

There and Back Again

Dealing with highly-dynamic scenes and long-term change
during topological/metric route following

Paul Furgale¹, Philipp Krüsi¹, François Pomerleau², Ulrich Schwesinger¹, Francis Colas¹, and Roland Siegwart¹

Abstract—Topological/metric route following, also called teach and repeat (T&R), enables long-range autonomous navigation even without globally consistent localization. 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. This talk will present our efforts to develop a T&R system suitable for long-term robot autonomy in highly dynamic, unstructured environments. We use the fast iterative closest point (ICP) algorithms from *libpointmatcher*¹ to build a T&R system based on a spinning laser range finder. The system deals with dynamic elements in two ways. First, we employ a system-compliant local motion planner to react to dynamic elements in the scene during route following. Second, the system infers the static or dynamic state of each 3D point in the environment based on repeated observations. The velocity of each dynamic point is estimated without requiring object models or explicit clustering of the points. At any time, the system is able to produce a most-likely representation of underlying static scene geometry. By storing the time history of velocities, we can infer the dominant motion patterns within the map. The result is an online mapping and localization system specifically designed to enable long-term autonomy within highly dynamic environments. We validate the approach using data collected around the campus of ETH Zurich over seven months and at an outdoor 3D test site in Thun, Switzerland.

Index Terms—Long-term mapping, dynamic obstacles, ICP, kd-tree, registration, scan matching, robot, SLAM.

I. INTRODUCTION AND RELATED WORK

This short paper is an overview of our work at the ETH Zurich Autonomous Systems Lab to build a system capable of long-term topological/metric route following (often called teach and repeat (T&R)) that is suitable for navigation in rough terrain and dynamic environments. Our goal is to be able to deal with dynamic environments over all time scales, both slow scene changes over time, and the presence of moving obstacles such as pedestrians or cars. Our work follows the basic structure of the stereo-camera-based T&R algorithm described in [1]. However, as camera-based localization algorithms are inherently sensitive to changes in lighting, we have decided to build our system around a Velodyne HDL-32E spinning laser rangefinder (Fig. 1). While there have been other rough-terrain-capable T&R algorithms based on lidar in the past [2], [3], these systems used



Fig. 1. ARTOR, a search and rescue robot specialized for outdoor applications navigating in highly dynamic urban environments. Typical mobile elements include pedestrians, bikers, cars, trucks and trams.

an appearance-based approach that requires high-resolution intensity/range images. Instead, our work uses the iterative closest point (ICP) algorithm which is more suitable for our chosen lidar and produces highly accurate motion estimates.

This paper is not meant to present novel research. Rather, it summarizes two of our recent papers (often verbatim) and interested readers are referred to these papers for more details. Krüsi et. al [4] presents our ICP-based T&R system and describes how we included the capability for local motion planning to be able to repeat routes in highly dynamic environments, or when short sections of the route are blocked. Pomerleau et al. [5] describes our efforts to update maps to encode dynamic and static parts of the scene. These systems operate on the same input data but are currently separate. Our future work will aim to bring them together to become a T&R system that is capable of long-term operation in dynamic environments.

II. SYSTEM OVERVIEW

A. Fast ICP: *libpointmatcher*

Offline 3D mapping can be realized with many open source software packages currently available, such as Point Cloud Library (PCL), MeshLab and CloudCompare. When it comes to online registration applied to a mobile system, which needs to take real-time decisions based on its surroundings, specific requirements have to be handled carefully. For that reason, we based most of our registration solutions on *libpointmatcher* [6]. Its modularity allows the generation of solutions tailored to different kinds of platforms and locomotion, while maintaining real-time

The authors are with ¹ the Autonomous Systems Lab, ETH Zurich, and ² Laval University. Correspondence: paul.furgale@mavt.ethz.ch

¹ <https://github.com/ethz-asl/libpointmatcher>

capability even with large bandwidth sensors. Working applications have been presented for ground, water, and aerial platforms using multiple sensors ranging from the Kinect to the Velodyne HDL-32E, including the more traditional Hokuyo and Sick lidars [7].

B. ICP-based T&R

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. We use point clouds from a spinning 3D laser scanner (Velodyne HDL-32E) and ICP for incremental localization, map building, and map-based localization. Our system includes an obstacle avoidance scheme employed in the repeat pass (cf. Section II-C). The algorithm is able to detect obstacles obstructing the path and to plan avoidance maneuvers that temporarily leave the reference route whenever necessary, yielding an *adaptive* route following system.

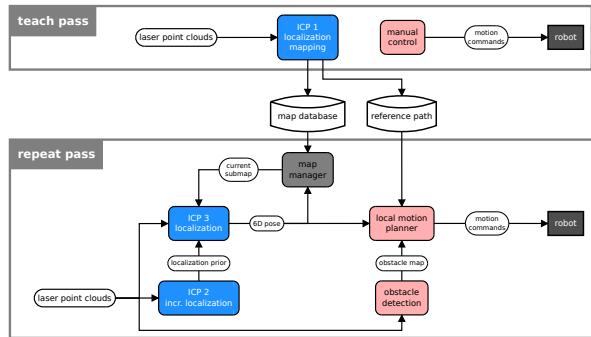


Fig. 2. System overview. Three main components are the basis of our adaptive route following system: a T&R framework, ICP-based registration modules, and an obstacle avoidance scheme (obstacle detection and local motion planning). The map built during the teach pass is saved in a database and used by 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 computes safe commands for the robot, making it follow the desired reference path while avoiding collisions with obstacles.

Figure 2 gives a on overview of the complete system. In the teach pass, an ICP-based localization and mapping module is used to build up a database of connected local maps (a *pose graph*), 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 previously created map. 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, shown in Figure 3. Based on this data, the planner computes a feasible local trajectory and the corresponding motion commands for the robot.

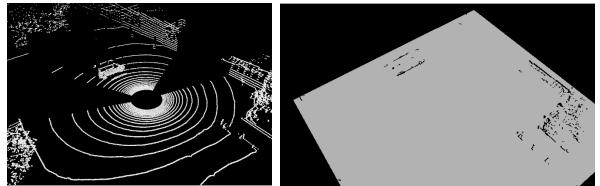


Fig. 3. An example of the input and the output of our obstacle detection module: a point cloud from the Velodyne 3D lidar (top), and the corresponding planar obstacle map (bottom). The scene shows a van on a large flat area, surrounded by buildings and some trees. The Velodyne consists of a series of 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.

C. Responding to Dynamic Elements: 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. The collision avoidance scheme on ARTOR is based on the sampling-based online planning framework presented in [8]. The framework enforces path alignment 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 models (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 candidate according to the cost function specified below. For the system described in this paper, we use a tree depth of one, as shown in Figure 4.

Collision Detection. Collision detection is performed on the two-dimensional binary occupancy grid sown in figure Figure 3. Objects are assumed to be static due to the lack of a classification module for dynamic objects in the

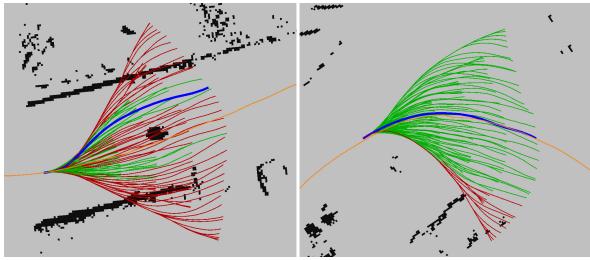


Fig. 4. Set of candidate trajectories shaped around the reference path, generated by the local planner for a fixed target speed and a single tree level. The figures show snapshots of the planner output during the T&R experiment described in Section III-A, containing the reference path (orange), the candidate trajectories (feasible: green, unfeasible: red), the selected trajectory (blue), 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).

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.

D. Responding to Dynamic Elements: Map Maintenance

Combining local motion planning with T&R allows us to react to dynamic scenes but it doesn't help us to update our maps as the scene changes or to infer common motion patterns attached to particular places. Fig. 5 presents the general overview of our map-maintenance system that was designed specifically for these tasks. Our eventual goal is for this module to take the place of ICP2 and ICP3 in the full T&R system shown in Fig. 2. The *Registration* module takes point clouds from the sensor and computes its coordinates in a global reference frame while correcting odometry. The *Global Map Maintenance* module estimates if points in the global map are dynamic or not and integrates the new point cloud into the existing map. Finally, the *Velocity Estimation* module estimates the velocity of the dynamic points by comparing the dynamic elements at current and previous times.

a) Dynamic Element Identification: More specifically, the estimation of whether a given point of the map is dynamic or not is based on visibility assumptions: if a new point is

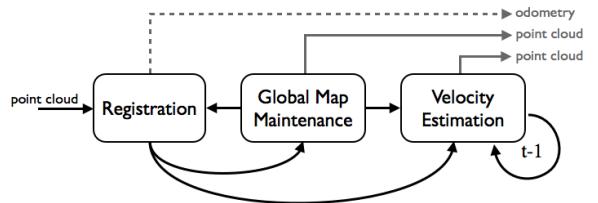


Fig. 5. Block diagram of the processing pipeline. Boxes represent separate processes running at different frequencies. Solid arrows represent point clouds being communicated to each module and the dashed line is odometry. The gray arrows show the output of each module.

observed behind a point in the map, that point was dynamic. In order to avoid the complex process of ray-tracing, we use spherical coordinates originating from the sensor. We can then use nearest neighbor search in this space to obtain points in the same conical aperture. It is implemented using the libnabo efficient kd-tree library [9].

We use a Bayesian approach to update the probability of a point to be dynamic:

$$P(Dyn|\theta, \phi, \delta) \propto \sum_{U, Odyn} \left| P(Odyn)P(U|\theta, \phi) \right| \times P(Dyn|Odyn)P(\delta|Dyn) \quad (1)$$

where Dyn is the probability for a point in the map of being dynamic at given time and $ODyn$ at previous time; θ is the angle between the point in the map and the new point; ϕ is the incidence angle of the point in the map, δ the distance between the points, and U a binary variable indicating if we need to update the point.

b) Velocity Estimation: Building on top of the dynamic object classification, one can estimate the velocity of moving objects. Most approaches rely on the clustering of the points into objects for which the velocity is then estimated by looking at, for example, the change in position of the center of mass [10], [11]. From a newly acquired point cloud P_t at time t , we associate all of its points to the global map. A subset of mobile points M_t is generated from P_t , fulfilling the requirement of being a dynamic obstacle. Those dynamic obstacles M_t can then be compared to the last subset M_{t-1} to extract velocity vectors. We based our approach on point-cloud registration using ICP, where M_t is the *reading* point cloud and M_{t-1} is the *reference*. Having different transformation parameters for each point is known as non-rigid ICP [12]. We reuse the underlying principles but extracted only translation components instead of the full 6 degrees of freedom (DOF) transformation. In essence, we propose to do dual non-rigid ICP—both from reading to reference and from reference to reading—and, given that we have a timestamp per point, divide the alignment error with the difference of acquisition time to estimate the velocity vectors. We use a weak spatial smoothness prior to harmonize the velocities across nearby points.

III. EXPERIMENTAL RESULTS

A. ICP-based adaptive T&R

We tested our adaptive route following system in a dynamic urban environment around the ETH campus in Zurich.

The experiments were conducted with the robot ARTOR, a six-wheeled, skid-steered electric vehicle, equipped with—among other sensors—a 3D Velodyne laser scanner. We first evaluated the localization and mapping system, assessing its accuracy and its robustness to deviations from the path driven in the teach pass. The latter is crucial in the repeat pass when deviating from the original path to avoid dynamic obstacles. We further compared our approach to a state-of-the-art stereo-vision-based T&R system [1]. Ground truth with millimeter precision was provided by a Leica Total Station.

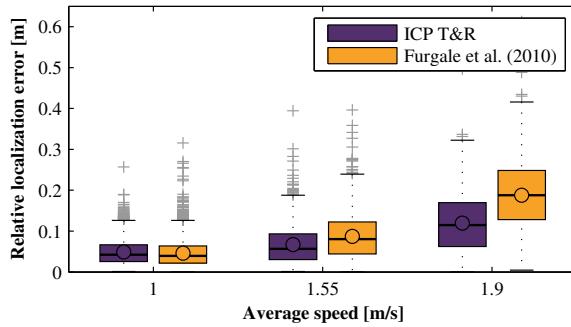


Fig. 6. Relative localization errors of the ICP-based T&R system in a dynamic urban environment. The bottom and the top of the boxes represent the first and the third quartile, respectively. That is, the box contains 50% of all measurements. The line inside the box is the median, the circle stands for 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.

The results of the accuracy analysis are shown in Figure 6. Motivated by the relative nature of our system, we measured the *relative* localization error, that is, the error in the position estimate relative to the current map in the pose graph. For this experiment, we manually drove the robot seven times along a path of around 130 m, 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).

To assess the system’s ability to detect and avoid obstacles, as well as the robustness of the localization to path deviations, we conducted an experiment where differently sized obstacles were put in the robot’s way. Using the full adaptive T&R system (ICP and obstacle avoidance), we first taught the system a straight path of around 23 m. We then let it autonomously repeat this route several times, while blocking the path with increasingly large obstacles. Figure 7 shows the four different obstacles used in the experiment, as well as the paths driven by our robot to avoid them. It can be seen that our obstacle avoidance scheme enables the robot to safely and smoothly drive around large obstacles. In this experiment, we deviated from the taught path by up

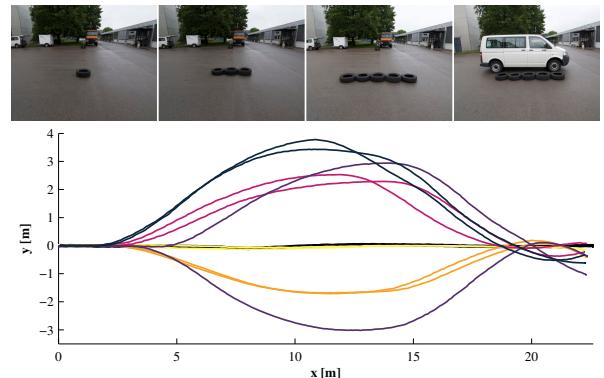


Fig. 7. Obstacle avoidance experiment. The experiment consisted of teaching a straight path, followed by autonomous repeat passes with an increasingly large obstacle in the middle of the path. The photographs on the top show the four different obstacle settings. The starting point of the robot was left 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.

to 4 m, and the robot reliably found its way back onto the path after the obstacle had been passed. Figure 8 shows the relative localization errors of our system and the stereo-based framework as a function of the maximum deviation from the path. When the robot deviated from its original path, the error of the stereo-based system increased dramatically, while the performance of our laser-based approach remained nearly constant.

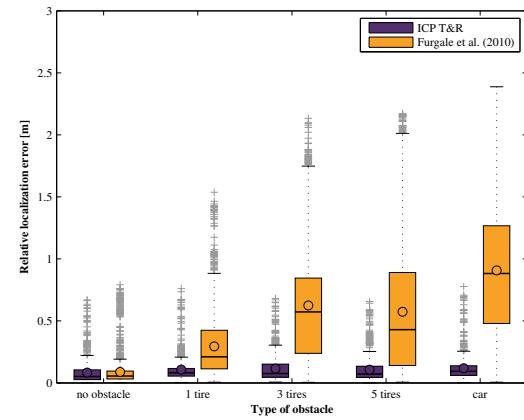


Fig. 8. Localization accuracy depending on the type of obstacle on the path. The size of the obstacle and the resulting deviation from the taught path increase from the left to the right. Each box plot contains the data of six experiments: the robot autonomously drove twice in each obstacle setting (using 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.

Having verified our system’s conformance to the requirements in terms of localization (accuracy, robustness to path deviations) 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 (Figure 9), and let it repeat the route eight times in autonomous mode. The first half of the path consisted of a quiet side street with parked cars on either side, the other half was a rather busy road with a lot of traffic including cars, trucks, bicycles and trams, as well as pedestrians. In total, our robot autonomously drove a distance of more than 10 km.

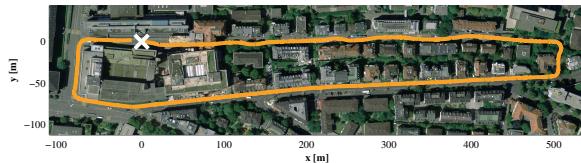


Fig. 9. The route in our long-range field tests in the urban environment around ETH Zurich. The path was a loop of around 1.3 km, which we drove in clockwise direction at a speed of approximately 1 m/s. Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000

The teach pass of the experiment was done during the night on March 12, 2013. The repeat runs were conducted in May, 2013 using the map recorded more than two months earlier. 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 in the same place. Neither these changes, nor the dynamic elements (pedestrians, cars, trams), nor the different weather and lighting conditions (sunny/rain/snowfall, day/night) had any notable influence on the performance of our ICP-based localization system. Moreover, all the relevant static and dynamic obstacles (including strangers deliberately stepping into the way) have been detected and avoided reliably. Except from one single situation², the robot drove the entire distance in all eight repeat passes completely autonomously, without any intervention of the operator.

B. Responding to Dynamic Elements: Map Maintenance

The map-maintenance module was developed and tested independently from the T&R system. To test the identification of dynamic elements, we used the ARTOR robot to survey a visitor parking lot (see Fig. 1). This survey consisted of driving the robot around the area at different times during three consecutive days.

The resulting segmentation is shown in Fig. 11. The top panel shows the static map in which we can see the trees and the ground. The bottom panel shows the dynamic map in which we can see the cars and many points in the middle corresponding to bikes and pedestrians. More precisely, in (1) the static map shows a path while the dynamic map shows the trails of pedestrians. In (2) we can distinguish, in the dynamic map, the flow of bikes and pedestrians around parked cars.

As a second application of this framework, we surveyed a 1.3 km long route in the city of Zurich at three different times

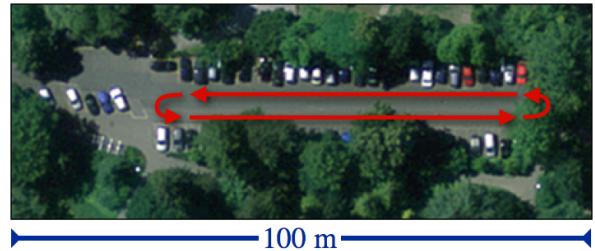


Fig. 10. Aerial view of the parking lot used for the segmentation experiment. In red, the survey path realized by the robot. Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000.

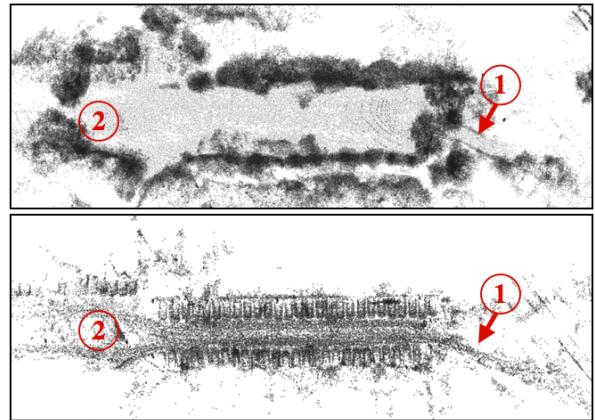


Fig. 11. Result of the segmentation after 9 surveys over the course of 3 days. Top: Reconstruction with $P(Dyn = true) < 0.5$. Bottom: Reconstruction with $P(Dyn = true) > 0.5$. The flow of pedestrians and bikes can be seen using a path (1) and splitting to avoid another row of parked cars (2).

in seven months (March, May and September 2013). The result is shown in Fig. 12. The middle plot shows the static map reconstructed with our algorithm. In the bottom plot, we show the amount of dynamic elements that can be used to indicate areas that can be potentially difficult or hazardous to navigate in.

The other experiment was conducted in front of the main building of ETH Zurich. It took place during the *information day*, which meant that many young students gathered in the streets, with sometimes as many as 15 persons in the vicinity of the robot. The main street consists of two large sidewalks, two lines for cars, and two lines for trams. The robot surveyed the area twice within 20 minutes, each time driving on the sidewalks on both sides of the street. Fig. 13 presents the results of the experiment. The two lower graphs show the extracted dynamic objects over the course of the survey, with their estimated speed and direction of motion. In the speed graph (left), blue corresponds to the range of typical walking speeds of pedestrians. The sidewalks and the pedestrian crossings (the latter marked with red arrows) can be clearly identified by looking at the blue objects. Furthermore, there are two lines of faster objects (yellow to red), which designate the car lanes. In the orientation graph (right), the two main directions of the cars are clearly visible. On the sidewalks the situation is naturally more chaotic, as

²Probably due to reflections of the laser on the wet pavement, the obstacle detection module reported an obstacle that did not actually exist.

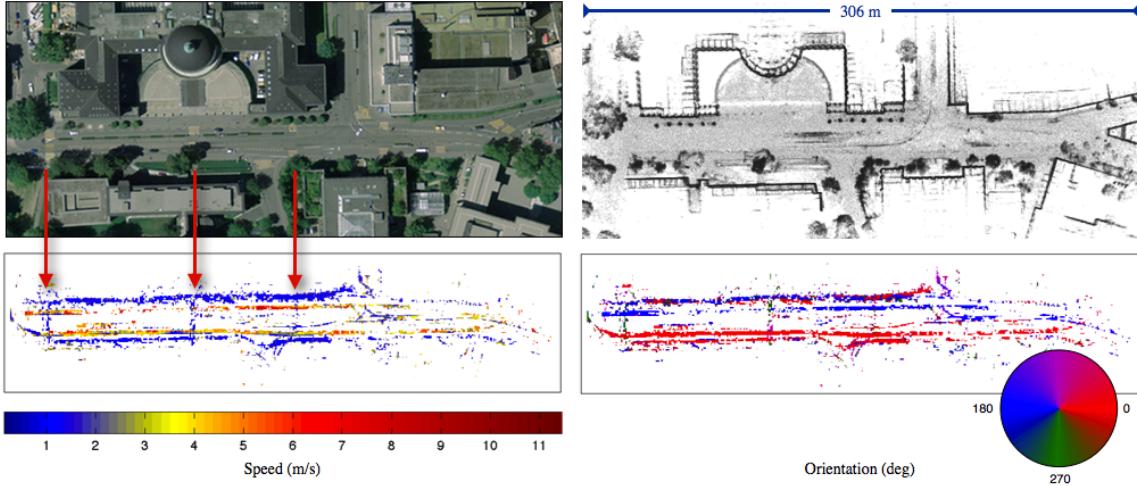


Fig. 13. Extraction of velocity information at a global scale. *Top left*: Aerial view of the street in front of ETH Zurich. *Source*: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000. *Top right*: 3D reconstruction after dynamic element removal. *Bottom left*: Average speed of the moving objects. *Bottom right*: Average orientation of the moving objects. The red arrows highlight the pedestrian crossings.

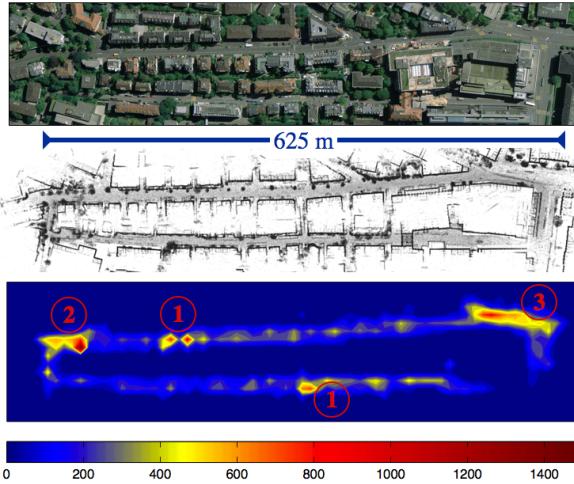


Fig. 12. Long range survey over a 1.3 km long path. The environment was monitored over a period of seven months. *Top*: Aerial view of surveyed area. *Source*: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000. *Middle*: 3D reconstruction after dynamic elements removal. *Bottom*: Occurrence of dynamic elements. The graph highlights the position of (1) construction sites, (2) a large tree and (3) a busy street intersection. Color represent to number of dynamic points over a cell size of 10m.

pedestrians do not walk on distinct lanes.

IV. CONCLUSION

This paper provided a summary of our efforts to build a T&R system capable of adaptation when faced with dynamic scenes. For full details and more extensive experimental results, please refer to references [4] and [5].

V. ACKNOWLEDGMENTS

This work was supported by the EU FP7 IP projects Natural Human-Robot Cooperation in Dynamic Environments (ICT-247870), TRADR (ICT-609763), the armasuisse S+T UGV research program, and the EU FP7 2007-2013 Program, Challenge 2, Cognitive Systems, Interaction, Robotics,

under grant agreement No 269916, V-Charge. The authors are grateful to M. Rufli and M.-È. Garneau for useful comments.

REFERENCES

- [1] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [2] C. McManus, P. Furgale, and T. D. Barfoot, "Towards lighting-invariant visual navigation: An appearance-based approach using scanning laser-rangefinders," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 836–852, 2013.
- [3] C. McManus, P. T. Furgale, B. E. Stenning, and T. D. Barfoot, "Lighting-invariant visual teach and repeat using appearance-based lidar," *Journal of Field Robotics*, vol. 30, no. 2, pp. 254–287, 2013.
- [4] P. Krüsi, B. Bücheler, F. Pomerleau, U. Schwesinger, R. Siegwart, and P. Furgale, "Lighting-invariant adaptive route following using icp," *Journal of Field Robotics*, 2014, conditionally accepted on March 17, 2014. (video).
- [5] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3d map maintenance in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 3 – June 7 2014. [Online]. Available: [bib/pomerleau_icra14.pdf](#)
- [6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, Feb. 2013.
- [7] F. Pomerleau, "Applied Registration for Robotics - Methodology and Tools for ICP-like Algorithms," Ph.D. dissertation, ETH Zurich, 2013.
- [8] U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart, "A sampling-based partial motion planning framework for system-compliant navigation along a reference path," *Proceedings of the IEEE Intelligent Vehicles Symposium (IVS)*, 2013.
- [9] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter, "Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration," *Journal of Software Engineering for Robotics*, vol. 3, no. 1, pp. 2–12, Mar. 2012.
- [10] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 142–147.
- [11] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, Sept. 2007.
- [12] J. Feldmar and N. Ayache, "Locally affine registration of free-form surfaces," in *Computer Vision and Pattern Recognition, 1994. Proceedings of the IEEE Computer Society Conference on*, 1994, pp. 496–501.