a directed sensor. In contrast, our system

³Although metric accuracy is not strictly necessary for route following (see (Furgale and Barfoot, 2010) for a discussion of this topic), it is just generally useful to robot autonomy as it can help to suggest loop closures, achieve planning goals, allow the robot to accurately drive through tight spaces, and so on.

maintains the lighting invariance while both increasing the robustness to path deviations and increasing the metric accuracy, making it ideal for the search and rescue application.

Building laser-based topological/metric maps that are sufficiently large to enable localization well beyond the robot’s path at the time of map recording has been addressed in many prior works (Thrun et al., 1998; Tomatis et al., 2003; Modayil et al., 2004; Bosse et al., 2004). The *Atlas* framework presented in (Bosse et al., 2004) is a hybrid topological/metric SLAM system for large-scale environments. Similar to the representation in our approach, its map consists of a graph, where vertices are local frames with attached metric maps, and edges represent transformations between adjacent frames. Map matching in a post processing step enables building globally consistent metric maps. (Marshall et al., 2008) shows the application of a similar map structure based on laser ranging to navigation of underground mining vehicles in a T&R framework. The *manifold maps* introduced in (Howard, 2004) are an alternative representation of topological/metric maps. Instead of constructing a graph of connected submaps, the map is augmented with an additional dimension. This solves the problem of inconsistent maps at crossovers due to localization drift: the additional dimension separates two places in the map that spuriously appear to be at the same metric location. All of these approaches have in common that they are tailored to planar environments, building 2D, often binary maps of the obstacles in the surroundings. This may be adequate for indoor navigation or driving on paved roads, but it is not sufficient for operating in unstructured, 3D terrain. Our ICP-based mapping and localization framework builds a topological/metric graph of large, *three-dimensional* maps, which enable precise localization in all six degrees of freedom.

Most state-of-the-art route following frameworks do not allow the robot to leave its reference path. However, we believe that the ability to adapt to changes in the environment is a key ingredient for long-term autonomous navigation in any environment that is not completely static. Apart from the requirements on robustness of the localization to path deviations, this also necessitates obstacle detection and motion planning capabilities. The approach of (Cherubini and Chaumette, 2013) is similar to ours in the sense that a laser-based obstacle detection and a local planner are integrated with a route following framework. Their local planning is based on the evaluation of a set of tentacles (arcs) originating at the robot’s position, representing possible trajectories. Our planner uses model-based dynamic forward simulation to compute a set of candidate trajectories. This yields a greatly increased variety in the shape in of the local paths, and may therefore enhance the performance in situations cluttered with obstacles. Moreover, it reduces the probability of falling into local minima (cul-de-sacs). The trail following framework presented in (Rasmussen et al., 2012) includes obstacle detection and local motion planning as well. Their planner produces candidate paths that are concatenations of Dubins paths (Dubins, 1957), therefore accounting for differential motion constraints and allowing for a certain flexibility in terms of path shapes. However, unlike our planner, it produces purely geometric paths without considering the vehicle dynamics. We believe that the latter is essential for both smoothness and safety of robot motions. An alternative approach to integrating planning with a route following system is presented by Stenning et al. (2013). Their approach is to build a network of paths that can be used to more flexibly navigate the environment by choosing among different routes. For driving in dynamic and constantly changing environments, this might however not be sufficient: in theory we would need an infinitely dense network to be guaranteed to find a path around an obstacle if one exists. The large FOV of our sensor and the resulting increased robustness to path deviations of our system allow us to significantly deviate from the taught path when avoiding obstacles, without actually extending the network of paths.

Autonomous navigation in urban pedestrian environment has received considerable attention in recent years. Siagian et al. (2013) present an approach based on both vision and laser ranging. Their navigation system uses monocular vision-based road and vanishing point recognition for estimating the robot’s pose, while obstacles are detected with a laser range finder. In contrast to this approach, our system does not require any man-made structure such as roads for localization and is invariant to lighting conditions. A purely laser-based technique for navigating in crowded environments is shown in (Trulls et al., 2011). Unlike our approach, this system is tailored to navigation in known areas, thus does not include mapping of the environment. Localization is achieved using a particle filter. Remarkable results in this area have been achieved in the European-Commission-funded EUROPA project (Kümmerle et al., 2013). The resulting navigation system

is based on two planar laser range finders as the primary sensors, which in experiments has shown to enable navigation in crowded city environments over several kilometers. The system includes obstacle detection, motion planning, and SLAM to learn large-scale maps, that are represented in a hybrid topological/metric manner. However, the mapping of the environment is only 2D. This differentiates it from our ICP-based mapping technique, which is inherently 3D.

3 System overview

Robotic navigation in the context of T&R involves two distinct phases: the *teach pass* and the *repeat pass*. In the former, the robot is manually steered along the desired route, and the system builds up a map of the environment. In the latter, the robot autonomously repeats the route, using its sensor readings to localize within the map recorded in the teach pass. The building blocks of a T&R system are thus four modules, accomplishing the following tasks:

- (1) incremental localization;
- (2) mapping of the environment;
- (3) map-based localization; and
- (4) path tracking control.

The functions (1) and (2) are required in the teach pass for building the map of the environment, (1), (3) and (4) are employed in the repeat pass for localization within the previously recorded map, and for controlling the robot to follow the desired route. Existing T&R systems suitable for navigation in non-planar (3D) environments implement the functions (1) to (3) using appearance-based methods with either cameras (e.g. Furgale and Barfoot (2010)) or lasers (McManus et al., 2013b). However, this is a implementation choice rather than a general characteristic of the T&R concept. In this paper, we use geometric measurements instead of appearance data, namely 3D point clouds from a spinning laser. We show how ICP can be employed for incremental localization, map building, and map-based localization.

In a classical route following system, the robot is supposed to precisely follow the learned path when repeating the route. This is typically achieved with a path tracking controller, generating motion commands based solely on the robot's pose relative to the reference path. In order to enable driving in dynamic environments, we replace path tracking control by an obstacle avoidance scheme, which takes as input the same point cloud data as the localization and mapping modules. Our algorithm is able to detect obstacles obstructing the path and to plan avoidance maneuvers to temporarily leave the reference route whenever necessary.

Our system thus consists of three main ingredients: a T&R framework (Section 5), ICP-based localization and mapping (Section 4), and an obstacle avoidance scheme (Section 6). Figure 2 illustrates how these components are assembled to form an adaptive route following system. In the teach pass, the robot is manually controlled. An ICP-based localization and mapping module is used to build up a database of connected local maps, and the driven route is stored as reference path. In the repeat pass, there are two different ICP processes: one for incremental localization, and one for localization within the map recorded in the teach pass. These processes are further explained in Section 4. A map manager module accesses the map database and in each iteration selects the appropriate submap, based on the current pose estimate. The latter is input to the local motion planner, along with the reference path and a local obstacle map produced by the obstacle detection module. Based on this data, the planner computes a feasible local trajectory and the corresponding motion commands for the robot.

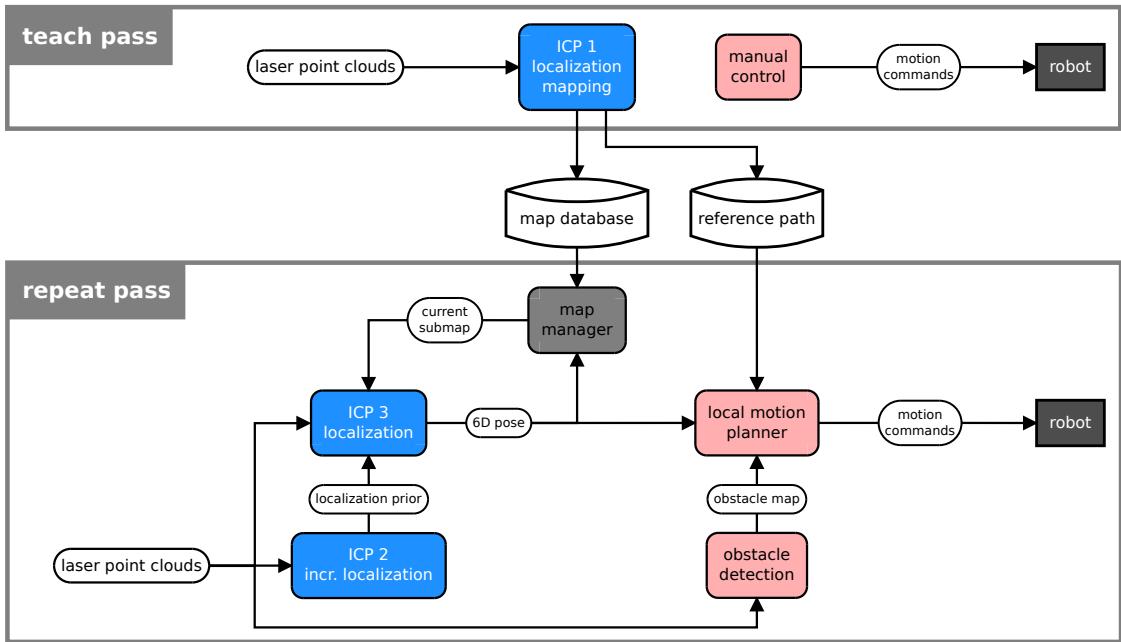


Figure 2: System overview. Three main components are the basis of our adaptive route following system: a T&R framework, ICP-based registration modules, and an obstacle avoidance scheme (obstacle detection and local motion planning). The map built during the teach pass is saved in a database and used by an ICP-based localization module in the repeat pass, to obtain an estimate of the robot's pose. Based on this and on a map of obstacles in the vicinity, the local motion planner finally computes safe commands for the robot, making it follow the desired reference path while avoiding collisions with obstacles obstructing the same.

4 Iterative closest point (ICP)

This section describes the ICP processing pipelines needed by the T&R system presented in Section 5. The specific processing steps were chosen with two goals in mind. First, we had to use ICP to construct real-time capable building blocks needed for a T&R system. Second, the processing pipeline had to be tuned for a high-rate spinning lidar producing approximatively 70,000 points at 10 Hz. To handle the real-time computation, we use the open source registration library `libpointmatcher` (Pomerleau et al., 2013). This library has previously demonstrated 3D registration maintaining real-time capability using a Kinect (i.e., 19,200 points at 30 Hz) (Pomerleau et al., 2011) and its configuration flexibility allowed us to rapidly deploy a registration solution adapted for a search and rescue system.

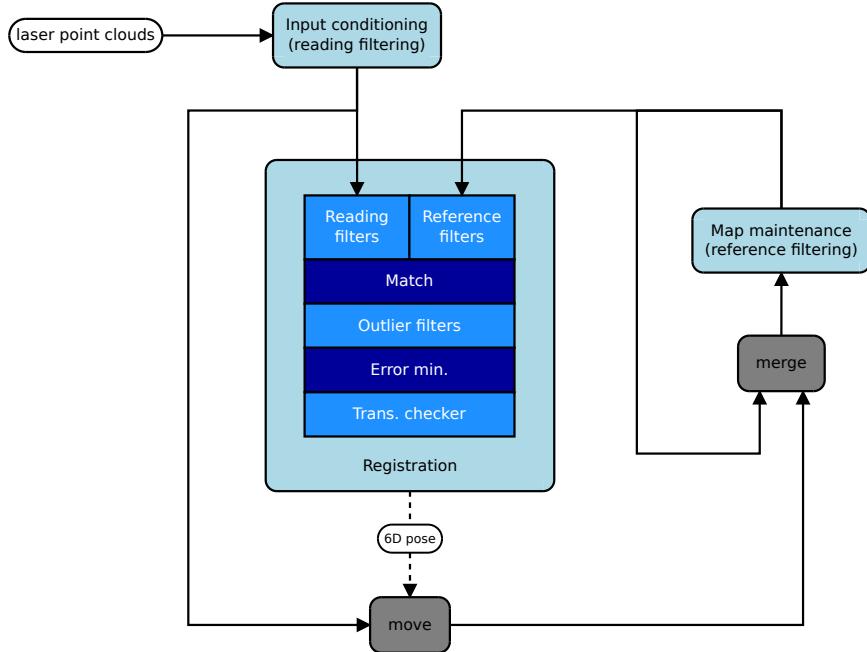


Figure 3: ICP framework used to compute the odometry based on laser readings and to build a 3D map of the environment. The arrows represent the flow of information, with solid black being point clouds and dashed the 6D pose computed by the registration module. The three light blue boxes are configurable modules with their parameters presented in Table 1.

As shown in Figure 2, our T&R system involves three slightly different ICP modules. In this section, we focus on the generic case of concurrent localization and global map building. This setup is used for the incremental localization in the repeat pass (ICP 2 in Figure 2). The other two ICP modules only differ from this in that they use a database of local submaps instead of one global map (details are explained in Section 5). The registration and filtering techniques as well as the parameter values are the same in all three modules.

Figure 3 shows the framework put in place around `libpointmatcher` allowing the localization and the map building. The input to the ICP framework are point clouds of complete revolutions of the sensor (360°), which we build by assembling all the measured points in a naive way, i.e., without considering the robot's motion during the acquisition (refer to Section 8.2 for a discussion of this approach). The point clouds are first filtered and preconditioned before being processed by the registration module. The registration module takes the filtered readings and computes a 6D pose that minimizes the alignment error of the reading point cloud to a reference point cloud. The module applies the standard ICP steps based on the nomenclature of Rusinkiewicz and Levoy (2001). The reference point cloud is represented by the global map that is updated every time the overlap between the reading and the reference goes under 80 %. The map maintenance module

applies a sequence of filters to keep its density constant and updates the surface normals leading to a better surface orientation estimation where the measurements were too sparse or too dense. The registration and map maintenance processes are decoupled and, in the deployed system, were running at approximately 5 Hz and 1 Hz respectively.

For experimental reproducibility and given that the library is available online⁴, we present the full list of parameters used for the configuration of the ICP framework in Table 1. This table follows the library documentation and will not be explained in full details, but we will lead the attention of the reader to a few important points. To ensure real-time operation, the first module in the Input filter chain randomly removes 85 % of the points from the laser scan. Also to gain speed, we use the approximate kd-tree⁵ search (Arya and Mount, 1993), with a tolerance of $\epsilon = 3.16$, when estimating surface normals based on local neighborhood and during ICP data association step. Those parameters represent a good tradeoff between the accuracy of localization and the speed at which the registration can be computed. To avoid local minima caused by the circular shape of the Velodyne point clouds, we also apply a filter aiming at a uniform density of points. The same filter is used in the map maintenance module to avoid accumulation of unnecessary points through time. This filter retains a low density of points (10 pts/m^3) because the platform usually covers very large areas during a typical outdoor deployment. Having a uniform distribution of points reduces the impact of dynamic elements passing close by the robot. Also, the low density of points used leads to few points on small objects, like cars and pedestrians, bringing more weight to large structures (i.e., houses, ground, hills, etc.) that are typically static. Dynamic elements are also handled at the registration level. In the Data association module, pairs of points with a matching distance larger than 2 m are not considered and only the lowest 90 % of all the remaining distances are kept. In that configuration, disturbances due to cars, pedestrians, snow, or reflections caused by water have a reduced impact on the error minimization.

Table 1: Configurations of ICP chains for the ARTOR mapping applications.

<i>Step</i>	<i>Module</i>	<i>Description</i>
Input	Read. filtering	RandomSampling SurfaceNormal ObservationDirection OrientNormals MaxDensity
		keep randomly 15 % of the points surface normals based on 20 nearest neighbors, $\epsilon = 3.16$
		add vector pointing toward the laser
		orient surface normals toward the obstacle direction
		subsample to a point density of 50 pts/m^3
Registration	Ref. filtering	- points used as reference are conditioned by the filters from the rows MAP
		MaxDist keep points within a radius of 70 m
	Read. filtering	- points used as reading are conditioned by the filters from the rows INPUT
		RandomSampling keep randomly 25 % of the points
	Data association	KDTree kd-tree matching with 2.0 m max. distance, $\epsilon = 3.33$
	Outlier filtering	TrimmedDist keep 90 % closest points
		SurfaceNormal remove paired normals angle $> 90^\circ$
	Error min.	PointToPlane point-to-plane
	Trans. checking	Differential min. error below 0.01 m and 0.001 rad
Map		Counter iteration count reached 40
		Bound transformation fails beyond 2.0 m and 0.4 rad
	Ref. filtering	SurfaceNormal MaxDensity MaxPointCount Update normal and density, 20 nearest neighbors, $\epsilon = 3.16$ subsample to keep a point density of 10 pts/m^3 subsample 70 % if more than 600,000 points (30,000 in ICP 2)

5 Teach and repeat system

In this section, we describe the T&R system which employs the ICP framework presented above. In the teach pass, the robot is steered along the desired route by an operator. Meanwhile, it scans the environment and builds up a sequence of connected local maps (called submaps). In the repeat pass, the robot autonomously

⁴<https://github.com/ethz-asl/libpointmatcher>

⁵The implementation uses **libnabo**: <https://github.com/ethz-asl/libnabo>

navigates along the previously driven path (in either direction), localizing itself relative to the nearest submap at every point and trying to minimize the path-following error within this submap.

The map representation used in our algorithm is a *pose graph*, an undirected graph in which vertices implicitly represent coordinate frames with attached metric maps, and edges represent transformations between coordinate frames. During the map building process, the robot estimates its motion and adds vehicle frame poses encoding the trajectory of the robot to the pose graph. The resulting representation may be classified as hybrid topological/metric. Local submaps are interpreted as cohesive metric spaces that the robot can plan and reason in. As the scale of the map grows, so does its uncertainty, and the metric accuracy of the map may be incorrect at large scales. However, the topology remains correct and the robot is able to travel over long distances from one physical place to another simply by making progress within the individual submaps. As noted for other T&R systems, this results in a highly scalable map representation as only a small portion of the full map needs to be kept in memory at any time.

5.1 Teach step — mapping

Figure 4 illustrates the map-building process during the teach step. Whenever we get a new point cloud from the lidar sensor, we try to match it against the most recently completed submap (here *submap i*). The matching process provides an estimate of the pose of the robot within this submap. We transform the input point cloud with this estimate, and insert it into the map that we are currently building (here *submap i+1*). Depending on the available computational resources, updating the map may not be feasible in real time, due to the rather expensive map maintenance process. Therefore, if new point clouds arrive before the last map maintenance step is completed, the points are used only for localization, and they are not added to the map.

Periodically, the submap being used for localization is saved to disk and replaced by the submap that was being built. The algorithm then initiates the construction of a new submap. The initialization of a new submap can be triggered by two events:

- the distance between the last submap and the current robot position exceeds a threshold (we set this threshold to 10 m)
- the overlap between the current scan and the map that we are matching it against falls below a certain threshold

The latter can happen for example when driving around the corner of a house, when suddenly large parts of the reference submap disappear from the FOV. The threshold value for the overlap required some tuning. We finally found 84 % to be a reasonable value for our application. In all of our outdoor experiments, the overlap usually remained high, and only fell below the threshold in special cases as the one described above. The vertices in our pose graphs are therefore typically 10 m apart from each other. Since the robot tries to follow a straight line from one vertex to the next in the repeat step, this wide spacing would not allow it to precisely follow the route taught by the operator. In the pose graph, we therefore store many intermediate poses encoding the robot’s path between the submaps.

5.2 Repeat step — following

During route following, a portion of the pose graph is loaded up and the robot attempts to localize against the nearest submap. For motion control, that is, to make the robot follow the reference path encoded in the pose graph, we use either a path tracking controller, or the local motion planner described in Section 6.2. The former is similar to the one used in (Furgale and Barfoot, 2010) and (McManus et al., 2013b), and has originally been developed in (Marshall et al., 2008). It is a nonlinear controller using state feedback linearization and a unicycle model of the vehicle.

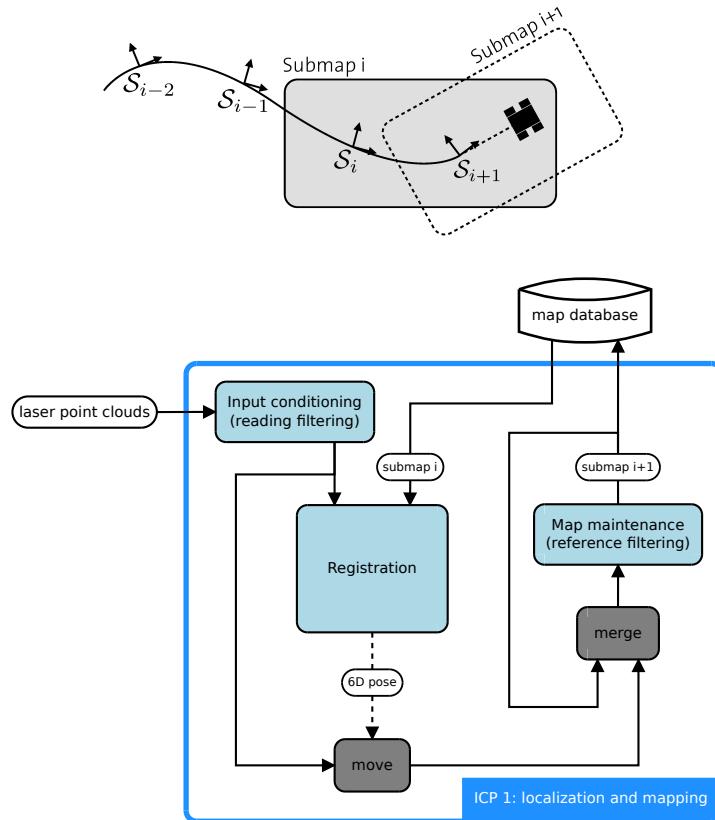


Figure 4: Building a database of local submaps during the teach pass. Illustration of the process (top) and setup of the ICP module (bottom). Whenever the robot gets a new point cloud, the registration module tries to match it against the current reference (here *submap i*). Once the localization is computed, the current point cloud is added to the map that is being built (here *submap i+1*). The difference to the ICP module shown in Figure 3 is that here we build a database of submaps (a pose graph) instead of a single global map.

In both (Furgale and Barfoot, 2010) and (McManus et al., 2013b), it was found that interleaving VO and localization provided the best robustness to environmental changes; where localization failed, VO would provide an accurate relative motion estimate over short distances. We have developed a similar system based on ICP that we call laser odometry (LO). We maintain a small *sliding map* of laser points for computing the LO. The sliding map is built by a separate ICP process, and is essentially a global point cloud map with a relatively low limit on the number of points (30,000 in our setup). This limit is enforced by randomly subsampling the map whenever the number of points exceeds the threshold. As we remove points in the entire map, but only add new points close to the robot (sensor readings), this results in a map that is very sparse everywhere except in the vicinity of the robot. It can thus be seen as a local map moving with the robot, although it is represented in a global coordinate frame attached to the starting point of the repeat pass. As illustrated in Figure 5, every incoming point cloud is first matched against the sliding map to estimate the relative motion of the robot. This transformation is used to add the points to the sliding map. The transformation from the LO is then used as a prior when matching against the closest submap in the pose graph. This interleaving of LO and localization allows to continue operation when localization against the map fails.

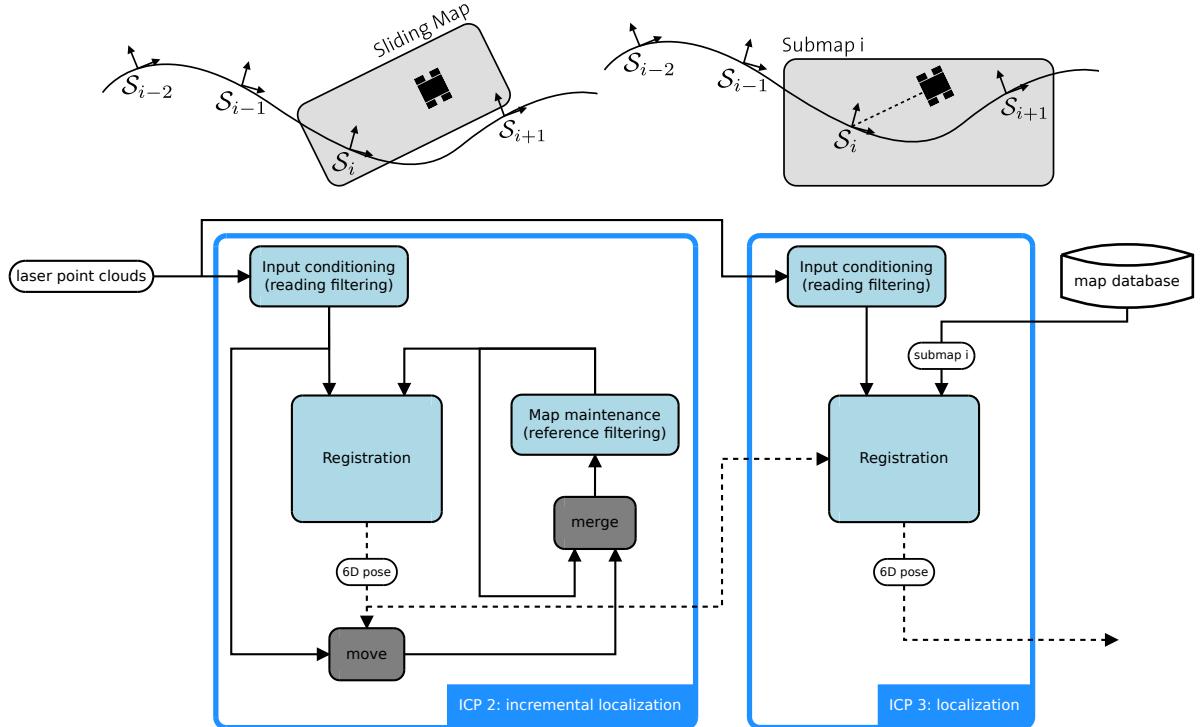


Figure 5: ICP-based localization in the repeat pass. Illustration of the process (top) and setup of the ICP modules (bottom). Our framework includes two separate ICP modules, one for incremental localization (LO, left) and one for localization against the map recorded in the teach pass (right). The LO module maintains its own local map (*sliding map*). Incoming point clouds are first processed by the LO pipeline to produce an estimate of the relative motion since the last time step, which is then used as a prior for localization against the closest submap in the pose graph. The localization module (ICP 3) uses the existing map database and does not build a map.

6 Obstacle avoidance

Many existing T&R systems aim at following a previously learned route regardless of changes in the environment. Given the localization of the robot and a reference path, this can be accomplished by a relatively simple path tracking controller. In reality however, hardly any environment is completely static. If we aim

at fully autonomous route following, this means that the robot must be able to (at least locally) react to changes in the environment, such as obstacles blocking the path. The most simple solution would be a safety stopping module, which brings the robot to a halt in case of an imminent collision. However, we aimed at going one step further and developed an obstacle avoidance module, which enables our robot to detect and circumnavigate objects that occur on the route.

The system consists of two main components: an obstacle detection module (Section 6.1) and a local motion planner (Section 6.2). Using the 3D range data provided by the Velodyne sensor, the obstacle detection module builds a planar map of the robot’s vicinity, classifying the area into free and occupied space. This obstacle map, as well as the global reference path and the localization of the robot are the inputs to the motion planner. Based on this information, the planner computes a feasible and suitable local trajectory and generates appropriate velocity commands, which make the robot avoid collisions with obstacles, while still following the reference path as closely as possible.

6.1 Obstacle detection

Inspired by the work presented in (Hähnel and Thrun, 2008), we developed an obstacle detection algorithm that is based on the Velodyne 3D range scans. The algorithm works directly on the raw laser data without any filtering or subsampling. It is completely independent of the ICP-based localization and mapping module, does not require any external state estimation, and we do not make any assumptions about the robot’s attitude. The Velodyne lidar emits 32 laser beams rotating around the vertical axis. On a flat ground, the measurements therefore consist of several concentric rings around the sensor, as illustrated in Figure 6. The core idea of our obstacle detection scheme is to look at the horizontal distance between subsequent rings: positive obstacles will locally reduce this distance, while negative obstacles cause the rings to be further apart from each other.

We make two main assumptions for our obstacle detection module: first, the environment is supposed to be locally planar, and second, we assume the height of the laser sensor above ground to be constant, regardless of the vehicle’s attitude. As illustrated in Figure 7, this allows us to compute the expected distance between two rings, $\Delta d_{i,e}$, and the actual distance, Δd_i , using only the measured ranges of two successive laser beams. The angle φ between those measurements is constant and known ($4/3^\circ$ in the case of our sensor). Note that computing the desired distance values does not require the absolute angle of the laser beam to the vertical to be known, if we assume the height h to be constant. The computations are therefore independent of the robot’s pitch and roll angle.

Whenever a new scan arrives, we go through each vertical line of measurements and classify every point as either “obstacle” or “free”, by computing the actual and the expected horizontal distance to its predecessor from the two range values r_i and r_{i-1} . We use a relative measure for this classification: a point is counted as an obstacle if the rings at that point are compressed to less than $1 - P$, $P \in [0, 1]$ of the nominal distance, or expanded by a factor higher than $1 + P$, that is, if

$$\frac{|\Delta d_{i,e} - \Delta d_i|}{\Delta d_{i,e}} > P \quad (1)$$

Taking the absolute value in Equation 1 allows to detect negative obstacles, such as steps, or a hole in the ground, which yield an increased distance between the rings. On our robot, the Velodyne lidar is at a height h of 1.12 m, and we found a suitable value for P to be 0.85. This means that we consider a point as an obstacle if the horizontal distance to its predecessor is less than 15% or more than 185% of the expected distance. The relative nature of this threshold by construction leads to an obstacle detection that is more tolerant further away from the robot, meaning that the minimum height of an object to be considered as an obstacle increases with distance. This behavior arises from the fact that the resolution of the sensor decreases with increasing distance (the nominal distance between the rings increases). Yet we believe that being less strict about the obstacle height at further distances is a reasonable policy in general, as the uncertainty of the measurement may be higher as well. Figure 8 illustrates the classification behavior of our obstacle

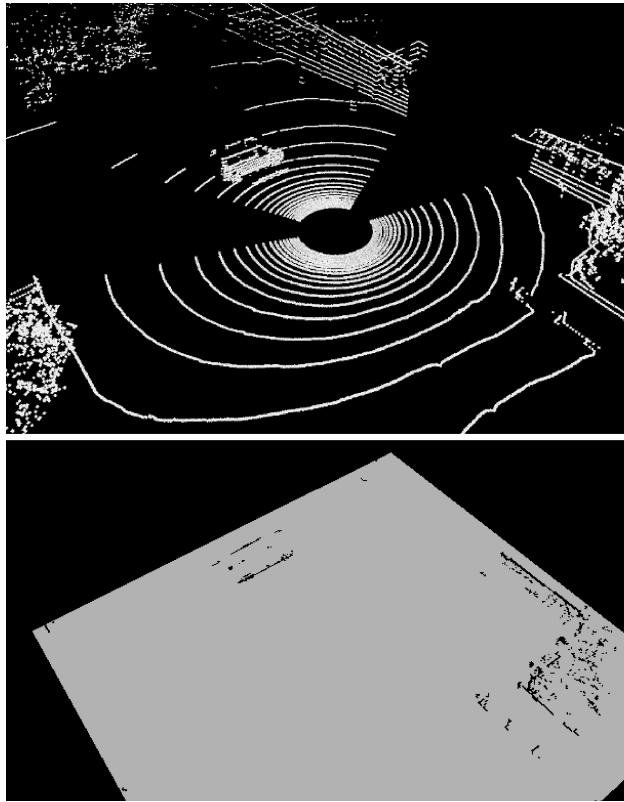


Figure 6: An example of the input and the output of our obstacle detection module: a point cloud from the Velodyne 3D lidar (top), and the corresponding planar obstacle map (bottom). The scene shows a van on a large flat area, surrounded by buildings and some trees. The Velodyne consists of a series of individual lasers, spinning around the vertical axis. In a planar area, this results in an array of concentric circles. For obstacle detection, we exploit the fact that objects sticking out from the plane locally compress these rings.

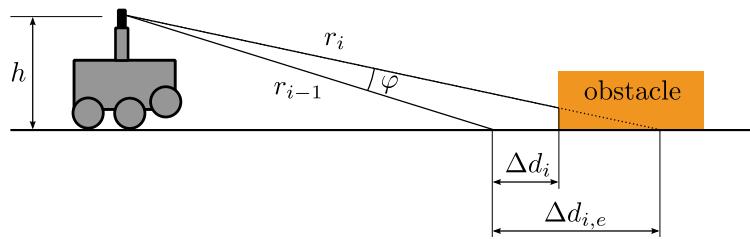


Figure 7: In the case of a positive obstacle, the laser rings are locally compressed. Assuming the height of the sensor above ground, h , to be constant and the environment to be (locally) planar, the expected and the actual distance between two subsequent rings can be computed based solely on the two corresponding range measurements r_i and r_{i-1} , and the angle between the beams, φ . The attitude of the robot is not required.

detection algorithm for the aforementioned values of h and P .

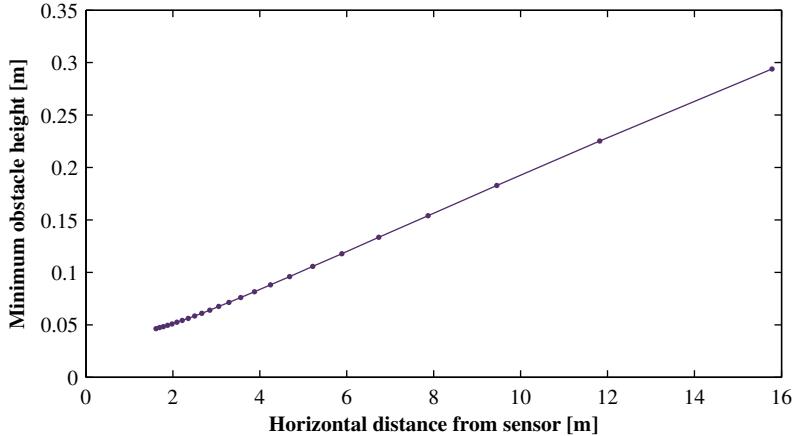


Figure 8: The minimum height (or depth) of an object to be classified as an obstacle by our algorithm increases about linearly with the distance from the sensor, starting at a value of 5 cm at a distance of 2 m. The graph shows the classification behavior for the parameters we chose for the experiments in this article. The marks on the line represent the individual laser beams of the Velodyne lidar. The distribution of these points shows that the sensor’s resolution drastically decreases with distance.

We keep a list of all points that were classified as obstacles. Each point consists of a triple of coordinates and an associated time stamp. Obstacle points older than a user-defined threshold Δt_{max} are deleted from the list. In regular intervals, we project all the obstacle points onto a planar grid map. We count the number of points in each cell of the map, and label a cell as occupied whenever this number exceed a certain threshold value. In the experiments presented in Section 7, we used square obstacle maps of 30 by 30 m, consisting of cells with a side length of 10 cm. The threshold Δt_{max} was set to 0.5 s, and we updated the obstacle maps with a frequency of 3 Hz.

6.2 Local motion planning

Fully autonomous repeat runs necessitate functionality for addressing unforeseen changes and unmapped objects in the environment online. To this end, the navigation system needs to be able to autonomously bypass newly appearing objects obstructing any parts of the teach run’s reference path. At the same time motion should be constrained to areas where a significant overlap of consecutive frames’ FOV can be guaranteed, so as to retain a large number of previously detected environmental features for the localization module. This is especially important for sensors with limited FOV (such as the stereo vision system we compare our ICP-based localization system against). The collision avoidance system presented in this section simultaneously addresses both of these requirements.

The collision avoidance scheme on ARTOR is based on the sampling-based online planning framework presented in (Schwesinger et al., 2013). The framework enforces path alignment—and therefore a large overlap in the FOV—by shaping a tree of system-compliant motions along a reference path. It is based on a user-supplied system model and control law. The system model is (numerically) forward simulated towards samples drawn from a state-manifold aligned with the reference path. Within the framework, the state-manifold is defined as the heading and curvature aligned subspace of the robot’s state-space. An internal control law then regulates the simulated system towards a sampled version of this manifold. Within the forward simulation of the system model, any kind of system constraints can be enforced, making the approach appealing to constrained (such as non-holonomic) systems.

System Model. The highly configurable planning framework is able to incorporate arbitrary system mod-

els (suitable for forward simulation). The ARTOR system is modeled as a kinodynamic differential drive robot with constraints in both longitudinal and rotational velocity as well as longitudinal and rotational acceleration.

Sampling Scheme. The sampling scheme of the state-manifold used within the framework is also partially configurable. To generate candidate evasive trajectories, the framework samples lateral offsets from the reference path. To allow for longitudinal speed adaptations while driving, several target speeds of the robot are created by sampling speed offsets from the reference speed. The result of the state-manifold sampling is a tree of candidate trajectories, which is searched for the optimal alternative according to the cost function specified below. For the system described in this article, we use a tree depth of one, as shown in Figure 9.

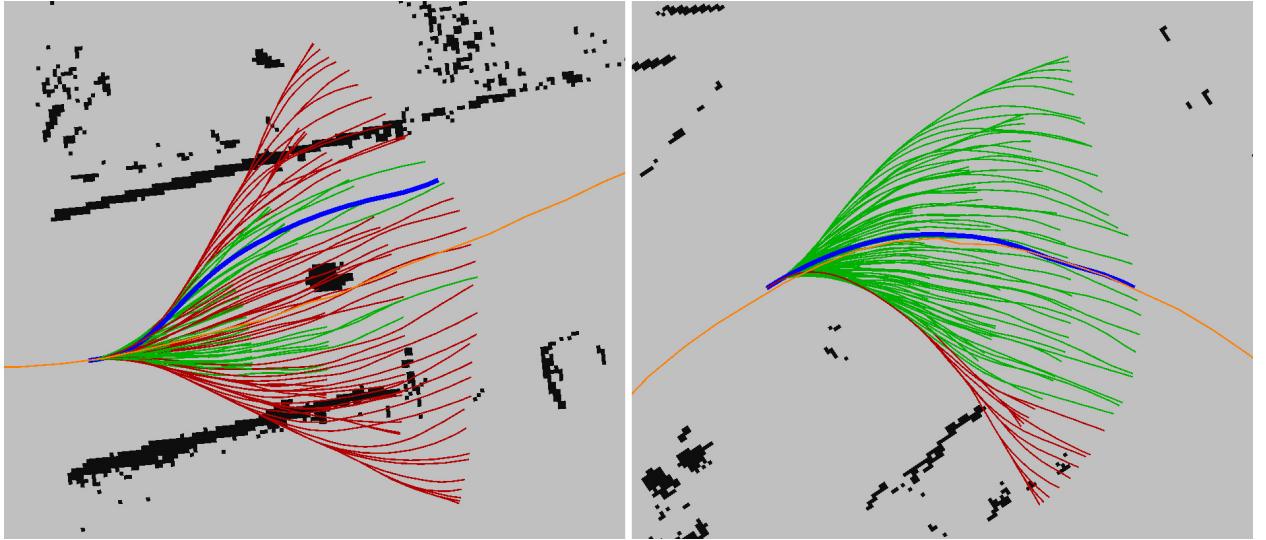


Figure 9: Set of candidate trajectories shaped around the reference path, generated by the local planner for a fixed target speed and a single tree level. The figures show snapshots of the planner output during the T&R experiment described in Section 7.4, containing the reference path (orange), the candidate trajectories (feasible: green, unfeasible: red), and the underlying obstacle grid map. In the left figure, an evasive maneuver is planned due to an oncoming pedestrian obstructing the reference path. The trajectory sets contain 495 trajectories generated from 11 speed offset samples and 45 lateral offset samples (with a maximum lateral offset of eight meters).

Collision Detection. Collision detection is performed on the two-dimensional binary occupancy grid described in the previous section. Objects are assumed to be static due to the lack of a classification module for dynamic objects in the current implementation of the obstacle detection, but are sufficiently accounted for through fast replanning at approximately 10 Hz.

Optimization Criterion. Naturally, the robot should follow the reference path as closely as possible while trying to keep a user-defined reference speed. The cost function is therefore divided into a lateral and longitudinal cost term. In our implementation, the lateral part is composed of the integrated distance between reference path and candidate trajectory over time plus a terminal cost that penalizes distance from the reference path at the end of a sample. Similarly, the longitudinal cost term integrates speed offsets to the reference speed and adds a terminal speed error.

Trajectories in collision with any object in the scene are assigned infinite costs. We include (soft) penalization of trajectories passing in close proximity to any collidable object in order to keep a desirable safety distance whenever possible.

7 Experiments

We tested our system extensively in different environments, and under a variety of weather and lighting conditions, using our robot ARTOR. In a first set of experiments, we assessed whether the system was able to fulfill all our requirements: localization accuracy, obstacle avoidance capability, robustness of the localization to path deviations, lighting invariance, and operational reliability (Section 7.3). In all of these experiments, we further compared our system to the stereo-vision-based T&R framework of Furgale and Barfoot (2010) in terms of localization performance. In a second step, we investigated the performance of the complete adaptive T&R system in a dynamic urban environment featuring both static and dynamic obstacles (Section 7.4). Our robot autonomously drove more than 22 kilometers in total, 7 in rough terrain (Section 7.3.3) and 15 in the urban environment (Section 7.4), at day and night, and under different weather conditions (sunlight, rain, snowfall).

The field tests have been conducted in two different locations: in a dynamic urban environment and in rough, off-road terrain (Figure 10). The experiments in the city context were carried out around the ETH campus in Zurich. The robot drove on narrow side streets as well as on sidewalks and bike lanes along busy major roads. The static part of the environment consisted of high buildings, several construction sites, and parked cars along the streets. The latter two yielded significant changes over time in the perceived environment. Moreover, both the localization system and the obstacle avoidance module had to deal with dynamic objects, such as cars, trucks and pedestrians.

The second set of experiments was conducted on a military testing ground for off-road vehicles in Thun, Switzerland. The challenges in this environment were of a different kind. There were fewer dynamic objects in the scene, but, unlike on urban roads, the terrain was neither flat nor smooth. The area consisted of rugged gravel paths with steep inclines, yielding a highly 3D motion. Furthermore, the unstructured nature of the environment—there were no faces of buildings, no paved roads, only trees, grass and bushes—rendered localization and mapping more challenging.



Figure 10: The experiments presented in this section were conducted in two different environments: on busy city streets around the ETH campus in Zurich (left), and on a testing ground for off-road vehicles in Thun (right). The main challenges in the former were dynamic objects such as moving cars and pedestrians, as well as significant changes over time in the static part of the environment. In the latter, the robot had to cope with unstructured, rugged, 3D terrain.

7.1 Robotic hardware

The robot used in these experiments, ARTOR, is based on the LandShark platform manufactured by Black-I Robotics, USA. It is a six-wheeled, skid-steered vehicle powered by two electric motors. This configuration yields traction comparable to tracked vehicle, while simplifying the maintenance of the locomotion system. The basic platform has been modified by RUAG Land Systems (Thun, Switzerland) to be suitable for autonomous navigation. The dimensions of the complete vehicle are 1.32/0.71/1.22 m (length/width/height) and it has an approximate weight of 350 kg. The robot can drive at a maximum speed of 3.5 m/s, but we typically set the speed to around 1 m/s.

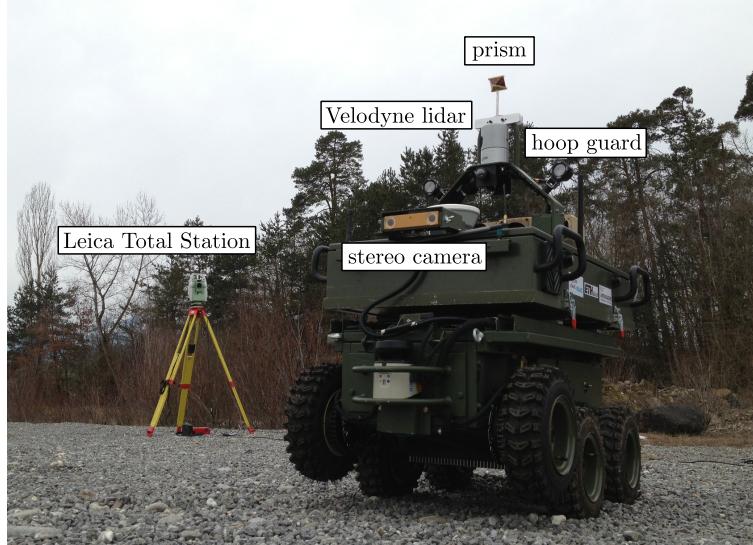


Figure 11: Our robot ARTOR, an electrically powered, skid-steered vehicle, equipped with an array of sensors. For our ICP-based T&R system we use the Velodyne 3D lidar mounted on top of the robot. For protection of the sensor we mounted a metal hoop around it. A Leica Total Station is used for ground truth in the experiments. It tracks the 3D position of a prism mounted on the robot and provides measurements with millimeter accuracy.

ARTOR is equipped with an array of different sensors, among which only the Velodyne HDL-32E 3D laser range finder is used by our ICP-based registration module and the obstacle detection system. We do not employ any other sensor measurements, such as wheel odometry or inertial measurements. The Velodyne HDL-32E produces around 70,000 point measurements at 10 Hz, and has a maximum range of 80 m. It is equipped with 32 laser/detector pairs oriented at different inclination angles between +10.7 and –30.7° from the horizontal. The lasers are contained in a spinning head, which yields a 360° FOV. However, the hoop guard mounted around the sensor on ARTOR causes two blind spots of around 43°, one on either side of the robot. These are well visible in Figure 6. We further discard the lowermost two laser beams, as these mainly hit the robot itself. The number of range measurements per revolution is thus reduced to around 42,000 in average.

In the experiments comparing to the stereo-vision-based approach, we used a Point Grey Bumblebee2 stereo camera, mounted at the front of the robot. It has a 100° FOV and acquires pairs of color images with a resolution of 1024×768 pixels at 20 Hz. In our experiments, we used grayscale images subsampled to 512×384 pixels to allow for real-time performance.

All data processing is executed onboard, on a laptop featuring an Intel Core i7 processor running Ubuntu Linux and the Robot Operating System ROS. All sensor measurements are provided in the form of ROS messages, and the different software components, such as the localization module, the obstacle detection, or the local planner, communicate with each other via ROS topics and ROS services.

7.2 Ground truth and performance measures

In all the experiments, except the long-range field tests, we compared our system's pose estimates to highly accurate ground truth. The latter was provided by a Leica Total Station TS15 I, measuring the robot's 3D position with millimeter accuracy at approximately 7Hz. More precisely, it tracked the position of a prism mounted on top of the robot (cf. Figure 11). Since this system does not provide the orientation of the robot, our evaluation concentrates on the error in *position* estimates.

Motivated by the *relative* nature of our mapping and localization system, we focus the analysis on the accuracy of our pose estimate *relative* to the current map in the pose graph, rather than computing a global localization error. For proper operation, our system only requires the former to be sufficiently accurate. Our main localization accuracy measure is a combination of the position estimate errors in lateral and longitudinal direction along the path, relative to the current vertex in the pose graph. We denote these by e_{lat} and e_{long} , respectively, and call the combination of both the *relative localization error* e_{rel} .

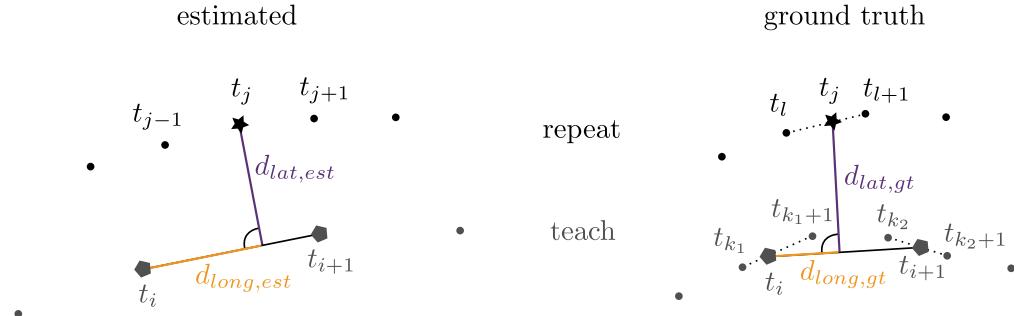


Figure 12: Computation of offsets from the pose graph in lateral and longitudinal direction. The points drawn in this plot are in fact 3D position estimates (left), and measurements (right). All the computations are made in 3D space. We use the following subscripts for time stamps of pose estimates (E) and ground truth measurements (G): E/teach: i , E/repeat: j , G/teach: k , G/repeat: l . Left: the estimated offsets are obtained by projecting the estimated position at time t_j to the pose graph at the reference vertex at time t_i (vertex i). Right: the ground truth values of the offsets are computed using time-based linear interpolation between ground truth position measurements in both the teach pass ($t_l \leq t_j \leq t_{l+1}$) and the repeat pass ($t_{k_1} \leq t_i \leq t_{k_1+1}$ and $t_{k_2} \leq t_{i+1} \leq t_{k_2+1}$).

For a pose estimate at time t_j , we compute this value as follows. We get the estimated lateral and longitudinal offsets $d_{lat,est}$ and $d_{long,est}$ by projecting the estimated robot position at time t_j onto the line connecting the current reference vertex and its successor (Figure 12, left). These vertices have been created during the teach pass, and their respective time stamps are t_i and t_{i+1} . In order to compute the true values of the offsets $d_{lat,gt}$ and $d_{long,gt}$, we need the ground truth positions at times t_j , t_i , and t_{i+1} . However, our localization estimates and the ground truth measurements are not synchronized. We therefore use linear interpolation to compute the ground truth positions of the robot at the required time stamps, as shown in Figure 12 (right). Finally, the relative localization error at time t_j is computed according to Equation 2.

$$\begin{aligned}
 e_{lat} &= |d_{lat,gt} - d_{lat,est}| \\
 e_{long} &= d_{long,gt} - d_{long,est} \\
 e_{rel} &= \sqrt{e_{lat}^2 + e_{long}^2}
 \end{aligned} \tag{2}$$

7.3 Analysis experiments

In this first set of experiments, we analyzed our ICP-based localization and mapping system in depth, to assess whether it fulfills our requirements. We conducted the following three types of experiments.

- An evaluation of the accuracy of our localization and mapping system in different environments and at different driving speeds, using precise ground truth measurements (Section 7.3.1).
- An analysis of the robustness of our localization system to deviations from the initial path, in both structured and unstructured environments (Section 7.3.2). We show that our system is able to detect and drive around large obstacles, and to stay precisely localized while doing so.
- A long-range T&R experiment in rough terrain over the course of an entire day, demonstrating the lighting invariance and reliability of our system (Section 7.3.3).

Throughout these experiments, we compared our ICP-based system to the stereo-vision-based T&R framework. We recorded images of our robot’s onboard stereo camera along with the laser point clouds in all the experiments. This allows for a fair comparison of the two systems, as their respective performance can be investigated under exactly equal conditions. The purpose of this comparison is twofold. First, we would like to assess the comparative localization accuracy of our system in the nominal case, that is, the conditions that the well-established stereo-based approach was designed for: only small deviations from the taught path, and sufficient ambient light. Second, we are interested in evaluating how much robustness to path deviations can be improved and dependency on lighting conditions can be mitigated by using an omnidirectional active sensor instead of a directed passive device.

7.3.1 Localization accuracy evaluation

Building a map of the environment and being able to localize accurately within the previously recorded map is a basic requirement for reliable autonomous navigation. We therefore first assessed the accuracy of our system in the two different environments, and at different driving speeds. Furthermore, we compared its accuracy to that of the stereo-vision-based framework. The urban and the off-road environment did not only differ in terms of what the sensors perceived (dynamic vs. static, structured vs. unstructured), but also in the motion of the robot—which in turn influenced the sensor readings. Figure 13 illustrates these differences. Moving in a 3D, unstructured environment involves much higher angular displacements and velocities than driving on paved city roads.

The experimental setup was as follows. In both the urban and the unstructured environment we manually drove the robot seven times along a path of around 130m, recording the laser point clouds as well as the stereo images. The first three runs were driven at the robot’s nominal speed of 1 m/s, the third and the fourth at around 1.5 m/s, and the last two at up to 2 m/s. All computations were run offline, but in real time on the same computer that is used on our robot. The first run was used to build the map (teach), and the remaining six passes to localize within this map (repeat). Ground truth was provided by the Leica Total Station during all these experiments.

The results of the accuracy analysis are shown in Figure 14 and Figure 15. First of all, we see that our ICP-based T&R system provides highly accurate relative localization: at the nominal speed of 1 m/s, the mean error is below 5 cm in the urban environment, and a little bit higher (but still well below 10 cm) in rough terrain. We also notice that the errors are overall higher in the unstructured environment, independent of the speed setting and for both the laser-based and the stereo-based system. One reason for this is the shaky motion involving high variation in roll and pitch angle, which may lead to motion blur and less overlap between the recorded map and the sensor readings in the repeat passes. For the ICP-based system, the unstructured nature of the surroundings additionally had a negative impact on the accuracy.

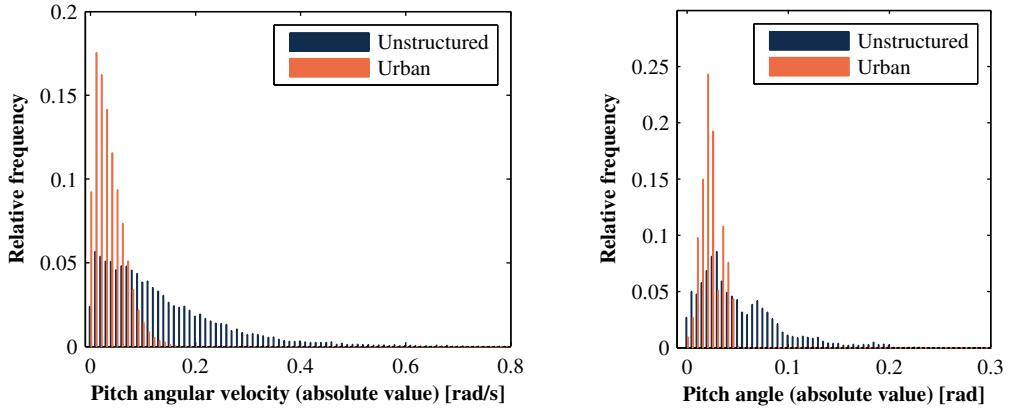


Figure 13: Distributions of the pitch angular velocity (left) and displacement (right) on the paths driven for the localization accuracy evaluation. The plots show the values recorded by the onboard IMU on a 130 m path driven at 1 m/s. It is clearly visible that when moving on paved roads the robot stays horizontal most of the time, while driving on an uneven gravel path yields high angular velocities and significant variation in the vehicle’s pitch angle.

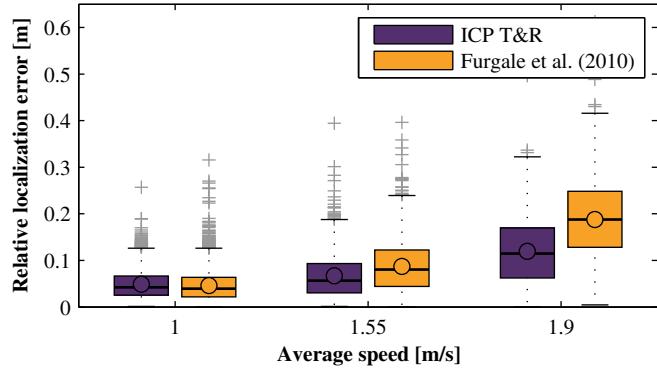


Figure 14: Relative localization errors in the urban environment. The bottom and the top of the boxes represent the first and the third quartile, respectively. That is, the box contains 50 % of all measurements. The line inside the box is the median, the circle stands for the mean value, and the gray plus signs are outliers. The plot shows the results of six different runs (2 at each speed setting) along the same 130 m path. Both the laser-based and the stereo-based systems are very accurate at the typical speed of our robot. When increasing the speed, the our system outperforms the stereo-based approach.

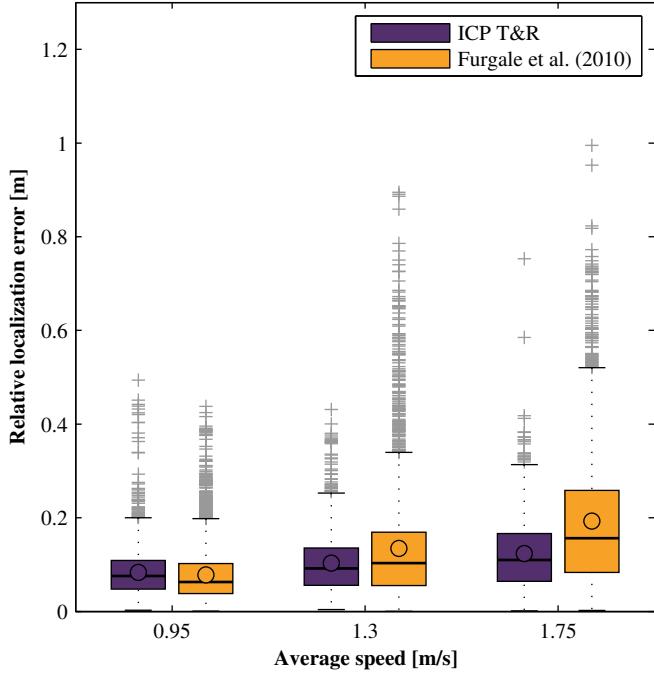


Figure 15: Relative localization accuracy in the unstructured environment. The plot shows the results of six different runs (2 at each speed setting) along the same 130 m path. The accuracy in this kind of terrain is slightly lower than in the urban context, due to the more bumpy motion and less structure in the environment.

7.3.2 Evaluation of the robustness to path deviations

In the classical T&R scenario, the robot is supposed to follow a previously taught route as closely as possible. However, in practical applications this is often not possible, as the environment is typically not static and obstacles might appear on the path that were not present at teach time. In these situations, the robot temporarily needs to deviate from its original path. We therefore wish to assess the robustness of the localization to deviations from the taught path. In the experiments presented in this section, we did so for our ICP-based T&R system as well as for the stereo-vision-based framework, and we compared the localization accuracy of the two different approaches using precise ground truth provided by the Leica Total Station. As in the accuracy evaluation of the previous section, we analyzed the robustness to path deviations of the two T&R systems in both urban and unstructured environment.

For the experiment in the urban environment, we chose a paved place of around 40 by 40 m, surrounded by several buildings. We used the full adaptive T&R system (ICP and obstacle avoidance). In the teach phase, we manually drove the robot over a straight path of around 23 m. We then let it autonomously repeat this path several times, while putting a single obstacle in the way and gradually increasing its size. The obstacle avoidance system described in Section 6 was used to detect the obstacles and find a suitable path around them. The path was repeated twice with each obstacle setting. We started with the unobstructed path, then placed a single car tire on the path, and added more tires aside of the first one in the subsequent two runs. In the last run, we placed our van in the middle of the path. Figure 16 shows the four different obstacles used in the experiment⁶, as well as the paths driven by our robot to avoid them. The first conclusion that we can draw from this experiment is that our local motion planner—together with the obstacle detection module, which reliably detected all the objects—enables the robot to safely and smoothly drive around large obstacles. In this experiment, we deviated from the taught path by up to 4 m, and the robot reliably found

⁶Note that, unlike the photographs may suggest, the experiment has been conducted on a sunny and dry evening. The pictures had been taken during an earlier experiment with the same setup.

its way back onto the path after the obstacle had been passed.

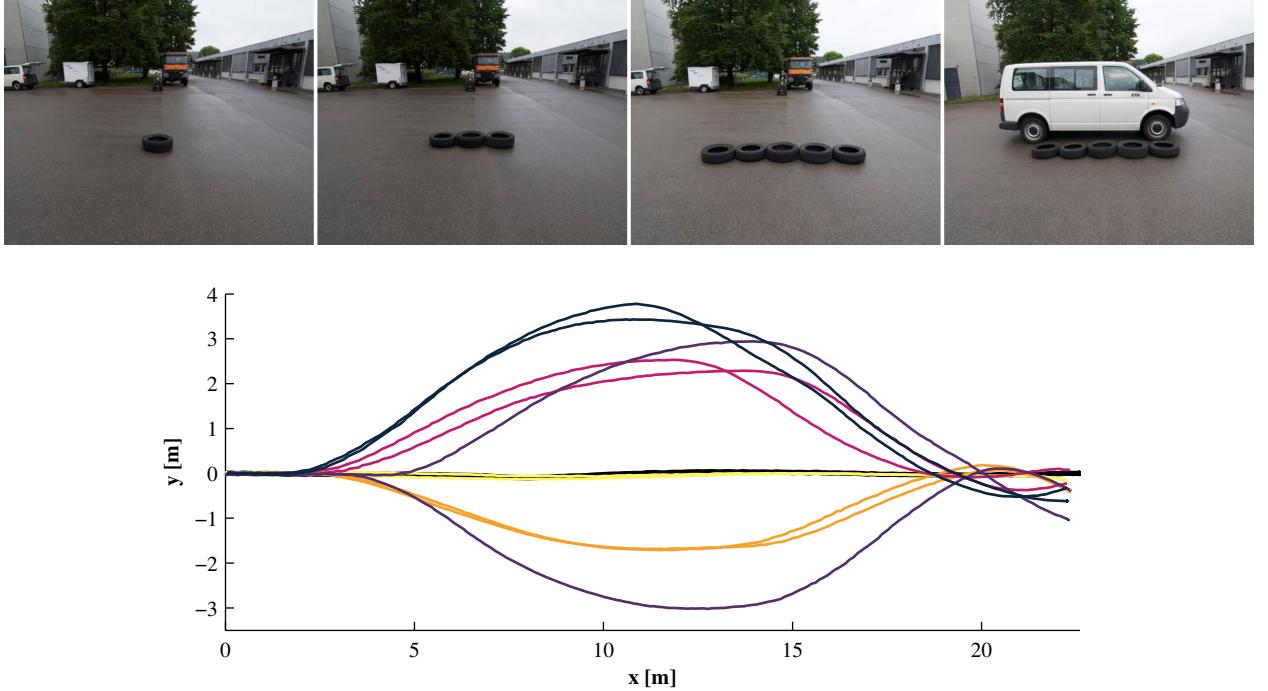


Figure 16: Obstacle avoidance experiment in the urban environment. The experiment consisted of teaching a straight path, followed by autonomous repeat passes with an increasingly large obstacle in the middle of the path. The photographs on the top show the four different obstacle settings. The starting point of the robot was left of the orange truck in the background, the goal was behind the photographer. The graph on the bottom shows the result of our local planner: the paths autonomously driven by the robot to avoid the different obstacles, where brightness of the lines decreases with increasing obstacle size.

In order to analyze the localization accuracy, we recorded the raw sensor data (laser point clouds and stereo images) of all runs during the experiment, and ran both the ICP-based and the stereo-based T&R system on the respective data in offline processing. There is some small variation in the localization results due to slightly different starting conditions when playing back the recorded data. In the vision-based system, this is due to the use of the RANSAC algorithm. In the laser-based system, this is due to the random subsampling of points. We therefore ran every experiment three times with the same data, which resulted in a total of six evaluations per obstacle setting (since we drove the path twice for each obstacle). Figure 17 shows the relative localization error of both T&R systems as a function of the size of the obstacle (and therefore as a function of the maximal deviation from the path). In the nominal setting, both systems exhibit similar performance, which could be expected based on the experiments shown in the previous section. When the robot deviates from its original path, however, the error of the stereo-based system increases dramatically, while the performance of our laser-based approach remains nearly constant. In the most challenging case, with the van obstructing the path and the view, the mean error of the stereo-based system is as large as 1 m, while our laser-based technique is still able to localize with an accuracy of around 10 cm.

In every time step during the repeat pass, both T&R systems attempt to localize with respect to the pose graph, using the current sensor readings. Both the laser-based and the stereo vision system fall back on incremental localization whenever map localization fails (LO and VO, respectively). Ideally, however, we would like to always localize against the map, as this yields more precise results and decreases the probability of a complete system failure. Our results show that the stereo-based system requires the robot to stay very close to the original path to enable localization against the map (the high localization errors arise from inaccurate incremental localization). The main reason for this is the small FOV of stereo cameras. If the

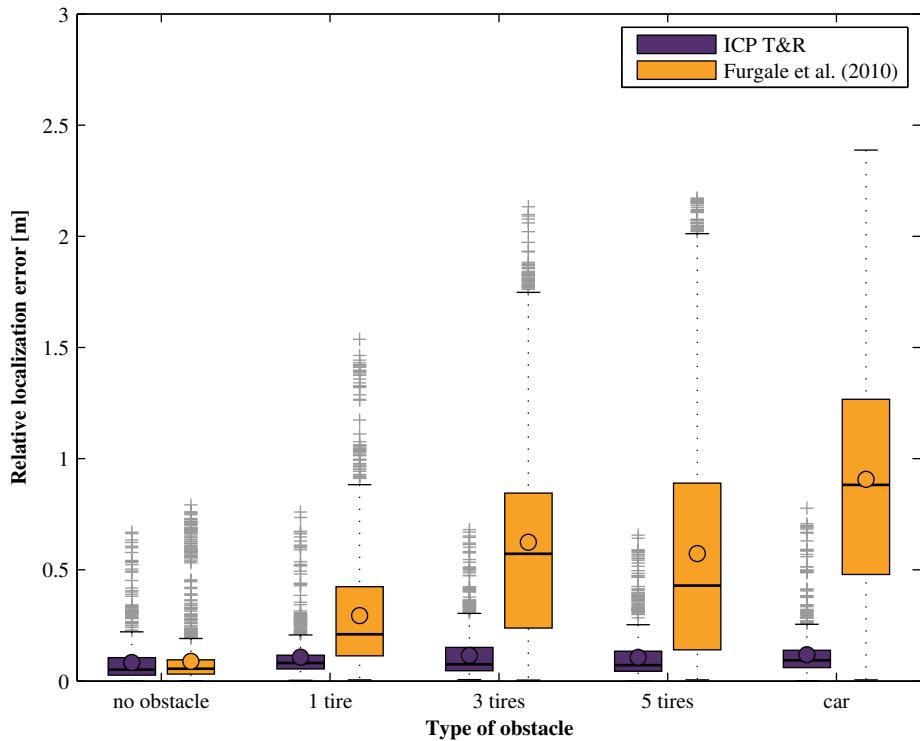


Figure 17: Localization accuracy depending on the type of obstacle on the path (urban environment). The size of the obstacle and the resulting deviation from the taught path increase from the left to the right. Each box plot contains the data of six experiments: the robot autonomously drove twice in each obstacle setting (using our ICP-based T&R scheme and the obstacle avoidance method), and we evaluated the data of each run three times with both T&R systems. The results clearly show the advantages of an omnidirectional, high-range sensor such as the Velodyne lidar over a directed sensor such as the stereo camera: it enables precise localization even when deviating from the taught path by several meters.

rotation of the camera relative to the taught path is too high, or when the camera is too far off the path, there is not enough overlap between the current image and the map, hence we do not have enough common visual features. The omnidirectional nature of the laser sensor makes our approach robust to rotations. Moreover, and in contrast to the stereo camera, the accuracy of range measurements does not significantly decrease with distance. As a consequence, our system does not fall back on incremental localization, but it is able to localize against the map even when deviating considerably from the reference path. Looking closely at the results in Figure 17, we see that also the error of the laser-based system increases with the distance from the path, yet only marginally. We see two reasons for this behavior. First, leaving the original path involves slightly different views on the environment, as well as larger and faster steps in the yaw angle. Second, due to the hoop guard around the lidar on our robot and the resulting two blind spots, we no longer have a truly omnidirectional sensor. This means that the orientation of the robot can indeed play a role for the localization, albeit much less significantly than in the stereo-based system.

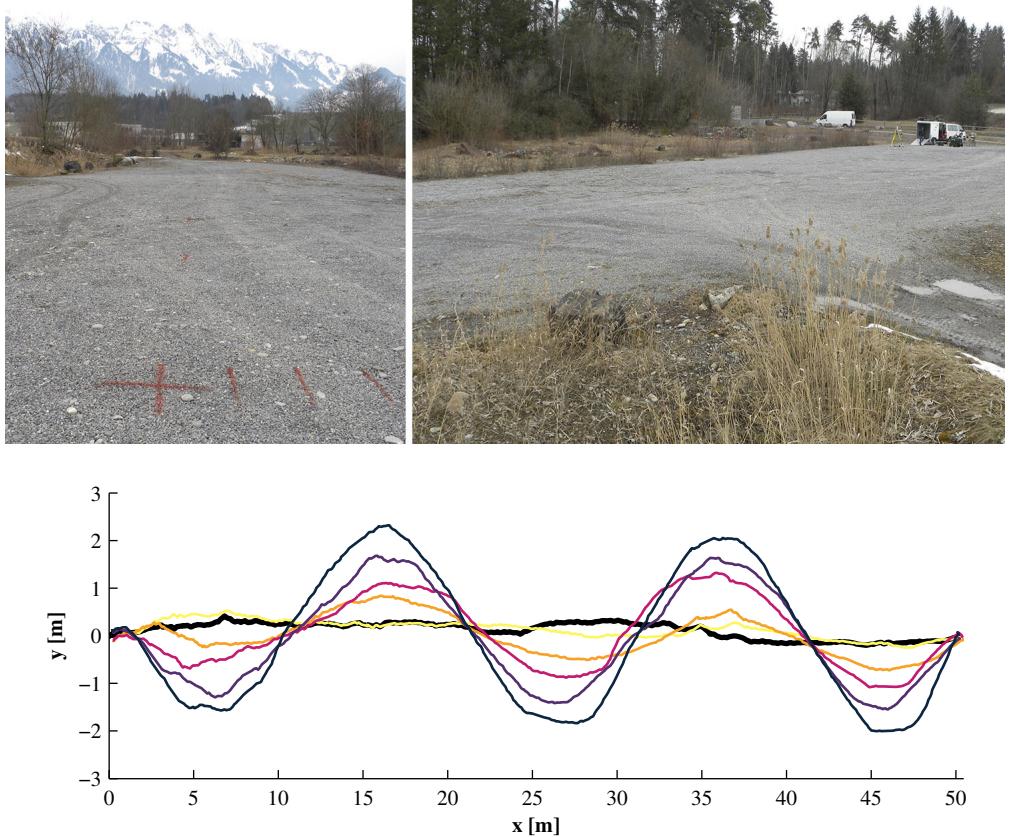


Figure 18: Experiment for evaluation of the robustness to path deviations in unstructured environment. The experiment has been conducted on the gravel area visible on the photographs. The robustness of the localization to deviations from the taught path was evaluated by driving the robot manually with sinusoidal oscillations around the original straight path. The experiment was repeated several times with different amplitudes, which had been marked on the ground with spray paint (visible in the left image). The graphs on the bottom show the paths driven by the robot (ground truth).

We conducted a similar experiment in the unstructured environment in Thun. However, we did not use our obstacle avoidance scheme to make the robot deviate from its learned path. Rather we repeatedly steered it manually along the taught route, but with sinusoidal oscillations around the original straight path. The amplitude of these oscillations was increased in every pass, while the speed of the robot was kept constant at around 1 m/s in all runs. The experiment was done on a relatively flat gravel area, over a distance of 50 m (cf. Figure 18). Again, we recorded the raw sensor data from both the laser and the stereo camera, and did the T&R evaluation in offline processing. To account for the aforementioned slight variations in the results,

we ran every experiment five times. The results of the experiment are shown in Figure 19. They are very similar to those in the urban environment. The mean error of the ICP-based system stays nearly constant at around 10 cm. The stereo-based system is slightly more accurate in the nominal case (very close to the taught path), but its error increases approximately linearly with the maximum offset from the original path. Angular variations in terms of maximum rotation of the robot relative to the straight line path were in the same range as in the previous experiment. However, the maximum lateral deviation from the path was 2 m in this experiment, whereas it had been almost 4 m in the previous test. This may be the main reason why the error of the stereo-based system does not reach equally high values as in the urban environment.

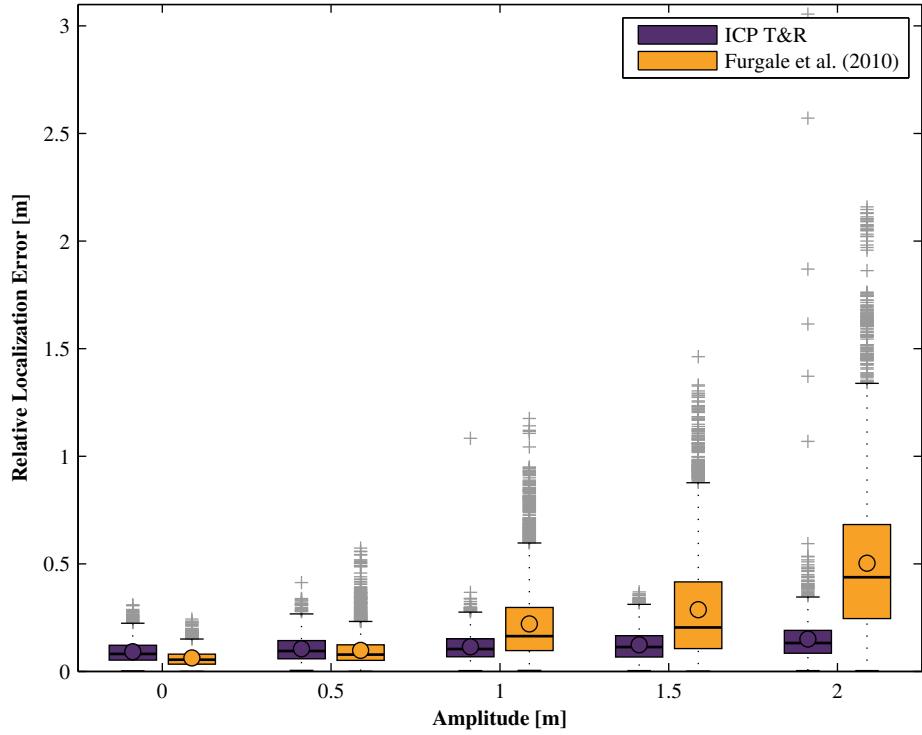


Figure 19: Localization accuracy as a function of the maximum deviation from the original path (unstructured environment). Each box plot contains the data of five experiments, all run offline and with the same data. The stereo-based system suffered from the small FOV of the stereo camera. When the robot turned away from the path, or when it was too far off, localization had to be accomplished by incremental techniques only, due to the lack of overlap between the sensor readings and the map. This lead to comparatively high localization errors, which increase about linearly with the maximum distance to the path. The ICP system is however able to localize precisely even when deviating from the path by several meters, and despite the unstructured and relatively open nature of the terrain without much 3D structure.

7.3.3 Lighting invariance and system reliability evaluation

To demonstrate the lighting invariance of our approach and the operational reliability of the system, we conducted a long-range T&R field test that took place over the course of an entire day. The experiment was carried out in the unstructured, rough terrain environment in Thun. We manually steered the robot over a path of around 900 m, and then let it repeat this route eight times autonomously. There were no dynamic or static obstacles present in the repeat passes. The obstacle avoidance module was not active: the robot was controlled by a simple path tracker (cf. Section 5.2), which made it follow the path that it had driven in the teach run.

Our ICP-based localization and mapping system had to deal with rugged gravel paths, 3D terrain with steep

slopes, and the lack of man-made structure in the surroundings (cf. Figure 22). Figure 20 shows an aerial view of the area and the route that our robot drove. It started and ended at the place marked with a white cross. The first part of the route (to the left of the cross) was the same path that was used in the experiments evaluating the accuracy of our localization system (Section 7.3.1). In the beginning and towards the end of the route, the path was surrounded by small trees and bushes, which hardly had any leaves at the time of the experiment. The middle part of the route ran on more open terrain with less vegetation and higher visibility range.

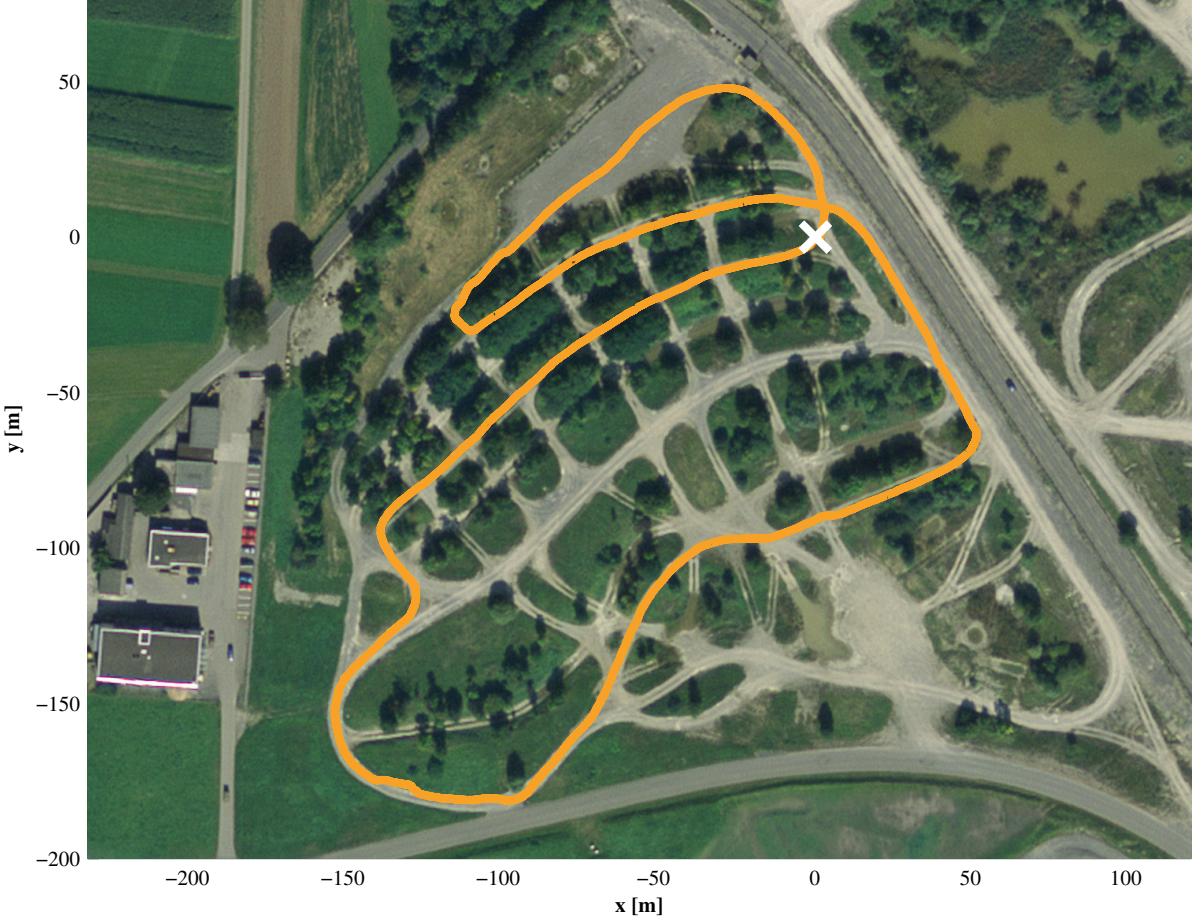


Figure 20: The environment and the robot’s route in our long-range field tests in Thun. The area is a military testing ground for off-road vehicles. It consists of a network of gravel paths, surrounded by trees, bushes and grassland. The paths in the upper part of the image are highly 3D and exhibit considerable inclination, whereas the terrain visible in the lower part is flatter and more open. Our route was a loop of around 900 m, starting and ending at the place marked with the white cross. *Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000*

The experiment was conducted on April 26, 2013. We did the teach run (producing the map shown in Figure 21) in the early afternoon around 1 PM, and started with the first repeat pass shortly after 2 PM. The route was repeated eight times in total, spread over the afternoon and the evening. The weather was sunny in the beginning, but clouds came up later in the afternoon, and there were several short showers. The first five repeat passes were done during daylight between 2 PM and 7 PM. At 8 PM it slowly started to get dark. We did two runs immediately after each other in this twilight time, one at 8:24 PM and the other one at 8:51 PM. We then waited until it was completely dark, to repeat the route once more at 9:52 PM.

Table 2 summarizes the results of the experiment. All eight repeat runs were driven with complete autonomy.

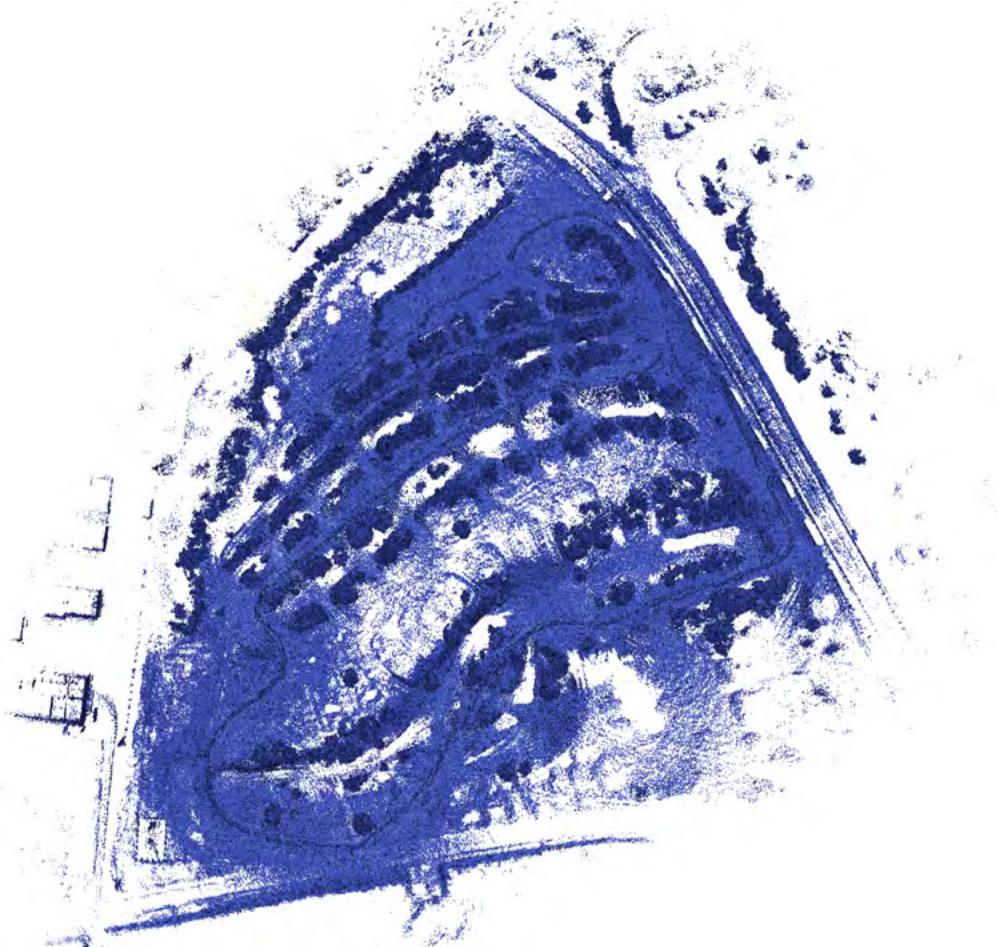


Figure 21: A top view of the pose graph and the laser points included in all submaps created in the Thun long-range experiment. Although there has been no loop closure, the consistency of the map demonstrates the accuracy of the laser-based motion estimation over the kilometer scale. While large-scale metric accuracy is not strictly necessary for T&R, we believe that the high accuracy of our ICP based system can be helpful for many applications, such as path planning or interaction with human operators.

The robot successfully completed the route every time, and it reached its goal position at the start of the route with an accuracy in the range of some centimeters. All parts of our system worked without any failure during the entire test. The robot never deviated from its taught path, nor did any other situation require manual intervention. Overall, our robot covered a distance of more than 7 km in rugged, 3D terrain with an autonomy rate of 100 %.

Table 2: Long-range T&R experiment in unstructured environment: list of repeat runs.

ID	Time (start)	Lighting	Weather	Autonomy
-	01:01 PM	day (cloudy)	dry	<i>manual control (teach pass)</i>
1	02:03 PM	day (sunny, some clouds)	dry	100 %
2	03:20 PM	day (cloudy)	dry	100 %
3	04:14 PM	day (cloudy)	dry	100 %
4	05:41 PM	day (cloudy)	dry	100 %
5	06:43 PM	day (cloudy)	dry	100 %
6	08:24 PM	dawn (cloudy)	dry	100 %
7	08:51 PM	dawn (cloudy)	dry	100 %
8	09:52 PM	night	dry	100 %

In order to compare our ICP-based system to the stereo-based framework, we recorded stereo images of the onboard camera during the entire experiment. In offline processing, we then ran the experiment once more, this time using the camera images and the stereo-based T&R system. Here we focus the comparison on the ability of both systems to localize within a previously built map, over the course of a day, in different weather and lighting conditions. The measurement principle of the two systems is extremely different and, in the absence of ground truth, it is very difficult to provide direct quantitative evaluation of the algorithm performance. Rather, we provide an empirical evaluation based on the values emanating from each system that describe the quality of the localization. The two values of interest are the *localization overlap* in the ICP system (the fraction of the current laser scan that overlaps with the map), and the *number of map matches* in the stereo-based system (the number of visual features in the current image that can be matched with features in the map). In both systems we get a number for these two quantities every time a sensor reading is processed during the repeat passes.

For the ICP system, the mean overlap remains almost constant over all eight repeat runs (Figure 23). This was expected, as the output of the Velodyne laser is very consistent over widely varying lighting conditions. Moreover, the overlap values exhibit very little variance: with the exception of a few outliers, the overlap always remains between 0.89 and 0.9 in the entire experiment.

With the stereo-based method, the picture is completely different. Figure 24 shows how the number of map matches is dependent on ambient lighting, and therefore on the time of day. Sunset was at 8:35 PM, in the middle of repeat run 6, but it was only between run 7 and 8 that it got completely dark. Equipped with automatic exposure control, the stereo camera provided sufficiently exposed images during the entire day and even in the beginning of dawn; until run 6, the algorithm produced enough map matches for successful localization. Starting from repeat run 7 (15 minutes after sunset), there was too little ambient light for capturing images that would allow extraction of a sufficient number of visual features. As a consequence, the average number of map matches decreased to almost 0, and localization failed. In the last run, we turned on the robot’s headlights. However, they only lightened the vicinity of the robot, and therefore only a small portion of the image (cf. Figure 24, bottom right). The algorithm was able to detect some features in the lightened part, but far too few for localization within the map.

We can further observe that the variance in the number of map matches over a single run is much higher than the variance of the overlap in the ICP-based system, although the global lighting conditions can be assumed to be almost constant over the duration of a single repeat pass. Potential reasons for this are local lighting changes over the course of the path, motion blur due to bumpy vehicle motion, and the fact that the stereo camera is a directed sensor with a limited FOV, unlike the omnidirectional Velodyne laser scanner.



Figure 22: ARTOR in different sections of the long-range route following experiment in Thun. The environment consisted of gravel paths surrounded by trees, bushes and grassland. The pictures at the bottom show the high degree of three-dimensionality in the terrain. Unlike in the urban environment, driving on these paths included considerable pitching and rolling of the robot, as well as high vibrations due to the bumpy terrain and the lack of suspension on the robot.

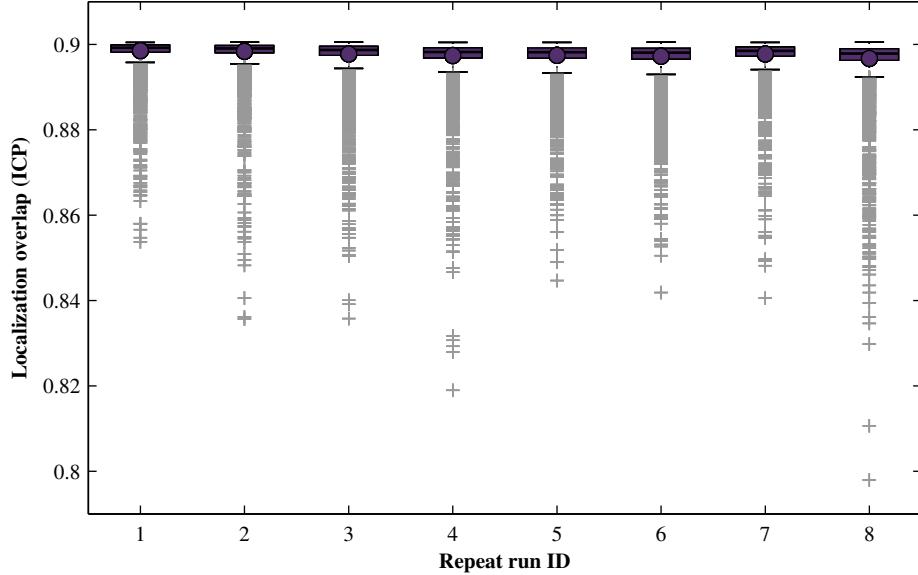


Figure 23: Localization overlap of the ICP-based system in the eight repeat runs. The overlap value is a measure of how consistent the laser scans acquired during the repeat runs are with the previously recorded map. The results show that the overlap is very stable; it exhibits very little variance over the course of any single repeat pass, as well as over the entire day.

7.4 The complete system: long-range adaptive teach and repeat in dynamic environments

Having verified our system’s conformance to the requirements in terms of localization (accuracy, robustness to path deviations, invariance to environmental conditions) and obstacle avoidance capability, we finally tested the performance of the complete system in an extensive T&R experiment in a dynamic urban environment. We taught the system a 1.3 km path by driving the robot manually, and let it repeat the route 12 times in autonomous mode. The repeat runs were conducted on five different days in March and in May, in a large variety of different weather and lighting conditions, during rush hour as well as late in the evening. The system had to cope with both static and dynamic obstacles obstructing the taught path. In total, our robot autonomously drove a distance of more than 15 km.

The teach pass of the experiment was done in dry weather during the night on March 12, 2013. The route was a loop of around 1.3 km (Figure 25 and Figure 26). The first half of the path consisted of a quiet side street with parked cars on either side (Figure 27), the other half was a rather busy road with a lot of traffic including cars, trucks, bicycles and trams, as well as pedestrians (Figure 29). During the second half, we drove the robot on the sidewalk or on the bike lane at the border of the street. In the first half, we drove in the middle of the road.

The first part of the experiment was conducted in March 2013. Our robot repeated the taught route four times: once immediately after teaching the path, and three times on the day after. Our system had to cope with a large variety of weather and lighting conditions. The first and the last repeat run were done at night, the remaining two during the day. The weather was sunny and dry during the second repeat run in the morning, but it closed in towards the evening. It was raining when we did the third run in the afternoon, and started to snow during the last pass. Neither these drastic variations in lighting and weather conditions, nor changes in the environment (e.g. parked cars and trucks), nor the dynamic elements of the scene (pedestrians, cars, trams) presented a problem to our ICP-based mapping and localization system. In all the runs the robot completed the route and reached its goal position with centimeter accuracy.

We only had to intervene manually when the path was obstructed by obstacles that had not been present

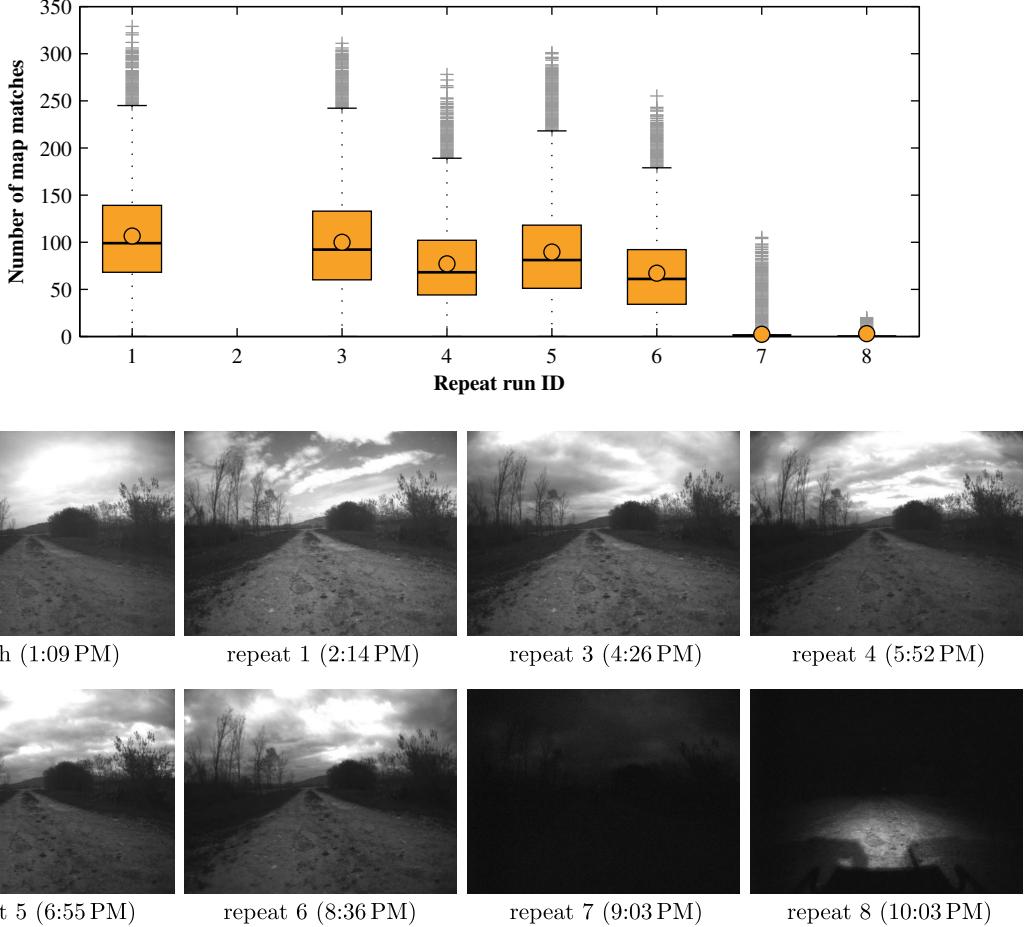


Figure 24: Number of map matches of the stereo-based system in the eight repeat runs (top), and images of the Bumblebee’s left camera in the middle of the repeat runs (bottom). The evaluation of the stereo-based T&R framework was done in offline processing, using stereo images recorded during the experiments with the ICP-based system. There is no evaluation of repeat run 2, since (due to a technical problem) no images were recorded during this run. The graph shows the dependency on ambient light of stereo-based localization. The last three repeat runs were done after sunset. The camera’s automatic exposure control was able to compensate for the fading light only until run 6. The headlights of the robot, that had been turned on for the last run, only lightened the vicinity of the robot, which was a too small portion of the image to provide enough visual features for localization within the map recorded at daylight.

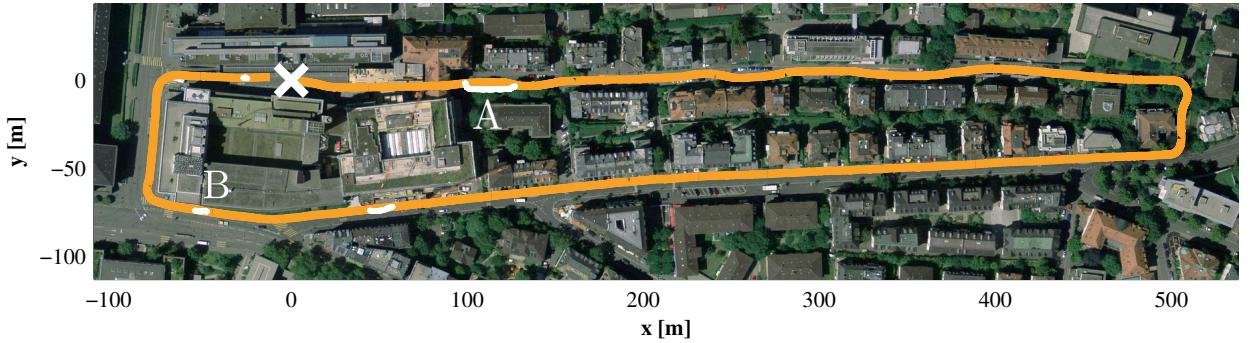


Figure 25: The route in our long-range field tests in the urban environment around ETH Zurich. The path was a loop of around 1.3 km, which we drove in clockwise direction at a speed of approximately 1 m/s. The four sections of the path highlighted in white are those parts that required manual intervention in one of the runs during the first part of the experiment (without obstacle avoidance), as the path was blocked by objects that had not been there in the teach run. Pictures of the obstacles at the locations marked with A and B are shown in Figure 28. *Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000*

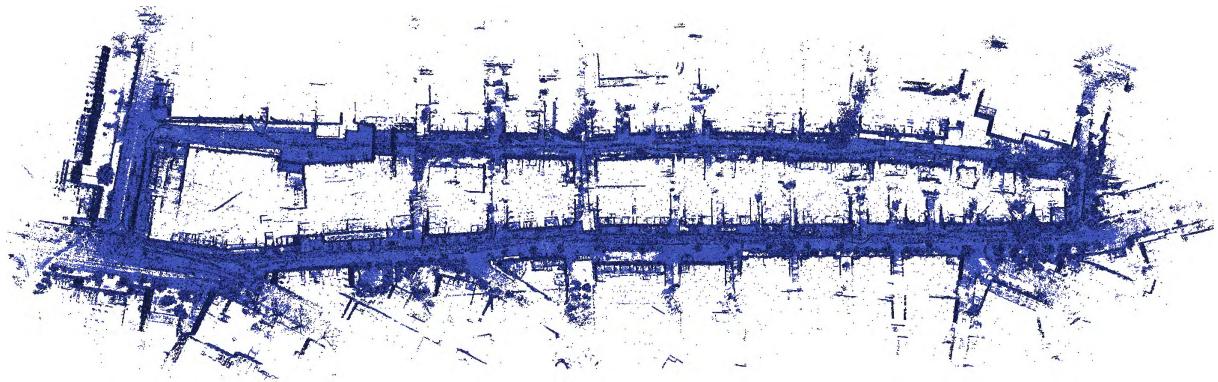


Figure 26: A top view of the pose graph and the laser points included in all submaps created in the Zurich long-range experiment. Once again, the consistency of the map demonstrates the accuracy of the laser-based motion estimation over the kilometer scale.

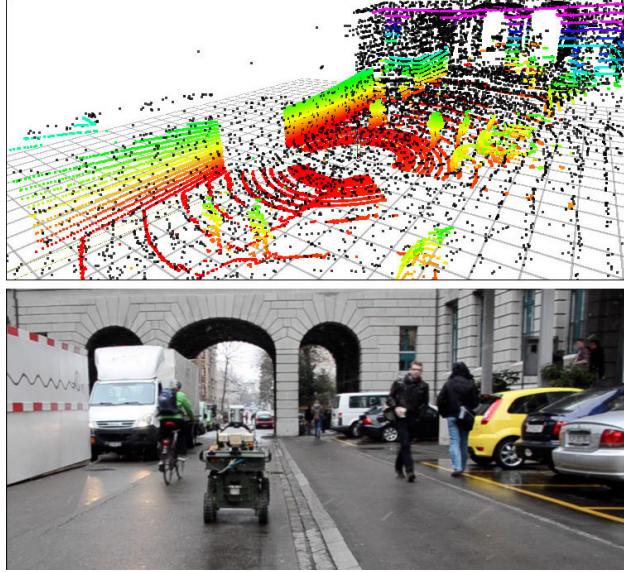


Figure 27: A typical scene encountered during the long-range experiment. As the map was recorded during night, several new elements were observed during the repeat passes. In this example from the repeat run at 3 PM, multiple pedestrians, a cyclist, a parked white truck and several cars were unknown to the robot. The upper picture shows the map recorded during the teach run (black points), and the current laser data (colored points) at the situation shown in the photograph below.

when the route had been taught. At the time of these experiments, we had not yet integrated our obstacle avoidance scheme into the system; when driving in autonomous mode, the robot was controlled by a path tracker, which made the robot follow the learned path irrespective of obstacles or other changes in the environment. In four cases—twice in the second and twice in the third repeat run—we had to steer the robot manually over a short distance, in order to avoid collisions. The passages that required manual control in some of the repeat runs are highlighted in Figure 25. Figure 28 gives an impression of the type of obstacles our robot encountered on the path. Although avoiding these obstacles required deviating from the original path by several meters, our system was able to localize correctly at all times. This meant that immediately after the obstacle had been passed we could switch back to the autonomous mode and the robot continued its route. Overall, our robot covered a distance of over 5 km, of which it drove 99 % autonomously.



Figure 28: Two examples of situations that required manual intervention in the experiment without obstacle avoidance. The letters A and B show the location of these places on the map in Figure 25. Situations like these are not only challenging in terms of obstacle avoidance, but also for localization algorithms, as considerable parts of the environment have changed with respect to the recorded map. In the example on the left, both the truck and the digger had not been there when we taught the path.

The described experiment has shown that our ICP-based mapping and localization system was able to cope with highly dynamic surroundings and significant changes in the environment. However, this was not true

for our system’s motion planning scheme (the path tracker). We have seen that a route following system without the capability of locally adapting the vehicle’s path is not feasible for applications in any non-static environment. For the second part of the experiments, we therefore integrated the obstacle detection module and the local motion planner described in Section 6, where the latter replaced the path tracker used in the previous experiment. We conducted the long-range experiment with integrated obstacle avoidance more than two months after the initial tests. Nevertheless, we did not record a new map of the route, but re-used the one recorded on March 12. In the meantime, however, the environment had changed considerably: trees and bushes had leaves again, construction sites on the route had changed, and (most probably) none of the cars parked along the streets were still at the same place.

Table 3: Long-range T&R experiment in urban environment: list of repeat runs.

Date	Time (start)	Lighting	Weather	Obstacle avoidance	Autonomy
March 12	10:33 PM	night	dry	<i>manual control (teach pass)</i>	
March 12	11:07 PM	night	dry	off	100 %
March 13	10:59 AM	day (sunny)	dry	off	97.8 %
March 13	03:05 PM	day (cloudy)	raining	off	98.3 %
March 13	07:12 PM	night	snowing	off	100 %
May 21	07:31 PM	dawn (cloudy)	raining	on	100 %
May 23	03:25 PM	day (cloudy)	raining	on	100 %
May 23	05:27 PM	day (cloudy)	raining	on	100 %
May 23	08:03 PM	dawn (cloudy)	raining	on	100 %
May 23	10:03 PM	night	dry	on	100 %
May 25	02:47 PM	day (cloudy)	raining	on	100 %
May 25	05:45 PM	day (cloudy)	dry	on	99.9 %
May 25	07:30 PM	day (sunny)	dry	on	100 %

We did eight repeat passes in total, which amounts to more than four hours of driving and an overall distance traveled of more than 10 km. A list of the repeat runs of the entire experiment is given in Table 3. When we started with the first run in the evening of May 21, 2013 it was raining and getting dark on the way. We continued the experiment on May 23, where we did four repeat passes, two in daylight in the afternoon, one during dusk at 8 PM, and the last one in the night at 10 PM. It was again a rainy day; only the last run was done in dry weather. On May 25 we did another three repeat passes, one around 3 PM in light rain, one at 5:45 PM, and one at 7:30 PM. During the second-to-last run it was not raining any more, but the roads were still wet, while for the last run it was completely dry. In most of the runs the robot had to cope with several static or dynamic obstacles. The static obstacles were mainly bicycles and cars parked at the side of the road or on the road. Dynamic obstacles were present in the form of pedestrians and oncoming cars. The latter were encountered on the first part of the route, where the robot drove on the road. Towards the end of the loop, the route ran on a rather narrow sidewalk for around 100 m (cf. Figure 10, left). During the afternoon runs, many pedestrians were passing in this section. Most of them avoided the robot, as they would do for a person crossing their paths. We also encountered several situations where strangers deliberately stepped in the way of our robot to test whether or not it would stop. Our obstacle detection and avoidance scheme handled all these cases without any intervention from our side.

There was one single situation where we had to drive the robot manually for about 1 m (in the second-to-last run), as the obstacle detection module reported an obstacle that did not actually exist, and only appeared when the robot was already too close to it to avoid it in forward motion. It would then reverse, go straight forward again, as the obstacle had disappeared, and stop again a few meters further, when it “saw” the obstacle again. This problem will be discussed further in Section 8.3.

Apart from this situation, the robot drove the entire distance in all eight repeat passes completely autonomously. We always walked next to it, but we never manually stopped it for safety reasons, nor did we encourage pedestrians, cyclists, or car drivers to move out of the way. All the relevant obstacles were



Figure 29: Examples of dynamic obstacles encountered during the long-range route following experiments in Zurich. A part of the route was driven on a sidewalk, and the robot had to detect and avoid people walking by or purposefully obstructing its path.

detected reliably, and the robot either drove around them when a safe path was available, or stopped when there was no room for an avoidance maneuver in forward motion (e.g., when a pedestrian abruptly stepped in its way).

8 Discussion

In this section, we summarize and discuss the results of the comparison of our ICP-based system to the stereo vision T&R framework, and discuss the impact of noise and dynamic environments on ICP-based registration. Moreover, we list the lessons learned during the long-range T&R experiments and discuss some important remaining challenges.

8.1 Comparison to stereo vision T&R system

In Section 7.3, we presented three different experiments where we compared our localization and mapping system to the stereo-based approach of Furgale and Barfoot (2010). It is important to see that the two systems are different in three aspects: perception technique (laser/active vs. camera/passive), sensor FOV (omnidirectional/360° vs. directed/100°), and registration method (point-based/ICP vs. appearance-based/feature extraction and matching). Nevertheless, we can draw a number of important conclusions from the comparison of the two approaches.

Localization accuracy in the nominal scenario. The experiments of Section 7.3.1 can be seen as “proof of concept” for using a spinning laser and ICP for localization and mapping in a T&R system, instead of an appearance-based method as in state-of-the-art works. We were able to show that we can achieve similar or even slightly better localization accuracy when repeating the previously driven route, both in unstructured 3D terrain and in dynamic urban environments. For successful operation, both registration methods place some requirements on the environment: appearance-based techniques rely on a certain amount of *texture*, ICP is dependent on sufficient *3D structure* in the surroundings. For our experiments, we chose places where both of these requirements were fulfilled, in order to allow for a fair comparison. However, it is clear that ICP would fail for example in a large, flat plane lacking any 3D structure within the range of the sensor. On

the other hand, appearance-based approaches would break down for example in a room where all surfaces are uniformly painted in the same color.

Sensor field of view and range. Deviations of some meters from the taught path, which typically include high rotations of the robot, are necessary in many real-world applications where obstacles may block the robot’s path. If a T&R system is supposed to work reliably, it must be able to provide accurate localization even in these difficult situations. The results of the experiments in Section 7.3.2, where we assessed the robustness to path deviations of both systems, have clearly demonstrated the advantages of omnidirectional sensors compared to directed perception devices. It has been shown that even small deviations from the route driven in the mapping stage force the 100°-FOV stereo system—unlike the ICP system—to fall back on incremental localization (VO), due to the lack of overlap between the camera images and the map. The consequence of this is the accumulation of error. While the extent of drift is dependent on the camera and on the specific VO algorithm used (our implementation yielded relatively high errors), it remains a fact that no incremental localization method is drift-free, and therefore localization against the map is required for a system to be truly scalable to routes of arbitrary length. In addition to its 360° FOV, our laser scanner has the advantage of providing precise distance data with a high maximum range of around 80 m, whereas the accuracy of stereo-based range measurements decreases quadratically with the distance from the camera (Lemaire et al., 2007). We therefore assume that we could go even further away from the taught path than we tested in the experiments, while still being able to localize within the map. However, this conjecture remains to be verified in future experiments.

Perception technique. The experiments in Section 7.3.3, where our robot repeated the same route over the course of an entire day, have clearly shown the superiority of laser-based systems over camera-based approaches in terms of dependency on environmental conditions. While the fact that cameras require a certain amount of ambient light is neither new nor surprising, it is still an important result that a laser scanner—being an active sensor—enables constant performance irrespective of the weather and the time of day. Adding artificial light sources could be a solution for vision-based systems to enable operation in darkness. Our robot is equipped with two headlamps, which we turned on for the last repeat pass of the experiment described in Section 7.3.3, where it was completely dark. In the end, we were not able to even start our stereo-based localization system on the images recorded in this run. Although the two headlamps are reasonably powerful (10 W LED each), only a small area in front of the robot was sufficiently illuminated. The visual features in the map, however, were spread over the entire image (except the sky), including points dozens of meters away from the robot. As a consequence, the lightened area did not provide enough features for localization against the map. Stronger light sources could improve the feature extraction, yet they will never range to the horizon in a generic outdoor setting. Moreover, we observed in our data set that the limited dynamic range of the camera likewise contributes to the problem. Nearby areas in the image are well illuminated and often over-exposed, while the background typically remains completely dark. Husmann and Pedersen (2008) built a system for camera-based navigation in the dark, using Point Grey Flea cameras and a 24 W LED spotlight. Our observations are consistent with their experiments. They computed that, to allow for a reasonably short exposure time (10 ms), illumination of a 90° FOV with a 10 m lookahead would require a 56 × increase in peak light power, which is clearly unfeasible. The problem was finally solved by computing high dynamic range (HDR) images, which however required the robot to stop for image acquisition.

Registration method. From our experiments, we cannot draw any general conclusions about point-based versus appearance-based methods for robotic localization. A meaningful comparison of ICP and appearance-based registration would require an omnidirectional laser sensor with a vertical resolution high enough for sparse feature-based approaches, such as the one of McManus et al. (2013b). Taking the opposite approach, that is, running ICP on point clouds with an (artificially) limited, narrow FOV, involves the risk of obtaining results that are highly dependent on the environment. With an omnidirectional sensor, it can be assumed that in most practice-relevant cases there will be sufficient 3D structure in the point cloud for ICP registration. This is no longer true with a narrow FOV: success or failure of registration may heavily depend on the environment and the orientation of the sensor relative to it. Irrespective of localization performance, there is one clear advantage of the point-based nature of our approach. The 3D point cloud maps built by the ICP

pipeline are large, dense, and accurate. Apart from their actual purpose of localization, they can be useful for many other applications, such as visualization, terrain assessment, or path planning.

8.2 ICP in dynamic outdoor environments

Under ideal conditions, that is, with a noise-free, static sensor in a static environment, sensor readings can be perfectly matched with the map recorded in an earlier phase. In real-world scenarios, as for example in our long-range experiments in the urban environment (Section 7.4), there are three different sources of disturbances: dynamic objects in the environment, the motion of the sensor during data acquisition, and noise due to rain, snow, reflections, or other effects. All these factors distort the acquired point clouds and therefore potentially have an impact on the localization performance.

We will start the discussion with the last point, the noise due to different environmental conditions and sensor characteristics. The teach pass for the long-range experiment in the urban environment has been recorded in dry weather, so the map could not be corrupted by noise due to rain or snow. However, some of the repeat runs have been conducted in rain and in snowfall. We inspected the point clouds recorded in both cases, and noticed that rain did not cause a mentionable amount of noise. We discovered, however, that our laser sensor produces considerable random noise at low temperatures (some of the experiments have been conducted around 0°C). In the snow, it is therefore hard to tell whether the noise originates from the snowflakes or just from the temperature. Overall, we can say that the noise did not have any visible impact on our system. The outlier rejection scheme in the matching step of the ICP pipeline (cf. Section 4) was able to adequately handle this data corruption.

The same is true in most cases for disturbances due to dynamic objects in the environment. In general, it can be stated that the ICP matching works correctly as long as noise and distortions due to dynamic scenes are not the main content of the point cloud. Otherwise, the outlier rejection would fail to separate the useful points from the disturbances. This can happen, for example, if a group of people is standing closely around the robot, obstructing nearly the entire FOV of the sensor. However, under typical operating conditions⁷, such as those in the route following experiment reported in Section 7.4, changes and dynamic elements in the environment did never cause the localization to fail. Particularly challenging situations were encountered when driving along a busy street (cf. Figure 10 (left)), where at times at least 10 cars were in the FOV of the laser, as well as several pedestrians and sometimes trams and cyclists (Figure 29). All of these elements were clearly not in the map that the ICP module used for registration. The experiments also involved several cases where a considerable part of the sensor’s FOV was blocked by parked vehicles (close to the robot’s path) that had not been present when teaching the route, such as the truck in Figure 28 (left).

The third source of disturbance is the distortion of the laser scans due to the vehicle motion. The Velodyne laser rotates around the vertical axis with a frequency of 10 Hz, providing around 180 individual data packets per revolution, each spanning an angle of approximately 2°. Currently, we build 360° point clouds by assembling these packets assuming that the sensor has not moved during one revolution (0.1 s). With the typical speed of our robot being 1 m/s, this yields an (erroneous) offset of 10 cm between the first and the last point of the revolution. The maximum error of point measurements can be reduced to 5 cm by defining the time stamp of the assembled point cloud appropriately, namely by taking the value after half a revolution. This error is in the same order of magnitude as the standard deviation of the range measurements according to the sensor specifications (2 cm at 25 m), which is why we currently neglect it. Nevertheless, the localization errors that we measured are in the same range as well. In the future, especially with regard to driving at higher speeds, we plan to account for the motion of the sensor when building the omnidirectional point clouds, for example, using an IMU and incorporating continuous-time motion estimation (Bosse et al., 2012; Anderson and Barfoot, 2013; Furgale et al., 2012). This might enable further improvements in terms of localization accuracy.

⁷We are targeting applications such as search and rescue or surveillance, which normally do not involve moving through dense crowds of people.

8.3 Lessons learned and remaining challenges

The long-range route following in unstructured 3D terrain and in a dynamic urban environment (Sections 7.3.3 and 7.4, respectively) have shown that our system enables reliable autonomous navigation and yields repeatable results under a variety of different environmental conditions. Despite these encouraging results, the system is not perfect. In the following, we point out the main difficulties we encountered during these long-range field tests, we show what situations could make the system fail, and we highlight the challenges that remain to be tackled.

ICP failure detection. A drawback of our current ICP implementation compared to appearance-based feature pipelines is the lack of appropriate means for failure detection in the registration. In each ICP registration step, we can see whether or not the algorithm converged, and we measure the percentage of overlap between the sensor readings and the map. However, convergence does not imply that the *correct* local minimum has been found, and the overlap is—unlike the number of feature matches in appearance-based methods—not a measurement for which an absolute minimum value for localization success can be defined. As a consequence, it is impossible to (re)localize, if the current pose estimate is too far away from the true value. We found a set of parameters for the registration module that in the experiments has shown to enable localization and mapping in different types of environments without such failures. The most important parameters in this respect are the upper bounds on translation and rotation jumps between two registration steps, which we set to 2 m and 0.4 rad, respectively. Nevertheless, a mechanism for detection of cases where ICP converges to a wrong local minimum would increase the robustness of the localization system and should therefore be addressed in future work. One possible approach is to involve incremental motion estimates produced by other sensors, such as wheel odometry or an IMU.

Robot control in 3D terrain. Although not relevant for the success of the route following missions, the experiments on 3D terrain (Section 7.3.3) disclosed a downside of the design of our robotic vehicle. Its raised front axle, together with the center of gravity location only slightly behind the middle wheels, causes the robot to tilt forwards when driving down a slope (cf. Figure 22, bottom left). The low-level motor controllers are tuned for skid-steering with both the middle and the rear wheels on the ground. Turning in this situation requires high forces, as the rear wheels have to move sideways (the center of rotation is very close to the middle axis). However, when only the middle axis has solid ground contact, much less torque is required on the motors for making the robot turn. As our controller does not adapt to these different conditions, it overshoots, which results in oscillations in the yaw angle. Although this behavior is certainly undesirable, we can report that it neither affected the vehicle’s safety, nor did the fast pitch and yaw motions present a problem for our ICP-based localization system.

Low-level vehicle software. We had some problems with the low-level vehicle control software. In two repeat runs, the robot stopped at some point, because of a low-level software crash. In order to restart it, we had to open the lid of the box on top of ARTOR. This meant that the Velodyne was tilted by about 60°. We opened and closed the lid very slowly to avoid that the ICP would lose track of the robot’s pose. In both cases we were able to continue the route without restarting any of the high-level programs, such as the ICP-based localization, the obstacle detection, or the motion planner.

Obstacle detection: false positives. In general, our obstacle detection method worked reliably. However, we have observed that sometimes spurious obstacles are detected when the pavement is wet and the robot is tilted onto its front wheels (the raised front axle of ARTOR yields a high pitch angle when the robot is standing on the middle and the front wheels instead of the middle and the rear wheels). We encountered this problem several times in preliminary tests; in the eight repeat runs with the complete system it only occurred once (as described in Section 7.4). We did not yet investigate the phenomenon in detail, but after a preliminary analysis of the sensor data recorded during such a situation we believe that the source of the problem is not our obstacle detection algorithm, but rather erroneous measurements of the laser scanner under these special conditions, possibly due to the laser reflecting off of the wet pavement.

Spurious negative obstacles. Cavities in the ground with a lattice-like cover—such as certain manholes,

for example—are often classified as negative obstacles by our algorithm. While the reason for this is obvious (some laser beams go through the lattice, as if it was not there), it is still undesirable and will have to be addressed in the future, for example using a solution similar to the one of Kümmeler et al. (2013). In our experiments, the misclassification of lattice-covered gaps merely resulted in the robot avoiding each manhole on the side of the street. However, in theory, it could lead to a completely blocked a path without reason.

Computational resources. All software components of our system currently run on a single laptop on the robot. When we added the local planner, we observed that the rate at which the ICP was able to compute localization updates dropped from 10 Hz to around 5 Hz. This means that roughly every second Velodyne scan is discarded. Although in the experiment we could not see an influence on the performance of our system due to the reduced update rate, this clearly shows that the onboard computer is working at its limit.

Delay and dynamic obstacles. In the current implementation, obstacles are treated as static objects: in every time step the obstacle map is updated according to the sensor readings, but the obstacles are not projected into the future. Together with relatively high delay in the entire pipeline, this sometimes leads to undesirable behavior. For example, we observed that when a cyclist was overtaking the robot and came close to it, it was detected as an obstacle, which lead to an avoidance maneuver when the cyclist was already out of the way again.

9 Conclusion

This paper presented a T&R algorithm designed for search and rescue and surveillance scenarios. We showed how to use ICP to construct the building blocks of T&R: laser odometry, submap construction, and localization. The resulting system is highly accurate and robust, and is invariant to lighting changes. Moreover, we developed an adaptive route following scheme that enables to reliably avoid collisions by temporarily leaving the taught path, yielding a system that is suitable for fully autonomous operation in changing and dynamic scenes. We demonstrated the gains in robustness to path deviations and environmental conditions by comparing against a state-of-the-art approach based on stereo vision in two different types of terrain. The experiments revealed the advantages of omnidirectional, active sensors for mobile robot localization and mapping. We finally demonstrated the system driving autonomously over seven kilometers in unstructured 3D terrain and over 15 kilometers in a highly dynamic scene, in a wide range of environmental conditions.

Our obstacle detection and local motion planning system is tailored to navigation in 2D environments. We did not test it yet in highly non-flat terrain, such as the testing area in Thun, but we assume that both the obstacle detection and the local planning method would require some adaptations to enable equally smooth and reliable navigation as in the urban environment. Another limitation of the current system is inherent in the local nature of our motion planner: despite allowing for deviations of some meters from the reference path, we are still restricted to remain relatively close to the taught route throughout the operation. In future work we plan to address these two limitations by using the constructed 3D maps for motion planning and extending the planner’s horizon, for navigating through large, unstructured, 3D environments without being restricted to a single reference path.

Acknowledgments

This work is supported by the armasuisse S+T UGV research program, the EU FP7 IP projects Natural Human-Robot Cooperation in Dynamic Environments (ICT-247870) and the EU FP7 2007-2013 Program, Challenge 2, Cognitive Systems, Interaction, Robotics, under grant agreement No 269916, V-Charge.

A Video Attachment

The video accompanying this article presents one repeat run of our robot in the dynamic urban environment (cf. Section 7.4), namely the one conducted at 3:25 PM on May 23. It shows the robot autonomously driving the entire 1.3 km route, and simultaneously visualizes all the important system components: the submap and the current scan used for localization, the obstacle map, the reference path, and the local trajectory generated by the motion planner.

References

- Anderson, S. and Barfoot, T. D. (2013). Towards Relative Continuous-Time SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1033–1040, Karlsruhe, Germany.
- Arya, S. and Mount, D. (1993). Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 271–280, Austin, TX.
- Baldwin, I. and Newman, P. (2012a). Laser-only road-vehicle localization with dual 2D push-broom LIDARS and 3D priors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2490–2497, Vilamoura, Portugal.
- Baldwin, I. and Newman, P. (2012b). Road vehicle localization with 2D push-broom LIDAR and 3D priors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2611–2617, Anchorage, AK.
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.
- Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The International Journal of Robotics Research*, 23(12):1113–1139.
- Bosse, M., Zlot, R., and Flick, P. (2012). Zebedee: Design of a Spring-Mounted 3-D Range Sensor with Application to Mobile Mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119.
- Chang, C.-K., Siagian, C., and Itti, L. (2012). Mobile Robot Monocular Vision Navigation Based on Road Region and Boundary Estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1043–1050, Vilamoura, Portugal.
- Cherubini, A. and Chaumette, F. (2013). Visual navigation of a mobile robot with laser-based collision avoidance. *The International Journal of Robotics Research*, 32(2):189–205.
- Churchill, W. and Newman, P. (2012). Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4525–4532, Staint Paul, MN.
- Diosi, A., Remazeilles, A., Segvic, S., and Chaumette, F. (2007). Outdoor visual path following experiments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4265–4270, San Diego, CA.
- Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516.
- Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560.
- Furgale, P., Barfoot, T. D., and Sibley, G. (2012). Continuous-time batch estimation using temporal basis functions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2095, Staint Paul, MN.
- Goedemé, T., Nuttin, M., Tuytelaars, T., and Van Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236.

- Hähnel, D. and Thrun, S. (2008). 3D Laser Based Obstacle Detection for Autonomous Driving. In *Workshop on 3D-Mapping, IEEE International Conference on Intelligent Robots and Systems (IROS)*, Nica, France.
- Howard, A. (2004). Multi-robot mapping using manifold representations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4198–4203, Barcelona, Spain.
- Husmann, K. and Pedersen, L. (2008). Strobe Lit High Dynamic Range Stereo Imagery for Dark Navigation. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Hollywood, CA.
- Kümmerle, R., Ruhnke, M., Steder, B., Stachniss, C., and Burgard, W. (2013). A Navigation System for Robots Operating in Crowded Urban Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Staint Paul, MN.
- Lemaire, T., Berger, C., Jung, I.-K., and Lacroix, S. (2007). Vision-Based SLAM: Stereo and Monocular Approaches. *International Journal of Computer Vision*, 74(3):343–364.
- Levinson, J. and Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378, Karlsruhe, Germany.
- Marshall, J., Barfoot, T. D., and Larsson, J. (2008). Autonomous Underground Tramming for Center-Articulated Vehicles. *Journal of Field Robotics*, 25(6-7):400–421.
- Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 83–88, Minneapolis, MN.
- McManus, C., Furgale, P., and Barfoot, T. D. (2013a). Towards lighting-invariant visual navigation: An appearance-based approach using scanning laser-rangefinders. *Robotics and Autonomous Systems*, 61(8):836–852.
- McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2013b). Lighting-invariant Visual Teach and Repeat Using Appearance-based Lidar. *Journal of Field Robotics*, 30(2):254–287.
- Meilland, M., Comport, a. I., and Rives, P. (2010). A spherical robot-centered representation for urban navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5196–5201, Taipei, Taiwan.
- Modayil, J., Beeson, P., and Kuipers, B. (2004). Using the Topological Skeleton for Scalable Global Metric Map-Building. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1530–1536, Sendai, Japan.
- Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148.
- Pomerleau, F., Magnenat, S., Colas, F., Liu, M., and Siegwart, R. (2011). Tracking a depth camera: Parameter exploration for fast ICP. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3824–3829, San Francisco, CA.
- Rasmussen, C., Lu, Y., and Kocamaz, M. (2012). A Trail-Following Robot Which Uses Appearance and Structural Cues. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Matsushima, Japan.
- Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, Quebec City, Canada.
- Schwesinger, U., Rufli, M., Furgale, P., and Siegwart, R. (2013). A Sampling-Based Partial Motion Planning Framework for System-Compliant Navigation along a Reference Path. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IVS)*, Gold Coast City, Australia.
- Siagian, C., Chang, C.-K., and Itti, L. (2013). Mobile Robot Navigation System in Outdoor Pedestrian Environment Using Vision-Based Road Recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.

- Stenning, B., Osinski, G. R., Barfoot, T. D., Basic, G., Beauchamp, M., Daly, M., Dong, H., Francis, R., Furgale, P., Gammell, J., Ghafoor, N., Lambert, A., Leung, K., Mader, M., Marion, C., McCullough, E., McManus, C., Moores, J., and Preston, L. (2012). Planetary surface exploration using a network of reusable paths. In *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Turin, Italy.
- Stenning, B. E., Mcmanus, C., and Barfoot, T. D. (2013). Planning using a Network of Reusable Paths: A Physical Embodiment of a Rapidly Exploring Random Tree. *Journal of Field Robotics*, 30(6):916–950.
- Stewart, A. and Newman, P. (2012). LAPS - Localisation using Appearance of Prior Structure: 6-DOF Monocular Camera Localisation using Prior Pointclouds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Staint Paul, MN.
- Thrun, S., Gutmann, J.-S., Fox, D., Burgard, W., and Kuipers, B. J. (1998). Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach. In *Proceedings of the National Conference on Artificial Intelligence*, pages 989–996, Madison, WI.
- Tomatis, N., Nourbakhsh, I., and Siegwart, R. (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1):3–14.
- Trulls, E., Murtra, A. C., Pérez-Ibarz, J., Ferrer, G., Vasquez, D., Mirats-Tur, J. M., and Sanfeliu, A. (2011). Autonomous Navigation for Mobile Service Robots in Urban Pedestrian Environments. *Journal of Field Robotics*, 28(3):329–354.
- Šegvić, S., Remazeilles, A., Diosi, A., and Chaumette, F. (2009). A mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 113(2):172–187.
- Zhang, A. M. and Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. *The International Journal of Robotics Research*, 28(3):331–356.