DR 1.1: Essential sensing, mapping and low-level memory

Tomáš Svoboda*, Dong-Uck Kong†, and the TRADR consortium

*CTU in Prague
†Fraunhofer IAIS, Sankt Augustin, Germany
⟨svobodat@fel.cvut.cz⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	December 2014
Actual submission date:	February 2015
Lead partner:	Czech Technical University in Prague
Revision:	final
Dissemination level:	PU

This report presents the results of WP1 for the first year of TRADR project. It describes the essential robot (UGV) perception functionalities and advancements in creating metrical maps.

Executive Summary

The key objective of WP1 is to provide sensory data from all involved robots registered in space and time, to keep creating and updating robot centric representations, and ground them into the world coordinate frame. The obtained representations are furnished to other WPs, which maintain higher level situation awareness.

In Year1 one we concentrated on essential UGV perception functionalities. Adaptive traversability changes robot morphology as the robot traverses various obstacles. We introduced a concept of safe exploration in order to assure the robot stays operational. We were working on a multimodal victim detector that should overcomes problems when temperature of the environment is similar to the human body temperature. In order to assure persistence of metrical maps we advance our mapping and robot localization for both UGV and UAV. During a longer mission there are almost always some anomalies in data. We explicitly detect these anomalies in a multimodal fusion scheme for robot localization. If needed we can also run two independent visual odometry algorithms.

Role of robot perception and metrical mapping in TRADR

The robot perception means the robot is able to analyze its neighborhood and act accordingly. Terrain recognition is essential for robot locomotion regardless whether the robot is teleoperated or moves autonomously. It is desirable the robot overcomes obstacles in a reasonable way - fast, safe, consuming less power, reducing cognitive load of a human operator. Automatic victim detection is important for many search and rescue scenarios. A human operator may provide final decision however, robots, when crawling through a disaster site should provide warning about possible victim locations.

The metrical mapping serves as the very basis for modeling the world. It is also the basis for sharing informations between robots and also among several sorties even missions.

Contribution to the TRADR scenarios and prototypes

The learnable adaptive traversability contributes to the models for acting (WP2) and also supports the human-robot teaming (WP5). As the metrical mapping establishes a common ground for sharing data it naturally supports almost all packages. Multi-robot mission will employ the robots

in various roles. We implemented several software packages support flexible deployment of the UGVs, e.g. lidar-camera calibration and IP PTZ camera interagration. During the TRADR Joint Excercise (TJEx) event in 2014 we deployed and tested multimodal data fusion and adaptive traversability algorithm.

1 Tasks, objectives, results

1.1 Planned work

In Year 1, WP1 planned to investigate "Essential sensing, mapping and low-level memory" (Milestone MS1.1). The work was divide into two WP1 tasks:

- Essential sensing functionality (T1.1)
- Robot centric metrical maps and models storage (T1.2)

Essential sensing concetrates on data fusion for making the localization and mapping more robust. As the UGVs were planned to be singificantly upgraded (WP6) significant part of the effort was allocated for transferring the background knowledge/codes to the upgraded robots.

The metrical maps are planned to be the core for sharing "knowledge" between robots, sorties, missions and human operators. In Year1, the main effort was planned for creating maps for one robot and in a robot-centric view.

1.2 Actual work performed

1.2.1 Task1.1 (Essential sensing functionality)

Our work on essential sensing functionalities can be roughly categorized into three pieces. We worked on fusing multiple sensory data for more persistent localizaiton and mapping. Adaptive traversability and safe exploration are the steps toward autonomy and smooth teleoperation on very difficult terrains. Fusing sensory data should also make a victim detector robust.

Data fusion for localization and mapping During our work in urban search and rescue (USAR) scenario, we encountered frequent abnormal patterns in the visual odometry attitude (VO), laser range finder attitude and position estimates. These usually occurred as a consequence of unexpected environmental effects or modality failures (e.g., dynamically changing conditions, terrain obstacles, limited range of view, low visibility), see Figure 1 for few examples. Such anomalies would significantly deteriorate estimates of attitude and position of the UGV and consequently propagate to higher-level functions depending on localization functionality. This finding motivated us to focus on those special conditions that would interfere with localization and task execution during the intended long-term missions. To correctly identify the actual anomalies which were to be rejected we went beyond the standard statistical tests by exploring different state-of-the-art machine learning approaches. We demonstrated the implications of our research both indoor (with precise reference from a Vicon system) as well as in challenging

Figure 1: Few examples of challenging environments causing anomalies. From left: limited field of view, dynamically changing conditions, low visibility. The cases often combine.

outdoor environment on a set of experiments designed to imitate search and rescue mission conditions (significant part of the dataset consists of experiments recorded during the joint exercise with end-users). The results are presented in [32] (Annex Overview 2.11). The anomaly detection handled serious local disturbances and vastly improved the overall performance, even in case of driving inside a pipe or through a dense forest.

A robust reliable robot perception essentially forces exploiting/fusing all available sensory data. Multimodal data fusion requires resolving several problems, significantly different sampling frequencies of the individual modalities being one of them. Our contribution [16] (Annex Overview 2.12) lies in the development of a model for such multi-modal data fusion using the Extended Kalman Filter (EKF), especially in the way we incorporated sensors with slow and fast measurement update rates (e.g. the IMU sampling rate is 90 Hz compared to the laser rangefinder output at 0.3 Hz). For this purpose, we proposed and investigated three different possible methods - one of them, the trajectory approach, is our contribution that we compared to a common state-of-the- art practice. We showed that a standard EKF designed with the velocity approach does not cope well with such significant differences in the frequency, whereas our proposed trajectory approach does. We compared our proposed solution with the well-proven and popular Rauch- Tung-Striebel smoother for the Extended Kalman filter. Furthermore, we improved the precision of our data fusion by incorporating scale estimation for the visual modality. We tested our algorithm on approximately 4.4 km of field tests (over more than 9 hours of data) both in- and outdoors. In order to have precise quantitative analysis, we recorded ground truth using either a Vicon motion capture system (indoors) or a Leica theodolite tracker (outdoors). This way we proved that our scheme is a significant improvement upon standard approaches. Combining all four modalities: IMU, tracks odometry, visual odometry and ICP-based localization, we achieved precision in the total distance driven of 1.2% error in the indoor environment and 1.4% error in the outdoor environment.

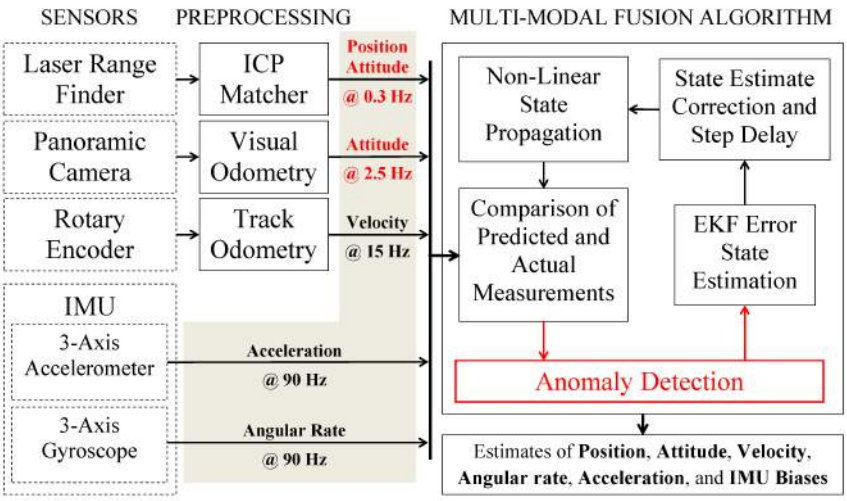


Figure 2: Fusing modalities/measurements that vary in multiple aspects, e.g. sampling frequency, robustness.

Adaptive traversability, exploration We were working on various aspects of adaptive traversability (AT). By adaptive traversability we understand autonomous motion control adapting the robot morphology configuration of articulated parts together with their compliance to traverse unknown complex terrain with obstacles in an optimal way see Figure 3. We verify this concept by proposing a reinforcement learning based AT algorithm for mobile robots operating in such conditions. We demonstrate the functionality by training the AT algorithm under lab conditions on simple EUR-pallet obstacles and then testing it successfully on natural obstacles in a forest. For quantitative evaluation we define a metric based on comparison with expert operator. Exploiting the proposed AT algorithm significantly decreases the cognitive load of the operator. The approach and the results are presented in [36] (Annex Overview 2.2) and [37] (Annex Overview 2.3), and also [22] (Annex Overview 2.6).

The robot is typically required to operate in an unknown environment and with imprecise sensory data. However, it is highly desired that the robot only act in a safe manner and do not perform actions that could probably make damage to them. To train some tasks with the robot, we utilize reinforcement learning (RL) [36] (Annex Overview 2.2). The machine learning method however requires the robot to perform actions leading to unknown states, which may be dangerous. We suggest to train a safety function which constrains possible actions to a subset of really safe actions. Our approach utilizes two basic concepts. First, a core of the safety function is given by a cautious simulator and possibly also by manually given examples. Second, a

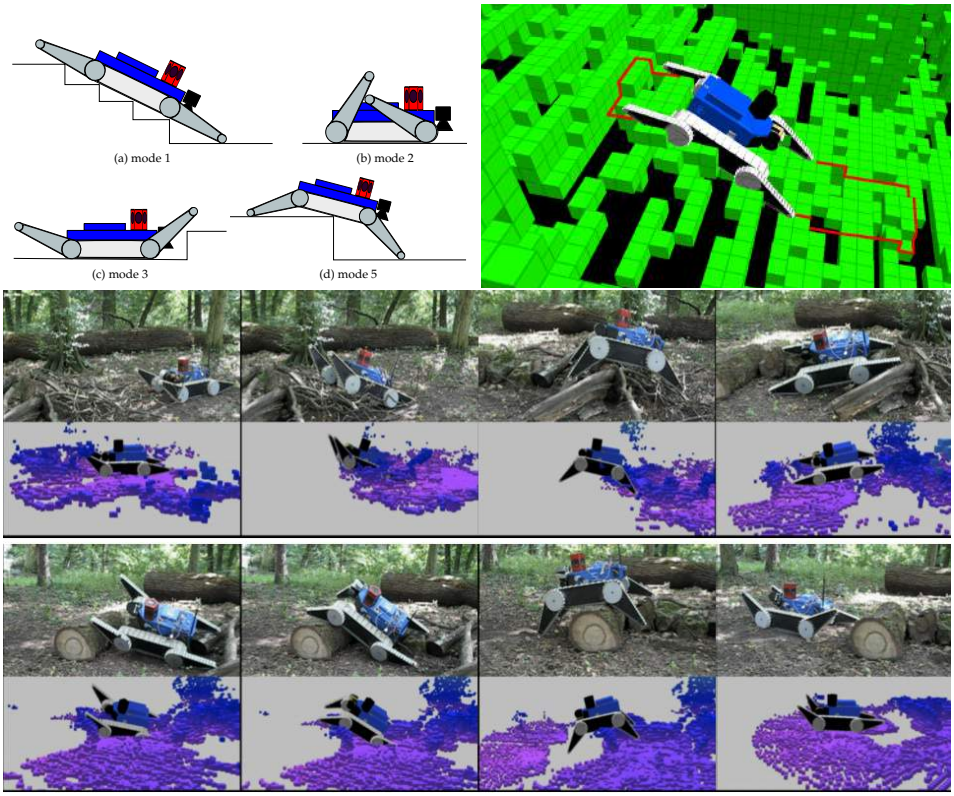


Figure 3: Traversability modes (left) and Robot on the terrain digital elevation model (right). Robot traversing difficult obstacles (bottom). Note that the DEM is often only partially observable.

Figure 4: Examples of a safe state (left) and a critical state (right).

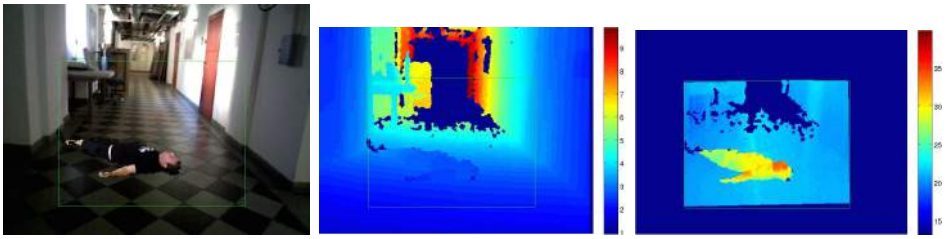


Figure 5: Dataset for training the victim detector. From left: RGB data, depth, and thermal data. Data were collected in a controlled environment and merged real data. As the scene depth is also known, person is put correctly lying on the floor.

classifier training phase is performed (using Neyman-Pearson SVMs), which extends the safety function to the states where the simulator fails to recognize safe states, see [23] (Annex Overview 2.4) and [24] (Annex Overview 2.5).

Visual localization In order to support the performance of the main fusion scheme [16] (Annex Overview 2.12) we significantly improved the already integrated visual odometry [35] (Annex Overview 2.8) and also developed and integrated and alternative approach based on FastSLAM 2.0 approach [29] (Annex Overview 2.7).

We also started working on an algorithm that would recognize previously visited places. We tested holistic approaches based on using GIST descriptors and descriptors computed a deep neural network [7] (Annex Overview 2.9).

Victim detector Many supervised learning methods requires huge training data. Since obtaining a large number (i.e., millions) of multimodal images of victims, with accurate ground truth needed for applying supervised learning methods, seems infeasible in near future, we decided to generate and use a partially synthetic dataset to train the classifier. First, a set of positive training examples was recorded in controlled environment, with the ground-truth segmentation obtained via a chroma key compositing technique. Then, these were combined with separately recorded background data to generate synthetic examples with known segmentation masks. Motivated by recent promising results in pedestrian detection, preliminary experiments with decision trees were carried out with a rich feature set covering all the modalities (i.e., color, depth, and temperature) and taking into consideration their specifics. An efficient method was developed for learning the structure of decision trees on large datasets with exact probability criterion for selecting split-node features.

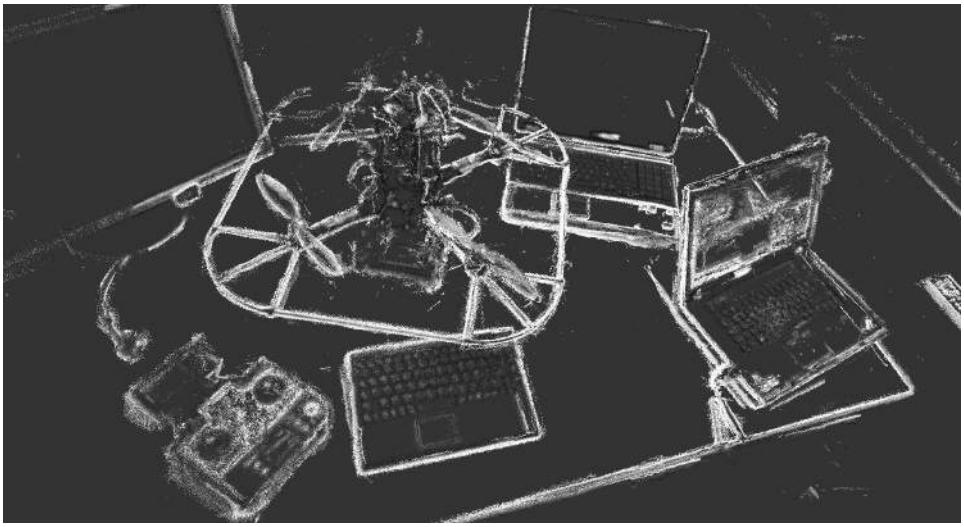


Figure 6: Screenshot of an experiment applying the semi-dense mapping method. The image potraits an accumulated point cloud. The semi-dense method processes only image regions with sufficiently large gradients.

1.2.2 Task1.2 (Robot centric metrical maps and models storage I)

UAV-based mapping One of the key aspects of WP1 is to extract useful metric environment models from multiple robots' perceptional experience and to make the models persistent. Based on the requirements for the TRADR multi-robot system, the central question was formulated:

How can particularly efficient mapping and localization methods be developed with respect to measurement quality, sensor data quantity and long-term data analysis for both robot and human action forces?

It is an explicit goal to exploit the advantages of visual sensors in environment perception and mapping (SLAM) because of their richness of information, light weight and availability. With the common goal to enable robots to work in arbitrary environments by visual sensors, several possible concepts have been investigated. Keypoint-based SLAM methods, dense surface-based modeling methods, semi-dense mapping method and their combinations were thoroughly investigated (Figure 6). Also, we investigated several image acquisition systems. More survey details can be found in the technical report [15] (Annex Overview 2.13).

UGV mapping We continued in the work done within the NIFTi project on aligning 3D pointclouds [27, 28, 25]. The main motivation is to go beyond the usual iterative scheme that requires the point clouds to be closely related. This may suffice for sequential map building in a one robot mission. We

used the large data collections assembled within the NIFTi project and added few new datasets. We experimentally compared the Iterative Closest Point (ICP) method with approaches based on matching local features. We also experimented with newly defined color features [12] (Annex Overview 2.10). The main conclusion is that, providing sufficient overlap of the point clouds, performance of the feature based method is largely independent of the transformation between point clouds. This may be valuable in case when multiple robots are mapping common terrain but from different viewpoints (perspectives).

Model storage In TRADR we also adressed the persistence of our world model, esp. the maps created by the robots. We are able to store and load maps. We succeeded in continuing mapping on a map loaded from another mission. With our storing mechanism, we are also able to travel 'back in time', which is very useful for the end users in longer running missions.

At the moment the world model is stored as a single big pointcloud, which is not meant to be refined, but rather to be used as a starting point for the new map. In order to utilize the stored maps even more, we will have to save the informations which were used to create the map rather than the map itself. We already identified the key topics, which have to be considered while designing the database schema.

location of objects If we split up the map, we will have to decide, in which submap an object will be located. A moving object might 'jump' to another map.

non rigid transformations How do we store non rigid transformations (between two maps) in the database.

versioning and time As the scene will change over time, we will have to cope with that in our representation.

peformance Since we are only saving the data used to create the map, rather than the map itself, there will be always some effort needed to reconstruct the map. We are trying to minimize this effort by using a suitable representation (e.g. saving already transformed pointclouds).

synchronization As we will need to map on the robots (at least in a small area), we will have to integrate that into the global map stored in the central database.

1.2.3 Various auxiliary tasks

Since representation of the environment in form of a triangular mesh is better suited for simulation e.g., in order to assess robot safety during a planned

Figure 7: Camera to lidar calibration. A sw tool for automated calibration.

obstacle traversal or visualization purposes, a module (a ROS node) for automatic conversion of point clouds to triangular meshes was developed.

While seeing the mobile robot itself in its cameras is beneficial for teleoperation, it is typically undesirable for computer vision methods such as object detection or visual odometry. Being able to distinguish the parts of image covered by the robot would presumably improve performance of these methods. As this becomes particularly important with the Kinova manipulator being mounted on the UGV, as the arm may operate in the entire field of view, we developed a module masking out the robot in calibrated cameras which uses a known model and current state of the joints.

New drivers for Optris infrared cameras were developed which are better suited for capturing synchronized video streams from multiple devices. These were used in capturing the multimodal victim datasets, see Figure 5.

The newly upgraded UGV allows flexible mounting of additional cameras directly on the upper board. Foreseeing this flexibility we prepared an automated tool for camera-lidar calibration [3] (Annex Overview 2.1). As the mounting point of the swiveling Lidar is fixed it may well serve as the calibration basis. The tool reads a standard bag-file and guides a user when assigning the necessary point-to-point correspondences, see Figure 7. Various point cloud visualizations are implemented.

We listen to the end-users and they sometimes complain about limited zoom possibility when using the standard Ladybug3 omnidirectional camera. We developed ROS nodes for operating an IP PTZ camera and we will integrate it onto the UGV board during the Year2 of the project.

An auxiliary library for ROS-Matlab interaction were developed to allow fast prototyping of ROS nodes in Matlab.

1.3 Relation to the state-of-the-art

We progressed our work on multimodal data fusion [16] (Annex Overview 2.12) towards detecting anomalies in data. *We go beyond the standard statistical tests by exploring different state-of-the-art machine learning approaches and exploiting our rich dataset that we share with the robotics community.* Failures in the exteroceptive perception systems are one of many sources of uncertainty in mobile robot localization. Anomaly detection addresses the task of finding patterns that do not conform to expected behavior [5].

There has been a lot of effort put into rejection of such disturbances in a broad range of different applications [8]. Within the field of robotics the problem of system failures identification is often referred to as fault detection and isolation (FDI) [26]. Fault is defined as an anomaly in behavior of the monitored system and can be detected, isolated (locating the faulty component) and identified (determination of magnitude of the fault) [9]. A reliable robotic system must deal with many uncertainties that can be handled by FDI, e.g., by quality inspection of the provided information [4], comparing information providers [33], using information flow between the control and actuation [6], monitoring the reliability of resources [20], or proper recognition and modeling of the sensor and mechanical failures [11].

We advanced our work on adaptive traversability [36] (Annex Overview 2.2) towards robust handling of uncomplete or partially observed data. Many authors estimate terrain traversability only from exteroceptive measurements (e.g. laser scans). For example Kim et al. [14] estimate whether the terrain is traversable or not and plan the trajectory over the traversable terrain. In our experience, when the robot is teleoperated in the real environment, it is not possible to plan the flipper motion in advance only from the exteroceptive measurements. When the terrain collapses unexpectedly, captured terrain profile must be updated without exteroceptive measurements. Reactive control has been successfully used for learning the acrobatic tricks with an RC helicopter [1, 2]. Since it is possible to model the helicopter-air interactions well, one can use the model to improve the reinforcement learning. In our case, analytical modeling of the robot-terrain interaction is very difficult. Contrary to [1] we rather focus on a model-free reinforcement learning technique. The idea of inference from incomplete data via some kind of sampling [18, 34] or EM algorithm [10] has been known for several decades. *In contrast to others, we demonstrate the proposed approach on a real robotic platform equipped with many different sensors.*

Many robotic tasks are tackled by RL with iterative state-action space exploration (RC helicopter acrobacy [1], adaptive traversability [36] (Annex Overview 2.2), etc.). RL essentially needs to exhaustively sample the state-action space (which is called "exploration"), and the exploration strategy is represented by a stochastic policy. While manually-driven exploration is often prohibitively time consuming, autonomous exploration is usually only

applied to inherently safe systems (pendulum) or to simulators [30]. *We propose a framework for making autonomous exploration even for general systems, [24] (Annex Overview 2.5).*

The upgraded visual odometry codes are based on the state of the art approaches [19, 31, 17]. Our ongoing work on visual place recognition experiments with GIST descriptors [21] and a deep learning network [13].

References

[1] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *In Advances in Neural Information Processing Systems 19*, page 2007. MIT Press, 2007.

[2] Pieter Abbeel and Andrew Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proc. of the 22nd int. conf. on Machine learning*, ICML '05, pages 1–8, New York, NY, USA, 2005. ACM.

[3] Jan Brabec. Automated camera calibration from laser scanning data in natural environments, June 2014.

[4] Christopher Brunner, Thierry Peynot, Teresa Vidal-Calleja, and James Underwood. Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *Journal of Field Robotics*, 30(4):641–666, 2013.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[6] Anders Lyhne Christensen, Rehan OGrady, Mauro Birattari, and Marco Dorigo. Fault detection in autonomous robots based on fault injection and learning. *Autonomous Robots*, 24(1):49–67, 2008.

[7] Erik Derner and Tomáš Svoboda. Indexing images for visual memory by using DNN descriptors – preliminary experiments. Research Report CTU–CMP–2014–25, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2014.

[8] Sumeet Dua and Xian Du. *Data mining and machine learning in cybersecurity.* Taylor & Francis, 2011.

[9] Janos Gertler. *Fault detection and diagnosis in engineering systems.* CRC press, 1998.

[10] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an em approach. In *Neural Information Processing Systems (NIPS)*, pages 120–127. Morgan Kaufmann, 1994.

[11] Puneet Goel, Göksel Dedeoglu, Stergios I Roumeliotis, and Gaurav Sukhatme. Fault detection and identification in a mobile robot using multiple model estimation and neural network. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2302–2309. IEEE, 2000.

[12] Aleš Hrabalík. 3d point cloud registration, experimental comparison and fusing range and visual data, June 2014.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[14] Dongshin Kim, Jie Sun, Sang Min, Oh James, M. Rehg, and Aaron F. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *In Proc. of International Conference on Robotics and Automation (ICRA)*, pages 518–525, 2006.

[15] Dong-Uck Kong. Persistent mapping of dynamic environments by multiple aerial robots with multi-camera systems. Technical report, Fraunhofer IAIS, 2015. unpublished.

[16] Vladimír Kubelka, Lorenz Oswald, François Pomerleau, Francis Colas, Tomáš Svoboda, and Michal Reinstein. Robust data fusion of multimodal sensory information for mobile robots. *Journal of Field Robotics, in press*, 2015.

[17] Rainer Kümmerle, Giorgio Grisetti, Strasdat Hauke, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[18] Daniel J. Lizotte, Lacey Gunter, Eric Laber, and Susan A. Murphy. Missing data and uncertainty in batch reinforcement learning. In *Neural Information Processing Systems (NIPS)*, 2008.

[19] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.

[20] Yoichi Morales, Eijiro Takeuchi, and Takashi Tsubouchi. Vehicle localization in outdoor woodland environments with sensor fault detection. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 449–454. IEEE, 2008.

[21] Ana C. Murillo, Gautam Singh, Jana Košecká, and J.J. Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, February 2013.

[22] Tomáš Nouza. Safe adaptive traversability learning for mobile robots, June 2014.

[23] Martin Pecka and Tomáš Svoboda. Safe exploration techniques for reinforcement learning – an overview. In Jan Hodicky, editor, *Modelling and Simulation for Autonomous Systems*, volume 1 of *Lecture Notes in Computer Science*, pages 357–375, Cham, Switzerland, May 2014. NATO Modelling and Simulation Centre of Excellence, Springer.

[24] Martin Pecka, Karel Zimmermann, and Tomáš Svoboda. Safe exploration for reinforcement learning in real unstructured environments. In *Computer Vision Winter Workshop, in press*, 2015. to appear.

[25] Tomas Petricek and Tomas Svoboda. Area-weighted surface normals for 3d object recognition. In *Pattern Recognition (ICPR), 2012 21th International Conference on*, pages 1492–1496, November 2012.

[26] Ola Pettersson. Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, 53(2):73–88, 2005.

[27] Francois Pomerleau, Francis Colas, Roland Siegwart, and Stephane Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[28] Francois Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 2012.

[29] Pavel Potoček. Probabilistic approach to landmark management in visual odometry, June 2014.

[30] S Ross and JA Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

[31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564 –2571, Nov. 2011.

[32] Jakub Šimánek, Vladimír Kubelka, and Michal Reinštein. Imroving multi-modal data fusion by anomaly detection. *Autonomous Robots, in press*, 2015. accepted 5 January 2015.

[33] Paul Sundvall and Patric Jensfelt. Fault detection for mobile robots using redundant positioning systems. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3781–3786. IEEE, 2006.

[34] M. A. Tanner and W.H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82:528–540, 1987.

[35] Jan Vakula. Keypoint localization and matching in difficult scenes for visual odometry, Janyary 2015.

[36] Karel Zimmermann, Petr Zuzánek, Michal Reinstein, and Václav Hlaváč. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In Lynne Parker, editor, *ICRA2014: Proceedings of 2014 IEEE International Conference on Robotics and Automation*, pages 5177–5182, Piscataway, USA, June 2014. IEEE Robotics and Automation Society, IEEE.

[37] Petr Zuzánek, Karel Zimmermann, and Václav Hlaváč. Accepted autonomy for search and rescue robotics. In Jan Hodicky, editor, *Modelling and Simulation for Autonomous Systems*, volume 1 of *Lecture Notes in Computer Science*, pages 231–240, Cham, Switzerland, May 2014. NATO Modelling and Simulation Centre of Excellence, Springer.

2 Annexes

2.1 Brabec-bc2014, "Automated camera calibration from laser scanning data in natural environments"

Bibliography Jan Brabec. Automated camera calibration from laser scanning data in natural environments, Bachelor Thesis, June 2014. Czech Technical University in Prague.

Abstract We have developed an application for extrinsic camera calibration from the data acquired by the LIDAR scanner. Since the raw range images from the LIDAR scanner do not possess enough detail, we processed the range images to highlight edges and corners and allow the operator to create correspondences between the world points and the image points. We have also developed a technique for local correction of the correspondences in case the operator makes a slight mistake. The application is implemented as a node in Robot Operating System (ROS). We have performed experiments on a mobile robot intended for urban search and rescue. We experimentally show that the application can be used outside the laboratory to quickly calibrate a new camera in the system or recalibrate an already present camera. That is a big advantage compared to the present tools available in ROS that usually require the use of special calibration patterns and are restricted to the laboratory environment only.

Relation to WP A ROS node for mutual camera to Lidar calibration. A friendly GUI for an untrained user.

Availability Unrestricted. A public document `https://cyber.felk.cvut.cz/research/theses/papers/494.pdf` or `https://dspace.cvut.cz/handle/10467/24152`

2.2 Zimmermann-ICRA2014, "Adaptive Traversability of Unknown Complex Terrain with Obstacles for Mobile Robots"

Bibliography Karel Zimmermann, Petr Zuzanek, Michal Reinstein, and Vaclav Hlavac. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In Lynne Parker, editor, *ICRA2014: Proceedings of 2014 IEEE International Conference on Robotics and Automation*, pages 5177-5182, Piscataway, USA, June 2014. IEEE Robotics and Automation Society, IEEE.

Abstract In this paper we introduce the concept of Adaptive Traversability (AT), which we define as means of autonomous motion control adapting the robot morphology config uration of articulated parts and their compliances to traverse unknown complex terrain with obstacles in an optimal way. We verify this concept by proposing a reinforcement learning based AT algorithm for mobile robots operating in such conditions. We demonstrate the functionality by training the AT algorithm under lab conditions on simple EUR-pallet obstacles and then testing it successfully on natural obstacles in a forest. For quantitative evaluation we define a metrics based on comparison with expert operator. Exploiting the proposed AT algorithm significantly decreases the cognitive load of the operator.

Relation to WP It directly contributes to Task1.1 - essential robot perception.

Availability Unrestricted. Included in the public version of this deliverable.

2.3 Zuzanek-MESAS2014, "Accepted Autonomy for Search and Rescue Robotics"

Bibliography Petr Zuzánek, Karel Zimmermann, and Václav Hlaváč. Accepted Autonomy for Search and Rescue Robotics. In Jan Hodicky, editor, Modelling and Simulation for Autonomous Systems, volume 1 of Lecture Notes in Computer Science, pages 357-375. Springer.

Abstract Since exploration of unknown disaster areas during Search and Rescue missions is often dangerous, teleoperated robotic platforms are usually used as a suitable replacement for a human rescuer. Advanced robotic platforms have usually many degrees of freedom to be controlled, e.g. speed, azimuth, camera view or articulated sub- tracks angles. Manual control of all available degrees of freedom often leads to unwanted cognitive overload of the operator whose attention should be mainly focused on reaching the mission goals. On the other hand, there are fully autonomous systems requiring minimal attention but allowing almost no interaction which is usually not acceptable for the operator. Operator-accepted level of autonomy is usually a trade-off between fully teleoperated and completely autonomous robots. The main contribution of our paper is extensive survey on accepted autonomy solutions for Search and Rescue robots with special focus on traversing unstructured terrain, however brief summary of our system is also provided. Since, integral part of any Search and Rescue robot is the ability to traverse a complex terrain, we describe a system for teleoperated skid-steer robot with articulated sub-tracks (flippers), in which the operator controls robot speed and azimuth, while flipper posture and stiffness are controlled autonomously. The system for autonomous flipper control is trained from semi-autonomously collected training samples to maximize the platform stability and motion smoothness on challenging obstacles.

Relation to WP It directly contributes to Task1.1 - essential robot perception.

Availability Unrestricted. Included in the public version of this deliverable.

2.4 Pecka-MESAS2014, "Safe Exploration Techniques for Reinforcement Learning – An Overview"

Bibliography Martin Pecka and Tomáš Svoboda. Safe exploration techniques for reinforcement learning - an overview. In Jan Hodicky, editor, Modelling and Simulation for Autonomous Systems, volume 1 of Lecture Notes in Computer Science, pages 357-375. Springer.

Abstract We overview different approaches to safety in (semi)autonomous robotics. Particularly, we focus on how to achieve safe behavior of a robot if it is requested to perform exploration of unknown states. Presented methods are studied from the viewpoint of reinforcement learning, a partially-supervised machine learning method. To collect training data for this algorithm, the robot is required to freely explore the state space - which can lead to possibly dangerous situations. The role of safe exploration is to provide a framework allowing exploration while preserving safety. The examined methods range from simple algorithms to sophisticated methods based on previous experience or state prediction. Our overview also addresses the issues of how to define safety in the real-world applications (apparently absolute safety is un achievable in the continuous and random real world). In the conclusion we also suggest several ways that are worth researching more thoroughly.

Relation to WP It directly contributes to Task1.1 - essential robot perception.

Availability Unrestricted. Included in the public version of this deliverable.

2.5 Pecka-CVWW2015, "Safe Exploration for Reinforcement Learning in Real Unstructured Environments"

Bibliography Martin Pecka, Karel Zimmermann, and Tomáš Svoboda. Safe Exploration for Reinforcement Learning in Real Unstructured Environments. In Computer Vision Winter Workshop 2015.

Abstract In USAR (Urban Search and Rescue) missions, robots are often required to operate in an unknown environment and with imprecise data coming from their sensors. However, it is highly desired that the robots only act in a safe manner and do not perform actions that could probably make damage to them. To train some tasks with the robot, we utilize reinforcement learning (RL). This machine learning method however requires the robot to perform actions leading to unknown states, which may be dangerous. We develop a framework for training a safety function which constrains possible actions to a subset of really safe actions. Our approach utilizes two basic concepts. First, a "core" of the safety function is given by a *cautious simulator* and possibly also by manually given examples. Second, a classifier training phase is performed (using Neyman-Pearson SVMs), which extends the safety function to the states where the simulator fails to recognize safe states.

Relation to WP It directly contributes to the Task1.1 - Robot perception.

Availability Unrestricted. Included in the public version of this deliverable.

2.6 Nouza-ms2014, "Safe adaptive traversability learning for mobile robots"

Bibliography Tomáš Nouza. Safe adaptive traversability learning for mobile robots. June 2014, Master Thesis, Czech Technical University in Prague.

Abstract In mobile robotics it is necessary to predict a robot pose on a terrain to guarantee its stability when traversing an obstacle. Usual methods are based on an exact sim- ulation of a robot-surface interaction, but this requires a precise physical model, which can be hard to solve or can be too much complex. The aim of this thesis is to propose and experimentally evaluate an algorithm, based on machine learning methods, which predicts attitude of the robot (roll and pitch angles) in natural environment. The main contribution of this work lies in development and eval- uation of models, which can be used for predicting the safety of robot states and actions while interacting with the environment. Three models based on different multidimensional regression methods (linear, piecewise constant and Gaussian process) were trained and compared. As a part of this work, testing dataset was created and will be relased for the robotic community.

Relation to WP It contributes to the Task1.1 - Essential robot perception.

Availability Unrestricted. A public document `https://dspace.cvut.cz/handle/10467/24431` or `https://cyber.felk.cvut.cz/research/theses/detail.phtml?id=479`

2.7 Potocek-bc2014, "Probabilistic Approach to Landmark Management in Visual Odometry"

Bibliography Pavel Potoček. Probabilistic approach to landmark management in visual odometry, June 2014. Bachelor Thesis, Czech Technical University in Prague.

Abstract We implemented a FastSLAM 2.0-based algorithm for visual trajectory estimation and applied it to the NIFTi robot. We performed multiple experiments to validate the algorithm and measured its performance in various settings. We showed that our algorithm outperforms the existing solution in many of them. We proposed future changes to the algorithm that have a potential to further increase the performance.

Relation to WP It contributes to the problem of robot localization which is a one of the crucial steps when creating a metrical map.

Availability Unrestricted. A public document `https://cyber.felk.cvut.cz/research/theses/detail.phtml?id=503` or `https://dspace.cvut.cz/handle/10467/24249`

2.8 Vakula-ms2015, "Keypoint localization and matching in difficult scenes for visual odometry"

Bibliography Jan Vakula. Keypoint localization and matching in difficult scenes for visual odometry, January 2015, Master Thesis, Czech Technical University in Prague.

Abstract In the thesis we tested and improved an algorithm developed by Jiri Divis. Initially, the previous algorithm was carefully tested and after assessment the main imperfections also improved. The biggest improvements was achieved by a pairing improving so-called key-points. After the modifications we tested the algorithm in various types of environments in which the USAR robot can get. The best results of the algorithm was observed in an outdoor environments where there was sufficient amount of a visible edges in the scene. In an indoor environments we encoun- tered a problem with the lack of lighting, this has led to recording of blurry images. Therefore we added a detection of blurry images to the algorithm under which the blurry images were deleted. By the testing of the previous and the improved algorithms was found out a significant improvement in its behaviour.

Relation to WP It contributes to the problem of robot localization which is a one of the crucial steps when creating a metrical map.

2.9 Derner-TR-2014-25, "Indexing Images for Visual Memory by Using DNN Descriptors – Preliminary Experiments"

Bibliography Erik Derner and Tomáš Svoboda. Indexing Images for Visual Memory by Using DNN Descriptors – Preliminary Experiments. Technical Report CTU–CMP–2014–25. Czech Technical University in Prague.

Abstract Visual memory in mobile robotics is important to make the local- ization of a robot robust to situations, when GPS or similar localization methods are not available. Unlike many conventional approaches us- ing local features, we use a holistic method that employs deep neural networks (DNNs) to calculate a global descriptor of the whole image. We consider a scenario in which a robot equipped with an omni- directional camera calculates and stores DNN descriptors of images together with the positions as it moves in the environment. When the position is unknown to the robot, the algorithm estimates it for a given omnidirectional image by matching it with the most similar database image. We compared our approach with a recently tested GIST-based ap- proach on the same dataset and we found out that the DNN-based approach yields better results. The experiments also show that the DNN-based algorithm is quite robust to partial occlusion, rotation and changes in lighting conditions.

Relation to WP It contributes to the problem of robot localization, Task1.1

Availability Unrestricted. A public document `ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/Derner-TR-2014-25.pdf`

2.10 Hrabalik-bc2014, "3D Point Cloud Registration, Experimental Comparison and Fusing Range and Visual Data"

Bibliography Aleš Hrabalík. 3D point cloud registration, experimental comparison and fusing range and visual data, June 2014. Bachelor Thesis, Czech Technical University in Prague.

Abstract Point cloud registration is an important process in mobile robotics, serving as the cornerstone of simultaneous localization and mapping. The contribution of our work is twofold: firstly, we compare local registration methods using high-quality datasets and a custom protocol. In terms of precision and robustness to initial pose displacement, the capabilities of the methods are explored in an unprecedented detail, overcoming any previous work that we know of. Secondly, we propose enhancements to a global, feature-based registration method that take advantage of visual information, specifically camera imagery. Proposed changes include an extension of the feature descriptor, and a modification of reference frame determination. To investigate the modified methods, a dataset containing visual data is created. Experimental results indicate a significant improvement over the original method.

Relation to WP It contributes to the Task1.2 - Metrical mapping.

Availability Unrestricted. A public document `https://cyber.felk.cvut.cz/research/theses/detail.phtml?id=508` or `https://dspace.cvut.cz/handle/10467/24268`

2.11 Kubelka-AURO2015, "Improving multi-modal data fusion by anomaly detection"

Bibliography Jakub Šimánek, Vladimír Kubelka and Michal Reinštein. Imroving multi-modal data fusion by anomaly detection. In *Autonomous Robots*. 2015. Accepted, to appear.

Abstract If we aim for autonomous navigation of a mobile robot, it is crucial and essential to have proper state estimation of its position and orientation. We already designed a multi-modal data fusion algorithm that combines visual, laser-based, inertial, and odometric modalities in order to achieve robust solution to a general localization problem in challenging Urban Search and Rescue environment. Since different sensory modalities are prone to different nature of errors, and their reliability varies vastly as the environment changes dynamically, we investigated further means of improving the localization. The common practice related to the EKF-based solutions such as ours is a standard statistical test of the observations—or of its corresponding filter residuals—performed to reject anomalous data that deteriorate the filter performance. In this paper we show how important it is to treat well visual and laser anomalous residuals, especially in multi-modal data fusion systems where the frequency of incoming observations varies significantly across the modalities. In practice, the most complicated part is to correctly identify the actual anomalies, which are to be rejected, and therefore here lies our major contribution. We go beyond the standard statistical tests by exploring different state-of-the-art machine learning approaches and exploiting our rich dataset that we share with the robotics community. We demonstrate the implications of our research both indoor (with precise reference from a Vicon system) as well as in challenging outdoor environment. In the final, we prove that monitoring the health of the observations in Kalman filtering is something, that is often overlooked, however, it definitively should not be.

Relation to WP It contributes to the Task1.1 - Essential robot perception.

Availability Unrestricted. Included in the public version of this deliverable.

2.12 Kubelka-JFR2015, "Robust Data Fusion of Multi-modal Sensory Information for Mobile Robots"

Bibliography Vladimír Kubelka, Lorenz Oswald, François Pomerleau, Francis Colas, Tomáš Svoboda, and Michal Reinštein. Robust Data Fusion of Multi-modal Sensory Information for Mobile Robots. In *Journal of Field Robotics*, to appear, accepted in 2014.

Abstract Urban Search and Rescue missions for mobile robots require reliable state estimation systems resilient to conditions given by the dynamically changing environment. We design and evaluate a data fusion system for localization of a mobile skid-steer robot intended for USAR missions. We exploit a rich sensor suite including both proprioceptive (inertial measurement unit and tracks odometry) and exteroceptive sensors (omnidirectional camera and rotating laser rangefinder). To cope with the specificities of each sensing modality (such as significantly differing sampling frequencies), we introduce a novel fusion scheme based on Extended Kalman filter for 6DOF orientation and position estimation. We demonstrate the performance on field tests of more than 4.4km driven under standard USAR conditions. Part of our datasets include ground truth positioning; indoor with a Vicon motion capture system and outdoor with a Leica theodolite tracker. The overall median accuracy of localization - achieved by combining all the four modalities - was 1.2% and 1.4% of the total distance traveled, for indoor and outdoor environments respectively. To identify the true limits of the proposed data fusion we propose and employ a novel experimental evaluation procedure based on failure case scenarios. This way we address the common issues like: slippage, reduced camera field of view, limited laser rangefinder range, together with moving obstacles spoiling the metric map. We believe such characterization of the failure cases is a first step towards identifying the behavior of state estimation under such conditions. We release all our datasets to the robotics community for possible benchmarking.

Relation to WP It contributes to the Taks1.2 - Building metrical maps.

Availability Unrestricted. Available in the public version of this deliverable.

2.13 Kong, Dong-Uck (2015), "Persistent mapping of dynamic environments by multiple aerial robots with multi-camera systems."

Bibliography Dong-Uck Kong. "Persistent mapping of dynamic environments by multiple aerial robots with multi-camera systems." Unpublished Technical Report, January 2015.

Abstract One of the key aspects of WP1 is to extract useful metric environment models from multiple robots perceptional experience and to make the models persistent. Based on the requirements for the TRADR multi-robot system, the central question was formulated: How can particularly efficient mapping and localization methods be developed with respect to measurement quality, sensor data quantity and long-term data analysis for both robot and human action forces?

Relation to WP This technical report presents the current state and detailed roadmap of the work on metrical maps in T1.2, in particular related to the UAV.

Availability Restricted. Not included in the public version of this deliverable.

2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Adaptive Traversability of Unknown Complex Terrain with Obstacles for Mobile Robots

Karel Zimmermann[1], Petr Zuzanek[2], Michal Reinstein[3], and Vaclav Hlavac[4]

***Abstract*— In this paper we introduce the concept of Adaptive Traversability (AT), which we define as means of autonomous motion control adapting the robot morphology—configuration of articulated parts and their compliance—to traverse unknown complex terrain with obstacles in an optimal way. We verify this concept by proposing a reinforcement learning based AT algorithm for mobile robots operating in such conditions. We demonstrate the functionality by training the AT algorithm under lab conditions on simple EUR-pallet obstacles and then testing it successfully on natural obstacles in a forest. For quantitative evaluation we define a metrics based on comparison with expert operator. Exploiting the proposed AT algorithm significantly decreases the cognitive load of the operator.**

I. INTRODUCTION

Tracked robots with several articulated parts such as legs or subtracks—referred to as flippers, see Fig. 1—have been studied intensively since the design of robot morphology directly influences the ability to traverse complex terrain, especially with natural unstructured obstacles. Possessing a high number of articulated parts inevitably yields more degrees of freedom that have to be controlled. To reach a suitable pose to traverse such terrain in a safe way may become easily intractable, even for an expert operator. Controlling such many degrees of freedom also requires more time and poses a significant cognitive load onto the human operator. This may have crucial effect on the success of any Search & Rescue mission, that we primarily aim for [1], as well as influence on the robot safety.

We call this task *Adaptive Traversability* (AT), which we define as means of autonomous motion control adapting the robot morphology (configuration of flippers and their compliance) to traverse unknown complex terrain with obstacles in an optimal way. Our metrics for optimality is based on comparison of the AT autonomous regime to the control of an expert operator with respect to time taken for traversal, robot safety and smoothness of transitions. Beside having an edge in these criteria, the ultimate merit of using AT lies in minimal cognitive load for the operator.

Many approaches focus on optimal robot motion control in an environment with a known map, leading rather to the research field of trajectory planning. In contrary to planning [2] [3], the AT can easily be exploited in unknown environment and hence provide a crucial support to the actual procedure of map creation. From the conceptual point of view, the AT is intended to run one level below any SLAM or trajectory planning algorithms and its input commands can either be directly from the operator (usually unknown-map case) or from a planner. We would like to emphasize to perceive AT rather as independent complement to trajectory planning and in no way a substitution. If the task of AT was to be solved by means of trajectory planning, a reliable map is required, providing detailed information on Robot-Terrain Interaction (RTI) (e.g. estimation of stability, slippage coefficient, power consumption, robots full 3D pose etc). Such RTI can be theoretically estimated from the terrain shape and a physical model of the robot and used to build traversability maps [4] [5]. However this modeling is analytically very complex, computationally demanding and in specific cases such as high slippage or aerial motion phases often inaccurate and unreliable. This is not viable solution for many applications, especially when the robot is controlled in an unknown environment. Therefore, in our approach to AT we rather propose to process only the instantaneous RTI properties locally as the robot traverses and explores the environment. The only way to obtain such RTI properties is prediction online using machine learning techniques [6], [7], [8]. We adopted this approach to RTI already in applications such as predicting correction coefficients of robot odometry [9] or estimating stride length of a legged robot while slipping [10] [11]. Since the RTI is predicted only locally, greedy optimization of inaccurately estimated RTI criterion can easily lead the robot into the

Fig. 1. **Adaptive traversability:** Robot configuration of the 4 flippers (subtracks) and their compliance is controlled autonomously in order to adapt to the terrain and traverse it in an optimal way. The robot is equipped with rotating laser scanner SICK LMS-151, Ladybug 3 omnicam, Xsens MTi-G IMU, and independent flipper stiffness control for each subtrack.

The authors were supported as follows: [1]K.Z. by the Czech Science Foundation Project 14-13876S, [2]P.Z. by the SGS13/142/OHK3/2T/13 of the CTU in Prague, [3]M.R. by the EU-FP7-ICT-609763 TRADR, and [4]V.H. by the Project TE01020197 of Technology Agency of the Czech Republic.

K.Z. and P.Z. are with the Center for Machine Perception, Dep. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague (e-mails: {zimmerk,zuzanp1}@cmp.felk.cvut.cz). M.R. and V.H. are with the Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague (e-mails: {michal.reinstein,hlavac}@ciirc.cvut.cz).

dangerous state. Therefore, we propose to rather compute the RTI criterion directly during the training phase and learn to predict the expected sum of RTI criterion values, which can be obtained with the robot from its current state for different controlling strategies. This formulation naturally leads to the reinforcement learning task, where the RTI criterion corresponds to the reward.

Our main contributions lie in (i) defining the concept of AT for mobile robots, and (ii) proposing a reinforcement learning based AT algorithm for autonomous morphology adaptation that improves the motion control even in complex unknown environment. We (iii) demonstrate the functionality by training the AT under lab conditions on simple EUR-pallet obstacles and testing it on natural obstacles in a forest. For quantitative evaluation we (iv) define a metrics based on comparison with expert operator.

The paper is structured as follows: Section II introduces the related work, Section III describes our proposed solution, Section IV summarizes the experimental evaluation and Section V concludes the implications of our work.

II. Related Work

An ample amount of work has been devoted to the field of Robot Terrain Classification (RTC) [12], [13], [14], [15] where terrain features are mapped on discrete classes of given properties. On the conceptual level, this is relevant to extracting the terrain properties for the RTI. However, these RTC defined classes are often weakly connected with the way the robot can actually interact with the terrain or the connection is lacking. Few papers describe the estimation of RTI directly, for example, Kim et al. [7] estimate whether the terrain is traversable or not, and Ojeda et al. [8] estimate power consumption on a specific terrain type. In the literature, the RTI properties can be specified explicitly (e.g. robot consumption [8]) or implicitly (e.g. state estimation correction coefficient [9]). The problem of the AT in the way we approach it using reinforcement learning is a road less traveled in robotics, but though the target application differs, highly relevant is the work of Abbeel et al. [16], [17]. There are also alternative solutions, based for instance on kinematic model of the robot [18], [19], or by achieved learning a direct mapping between terrain features and robot actions [20], [21]. However, analytical modeling of the RTI is in general very difficult and simplifications cannot be avoided. On contrary to [16] we omit this modeling since it is not needed in our approach and instead of using Value-based algorithms, we rather focus on Q-learning technique.

III. Adaptive Traversability by Reinforcement Learning

We solve the adaptive traversability problem for a tracked robot[1] equipped with four flippers, see Fig. 1. The sensor suite of the robot consists of a rotating 2D laser scanner (SICK LMS-151) mounted in front of the robot (the rotation of the scanner provides the 3D scans), a Point Grey Ladybug 3 omni-directional camera, and a Xsens MTi-G inertial measurement unit (IMU) with GPS.

It is expected that the speed and azimuth of the robot is controlled by the operator (or provided by a path planner), and the task is to control the configuration of the four flippers and their compliance. Compliance of flippers is obtained by measuring the actual current in flipper drives and setting a threshold on the maximum allowed current. This threshold is called the *compliance threshold*.

To simplify such 8-dimensional task, we defined five discrete flipper modes specifying the angle and the compliance threshold for all four flippers. The task is to switch between these flipper modes (denoted by $c \in \mathbb{Z}$) in order to collect maximum sum of *rewards* over the obstacle being traversed. We define reward function $r(c, s) : (\mathbb{Z} \times \mathbb{R}^n) \rightarrow \mathbb{R}$, which assigns a real valued reward for achieving state s while using mode c. We experimented with several types of the reward function, which are described and evaluated in Section IV. For now, we define the reward function as a weighted sum of (i) user denoted penalty (reward) specifying that the state is (not) dangerous, (ii) high pitch angle penalty (considering robot safety from flip-over), (iii) excessive flipper mode change penalty, (iv) robot forward speed reward (for making progress in traversing), and (v) motion roughness (smoothness) penalty (reward).

A. Reinforcement learning algorithm

To tackle this problem, the reinforcement learning technique is used. We define function $Q(c, s) : (\mathbb{Z} \times \mathbb{R}^n) \rightarrow \mathbb{R}$, which estimates expected sum of discounted rewards, when the robot is in state s and flippers are set to mode c and the robot will be controlled optimally from the following state onward. Such function allows for the following recursive definition:

$$Q(c, s) = \sum_{s'} p(s'|c, s) \Big[r(c, s) + \gamma \max_{c'} Q(c', s') \Big] \quad (1)$$

where $p(s'|c, s)$ is transition probability that the robot, which is in state s with flippers set to mode c will get to the following state s'. Discount factor $\gamma \in \{0, 1\}$ is used to reduce the influence of distant future rewards. If such function is known, the optimal flipper mode c^* for the robot in the state s is chosen as follows

$$c^* = \arg\max_{c} Q(c, s) \quad (2)$$

Since we want to avoid the learning of $p(s'|c, s)$, function $Q(c, s)$ is learned using modification of the *fitted Q-iteration* algorithm summarized in Alg. 1. The proposed algorithm repeats the Q-learning procedure for several episodes. Training data collected for the first episode (line: 3) are obtained by an expert operator. To speed up the learning process, also *reasonably* negative training samples (with negative rewards) are provided. Once we are satisfied with the performance on validation data (also collected and annotated by the expert operator), we start to collect the training data with autonomously chosen flipper modes, i.e. chosen according to Eq. (2). When a batch of the training data is collected, the

[1]Developed as part of NIFTi project `http://www.nifti.eu`

$Q(c, s)$ function is trained in lines 4-7. Since we defined the $Q(c, s)$ as a collection of piecewise constant functions, the solution of the problem from line 6 is detailed in Section III-B. Section III-C describes features representing the state and the feature selection method we used.

//Initialization
1: $Q(c, s) = 0 \quad \forall_{c,s}$
2: **while** (adaptive traversability is not good) **do**
3: Drive the robot over training obstacles and assign rewards. Captured training data consists of sequences $[(s^1, c^1, r^1), (s^2, c^2, r^2), \dots]$.
// Train $Q(c, s)$
4: **repeat**
5: $y^i = r^i + \gamma \Big[\max_{c'} Q(c', s^{i+1}) \Big] \quad \forall_i$
6: $Q(c, s) = \arg\min_{\overline{Q}} \sum_{i=1}^{N} \|\overline{Q}(c^i, s^i) - y^i\|_2^2$
7: **until** (convergence reached)
8: **end while**

Algorithm 1: Procedure of learning of the Q-function.

B. Piecewise constant function learning

In our approach, the $Q(c, s)$ is collection of mode specific functions $Q_c(s)$ corresponding to the number of flipper configuration modes. Since the procedure of learning is same for all functions $Q_c(s)$, we omit index c and focus on the learning of a regression function $Q(s)$ for N training samples $(s^1, y^1) \dots (s^N, y^N)$ prepared in the line 5 of Alg. 1. The upper index i is used to denote training samples, the lower index k is used to denote features. Concatenation is denoted by square brackets.

We define $Q(s) = \sum_{k=1}^{K} q_k(s_k)$ as the sum of piecewise constant functions $q_k(s_k)$ of features $s_k \in \mathbb{R}$, where K denotes the number of features. Features s_k are normalized to have zero mean and unit covariance. Feature values are divided into U equally sized bins [2]. To simplify the notation, we define a bin assigning function $\Omega(s_k) : \mathbb{R} \to \mathbb{N}$ which assigns corresponding bin u to feature value s_k.

Response of the regression function $Q(s)$ is then computed as follows:

$$Q(s) = \sum_{k=1}^{K} q_k(s_k) = \sum_{k=1}^{K} \Lambda_{k,\Omega(s_k)}, \tag{3}$$

where $\Lambda_{k,\Omega(s_k)} \in \mathbb{R}$ is the constant response of feature function q_k on feature value s_k.

Substituting Eq. (3) into the problem in Alg. 1, line: 6, we obtain the corresponding least squares problem:

$$\Lambda = \underset{\overline{\Lambda} \in \mathbb{R}^{\mathbb{K} \times \mathbb{U}}}{\arg\min} \sum_{i=1}^{N} \Big\| \sum_{j=1}^{k} \overline{\Lambda}_{j,\delta(s_j^i)} - y^i \Big\|^2. \tag{4}$$

To write the solution of (4) in a compact form, we further introduce a binary matrix

$$[\mathbb{A}]_{i,(ku)} = \begin{cases} 1 & \text{if } \Omega(s_k^i) = u \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where index i determines the row and indices (ku)[3] determine the column. We also introduce a vector $\lambda = [\Lambda_{1,1} \dots \Lambda_{K,U}]^\top$, which is concatenation of all unknown lambdas from all bins and all features. Finally, we form the vector $\mathbf{y} = [y^1 \dots y^N]^\top$ with all desired values. The solution of problem (4) is then

$$\lambda = \underset{\overline{\lambda}}{\arg\min} \Big\| \mathbb{A}\overline{\lambda} - \mathbf{y} \Big\|^2 = \mathbb{A}^+ \mathbf{y}, \tag{6}$$

where $\mathbb{A}^+$ denotes Moore-Penrose pseudo-inverse of matrix $\mathbb{A}$.

C. State representation and feature selection

We represent the mutual state of the robot and the local neighboring terrain as N-dimensional feature vector $s \in \mathbb{R}^N$. Features are selected from a feature pool, which consists of: **Terrain shape features:** Since the robot is equipped with the laser scanner, we merge individual scans into a point cloud 3D map exploiting the ICP algorithm [22]. The point cloud map in the local neighborhood of the robot is further transformed into the Digital Elevation Map (DEM), see Fig. 2, capturing the local spatio-temporal representation of the terrain. To represent the terrain shape in a compact form, Haar-like features are computed using the DEM values. In addition to that parameters of the plane fitted into the neighboring terrain are used.

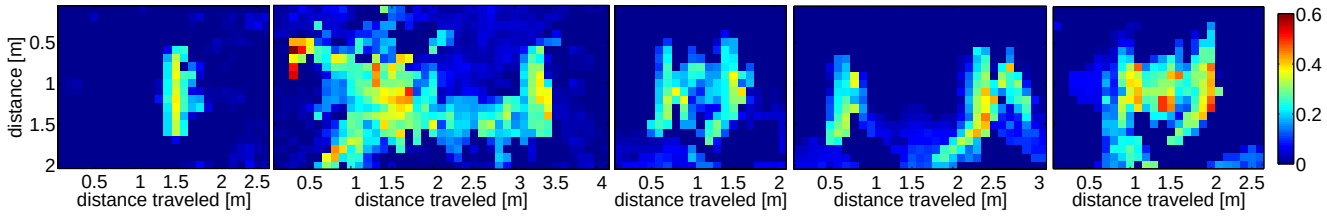
Robot state and configuration features: Robot speed (both actual and requested by the operator), pose (pitch, roll, yaw), flipper angles, compliance thresholds and actual flipper mode. To estimate the velocity of the robot, terrain adaptive odometry method [9] is used and combined with IMU data and information provided by the ICP using the Extended Kalman filter (EKF). The precise and stable pitch and roll angles are obtained using a complementary filter [23]. In addition to this information, currents in the flipper and the main track drives are used to provide the knowledge about the weight distribution and ground contacts.

We select a set of suitable features from the feature pool S by a forward stage-wise feature selection strategy [24] based on Gram-Schmidt orthogonalization process. More formally, we are given a training set $\{(\mathbf{S}^1, y^1), \dots (\mathbf{S}^N, y^N)\}$ consisting of N training samples, where $\mathbf{S}^i$ are M-dimensional vectors containing values of all features from the feature pool. Especially, we denote S_k^i as the k-th feature value of the i-th training sample.

This proposed feature selection method is summarized in Alg. 2. It successively builds the feature set from the features minimizing residuals Δy^i of all training samples $i = 1 \dots N$. Initially we equal residual of the i-th training sample $\Delta y^i = y^i$. In each training stage, the algorithm estimates coefficients for all features $k = 1 \dots M$ and greedily selects the feature with the lowest residual error. Such feature is added to the list of selected features and the algorithm continues while the validation error is decreasing.

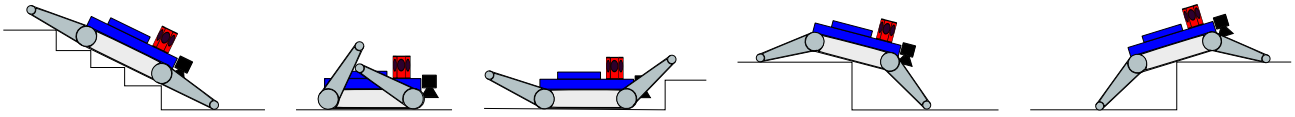
[2] except the size of border bins which are $[-\infty, \texttt{min\_value}]$ and $[\texttt{max\_value}, +\infty]$

[3] (ku) denotes a linear combination of indices k and u corresponding to the vectorization of Λ.



(1) simple obstacle (2) soft-terrain side-roll obstacle (3) long obstacle (4) double obstacle (5) complex obstacle

Fig. 2. **Obstacles:** Digital elevation maps (DEM) of testing obstacles constructed during experiments.



I-shape V-shape L-shape U-shape soft U-shape hard

Fig. 3. **Defined flipper modes:** Definition of five flipper modes corresponding to different morphological configurations of varying properties.

//Initialization

1: $\Delta y^k = y^k, \quad K = 1$

2: **while** (validation error is decreasing) **do**

// Select the feature k^* with the lowest residual error
// from the feature pool.

3: $k^* = \arg\min_{(k,\Lambda_K)} \sum_{i=1}^{N} \|Q([\mathbf{s}_1^i \dots \mathbf{s}_{K-1}^i S_k^i]) - y^i\|_2^2$

// Add the selected feature S_{k^*} into $\mathbf{s}$.

4: $\mathbf{s} = [\mathbf{s} \;\; S_{k^*}]$

// Update residuals

5: $\Delta y^i = y^i - Q(s^i) \; \forall_i$

6: $K = K + 1$

7: **end while**

Algorithm 2: Feature selection procedure

IV. EXPERIMENTAL EVALUATION

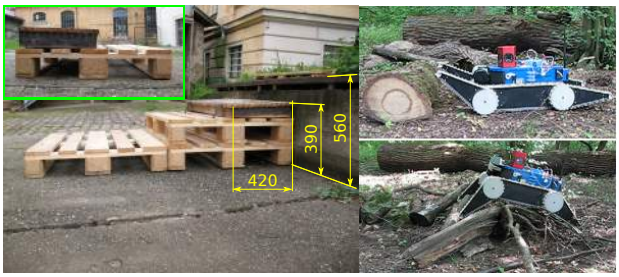
Our proposed solution to the AT was tested on five challenging obstacles created from woods and stones in an outdoor forest environment[4]; see the examples in Fig. 1, Fig. 4b and the digital elevation maps (DEM) of testing obstacles computed online by the robot Fig. 2. Each obstacle was traversed multiple times with autonomous flipper control (AC) following the Eq. (2).; obstacles $1, 2$ and 5 were also traversed with manual flipper control (MC) by the expert operator for the purpose of quantitative comparison. We emphasize that the complexity of testing obstacles was selected in order to challenge robots hardware capabilities. One of the testing obstacles even proved to be too complex to be traversed neither with the AC nor MC.

The rest of this section is organized as follows: Section IV-A describes training procedure, Section IV-B summarize the testing procedure. Section IV-C provides the comparison and evaluation.

A. Training Procedure

We define five morphological configurations—five different flipper modes (Fig. 3) (i) **I-shape** with unfolded flippers (useful for traversing holes or stairs), (ii) **V-shape** with flippers folded in order to provide the best observation

[4]For better comprehension, see the attached multimedia showing one testing drive over obstacle 2 from Fig. 2.



(a) Training objects (b) Testing objects

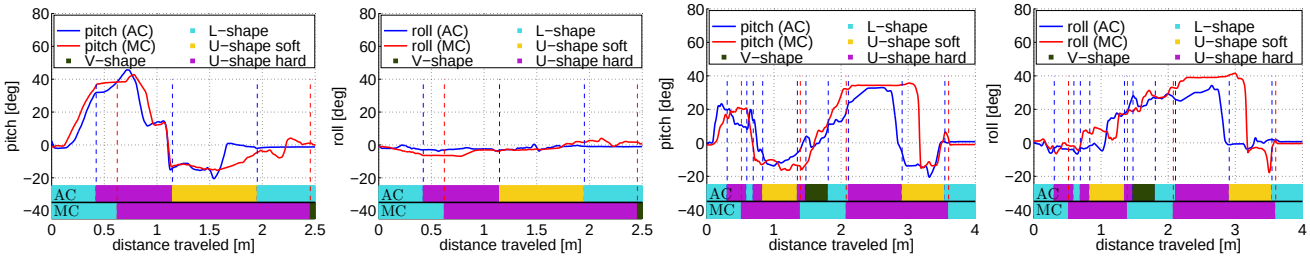
Fig. 4. **Obstacles:** (a) Three EUR pallets with one non standard pallet and concrete shoal used for training part. (b) Natural obstacles in an outdoor forest environment used for testing part.

capabilities to the robot, (iii) **L-shape** with front flippers raised (suitable for climbing up), (iv) **U-shape soft**, pushing the flippers down with low pressure—low compliance threshold (suitable for smooth climbing down), and (v) **U-shape hard**, pushing the flippers down with high pressure—high compliance threshold.

Our proposed approach to AT was trained in controlled lab conditions using only two artificial obstacles created from EUR pallets [5]. First training obstacle was just a single pallet, the second consisted of stairs created from three pallets (see Fig. 4a).

We trained the Q-function according to the Alg. 1 in three episodes, i.e. three iterations of the while-loop. In the first two episodes, the training data were collected with manual flipper control. To speed up the learning procedure *reasonably* negative (but not dangerous) training samples were provided. In the last episode, the training data were collected autonomously by the robot. Each training sample was accompanied by its reward. The best results were achieved for the reward function defined as a weighted sum of (i) manually annotated labels reflecting success of the operator's goal (either positive equal to 1 or negative equal to -1), (ii) thresholded exponential penalty for pitch angle, and (iii) roughness of motion penalty defined as $\sqrt{v_y^2 + v_z^2}$. In order to reduce oscillations between modes

[5]type EUR 1: 800mm $\times$ 1200mm $\times$ 140mm



(a) **Obst.1:** pitch + flipper modes (b) **Obst.1:** roll + flipper modes (c) **Obst.2:** pitch + flipper modes (d) **Obst.2:** roll + flipper modes

Fig. 5. **Pitch, roll, and flipper modes along the obstacle traversal:** All graphs show used flipper modes and pitch+roll angles reached by the robot during the obstacle traversal. Graphs (a) and (b) correspond to the obstacle 1, graphs (c) and (d) to the obstacle 2. Autonomous control (AC) is depicted by the blue color and manual control (MC) by the red color. Autonomously selected modes are shown in the first color bar and manual in the second.

with similar Q-values we (i) introduce additional penalty for changing the mode and (ii) evaluate the Q-values over 1 second long time interval. In each episode, the training of the Q-functions was iterated 30 times. The number of iterations was experimentally determined as sufficient for the convergence of the Q-values with $\gamma = 0.8$. To achieve a well conditioned training dataset, the training samples were artificially perturbed several times.

B. Testing Procedure

We tested the AC method on five challenging natural obstacles in a forest. Both the AC and MC allowed to traverse obstacles $1-4$ (see Fig. 2). Obstacle 5 consisted of two woods located in parallel with the mutual distance equaled approximately to the length of robot with folded flippers. Such obstacle turned out to be not traversable neither with the autonomous nor with the manual flipper control. For obstacles 1 and 2 quantitative comparison of the autonomous and manual flipper control is provided in Tab. I, II. To compare AC and MC traversability quality, five different metrics were proposed and evaluated: (i) average pitch angle (sum of absolute values of the pitch angle divided by the number of samples), (ii) average roll angle, (iii) traversal time (start and end points are defined spatially), (iv) average current in flipper engines (corresponds to flipper torque), (v) overall power consumption during the whole experiment, and (vi) number of mode changes.

C. Results

Tab. I shows that the average pitch, roll and the number of changes of the AC and MC on the obstacle 1 are comparable. However, the power consumption and the average current are both lower for the AC. This is achieved by more efficient mode selection—such as using the U-shape soft mode for going down from the obstacle—, see the flipper modes, pitch and roll angle plots of AC and MC in Fig. 5a,b.

Tab. II clearly demonstrates that the AC outperformed MC in most of evaluated metrics. The most significant difference can be seen in the actual time taken. While MC often required to stop the robot and wait for the end of the mode change procedure, the AC was continuous and proceeded as the robot was driven forward. Therefore, the traversal time of the AC is almost twice as short. In addition to that, since

TABLE I

COMPARISON OF AUTONOMOUS AND MANUAL ROBOT CONTROL ON THE OBSTACLE 1 (SIMPLE OBSTACLE).

	Pitch [°]	Roll [°]	Time [s]	Current [A]	Changes [−]	Consumption [Ah]
AC	11.2	1.8	35.7	3.4	3	0.07
MC	11.3	2.8	36.8	5.4	2	0.10

TABLE II

COMPARISON OF AUTONOMOUS AND MANUAL ROBOT CONTROL ON THE OBSTACLE 2 (CONTAINS SOFT-TERRAIN AND SIDE-ROLL).

	Pitch [°]	Roll [°]	Time [s]	Current [A]	Changes [−]	Consumption [Ah]
AC	10.2	10.6	75.3	3.9	10	0.17
MC	17.1	17.1	132.1	4.6	4	0.33

our definition of the reward function also contains penalty for being in extreme angles (accounting for robot safety), the AC achieved smaller pitch, roll, as well as flipper current—the ground/obstacle contacts were less frequent and less intense. The power consumption of AC compared to MC was hence much lower, enabling the robot to last longer while carrying out the mission.

On the other hand, the number of mode changes of AC is higher. To explain it, we need to analyze the actual motion trajectory and corresponding obstacle in detail (see also Fig. 5 c,d): The first part of the obstacle is created from many flexible sticks behaving as a soft terrain–and thus deforming under the robot weight. Since we used only the EUR pallets, this RTI property was not represented in the training set at all. The robot correctly starts in the L-shape mode to climb on the obstacle.Then it switches to the U-shape hard mode to lift its body on the obstacle. However, the soft terrain collapses under the body weight and L-shape must be used again to traverse the remaining hard part of the obstacle. Similar scenario repeats, when traversing the middle part of the obstacles and can be in general expected on similar terrain.

V. CONCLUSION

In this paper we have concentrated our efforts on defining the task of *Adaptive Traversability* (AT) as means of autonomous motion control adapting the robot morphology (configuration of flippers and their compliance) in order to

traverse unknown complex terrain with obstacles. Similar approaches have been deployed using trajectory planning for scenarios, where a map of the environment was available, hence providing an easy way to compute the Robot-Terrain Interaction. However, we propose a solution based on reinforcement learning that exploits only the information from local RTI and does it online, hence no map is needed—solution ideal for exploring unknown environments with obstacles. Having experience from real deployment of robots in Search & Rescue scenarios, we are aware of the crucial impact of cognitive load on the operator. Therefore, we define a metrics allowing us to compare our solution to an expert operator driving the robot manually. Beside outperforming the manual control in a number of criteria (time taken for traversal, power consumption, smoothness and safety of the robot), the main accomplishment lies in the minimal cognitive load required for the robot control while using our AT solution. Moreover, our approach can easily be used together with any trajectory planning algorithm in general in a complementary way. We would like to also point out, that for the actual training, only simple obstacles made of EUR pallets were used, but the actually testing was successfully performed using challenging natural obstacles in a forest environment.

To conclude, on the testing dataset, the proposed AT algorithm exhibited very stable behavior such as: (i) **Repeatability:** consistent flipper control over multiple traversals of the same obstacle, (ii) **Robustness:** training with similar parameters and similar training data yields similar behavior (iii) **Generalization:** reasonable and explainable flipper control on the challenging testing data—no deformable obstacles in the training dataset, yet surprisingly good performance on such deformable terrain during testing.

As a future work, we clearly see the opportunity in expanding all of our assumptions made: we can define more different modes, exploit more compliance thresholds, we can allow the robot to train by itself on much larger scale of obstacles, as well as to push the challenge of the testing environments. We also intend to integrate the AT algorithm with our SLAM solution and trajectory planner to expand the range of field applications and possibilities.

REFERENCES

[1] G.-J. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petricek, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, and S. Pratzler-Wanczura, "Experience in system design for human-robot teaming in urban search & rescue," in *Field and Service Robotics (FSR), 2012 8th International Conference on*, July 2012.

[2] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, "3d path planning and execution for search and rescue ground robots," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 722–727, Nov 2013.

[3] M. Brunner, B. Bruggemann, and D. Schulz, "Towards autonomously traversing complex obstacles with mobile robots with adjustable chassis," in *Carpathian Control Conference (ICCC), 2012 13th International*, pp. 63 –68, may 2012.

[4] S. Martin, L. Murphy, and P. Corke, "Building large scale traversability maps using vehicle experience," in *Internation Symposium on Experimental Robotics*, 2012.

[5] B. Cafaro, M. Gianni, F. Pirri, M. Ruiz, and A. Sinha, "Terrain traversability in rescue environments," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE Int. Symposium on*, pp. 1–8, Oct 2013.

[6] K. Ho, T. Peynot, and S. S. Sukkarich, "Traversability estimation for a planetary rover via experimental kernel learning in a gaussian process framework.," in *Internation Conference on Robotics and Automation (ICRA)*, 2013.

[7] D. Kim, J. Sun, S. Min, O. James, M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *In Proc. of International Conference on Robotics and Automation (ICRA)*, pp. 518–525, 2006.

[8] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, pp. 103–122, 2006.

[9] M. Reinstein, V. Kubelka, and K. Zimmermann, "Terrain adaptive odometry for mobile skid-steer robots," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2013. *accepted—to appear*.

[10] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pp. 617–624, 2011.

[11] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information," *IEEE Trans. on Robotics*, vol. 29, pp. 563–571, April 2013.

[12] C. Weiss, H. Frohlich, and A. Zell, "Vibration-based terrain classification using support vector machines," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, pp. 4429–4434, 2006.

[13] K. Kim, K. Ko, W. Kim, S. Yu, and C. Han, "Performance comparison between neural network and SVM for terrain classification of legged robot," in *Proc. SICE Annual Conf. 2010*, pp. 1343–1348, 2010.

[14] E. M. DuPont, C. A. Moore, and R. G. Roberts, "Terrain classification for mobile robots traveling at various speeds: An eigenspace manifold approach," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, pp. 3284–3289, 2008.

[15] P. Komma, C. Weiss, and A. Zell, "Adaptive bayesian filtering for vibration-based terrain classification," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, pp. 3307–3313, 2009.

[16] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *In Advances in Neural Information Processing Systems 19*, p. 2007, MIT Press, 2007.

[17] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proc. of the 22nd int. conf. on Machine learning*, ICML '05, (New York, NY, USA), pp. 1–8, ACM, 2005.

[18] Y. Okada, K. Nagatani, K. Yoshida, S. Tadokoro, T. Yoshida, and E. Koyanagi, "Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning," *J. Field Robot.*, vol. 28, no. 6, pp. 875–893, 2011.

[19] K. Ohno, S. Morimura, S. Tadokoro, E. Koyanagi, and T. Yoshida, "Semi-autonomous control system of rescue crawler robot having flippers for getting over unknown-steps," in *Intelligent Robots and Systems (IROS), 2007. IEEE/RSJ Int. Conf. on*, pp. 3012–3018, 2007.

[20] R. Sheh, B. Hengst, and C. Sammut, "Behavioural cloning for driving robots over rough terrain," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 732–737, 2011.

[21] E. Uğur and E. Şahin, "Traversability: A case study for learning and perceiving affordances in robots," *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, 2010.

[22] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, pp. 133–148, Feb. 2013.

[23] V. Kubelka and M. Reinstein, "Complementary filtering approach to orientation estimation using inertial sensors only," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pp. 599–605, 2012.

[24] F. W. Roush, "Applied linear regression : Snaford weisberg, new york: Wiley, 1980, pp. 283," *Mathematical Social Sciences*, vol. 3, July 1982.

Accepted Autonomy for Search and Rescue Robotics

Petr Zuzanek, Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, zuzanp1@cmp.felk.cvut.cz, Czech Republic
Karel Zimmermann, Center for Machine Perception, Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, zimmerk@cmp.felk.cvut.cz, Czech Republic.
Vaclav Hlavac, Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, hlavac@cmp.felk.cvut.cz, Czech Republic.

Abstract

Since exploration of unknown disaster areas during Search and Rescue missions is often dangerous, teleoperated robotic platforms are usually used as a suitable replacement for a human rescuer. Advanced robotic platforms have usually many degrees of freedom to be controlled, e.g. speed, azimuth, camera view or articulated sub-tracks angles. Manual control of all available degrees of freedom often leads to unwanted cognitive overload of the operator whose attention should be mainly focused on reaching the mission goals. On the other hand, there are fully autonomous systems requiring minimal attention but allowing almost no interaction which is usually not acceptable for the operator. Operator-accepted level of autonomy is usually a trade-off between fully teleoperated and completely autonomous robots.
The main contribution of our paper is extensive survey on accepted autonomy solutions for Search and Rescue robots with special focus on traversing unstructured terrain, however brief summary of our system is also provided.
Since, integral part of any Search and Rescue robot is the ability to traverse a complex terrain, we describe a system for teleoperated skid-steer robot with articulated sub-tracks (flippers), in which the operator controls robot speed and azimuth, while flipper posture and stiffness are controlled autonomously. The system for autonomous flipper control is trained from semi-autonomously collected training samples to maximize the platform stability and motion smoothness on challenging obstacles.

Keywords: Autonomy, Robot-terrain interaction, Search and Rescue, Traversability

1 Introduction

Figure 1: Search and Rescue mission: NIFTi UGV platform operating in simulated deployment. Earthquake disaster training field, Vigili del Fuoco Prato, Italy

Ground robotics systems offer huge potential for its utilization in Search and Rescue (SaR) missions. SaR robots may be used as suitable replacement for rescuers during investigation of poorly accessible locations or dangerous areas, and thus reduce the risk of injuries in target environments—Fig. 1 gives an example of NIFTi UGV deployment in SaR mission. Advanced robotic platforms consist of many degrees of freedom (DoF) that have to be controlled in time either manually or autonomously.

While the teleoperated platforms allow their full control, many DoF in such system might lead to unwanted cognitive overload of the operator. Even more in SaR scenarios. Rescuers operate under extreme cognitive load and stress as a consequent of dangerous environment they are deployed in, and their attention should be primarily paid for reaching the mission goals—such as victim search and rescue, analyzing risk of secondary disaster, etc. So, what can be done to improve such behavior? An ideal scenario is to use only artificial intelligence (AI) and let the robots follow the commands. So far, AI has not capabilities to be deployed in robots operating in complex dynamic environments and thus systems based on human-robot interaction still outperforms the autonomous solutions. The amount of cooperation and task-sharing can vary

depending on the scenario as well as platform construction and capability. If such human-robot cooperation should bring a benefit into search and rescue field, it is necessary to find an accepted level of autonomy, suitable for the SaR, which is usually trade-off between teleoperation and full autonomy.

The aim of this paper is: (i) summarize related solutions in motion control suitable for Search and Rescue missions and (ii) briefly introduce our solution to operator-accepted autonomy designed for skid-steer NIFTi UGV robot with articulated flippers.

Rest of the paper is organized as follows: Sec. 2 provides survey on motion control by considering both: (i) related works—see Sec. 2.1 and (ii) our solution—outlined in Sec. 2. Conclusion is given in Sec. 3.

2 Survey on motion control

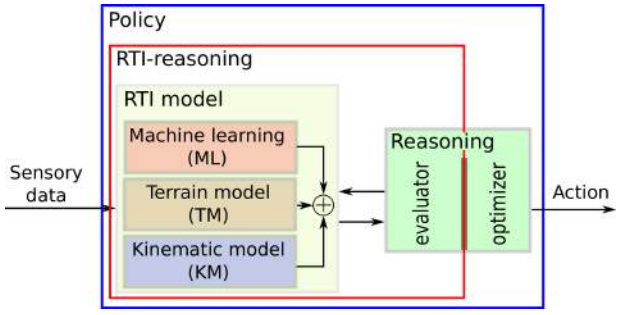


Figure 2: General design of motion control approaches: The mapping between robot state **s** coming as input and optimal action a to be taken can be done in several ways: (i) learning direct state-action association—policy (ii) reasoning about action quality based on the modeled robot-terrain interaction (RTI) or (iii) learning the state-action quality based on RTI—RTI-reasoning.

Over the last few years, there were published several motion control approaches whose solutions varied significantly depending on the application (target environment, autonomy level), platform capabilities, onboard computation power, etc. Instead of focusing on comparison of methods outcomes (which is nonsense due to above described differences), we provide comparison of motion control approaches on design level based on the framework depicted in Fig. 2.

The aim is to find a mapping (policy) between state described by sensory data and optimal motion control action. Since finding direct mapping might be difficult it is common to decompose the task into modeling and reasoning functions. While the modeling function aims in discovering robot-terrain interaction (RTI) based on the current state and possible action to take, reasoning function evaluates quality of the interaction and decides which action should be taken in order to ensure optimal behavior on terrain. Reasoning might be decomposed into evaluating and optimizing, but not every approach aims in such decomposition and rather design complex reasoning functions where both are hidden.

Note that retrieving real RTI is not feasible due to terrain and/or platform complexity and following can be considered as suitable replacement: (i) approximate models of interacting objects (often robot *Kinematic model (KM)* and simplified terrain representation—*Terrain model (TM)*), (ii) description substituted by knowledge—*Machine learning (ML)*. In this motion control framework, ML is a suitable method not only for learning RTI, but also for learning evaluator response (RTI-reasoning) or whole policy (as depicted in Fig. 2 by red and blue rectangles).

The above introduced terms can be formalized as follows:

- State $\ldots \mathbf{x} \in \mathcal{S} \subset \mathbb{R}^D$,
- Action$\ldots \mathbf{a} \in \mathcal{A} \subset \mathbb{R}^d$,
- Policy$\ldots \pi : \mathcal{S} \rightarrow \mathcal{A}$,
- RTI $\ldots \oplus : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}^n$,
- Reasoning $\ldots \mathcal{R} : \mathbb{R}^n \rightarrow \mathcal{A}$,
- Evaluating$\ldots \mathcal{E} : \mathbb{R}^n \rightarrow \mathbb{R}^m$,
- Optimizing$\ldots * : \mathbb{R}^m \rightarrow \mathcal{A}$,
- RTI-reasoning $\ldots \hat{\mathcal{E}} : \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}^m$,

where $\mathcal{S}$ is set of states, $\mathcal{A}$ set of actions, $\mathbb{R}^x$ denotes $x-$dimensional vector of real numbers. This notation is especially used in Tab. 1 for sake of compactness.

Tab. 1 do the categorization based on Fig. 2 including the description of expected motion control outcome for given task. The rest of this section is focused on detailed description of approaches lying in Tab. 1. Related work is described in Sec. 2.1 while our solution (Zimmermann et al. [24]) is described in Sec. 2.2.

2.1 Related work

Dube [6] modeled flipper-ground contact points using kinematic robot description and terrain slope to estimate flippers and manipulator poses of PackBot robot in order to enhance its stability in complex terrain. Ohno et al. [17] aimed in development of a semi-autonomous active control system for the rescue crawler Aladdin. The reasoning about optimal control is made using judgment based rules designed in advance by experience. Such judgment considers flipper-ground contact points determined reactively using KM and proprioceptive measurements. Nagatani et al.[16] and Okada et al. [18] were targeting in development of active flipper control for their tracked robots Kenaf and its second generation Quince. These works resulted in development of shared autonomy system [19] which already had real deployment [15]. RTI is described by means of kinematic model and local terrain properties gathered along the main tracks. Kinematic model and terrain properties are used for

Author	Platform	RTI model	Reasoning	Output
Brunner et al. [1]	Telerob Telemax robot	KM⊕TM	$\mathcal{E}$: motion cost $*$: arg min	robot conf., heading
Caforo et al. [3]	Nifti UGV	KM⊕TM	$\mathcal{E}$: traversability cost $*$: arg min	4x flipper angle, heading, velocity
Colas et al. [4]	NiftiBot	KM⊕TM	$\mathcal{E}$: motion cost $*$: arg min	4x flipper angle, heading, velocity
Dornhege and Kleiner [5]	Lurker	KM⊕TM⊕ML	$\mathcal{R}$: behavior maps	heading, skill
Dube [6]	PackBot	KM⊕TM	$\mathcal{E}$:stability index $*$: arg max	manipulator and 2xflipper angle.
Ho et al. [9, 8]	Mawson rover	KM⊕TM⊕ML	$\mathcal{R}$:modeling + GPR	attitude,chassis conf.
Iagnemma and Dubowsky [10]	Wheeled planar rover	KM	$\mathcal{E}$: traction, power consumption, $*$: arg min	wheel velocities
Kolter et al. [12]	LittleDog	KM⊕TM	$\mathcal{E}$: motion cost $*$ arg min	desired contact points
Mathur and Pandian[13]	*targeted to Mars rover*	ML	$\mathcal{E}$: navigability $*$ arg max	heading
Nagatani et al.[16, 15]	Kenaf, Quince	KM⊕TM	$\mathcal{E}$:NESM stability $*$: arg max	4xflipper angle
Ohno et al. [17]	Aladdin	KM	$\mathcal{R}$: control rules	4xflipper angle., 4xflipper ang. vel., moving velocity
Okada et al.[18, 19]	Kenaf, Quince	KM⊕TM	$\mathcal{E}$:NESM stability $*$: arg max	4xflipper angle
Papadakis and Pirri [20]	TALOS	KM⊕TM	$\mathcal{E}$: ground clearance, robot orientation, angle stability $*$: arg max	attitude, 4xflipper angle
Sheh [21]	Emu	ML⊕TM	$\hat{\mathcal{E}}$: probability $p(a\|\mathbf{s})$ $*$: arg max	atomic action
Zimmermann et.al [24]	NIFTi UGV robot	ML⊕TM	$\hat{\mathcal{E}}$: $Q(a,\mathbf{s})$, $*$: arg max	4xflipper angle, 4xflipper stiffness

Table 1: Overview (alphabetical order) of the related work on motion control including our solution of AT. Table corresponds to the motion control design framework depicted in Fig. 2. The system outputs are summarized also.

initial robot posture computation and then, flipper angles are determined so as they touch the ground. The optimal reasoning is based on NESM stability criteria [7], which implies minimization of the risk of rolling over.

Papadakis and Pirri [20] introduced RTI model for quantifying the static 3D traversability cost for tracked robot NIFTi in known terrain map. The motion quality is expressed as combination of intrinsic robot characteristics and articulating capabilities (robot orientation and angle stability), in combination with the terrain surface (ground clearance). The approach is limited by the necessity to have the terrain map in prior which is not often possible due to either limited sensor capabilities or object occlusions. This issue tackled Ho et al. [9] who proposed solution for predicting the complete configuration map from incomplete terrain map for Mawson rover using Gaussian Process Regression (GPR). RTI is modeled by kinematic model in visible areas and then, RTI is interpolated using GPR in occluded parts of the map. Even more not only the rigid terrain assumption, but also with considering deformation caused by robot movement over deformable terrain [8]. Despite of the traversability has been assigned through the estimated robot configuration, it is not further discussed how to turn out complete configuration map into motion control plans. Kolter et al. [12] used experience based database for assigning the terrain characteristics in the missing regions and used complete RTI description (using kin. model and terrain shape) for creating the cost map and appropriate motion plan for the quadruped robot LittleDog whose joints are PD controlled. Limited capabilities of the sensors (not uniform sample distribution) overcame Colas et al. [4] by using distance filter and tensor voting procedure [14] for robust terrain representation that is further used by path planner for NiftiBot. Novel map representation, able to deal with unknown structure (assume earthquake scenario) were introduced by Caforo et al. [3]. The map is build inductively from raw point cloud and at the end provides information about the traversable regions. These are used for generating the motion plan of Nifti UGV robot. Brunner et al. [1], in order to overcome

the exhaustive search in full configuration space of 4 tracked robot, divided terrain map into difficult and easy to traverse regions. Whereas motion planning within easy regions considers partial robot configuration, planning through the difficult areas reflects complete robot configuration already available. RTI and traversability (robot configuration) is simulated (kinematic model-terrain interaction). Dornhege and Kleiner [5] expressed the traversability by concept of Behavior maps where terrain properties are directly mapped to specific robot skill using Fuzzy rules (considering kinematic constraints on robot climbing capabilities) and Markov Random Field classification. The paths found in Behavior maps can be turned into robot motion plans.

Sheh [21] with a four wheel robot Emu used high dimensional terrain description for designing the controller by learning RTI-reasoning based on the decision tree that at the end, decided which one of eight atomic actions a (forward-left-turn, straight- forward-move, etc.) should be used in given high dimensional robot-terrain state $\mathbf{s}$ based on the posterior probability $p(a|\mathbf{s})$. Iagnemma and Dubowsky [10] presented a control methodology based on wheel-ground contact angle (computed from kinematic model and proprioceptive data) for improved traction (on uneven terrain) or reduced power consumption (on relatively flat terrain) for wheeled mobile robot on Mars-like soil. Mathur and Pandian[13] classified terrain using textural analysis of visual imagery into Navigable and Not navigable regions and used them for planning through the assessed terrain.

2.2 Our approach to motion control

Figure 3: NIFTi UGV Robot climbing capability: Articulated flippers and passive differential increase the platform stability on uneven terrain and enlarge its operability space.

We aimed in development of semi-autonomous motion control system for NIFTi UGV skid-steer robot [1] (technical specifications provide Sec. 2.2.2) whose purpose is to control the flipper configuration (angle and stiffness) autonomously while the moving velocity and azimuth will be controlled manually by the operator.

[1] developed within NIFTi/TRADR project—see `http://www.tradr-project.eu`

2.2.1 Methodology and design

Reinforcement learning (RL) technique (survey on RL in robotics provided Kober et al. [11]) was used to learn RTI-reasoning response through the use of collected data during the manual control.

The current RTIs were expressed as scalar rewards combining several terms: (i) too-low pitch and roll reward, (ii) smoothness reward (iii) speed reward (iv) user denoted reward (penalty) indicating the success (failure) of the particular maneuver. The reasoning in RL is based on (discounted) sum of such rewards. We choose to learn action-state value $\hat{\mathcal{E}}(\mathbf{a},\mathbf{s}) = Q(\mathbf{a},\mathbf{s})$ representing sum of discounted rewards when taking action $\mathbf{a}$ at state $\mathbf{s}$ and behaving optimally onwards. Since the exteroceptive part of the state $\mathbf{s}$ includes only locally available terrain information, such concept can support map creation which is crucial for exploration during search and rescue missions—of course precise localization is still required [22].

Since handling all possible Q values covering real high dimensional action-state space is not feasible, we adopted following simplifications: (i) finite number of possible morphological adjustments—each configures angle and stiffness per flipper, and (ii) Q values approximation (least-squares fitted Q-iteration with parametric approximation algorithm [2]) caused by high dimensional state $\mathbf{s}$ representation.

As experimentally shown in the challenging outdoor forest environment (see Fig. 4), proposed concept outperformed the manual control in terms of time and smoothness of traversal, which might be seen as quantitative measure of operator's cognitive load.

2.2.2 Robot Specification

NIFTi UGV robot (see Fig. 1, 3, 4) consists of two main tracks, equipped with passive differential, and four articulated subtracks (so called flippers). Although such construction has many DoF and accepted autonomy has to be considered in real deployment, the design has following foundations: (i) increased platform stability on rough or unstructured terrain and (ii) enlarged operability space (climbing up/down the obstacles)—both situations are depicted in Fig. 3. Platform is equipped with SICK LMS-151 range finder, Ladybug 3 omnicam and Xsens MTi-G IMU.

3 Conclusion

In this paper, we provided survey on motion control approaches with focus on robot-terrain representation which helps to reveal advantages or limitations of individual solutions.

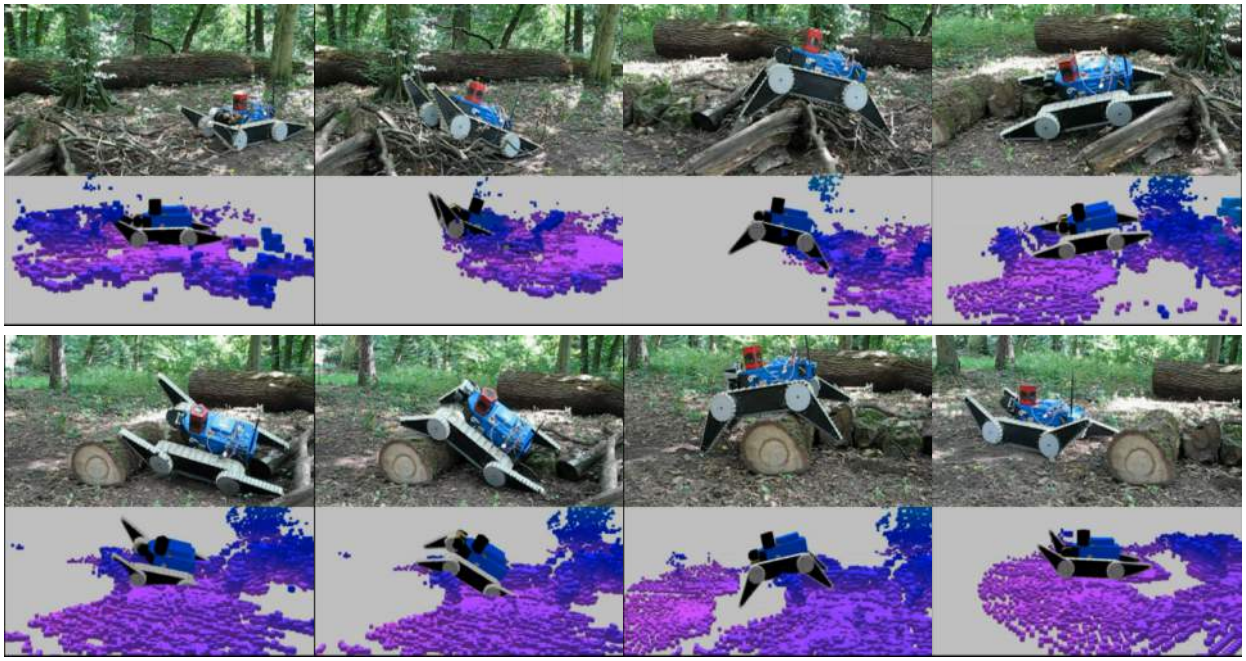


Figure 4: Motion control example in challenging outdoor forest environment: The sequence goes from left to right, from top to bottom. In the upper parts of the images, there are real scenes captured by external camera, in the bottom parts there are visualization of digital map sensed by robot.

Methods based on robot models, often considering only its kinematics, are able to achieve satisfactory good performance using robot-terrain contact properties. Some of them use just proprioceptive data [10, 17], others do combination with exteroceptive data resulting in simple terrain representation [6, 15, 16, 18, 19]. Such autonomy is accepted in an unknown complex terrain or running on platforms suited with poor exteroceptive sensors.

Once the environment is explored, more autonomy is accepted since there is no threat of failure caused by incomplete information and thus motion planning [1, 3, 4, 12, 20] can be executed. Such approaches are more demanding, require reliable sensor suite, enough computation power onboard, they are more complex which might go against its generalization and robustness.

Since there are a lot of difficulties to model, not only in such a complex SaR world, several simplifications are considered. To overcome these imperfection in modeling, machine learning technique are used to replace the description by the knowledge, as for example [5, 8, 9, 13].

Machine learning can also replace classical feedback control loop design and instead of modeling system dynamics by differential equations, which is difficult in case of nonlinear system, learn the mapping between states and actions from the collected training samples [21, 24]. Widely used is Reinforcement learning [23], since it guarantee optimal behavior from any state onwards [11].

4 Acknowledgement

The authors were supported as follows: Petr Zuzanek by the project EU-FP7-ICT-609763 TRADR, Karel Zimmermann by the project SGS13/142/OHK3/2T/13 of Czech Technical University in Prague, Vaclav Hlavac by the project TE01020197 of Technology Agency of the Czech Republic.

References

[1] Michael Brunner, Bernd Bruggemann, and Dirk Schulz. Towards autonomously traversing complex obstacles with mobile robots with adjustable chassis. In *Carpathian Control Conference (ICCC), 2012 13th International*, pages 63 –68, may 2012.

[2] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2010.

[3] Bruno Cafaro, Mario Gianni, Fiora Pirri, Manuel Ruiz, and Arnab Sinha. Terrain traversability in rescue environments. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–8, Oct 2013.

[4] F. Colas, S. Mahesh, F. Pomerleau, Ming Liu, and R. Siegwart. 3d path planning and execution for search and rescue ground robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ*

International Conference on, pages 722–727, Nov 2013.

[5] Christian Dornhege and Alexander Kleiner. Behavior maps for online planning of obstacle negotiation and climbing on rough terrain. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3005 –3011, 29 2007-nov. 2 2007.

[6] Chioniso Dube. Modeling the manipulator and flipper pose effects on tip over stability of a tracked mobile manipulator. In *Robotics and Mechatronics Conference*, 2011.

[7] Elena Garcia, Joaquin Estremera, and Pablo Gonzalez de Santos. A comparative study of stability margins for walking machines. *Robotica*, 20:595–606, 10 2002.

[8] Ken Ho, T. Peynot, and S. Sukkarieh. A near-to-far non-parametric learning approach for estimating traversability in deformable terrain. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2827–2833, Nov 2013.

[9] Ken Ho, Thierry Peynot, and Salah Sukkarich. Traversability estimation for a planetary rover via experimental kernel learning in a gaussian process framework. In *Internation Conference on Robotics and Automation (ICRA)*, 2013.

[10] Karl Iagnemma and Steven Martin Dubowsky. Mobile robot rough-terrain control (rtc) for planetary exploration. In *ASME*, 2000.

[11] Jens Kober, Andrew J. Bagnell, and Jan. Peters. Reinforcement learning in robotics: A survey. 2013.

[12] J.Z. Kolter, Kim Youngjun, and A.Y. Ng. Stereo vision and terrain modeling for quadruped robots. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1557–1564, 2009.

[13] Priyanka Mathur and KK Soundra Pandian. Terrain classification for traversability analysis for autonomous robot navigation in unknown natural terrain. *International Journal of Engineering Science and Technology (IJEST)*, vol. 4:38–49, 2012.

[14] Gérard Medioni, Chi-Keung Tang, and Mi-Suen Lee. Tensor voting: Theory and applications. In *Proceedings of RFIA*, 2000.

[15] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.

[16] Keiji Nagatani, Ayato Yamasaki, Kazuya Yoshida, Tomoaki Yoshida, and Eiji Koyanagi. Semi-autonomous traversal on uneven terrain for a tracked vehicle using autonomous control of active flippers. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2667 –2672, sept. 2008.

[17] Kazunori Ohno, Shouich Morimura, Satoshi Tadokoro, Eiji Koyanagi, and Tomoaki Yoshida. Semi-autonomous control system of rescue crawler robot having flippers for getting over unknown-steps. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3012 –3018, 29 2007-nov. 2 2007.

[18] Yoshito Okada, Keiji Nagatani, and Kazuya Yoshida. Semi-autonomous operation of tracked vehicles on rough terrain using autonomous control of active flippers. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2815–2820, 2009.

[19] Yoshito Okada, Keiji Nagatani, Kazuya Yoshida, Satoshi Tadokoro, Tomoaki Yoshida, and Eiji Koyanagi. Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. *Journal of Field Robotics*, 28(6):875–893, November 2011.

[20] Panagiotis Papadakis and Fiora Pirri. 3d mobility learning and regression of articulated, tracked robotic vehicles by physics-based optimization. In *VRIPHYS'12*, pages 147–156, 2012.

[21] Raymond Sheh, Bernhard Hengst, and Claude Sammut. Behavioural cloning for driving robots over rough terrain. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 732 –737, sept. 2011.

[22] Jakub Simanek, Michal Reinstein, and Vladimir Kubelka. Evaluation of the ekf-based estimation architectures for data fusion in mobile robots. *Mechatronics, IEEE/ASME Transactions on*, PP(99):1–6, 2014.

[23] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[24] Karel Zimmermann, Petr Zuzanek, Michal Reinstein, and Vaclav Hlavac. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In *Internation Conference on Robotics and Automation (ICRA)*, 2014. accepted for publication.

Lecture Notes in Computer Science: Safe Exploration Techniques for Reinforcement Learning – An Overview

Martin Pecka[1⋆], and Tomas Svoboda[1,2⋆⋆]

[1] Center for Machine Perception, Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic, martin.pecka@fel.cvut.cz, http://cmp.felk.cvut.cz/∼peckama2
[2] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic

Abstract. We overview different approaches to safety in (semi)autonomous robotics. Particularly, we focus on how to achieve safe behavior of a robot if it is requested to perform exploration of unknown states. Presented methods are studied from the viewpoint of *reinforcement learning*, a partially-supervised machine learning method. To collect training data for this algorithm, the robot is required to freely explore the state space – which can lead to possibly dangerous situations. The role of *safe exploration* is to provide a framework allowing exploration while preserving safety. The examined methods range from simple algorithms to sophisticated methods based on previous experience or state prediction. Our overview also addresses the issues of how to define safety in the real-world applications (apparently absolute safety is unachievable in the continuous and random real world). In the conclusion we also suggest several ways that are worth researching more thoroughly.

Keywords: Safe exploration, policy search, reinforcement learning

1 Introduction

Reinforcement learning (RL) as a machine learning method has been thoroughly examined since 80's. In 1981, Sutton and Barto [3] inspired themselves in the reinforcement learning discoveries in behavioral psychology and devised the *Temporal Difference* machine learning algorithm that had to simulate psychological classical conditioning. In contrast with *supervised learning*, reinforcement learning does not need a teacher's classification for every sample presented. Instead, it just collects rewards (or punishment) on-the-go and optimizes for the expected

⋆ Supported by the Grant Agency of the Czech Technical University in Prague under Project SGS13/142/OHK3/2T/13
⋆⋆ Supported by the EC project FP7-ICT-609763 TRADR

long-term reward (whereas supervised learning optimizes for the immediate reward). The key advantage is that the design of the rewards is often much simpler and straight-forward than classifying all data samples.

Reinforcement learning proved to be extremely useful in the case of state-space exploration – the long-term reward corresponds to the value of each state [17]. From such values, we can compose a *policy* which tells the agent to always take the action leading to the state with the highest value. As an addition, state values are easily interpretable for humans.

Since the early years, a lot of advanced methods were devised in the area of reinforcement learning. To name one, *Q-learning* [25] is often used in connection with safe exploration. Instead of computing the values of states, it computes the values of state–action pairs, which has some simplifying consequences. For example, Q-learning doesn't need any *transition model* (i.e. dynamics model) of the examined system.

A completely different approach is *policy iteration*. This algorithm starts with a (more or less random) policy and tries to improve it step-by-step [16]. This case is very valuable if there already exists a good policy and we only want to improve it [11].

What do all of these methods have in common, is the need for rather large training data sets. For simulated environments it is usually not a problem. But with real robotic hardware, the collection of training samples is not only lengthy, but also dangerous (be it mechanical wear or other effects). Another common feature of RL algorithms is the need to enter *unknown* states, which is inherently unsafe.

As can be seen from the previous paragraph, *safety* is an important issue connected with reinforcement learning. However, the first articles focused on maintaining safety during exploration started to appear much later after the "discovery" of RL. Among the first, Heger [15] "borrowed" the concept of a *worst-case criterion* from control theory community. In 1994 he created a variant of Q-learning where maximization of long-term reward is replaced with maximization of minimum of the possible rewards. That basically means his algorithm prefers to never encounter a bad state (or, at least to choose the best of the bad states). This approach has one substantial drawback – the resulting policies are far from being optimal in the long-term–reward sense [10].

In this paper we show the various approaches to safe exploration that have emerged so far. We classify the methods by various criteria and suggest suitable use cases for them. To better illustrate some of the practical details, we use the UGV (Unmanned Ground Vehicle) robotic platform from EU FP7 project NIFTi [6] (see Figure 1) as a reference agent. It may happen that in these practical details we assume some advantages of UGVs over UAVs (Unmanned Aerial Vehicles), like the ability to stand still without much effort, but it is mostly easy to convert these assumptions to UAVs, too.

Further organization of this paper is the following: in Section 2 we discuss some basics of reinforcement learning (the reader may skip it if he is familiar with reinforcement learning); Section 3 is an overview of the safety definitions

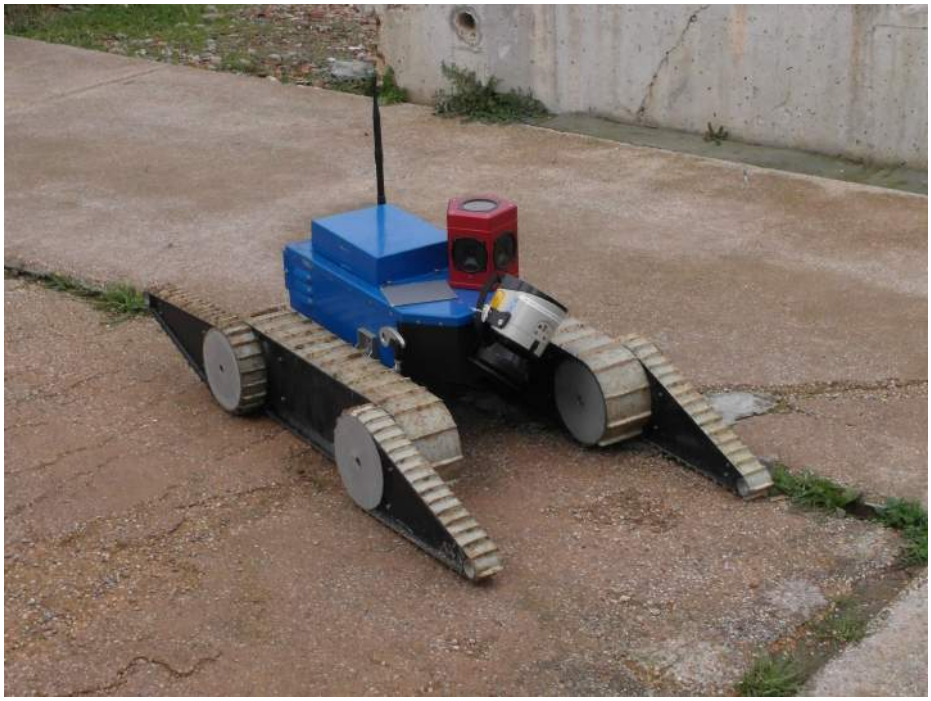


Fig. 1. NIFTi UGV robotic platform

used in literature; Section 4 is the main part concerning the various approaches to safe exploration, and in Section 5 we conclude the findings and we suggest some further areas of possible research.

2 Reinforcement learning basics

2.1 Markov Decision Processes

Markov Decision Processes (MDPs) are the standard model for deliberating about reinforcement learning problems. They provide a lot of simplifications, but are sufficiently robust to describe a large set of real-world problems.

The simplest discrete stochastic MDP comprises of: [17]

- a finite set of states $\mathbf{S}$
- a finite set of actions $\mathbf{A}$
- a stochastic *transition model* $\mathbf{P}$: $\mathbf{P}_t(s, a, s') = Pr(s_{t+1} = s' \mid s_t = s,\ a_t = a)$ for each $s, s' \in \mathbf{S}$, $a\ \in \mathbf{A}$, where Pr stands for probability
- and the *immediate reward function* $\mathbf{R}$: $\mathbf{S} \times \mathbf{A} \to \mathbb{R}$ (or $\mathbf{R}$: $\mathbf{S} \times \mathbf{A} \times \mathbf{S} \to \mathbb{R}$ if the reward depends on the stochastic action result)

To interpret this definition, we say that the at every time instant t the *agent* is in a state s, and by executing action a it gets to a new state s'. Furthermore, executing a particular action in a particular state may bring a *reward* to the agent (defined by $\mathbf{R}$).

The most important and interesting property of MDPs is the *Markov property*. If you have a look at the definition of the transition model, the next state only depends on the current state and the chosen action. Particularly, the next state is independent of all the previous states and actions but the current one. To give an example, the robot's battery level cannot be treated implicitly by counting the elapsed time, but rather it has to be modeled as a part of the robot's state.

Once the model is set up, everything is ready for utilizing an MDP. "The agent's job is to find a policy π mapping states to actions, that maximizes some long-run measure of reinforcement" [17]. The "long-run" may have different meanings, but there are two favorite optimality models: the first one is the *finite horizon* model, where the term $J = \sum_{t=0}^{h} r_t$ is maximized (h is a predefined time horizon and r_t is the reward obtained in time instant t while executing policy π). The dependency of r_t on the policy is no longer obvious from this notation, but this is the convention used in literature when it is clear which policy is used. This model represents the behavior of the robot which only depends on a predefined number of future states and actions.

The other optimality model is called *discounted infinite horizon*, which means we maximize the discounted sum $J = \sum_{t=0}^{\infty} \gamma^t r_t$ with $\gamma \in (0, 1)$ being the *discount factor*. The infinite horizon tries to find a policy that is the best one taking into account the whole future. Please note the hidden dependency on the policy π (and the starting state s_0) – it is the policy that decides on which action to take, which in turn specifies what will the reward be.

Other extensions of MDPs to continuous states, time or actions are beyond the scope of this overview. However, some of the referenced papers make use of these continuous extensions, which proved to be useful for practical applications.

2.2 Value iteration

Value iteration is one of the basic methods for finding the optimal policy. To describe this algorithm, it is first needed to define the essential notion of the *optimal value of a state*. In this whole subsection we suppose the discounted infinite horizon model, but analogous results can be shown for finite horizon, too. "The optimal value of a state is the expected infinite discounted sum of reward that the agent will gain if it starts in that state and executes the optimal policy." [17] Given a policy π, the induced value function is therefore defined as

$$\mathbf{V}_\pi(s) = E\left[\sum_{t=0}^{\infty} r_k \gamma^k\right], \tag{1}$$

where E denotes the expected value and r_k are the rewards for executing policy π. Taking the best value function over all policies then yields the *optimal value*

function $\mathbf{V}^*$: [17]

$$\mathbf{V}^*(s) = \max_{\pi} \mathbf{V}_\pi(s)\,. \tag{2}$$

Inversely, if we have the value function given, we can derive a policy from that. It is a simple policy that always takes the action leading to the most profitable neighbor state (with the highest value).

One useful formulation of the properties of the optimal value function is the formulation using the recurrent *Bellman equations* which define a dynamic system that is stable for the optimal value function. We can say a state's optimal value is the best immediate reward plus its best neighbor's optimal value: [17]

$$\mathbf{V}^*(s) = \max_{a} \left(\mathbf{R}(s,a) + \gamma \sum_{s' \in \mathbf{S}} \mathbf{P}(s,a,s') \mathbf{V}^*(s') \right). \tag{3}$$

Analogously, we can find the optimal policy using the same Bellman equation:

$$\pi^*(s) = \operatorname*{argmax}_{a} \left(\mathbf{R}(s,a) + \gamma \sum_{s' \in \mathbf{S}} \mathbf{P}(s,a,s') \mathbf{V}^*(s') \right). \tag{4}$$

The Value iteration algorithm is based on trying to compute the solution of Equation 4 using iterative Bellman updates (refer to Algorithm 1). In the algorithm, we use a structure called $\mathbf{Q}$ to store the "value" of state-action pairs. In Value iteration it is just a structure to save intermediate results, but it is the core of the Q-learning algorithm (described in Section 2.3). The stopping criterion of the Value iteration algorithm is not obvious, but Williams and Baird [26] derived an easily applicable upper bound on the error of the computed value function.

That said, after a sufficient number of those simple iterations, we can compute the almost optimal value function. The number of iterations needed for Value iteration to converge may be impractically high, but it is shown that the optimal policy converges faster [4], thus making Value iteration practical.

2.3 Q-learning

Just a small change to the Value iteration algorithm results in Q-learning. The basic algorithm is the same as Value iteration, just the update step is done differently (refer to Algorithm 2). The consequence of this change is that no model of the system (transition function $\mathbf{P}$) is needed. It is sufficient to execute all actions in all states equally often, and Watkins [25] proved that if Q-learning were run for an infinite time, the computed $\mathbf{Q}$ would converge to the optimal $\mathbf{Q}^*$ (an analogue of $\mathbf{V}^*$).

2.4 Policy iteration

Policy iteration is a completely different approach to computing the optimal policy. Instead of deriving the policy from the Value or Q function, Policy iteration

Algorithm 1 The Value iteration algorithm [17]

```
Input: an MDP (states S, actions A, rewards R, transition model P)
Output: the optimal value function V*, resp. the optimal policy π*
   derived from the value function

1. V(s) := arbitrary function
2. π := the policy derived from V
3. while π is not good enough do
4.   for all s ∈ S do
5.     for all a ∈ A do
   Update:
6.       Q(s,a) := R(s,a) + γ Σ_{s'∈S} P(s,a,s')V(s')
7.     end for
8.     V(s) := max_a Q(s,a)
9.   end for
10.   π := the policy derived from V
11. end while
12. V* := V, π* := π
```

Algorithm 2 The Q-learning algorithm (only the parts that differ from Value iteration when $\mathbf{V}$ is substituted with $\mathbf{Q}$) [17]

```
Input: an MDP (states S, actions A, rewards R, transition model may be
   unknown)
Output: the optimal state-value function Q*, resp. the optimal policy π*
   derived from the state-value function

6. Q(s,a) := Q(s,a)+
     α[R(s,a) + γ max_{a'} Q(s',a') − Q(s,a)]
8. line left out
```

works directly with policies. In the first step, a random policy is chosen. Then a loop consisting of policy evaluation and policy improvement repeats as long as the policy can be improved [17] (refer to Algorithm 3 for details). Since in every step the policy gets better, and there is a finite number of different policies, it is apparent that the algorithm converges [23].

Policy iteration can be initialized by a known, but suboptimal policy. Such policy can be obtained e.g. by a human operator driving the UGV. If the initial policy is good, Policy iteration has to search much smaller subspace and thus should converge more quickly than with a random initial policy [11].

Algorithm 3 The Policy iteration algorithm [17]

1. $\pi' =$ `arbitrary policy`
2. `repeat`
3. $\pi := \pi'$

   `Policy evaluation: (system of linear equations)`
4. $\mathbf{V}_\pi(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s' \in \mathbf{S}} \mathbf{P}(s, \pi(s), s') \mathbf{V}_\pi(s')$

   `Policy improvement:`

   $\pi'(s) := \operatorname{argmax}_{a \in \mathbf{A}} \Big[$
5. $\quad \mathbf{R}(s, a) + \gamma \sum_{s' \in \mathbf{S}} \mathbf{P}(s, a, s') \mathbf{V}_\pi(s') \Big]$
6. `until` $\pi = \pi'$

3 Defining safety

To examine the problems of safe exploration, it is first needed to define what exactly is the *safety* we want to maintain. Unfortunately, there is no unified definition that would satisfy all use cases; thus, several different approaches are found in the literature. An intuitive (but vague) definition could be e.g.: "State-space exploration is considered *safe* if it doesn't lead the agent to unrecoverable and unwanted states." It is worth noticing here that *unwanted* doesn't necessarily mean low-reward. In the next subsections we present the main interpretations of this vague definition.

3.1 Safety through labeling

The largely most used definition of safety is labeling the states/actions with one of several labels indicating the level of safety in that state/action. What varies from author to author is the number and names of these labels.

To start with, Hans [14] has the most granular division of state/action space. His definitions are as follows (slightly reformulated):

- an (s, a, r, s') tuple (transition) is **fatal** if the reward r is less than a certain threshold (s is the original state, a is an action and s' is the state obtained after executing a in state s, yielding the reward r),
- an action a is **fatal** in state s if there is non-zero probability of leading to a fatal transition,
- state s is called **supercritical** if there exists no policy that would guarantee no fatal transition occurs when the agent starts in state s,
- action a is **supercritical** in state s if it can lead to a supercritical state,
- state s is called **critical** if there is a supercritical or fatal action in that state (and the state itself is not supercritical),
- action a is **critical** in state s if it leads to a critical state (and the action itself is neither supercritical nor fatal in s),
- state s is called **safe** if it is neither critical nor supercritical,
- action a is **safe** in state s if it is neither critical, nor supercritical, nor fatal in state s,
- and finally a policy is **safe** if for all critical states it leads to a safe state in a finite number of non-fatal transitions (and if it only executes safe actions in safe states).

Since we will compare other definitions the the Hans', it is needed to define one more category. A state s is called **fatal** if it is an undesired or unrecoverable state, e.g. if the robot is considered broken in that state. The fatal transition can then be redefined as a transition ending in a fatal state. Opposite to the precisely defined terms in Hans' definition, the meaning of words "undesired" and "unrecoverable" here is vague and strongly task-dependent.

Continuing on, Geibel [12] defines only two categories – *fatal* and *goal* states. "Fatal states are terminal states. This means, that the existence of the agent ends when it reaches a fatal state" [12]. This roughly corresponds to our defined set of **fatal** states. *Goal states* are the rest of final states that correspond to successful termination. Since Geibel only considers terminal states for safety, his *goal* states correspond to a subset of **safe** states. The other Hans' categories need not be represented, since they are meaningless for final states.

An extension of Geibel's *fatal* and *goal* states is a division presented by García [10]. His *error* and *non-error* states correspond to *fatal* and *goal* states, but García adds another division of the space – the *known* and *unknown* states, where *known* states are those already visited (and *known* have empty intersection with *error*). He then mentions a prerequisite on the MDP that if an action leads to a known *error*/*non-error* state, then its slight modification must also lead to an *error*/*non-error* state (a metric over the state space is required).

In Ertle's work [9], again the two basic regions are considered – they are called *desired* and *hazardous* (corresponding to **safe** and **fatal**). However, due to the used learning technique, one more region emerges – the *undesired* region. It contains the whole *hazardous* region and a "small span" comprising of *desired* states, and denotes the set of states where no training (safe) samples are available, because it would be dangerous to acquire those samples. In particular, he says that "The hazards must be 'encircled' by the indications of the undesired approaching so that it becomes clear which area [...] is undesired" [9].

A summary of the labeling-based definitions is shown in Figure 3. We examined the apparent imbalance between the number of categories Hans defines, and the other definitions, and that led us to the following observations.

The first observation is that creating labels for actions or transitions is unnecessary. If we need to talk about the "level of safety" of an action, we can use the worst label out of all possible results of that action (which retains compatibility with Hans' definitions). Moreover, as "it is impossible to completely avoid error states" [22], we can ignore the effects of the action which have only small probability (lower than a safety threshold) – we will call such effects the *negligible effects*.

A second remark is that the **fatal** and **supercritical** sets can be merged. In Hans' work we haven't found any situation where distinguishing between **supercritical** and **fatal** would bring any benefit. Specifically, in his work Hans states that: "Our objective is to never observe supercritical states" [14], which effectively involves avoiding fatal transitions, too. And since we avoid both supercritical and fatal, we can as well avoid their union.

Third, safety of a state does not necessarily depend on the reward for getting to that state. E.g. when the UGV performs a victim detection task, going away from the target area may be perfectly safe, but the reward for such action should be small or even negative.

Putting these observations together, we propose a novelty definition of safety for stochastic MDPs, which is a simplification of Hans' model and a generalization of the other models:

- A state is **unsafe** if it means the agent is damaged/destroyed/stuck... or it is highly probable that it will get to such state regardless of further actions taken.
- A state is **critical** if there is a not negligible action leading to an unsafe state from it.
- A state is **safe** if no available action leads to an unsafe state (however, there may be an action leading to a critical state).

To illustrate the definition on a real example, please refer to Figure 2. In 2(a), the UGV is in a **safe** state, because all actions it can take lead again to safe states (supposing that actions for movement do not move the robot for more than a few centimeters). On the other hand, the robot as depicted in 2(b) is in a **critical** state, because going forward would make the robot fall over and break. If the robot executed action "go forward" once more, it would come to an **unsafe** state. Right after executing the action it would still not be broken; however, it would start falling and that is **unsafe**, because it is not equipped to withstand such fall and therefore it is almost sure it will break when it meets the ground.

3.2 Safety through ergodicity

An MDP is called *ergodic* iff for every state there exists a policy that gets the agent to any other state [20]. In other words, every mistake can be remedied in

(a) A safe state.

(b) A critical state – if the robot went still forward, it would fall down and probably break.

Fig. 2. An illustration of safe and critical states.

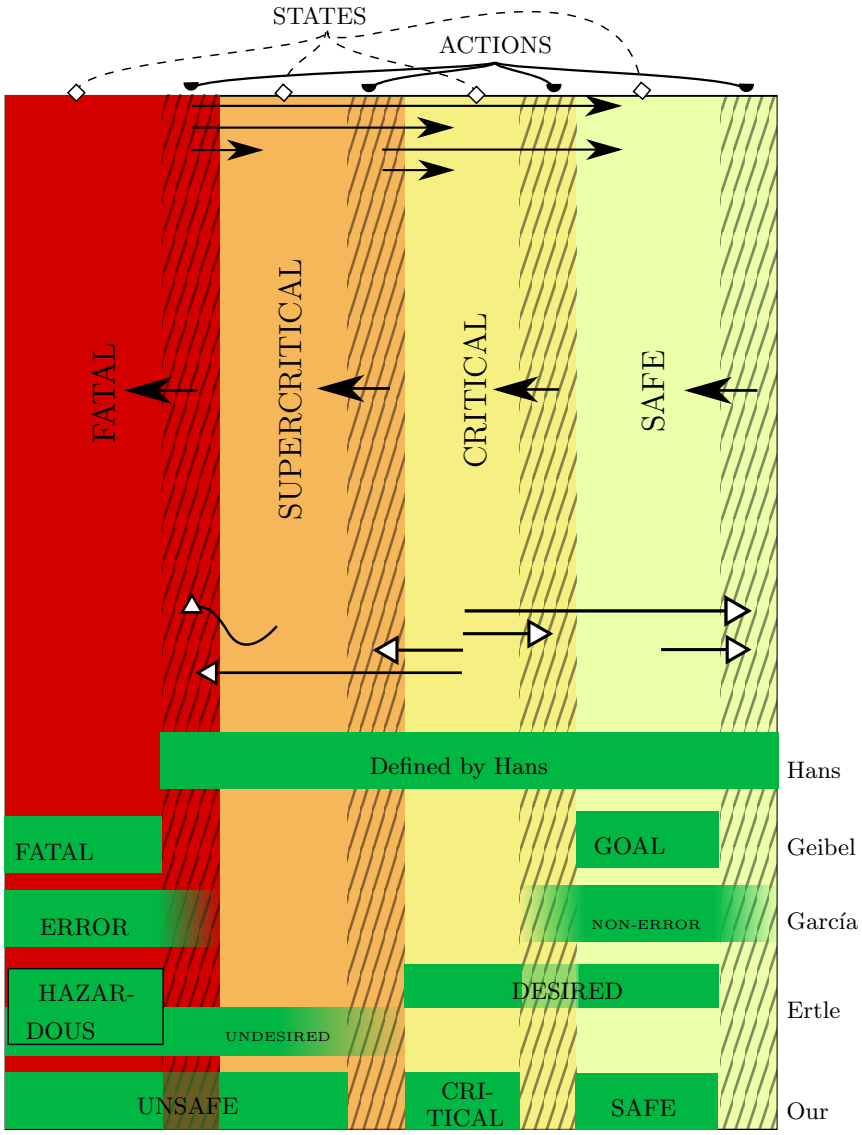


Fig. 3. *A summary of the definitions of safety.* The basic division is taken from Hans [14] and **fatal** states are added. States are drawn with solid background and white-headed arrows (⇾) denote the possible actions in the states. Actions are rendered with striped background and black-headed arrows (➔) end in states where it is possible to end up using the action.

such MDP. Moldovan [20] then defines *δ-safe policies* as policies guaranteeing that from any state the agent can get to the starting state with probability at least δ (using a *return policy*, which is different from the δ-safe one). Stated this way, the safety constraint may seem intractable, or at least impractical – it is even proved, that expressing the set of δ-safe policies is NP-hard [20]. An approximation of the constraint can be expressed in the terms of two other MDP problems which are easily solved [20]; that still leads to δ-safe policies, but the exploration performance may be suboptimal.

In our view, safety through ergodicity imposes too much constraints on the problems the agent can learn. It sometimes happens that a robot has to learn some task after which it is not able to return to the initial state (e.g. drive down a hill it cannot go upwards; a human operator then carries the robot back to the starting position). But the inability to "return home" in no means indicates the robot is in an unsafe state.

3.3 Safety through costs

Another definition of safety is to define a cost for taking an action/being in a state and minimize the worst-case cost of the generated policies (up to some failure probability). Such approach is presented in [15].

However, unless a threshold is set, this definition leads only to the *safest possible* policies, which are not necessarily *safe*. Expressing the safety using costs is natural for some RL tasks (e.g. when learning the function of a dynamic controller of an engine, the engine's temperature can be treated as a cost). Unfortunately, not all **unsafe** states can be described using such costs in general. In addition, specifying the right costs may be a difficult task.

3.4 Safety as variance of the expected return

An alternative to safety as minimization of a cost (either worst-case or expected) is minimizing both the cost and its variance. This approach is called *expected value-variance criterion* [15] and is used mainly in works prior 2000, e.g. [7]. A safe policy by this criterion can be viewed as a policy that minimizes the number of **critical** actions (because fatal transitions are expected to yield much larger costs than safe transitions, increasing the variance significantly).

As stated in [10], the worst-case approach is too restrictive and cautious. The other expected value-variance criteria suffer from the same disadvantages as safety through costs – mainly from the general difficulty to tune up the costs.

4 Safe exploration approaches

Finally, when the theoretical concepts have been shown and the various safety definitions have been presented, we can focus on the main part of this overview. Our categorization of safe exploration techniques is based on the work of García [10]. The basic division is as follows: approaches utilizing the expected return

or its variance (Sec. 4.1), labeling-based approaches (Sec. 4.2) and approaches benefiting from prior knowledge (Sec. 4.3).

4.1 Optimal control approaches

Techniques in this category utilize variations of the *expected value-variance* safety criterion. The most basic one is treating the rewards as costs (when a reward is denoted by r_t, the corresponding cost is denoted by c_t). Standard RL methods can then be used to solve the safe exploration task, as described e.g. in [7] for discounted infinite horizon.

The RL objective function

$$J = E\left[\sum_{t=0}^{\infty} \gamma^t c_t\right] \tag{5}$$

is called the *risk-neutral objective*. To make this objective *risk-sensitive*, we specify a *risk factor* α and rewrite the objective as: [15]

$$\begin{aligned} J = & \quad \tfrac{1}{\alpha} \log E\left[\exp\left(\alpha\gamma^t \textstyle\sum_{t=0}^{\infty} c_t\right)\right] \\ & \simeq E\left[\textstyle\sum_{t=0}^{\infty} \gamma^t c_t\right] + \tfrac{\alpha}{2} Var\left[\textstyle\sum_{t=0}^{\infty} \gamma^t c_t\right], \end{aligned} \tag{6}$$

which is also called the *expected value-variance criterion*. This approach is a part of theory using *exponential utility functions*, which is popular in optimal control [19]. To complete this section, the worst-case objective function (also called the *minimax objective*) is defined as

$$J = \sup\left[\sum_{t=0}^{\infty} \gamma^t c_t\right]. \tag{7}$$

As can be seen, the objective functions containing expectations cannot in fact assure that no unsafe state will be encountered. On the other hand, the minimax objective provides absolute certainty of the safety. However, it may happen that some of the unsafe states can only be reached with a negligible probability. In such cases, the *α-value* criterion defined by [15] can be used – it only takes into account rewards that can be reached with probability greater than α. In the work of Mihatsch [19], a scheme is presented that allows to "interpolate" between risk-neutral and worst-case behavior by changing a single parameter.

Delage's work [8] takes into account the uncertainty of parameters of the MDP. It is often the case that the parameters of the MDP are only estimated from a limited number of samples, causing the parameter uncertainty. He then proposes a possibility that the agent may "invest" some cost to lower the uncertainty in the parameters (by receiving some observations from other sources than exploration). A completely new research area then appears – to decide whether it is more valuable to pay the cost for observations, or to perform exploration by itself.

An approximation scheme for dealing with transition matrix uncertainty is presented in [21]. It considers a robust MDP problem and provides a worst-case, but also robust policy (with respect to the transition matrix uncertainty).

A theory generalizing these approaches can be found in [24]. The theory states that the optimal control decision is based on three terms – the *deterministic*, *cautionary* and *probing* terms.

> The deterministic term assumes the model is perfect and attempts to control for the best performance. Clearly, this may lead to disaster if the model is inaccurate. Adding a cautionary term yields a controller that considers the uncertainty in the model and chooses a control for the best expected performance. Finally, if the system learns while it is operating, there may be some benefit to choosing controls that are suboptimal and/or risky in order to obtain better data for the model and ultimately achieve better long-term performance. The addition of the probing term does this and gives a controller that yields the best long-term performance.[24]

To conclude this section, we think that these methods are not well suited for safe exploration – the expected value-variance and similar criteria provide no warranties on the actual safety. On the other hand, the worst-case approaches seem to be too strict.

4.2 Labeling-based approaches

The approaches utilizing some kind of state/action labeling (refer to Section 3.1 for the various labeling types) usually make use of two basic components – a *risk function* and a *backup policy*. The task of the *safety function* is to estimate the safety of a state or action. In the simplest case, the safety function can just provide the labeling of the given action; or it can return a likelihood that the action is safe; and in the best case, it would answer with a likelihood to be safe plus a variance (certainty) of its answer. The *backup policy* is a policy that is able to lead the agent out of the critical states back to the safe area. It is not obvious how to get such a policy, but the authors show some ways how to get one.

In the work of Hans [14], the most granular labeling is used, where fatal transitions are said to be the transitions with reward less than a given threshold. The safety function is learned during the exploration by collecting the so-called *min-reward samples* – this is the minimum reward ever obtained for executing a particular action in a particular state. The backup policy is then told to either exist naturally (e.g. a known safe, but suboptimal controller), or it can also be learned. To learn the backup policy, an RL task with altered Bellman equations is used:

$$\mathbf{Q}^*_{min}(s,a) = \max_{s'} \min \left[R(s,a,s'), \max_{a'} \mathbf{Q}^*_{min}(s',a') \right].$$

A policy derived from the computed $\mathbf{Q}^*_{min}$ function is then taken as the backup policy (as it maximizes the minimum reward obtained, and the fatal transitions

are defined by low reward). He defines a policy to be *safe*, if it executes only safe actions in safe states and produces non-fatal transitions in critical states. To learn such safe policy, he then suggests a level-based exploration scheme (although he gives no proofs why it should be better than any other exploration scheme). This scheme is based on the idea that it is better to be always near the known safe space when exploring. All unknown actions from one "level" are explored, and their resulting states are queued to the next "level". For exploration of unknown actions he proposes that the action should be considered critical until proved otherwise, so the exploration scheme uses the backup policy after every unknown action execution. A disadvantage of this approach is that the agent needs some kind of "path planning" to be able to get to the queued states and continue exploration from them.

García's PI-SRL algorithm [10] is a way to safeguard the classical policy iteration algorithm. Since the labels *error*/*non-error* are only for final states, the risk function here is extended by a so called *Case-based memory*, which is in short a constant-sized memory for storing the historical (s, a, $\mathbf{V}(s)$) samples and is able to find nearest neighbors for a given query (using e.g. the Euclidean distance). In addition to the *error* and *non-error* states, he adds the definition of *known* and *unknown* states, where *known* states are those that have a neighbor in the case-based memory closer than a threshold. A safe policy is then said to be a policy that always leads to *known non-error* final states. To find such policy, the policy iteration is initialized with the safe backup policy and exploration is done via adding a small amount of Gaussian noise to the actions. This approach is suitable for continuous state- and action-spaces.

Another approach is presented in the work of Geibel [12], where the risk and objective functions are treated separately. So the risk function only classifies the states (again only final states) as either *fatal* or *goal*, and the risk of a policy (risk function) is then computed as the expected risk following the policy (where *fatal* states have risk 1 and *goal* states have risk 0). The task is then said to be to maximize the objective function (e.g. discounted infinite horizon) w.r.t. the condition that the risk of the considered policies is less than a safety threshold. The optimization itself is done using modified Q-learning, and the optimized objective function is a linear combination of the original objective function and the risk function. By changing the weights in the linear combination the algorithm can be controlled to behave more safely or in a more risk-neutral way.

A generalization of Geibel's idea to take the risk and reward functions separately can be found in the work of Kim [18]. In this work, the constrained RL task is treated as a *Constrained MDP* and the algorithm *CBEETLE* for solving the Constrained MDPs is shown. The advantage of this work is that it allows for several independent risk (cost) functions and doesn't need to convert them to the same scale.

A similar approach of using constrained MDP to solve the problem can be found in the work of Moldovan [20]. He does, however, use the ergodicity condition to tell safe and unsafe states apart (that is, safe are only those states from which the agent can get back to the initial state). Moreover, this approach is only

shown to work for toy examples like the grid world with only several thousands of discrete states, which may not be sufficient for real robotics tasks.

The idea of having several risk functions is further developed by Ertle [9]. The agent is told to have several behaviors and a separate safety function is learned for each behavior. This approach allows for modularity and sharing of the learned safety functions among different types of agents. More details on this work will be provided in the next section, because it belongs to learning with teachers.

An approach slightly different from the previously mentioned in this section is using the methods of reachability analysis to solve safe exploration. Gillula in his work [13] defines a set of *keep-out states* (corresponding to **unsafe** in our labeling) and then a set called $Pre(\tau)$ is defined as a set of all states from which it is possible to get to a *keep-out* state in less than τ steps. Reachability analysis is used to compute the $Pre(\tau)$ set. *Safe states* are then all states not in $Pre(\tau)$ for a desired τ. This approach, however, doesn't utilize reinforcement learning, it computes the optimal policy using standard supervised learning methods with one additional constraint – that the system must use safe actions near the $Pre(\tau)$ set. On the other hand, the system is free to use whatever action desired when it is not near $Pre(\tau)$.

As was presented in this section, the labeling-based approaches provide a number of different ways to reach safety in exploration. They are, however, limited in several ways – some of them make use of the (usually hard-to-obtain) transition matrix, the others may need to visit the unsafe states in order to learn how to avoid them, or need the state-space to be metric.

4.3 Approaches benefiting from prior knowledge

The last large group of safe exploration techniques are the ones benefiting from various kinds of prior knowledge (other than the parameters of the MDP). We consider this group the most promising for safe exploration, because "it is impossible to avoid undesirable situations in high-risk environments without a certain amount of prior knowledge about the task"[10].

The first option how to incorporate prior knowledge into exploration is to initialize the search using the prior knowledge. In fact, several works already mentioned in previous sections use prior knowledge – namely the approaches with a backup policy (Hans [14], García [10]). Also, García suggests that the initial estimate of the value function can be done by providing prior knowledge, which results in much faster convergence (since the agent does no more have to explore really random actions, the estimate of the value function already "leads it" the right way) [10].

Another option how to incorporate prior knowledge is by using *Learning from Demonstration* (LfD) methods. Due to the limited space, we will not give the basics of LfD – a good overview of the state-of-the-art methods is for example in [2]. For our overview, it is sufficient to state that LfD methods can derive a policy from a set of demonstrations provided by a teacher. What is important, is that the teacher does not necessarily have to have the same geometrical and

physical properties as the trainee (although it helps the process if possible). It is therefore possible to use LfD to teach a 5-joint arm to play tennis, while using 3-joint human arm as the source of demonstrations (but the learned policy may be suboptimal; RL should then be used to optimize the policy).

In *Apprenticeship Learning* [1], the reward function is learned using LfD. The human pilot flies a helicopter at his best, and both system dynamics and the reward function are learned from the demonstrations. It is however apparent that the performance of the agent is no longer objectively optimal, but that it depends on the abilities of the human pilot.

Another way of incorporating prior knowledge into the learning process is to manually select which demonstrations will be provided, as in the work of Ertle [9]. In the work it is suggested that more teacher demonstrations should come from the areas near the unsafe set, in order to teach the agent precisely where the border between safe and unsafe is located.

The last technique described in our overview is interleaving autonomous exploration with teacher demonstrations. As in the previous case, some teacher demonstrations are provided in advance, and then the exploration part starts utilizing the teacher-provided information. After some time, or in states very different from all other known states, the agent requests the teacher to provide more examples [2,5]. The idea behind this algorithm is that it is impossible to think out in advance what all demonstrations will the agent need in order to learn the optimal policy.

Finishing this section, the algorithms utilizing prior knowledge seem to be the most promising out of all the presented approaches. They provide both a speedup of the learning process (by discarding the low-reward areas) and a reasonable way to specify the safety conditions (via LfD or interleaving).

5 Conclusion

In our work we have given a short introduction on the basics of *Markov Decision Processes* as well as the basic *Reinforcement Learning* methods like *Value Iteration*, *Q-learning* and *Policy Iteration*. In Section 3 we have summarized many recent approaches on how to define *safety* in the framework of optimal control and reinforcement learning. We have also proposed a novelty definition of safety, which divides the state space to *safe*, *critical* and *unsafe* states. We have shown that all other labeling-based safety definitions are covered by our new definition.

In Section 4 many different safe exploration methods are categorized into three basic groups – algorithms from optimal control theory, reinforcement learning algorithms based on state labeling, and algorithms utilizing extra prior knowledge. We have shortly summarized the advantages and disadvantages of the particular approaches. We have also stated that at least for difficult real-world problems, safe exploration without prior knowledge is practically impossible, and prior knowledge almost always helps to achieve faster convergence. Another observation has been that some of the safe exploration algorithms need to visit

unsafe states to correctly classify them later, which might discard them from some usage scenarios where the unsafe states are really fatal.

It seems to us that the field of safe exploration in reinforcement learning has been very fragmented and lacks an all-embracing theory. However, the question is, if it is even possible to find such theory – the main problem may be the fragmentation and differences of various RL methods themselves. At least, the safe exploration community would benefit from a unification of the terminology (and our proposal of the novelty safety labeling would like to help that).

Other ways of possible future research are for example the following. New ways of incorporating prior knowledge into methods not utilizing it yet could bring interesting speed-up of those algorithms. There is also a bottleneck in the estimation of the results of unknown actions – some advanced function approximation methods should be explored (we aim to investigate Gaussian Processes this way). There are not enough experiments from difficult continuous real-world environments, which would show for example how large problems can be solved using safe exploration. The interleaved learning needs some guidelines on how to cluster the queries for the teacher to some larger "packs" and "ask" them together, possibly increasing the fully autonomous operating time. Last, but not least, the possibility to share some learned safety functions among different kinds of robots seems to be an unexplored area with many practical applications (maybe robot-to-robot LfD could be used).

6 Acknowledgments

This work has been supported by the Grant Agency of the Czech Technical University in Prague under Project SGS13/142/OHK3/2T/13, and by EC project FP7-ICT-609763 TRADR. We would like to thank Karel Zimmermann and Michal Reinstein (Center for Machine Perception, CTU in Prague) for their valuable insights into reinforcement learning methods.

References

1. Abbeel, P., Coates, A., Quigley, M., Ng, A.Y.: An application of reinforcement learning to aerobatic helicopter flight. In: Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems 19. vol. 19, p. 1 (2007)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5), 469–483 (May 2009)
3. Barto, A.G., Sutton, R.S., Brouwer, P.S.: Associative search network: A reinforcement learning associative memory. Biological cybernetics (1981)
4. Bertsekas, D.P.: Dynamic programming: deterministic and stochastic models. Prentice-Hall (1987)
5. Chernova, S., Veloso, M.: Confidence-based policy learning from demonstration using Gaussian mixture models. In: AAMAS '07 Proceedings. p. 1. ACM Press (2007)
6. Consortium, N.: NIFTi robotic UGV platform (2010)

7. Coraluppi, S.P., Marcus, S.I.: Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. Automatica (1999)
8. Delage, E., Mannor, S.: Percentile optimization in uncertain Markov decision processes with application to efficient exploration. In: Proceedings of the 24th international conference on Machine learning - ICML '07. pp. 225–232. ACM Press, New York, New York, USA (2007)
9. Ertle, P., Tokic, M., Cubek, R., Voos, H., Soffker, D.: Towards learning of safety knowledge from human demonstrations. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5394–5399. IEEE (Oct 2012)
10. Garcia, J., Fernández, F.: Safe exploration of state and action spaces in reinforcement learning. Journal of Artificial Intelligence Research 45, 515–564 (2012)
11. Garcia Polo, F.J., Rebollo, F.F.: Safe reinforcement learning in high-risk tasks through policy improvement. In: 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). pp. 76–83. IEEE (Apr 2011)
12. Geibel, P.: Reinforcement learning with bounded risk. ICML pp. 162–169 (2001)
13. Gillula, J.H., Tomlin, C.J.: Guaranteed safe online learning of a bounded system. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2979–2984. IEEE (Sep 2011)
14. Hans, A., Schneegaß, D., Schäfer, A., Udluft, S.: Safe exploration for reinforcement learning. In: Proceedings of European Symposium on Artificial Neural Networks. pp. 23–25. No. April (2008)
15. Heger, M.: Consideration of risk in reinforcement learning. In: 11th International Machine Learning Conference (1994)
16. Howard, R.A.: Dynamic Programming and Markov Processes. Technology Press of Massachusetts Institute of Technology (1960)
17. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
18. Kim, D., Kim, K.E., Poupart, P.: Cost-Sensitive Exploration in Bayesian Reinforcement Learning. In: Proceedings of Neural Information Processing Systems (NIPS) (2012)
19. Mihatsch, O., Neuneier, R.: Risk-sensitive reinforcement learning. Machine learning 49(2-3), 267–290 (2002)
20. Moldovan, T.M., Abbeel, P.: Safe Exploration in Markov Decision Processes. In: Proceedings of the 29th International Conference on Machine Learning (May 2012)
21. Nilim, A., El Ghaoui, L.: Robust Control of Markov Decision Processes with Uncertain Transition Matrices. Operations Research 53(5), 780–798 (Oct 2005)
22. Peter Geibel, Wysotzki, F.: Risk-Sensitive Reinforcement Learning Applied to Control under Constraints. Journal Of Artificial Intelligence Research 24, 81–108 (Sep 2011)
23. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)
24. Schneider, J.G.: Exploiting model uncertainty estimates for safe dynamic control learning. Neural Information Processing Systems 9, 1047–1053 (1996)
25. Watkins, C.J., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (May 1992)
26. Williams, R.J., Baird, L.C.: Tight performance bounds on greedy policies based on imperfect value functions. Tech. rep., Northeastern University,College of Computer Science (1993)

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Safe Exploration for Reinforcement Learning in Real Unstructured Environments

Martin Pecka
Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
martin.pecka@fel.cvut.cz

Karel Zimmermann
Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
http://cmp.felk.cvut.cz/~zimmerk/

Tomas Svoboda, Center for Machine Perception
Department of Cybernetics, Faculty of Electrical Engineering
and
Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague

Abstract. *In USAR (Urban Search and Rescue) missions, robots are often required to operate in an unknown environment and with imprecise data coming from their sensors. However, it is highly desired that the robots only act in a safe manner and do not perform actions that could probably make damage to them. To train some tasks with the robot, we utilize reinforcement learning (RL). This machine learning method however requires the robot to perform actions leading to unknown states, which may be dangerous. We develop a framework for training a safety function which constrains possible actions to a subset of really safe actions. Our approach utilizes two basic concepts. First, a "core" of the safety function is given by a* cautious simulator *and possibly also by manually given examples. Second, a classifier training phase is performed (using Neyman-Pearson SVMs), which extends the safety function to the states where the simulator fails to recognize safe states.*

1. Introduction

Many robotic tasks are tackled by RL with iterative state-action space exploration (RC helicopter acrobacy [1], adaptive traversability [17], etc.). RL essentially needs to exhaustively sample the state-action space (which is called "exploration"), and the exploration strategy is represented by a stochastic policy.

While manually-driven exploration is often prohibitively time consuming, autonomous exploration is usually only applied to inherently safe systems (pendulum) or to simulators [16]. We propose a framework for making autonomous exploration safe even for general systems, and we test it on the task of autonomous control of articulated subtracks (flippers) of the USAR mobile robotic platform depicted in Figure 1.

1.1. Task description

The objective of our algorithm is to train a safety function that will allow to select only those exploration policies, that do not lead to unsafe states. Finding an efficient way to optimize the RL objective while using only safe policies is left for an upcoming research (a relevant approach for policy iteration is shown in [14]).

1.2. Contributions

The contributions of our paper can be summed up as follows. We introduce a novel term "cautious simulator" and show it is both simple to construct and useful in machine learning tasks. Next, we present a safe exploration algorithm based on NP-SVM that gradually discovers the safe region without visiting unsafe states or needing too much prior knowledge.

2. Limits of other safe exploration methods

2.1. Visiting of unsafe states

Many proposed safe exploration techniques require that the robot can visit unsafe states in order to get data with "negative" labels. This may be justifiable only if there is a precise model of the world where the dangerous steps can be simulated (e.g. [11, 10]), or if there is an unlimited number of robots to try with. In our work, the robot is never required to visit an unsafe state.

2.2. Coupling safety with rewards

Many safe exploration approaches try to utilize the existing RL methods to achieve safety. This is usually connected with some consequences unacceptable in USAR cases.

Either they claim a state is safe based on the minimum achievable reward – if it is high enough, the state should be safe [12]. This was proved to be highly non-optimal [8]. Or they just set negative rewards to unsafe states and run standard RL (e.g. [5]). However, no guarantees can be given this way, since RL only optimizes the expected outcome.

Tying safety with rewards seems to be unnecessarily constraining. Especially in the field of safe exploration, it does not hold that what is safe is also good from the task point of view (and vice versa, what is good for the task, is not necessarily safe).

We propose to decouple the terms safety and reward completely, as it is done in [9] (where, however, the safety and reward functions get combined together during learning using a weighted sum). In our work, the safety and reward functions are trained as independently as possible.

2.3. Too optimistic expectations

Our last remark is on what can be achieved at best by any safe exploration algorithm. Geibel mentioned that we can never achieve absolute safety [9]. Not only that the safety guarantees can be often only provided as an estimate (which can be erroneous), but we can also "protect" the robot only against some specified classes of risk.

This issue is covered in Ertle's system paper [7], along with the description of methodology and a general view on how the learning algorithms should look like. One implication of his work is that the safety implemented in robots should be *behavior-based* – e.g. each class of risk should include its own safety function and its own policy to avoid the danger. In our experiment, we only concentrate to the "behavior" of climbing down a step.

3. Precise formulation

We optimize a RL task given in the gradient policy search paradigm presented e.g. in [3]. The robot "lives" in a state space $\mathbf{X}$ and performs actions from $\mathbf{A}$ according to a stochastic policy $\pi : \mathbf{X} \to \mathcal{P}(\mathbf{A})$, and is rewarded by a real-number reward function $\mathbf{R} : \mathbf{X} \to \mathbb{R}$. Every policy can be evaluated by the expected performance given by

$$\mathbf{J}(\pi) = \mathbb{E}_{\xi \sim P_\pi} [\mathbf{R}(\xi)]$$

where ξ is a trajectory (sequence of states) created by following the policy from a common start state, and $\mathbf{R}(\xi)$ is a (possibly discounted) reward for the whole trajectory.

To simplify the learning, the policy is often assumed to be from a parametrized class of functions, and the learning is only performed on the parameter values. Thus we can write the policy as $\pi = \pi(\theta)$, and substitute just θ for the policy, yielding

$$\mathbf{J}(\theta) = \mathbb{E}_{\xi \sim P_\theta} [\mathbf{R}(\xi)]$$

Gradient policy search then searches for the policy parameters $\theta^\star$ which maximize the expected performance [2]:

$$\theta^\star = \arg\max_\theta \mathbf{J}(\theta)$$

And this is where safe exploration comes into play: during the gradient search, the examined values of θ are not restricted in any way, so that it can happen that the robot visits an unsafe state. With just a small alteration to the previous equation, we can "plug in" the safety:

$$\theta^\star = \operatorname*{argmax}_{\theta : \mathbf{S}(\theta) \geq s_{\min}} \mathbf{J}(\theta)$$

where $\mathbf{S}$ is the safety function and $s_{\min} \in \langle 0, 1 \rangle$ is a user-defined safety threshold. Finally, we define the state safety function $\mathbf{s} : \mathbf{X} \to \langle 0, 1 \rangle$, from which $\mathbf{S}$ is "composed" as

$$\mathbf{S}(\theta) = \min_{x \sim \pi(\theta)} \mathbf{s}(x)$$

The task is to construct a safety function closest to the real safety margins, and not to visit any unsafe states during the training.

4. Safe exploration system components

The basic background and motivation to our work has been presented, so now we can describe the main components of the algorithm.

A *cautious simulator* is the main component that differentiates our work from other safe exploration concepts. We use the simulator to predict safe states among the set of unvisited states (it may be e.g. a simple physical simulator). *Cautious* means that if the simulator labels a state as safe, it is also safe in the real world. On the other hand, it is allowed to wrongly label safe states (in the real world) as unsafe. Having a cautious simulator is a key to success of our algorithm, and creating such simulator is (much) easier than constructing a plausible physical simulator. Throughout all this work we suppose that running the simulator is (computationally) expensive, so we try to minimize the number of its uses, and we prohibit sampling the whole state space using the simulator.

Next, we need to have an experienced human operator that is capable of executing safe trajectories on the robot in the real world. We suppose that this operator has much more (prior or sensory) information than the robot has, and thus he or she can assess the safety of intended actions before executing them. These safe trajectories will be used to initialize the *safety function*. If we discover an area in the state space that is misclassified by the safety function as *unsafe*, the operator can reach these areas manually, which forces the algorithm to correct the safety estimates for that region.

Combining the simulator and operator results, we can construct the *safety function*. Such function takes the state of the robot (the extracted features), and labels it either safe or unsafe (by returning a number in the $\langle 0, 1\rangle$ interval, where values greater than s_{min} are considered safe). This component is implemented using Neyman-Pearson SVM classifier.

Finally, we need a safe policy extractor, that takes the current estimate of safety function and chooses a policy going only through safe states. Safe policies are then used to automatically gather new data.

The algorithm that combines all these components into a safe exploration scheme is shown in Alg. 1 and described in detail in the next section.

5. USAR safe exploration in detail

In this section we're going to go through the algorithm step-by-step and show what exactly is done in each step. In Table 1 we present the basic definitions used in the algorithm.

Algorithm 1 The safety function training algorithm

1. $\mathbf{X}^{real}$ = operator-generated initial trajectories
2. Update $\mathbf{T}$, $\mathbf{S_0} := \texttt{updateSVM}(\mathbf{T})$
3. $i := 0$
4. **while** learning should continue **do**
5. Generate an optimal policy π_i safe on $\mathbf{S_i}$, or use the operator "as a policy"
6. Drive using π_i, record visited states x_{new}
7. $\mathbf{X}^{real} = \mathbf{X}^{real} \cup x_{new}$
8. Update $\mathbf{T}$, $\mathbf{S'_i} := \texttt{updateSVM}(\mathbf{T})$
9. Perturb x_{new} several times, add the perturbed states to $\mathbf{X}^{sim}_{safe}$ or $\mathbf{X}^{sim}_{unsafe}$ depending on the result of simulation
10. Update $\mathbf{T}$, $\mathbf{S_{i+1}} := \texttt{updateSVM}(\mathbf{T})$
11. $i{+}{+}$
12. **end while**

Variable	Definition
n	The dimensionality of feature space
$\mathbf{X}$	$\mathbb{R}^n$, the feature (state) space
$\mathbf{A}$	$\mathbf{X} \times \mathbf{A} \to \mathcal{P}(\mathbf{X})$, the set of actions
$\mathbf{X}^{real}$	$\subset \mathbf{X}$, already visited states
$\mathbf{X}^{sim}_{safe}$	$\subset \mathbf{X}^{sim}$, states labeled *safe* by Sim
$\mathbf{X}^{sim}_{unsafe}$	$\subset \mathbf{X}^{sim}$, states labeled *unsafe* by Sim
$\mathbf{T}$	$\{\mathbf{X}^{real} \times \{\texttt{safe}\}\} \cup \{\mathbf{X}^{sim}_{safe} \times \{\texttt{safe}\}\} \cup \{\mathbf{X}^{sim}_{unsafe} \times \{\texttt{unsafe}\}\}$, the training set for SVM
Sim	$\mathbf{X} \to \{safe, unsafe\}$, the simulator
π_i	$\mathbf{X} \to \mathcal{P}(\mathbf{A})$, a stochastic safe policy
$\mathbf{S_i}$	$\mathbf{X} \to \{safe, unsafe\}$, a safety function (SVM)

Table 1. Notation used in the algorithm.

5.1. Initialization

On line 1 we first require the operator to generate some real-world trajectories. It is generally not necessary for them to be generated by the operator; they can also be substituted by a first run of the simulator or by prior knowledge (e.g. if a small part of safe states can be analytically expressed). It is important for this initial set to be sufficiently large – if it were not, the initial estimate of the safety function would be very poor. All the generated points are inserted into $\mathbf{X}^{real}$ which is represented either as a set of points, or as a spatial search tree (depending on the expected number of elements). Then we

update the training set $\mathbf{T}$ (according to its definition given in Table 1), and update the SVM model of the safety function ($\mathbf{S}_0$). Description of the SVM update is postponed for Section 5.6.

5.2. The stopping criterion

Line 4) represents the stopping criterion. It can be either a subjective measure (trading off safety function accuracy for time available for experimenting), or a qualitative measure (if the algorithm is no longer able to simulate more unvisited states, or if the safety function hasn't changed for some time).

5.3. Generating an optimal safe policy

On line 5 a policy is generated based on $\mathbf{S_i}$. There are several options on how to do that.

If the task is not only to train safety, but also to optimize a given criterion, it is needed to run a modified Reinforcement learning algorithm that optimizes the expected return subject to all the states selected by the policy are safe. Since computing such optimization problem efficiently is a large problem itself, we only give here a simple (and probably inefficient) way to solve the constrained RL problem. The easy solution is to set rewards for all unsafe states to negative infinity. This will surely find a policy that is safe, however we do not say whether it is optimal or not.

The other option is to randomly generate policies and verify their intersection with the safety function (e.g. by sampling). This is good if we are not interested in learning any specific task, and we just want to explore the state space ("optimal" here means any safe policy).

It can happen that there is no safe state for a particular feature value. Then we need to incorporate this into the policy and allow it to answer that a state is unreachable.

5.4. Policy execution

The step to be done next is to execute the safe policy (line 7 and further). This may need some additional work to be done, such as setting the robot to an initial position, changing the environment and so on. After the policy is executed, the newly visited states are added to $\mathbf{X}^{real}$ and an update of $\mathbf{T}$ and the SVM is run.

5.5. Simulation

The loop starting on line 9 specifies that we sample some perturbed states and simulate them in the simulator. Here is one important point – we assume that the further a perturbed state is from the current (real) state of the robot, the less precise the simulation is. Therefore we always try to perturb only in some small local neighborhood of the current state. How to perturb depends on the type of the features – it can be e.g. by Euclidean vector shifting. The magnitude of the shifts is one of the free parameters of this algorithm.

Once we have the simulations done, we record the simulated states to S^{sim}_{safe} or S^{sim}_{unsafe} depending on the results of the simulations (which are either binary classes or numbers from $\langle 0, 1 \rangle$). Then the training set and SVM are updated again (which is described in the next section).

This simulation and perturbation can also be run just after initialization, before the algorithm enters the learning loop. This way the initial estimate of $\mathbf{S}_0$ will be better.

5.6. Updating the safety function (`updateSVM`)

Representation and modification of the safety function are the key points of our algorithm. We need the safety function to generalize the set $\mathbf{X}^{real} \cup \mathbf{X}^{sim}_{safe}$ in continuous space, not containing any point from $\mathbf{X}^{sim}_{unsafe}$.

From our assumptions it follows that it is not necessary that a generalization over this set also denotes only safe regions (because we defined that safe are only visited states, and states tagged *safe* by Sim). However, if we assume continuousness of the safety function, it can be approximated very well.

To describe the representation of the safety function, we first define an auxiliary set $\mathbf{T}^{pruned} \subset \mathbf{T}$, which is basically the set of all visited or simulated states. To avoid serious problems in computation of the safety function, we need to prune $\mathbf{T}^{pruned}$ in such way, that there are no points from $\mathbf{X}^{sim}$ near to any point from $\mathbf{X}^{real}$. This in fact ensures that visited states have "priority" over states just simulated, which allows us to remedy states misclassified by Sim as *unsafe*, although they are *safe* in reality. Again, the distance function is a free parameter of this algorithm.

Now, $\mathbf{T}^{pruned}$ contains states of which no two cover each other, and are tagged either *safe* or *unsafe*. Finding a representation of $\mathbf{S_i}$ is now a binary classification task. To ensure safety of the estimated safety function, the classification has to be done in such a way that it never classifies an *unsafe* state as *safe*. This can be easily achieved by using the

Neyman-Pearson classification [15] with false negative rate limit set to zero (assuming *negative=safe*).

One of the possible implementations of this classification scheme is using $2\nu-$SVM presented in [6] utilizing LIBSVM [4]. There is a set of kernel functions that can be used with SVMs, and which one to choose again depends on the expected structure of the safety function. Preferring SVMs has one good reason against other binary classification tools – SVMs minimize structural risk (error on test data) rather than minimizing the error on training data. This should provide us with a robustly estimated safety function.

5.7. Remarks

The goal of this algorithm is to find a safety function closest to the real safety margins of the robot. The approximation of the real safety with the safety function should get better as the number of visited states increases, which can be confirmed taking into account how the training set for SVMs is built and how SVMs operate (assuming the kernel function is rich enough to represent the safety function).

Also we can conclude that the number of simulator runs is less than if we sampled the state space regularly, which could be another method of estimating the safety function. Furthermore, our approach has the advantage that it is always sufficient to simulate in local neighborhood of the state the robot is in, allowing for better simulations than if we ran the simulator in distant states.

6. Experiments

6.1. Platform description

To prove this concept of safe exploration we have set up an experiment on a real robot. In the experiment we train a safety function for the task of climbing down steps with various heights. The robot is in front of a terrain step and it receives the "go forward" command. The task is to find the safe flipper angles using which the robot climbs down safely (if it is possible at all).

The robot we used is the Absolem platform from EU FP7 projects NIFTi and TRADR (see Fig. 1). This is an actively articulated tracked platform with size about $60\,\text{cm}\times30\,\text{cm}\times30\,\text{cm}$ and weight 25 kg. The four articulated subtracks (two on each main track) are called *flippers*. The robot can actively control the rotation of each of the flippers (independently).

From the point of view of this experiment, the robot has two important sensors - an IMU (measuring rotation and acceleration), and a laser range finder with broad field of view (270° both horizontally and vertically). There is also a 3D map incrementally built from the laser data, so the robot knows the terrain under itself (which is occluded for the laser).

6.2. Experiment setup

For the experiment we have chosen the task of controlling front flippers when driving down a step (both flippers the same angle). This action is interesting because for different step heights there are different safe flipper configurations, and from a particular height up, there is no safe flipper configuration. The potentially unsafe states cover robot body breakage due to flipping over, gaining too high speed, or touching the terrain with one of its fragile parts (e.g. the laser scanner or camera).

So the state space consists of all possible step heights (also drop heights; measured at the point where the flipper is attached to the main track). The robot generates multiple data when driving down a step – first for height 0, then for the maximum height, and then for all the heights until it finishes climbing down the step (however, we assume only limited sampling capabilities, and this is why the data in Fig. 3 are that sparse). The action space then covers all possible flipper angles the robot can set when climbing down the step. For simplicity, we assume the robot can switch quickly between two different flipper configurations.

The policies are from the deterministic linear policy class of the form $\pi(x) = \theta_0 + \theta_1 x$. The reinforcement learning objective we minimize is $\mathbf{J}(\theta) = \theta_1^2$ (to prefer policies with less flipper motion, e.g. to save power). We seek for a safety function that would discriminate which flipper configurations are safe for which step heights. The safety function is represented by a 2C-SVM (equivalent to 2ν-SVM) with Radial Basis Function kernel.

For executing the simulations, we created a simple model of the robot for use with the Gazebo simulator. Gazebo is a physical simulation library, however our model contains only the basic physical properties. Namely, we have created plausible collision links for the real robot links (simple enough to allow for fast collision checking, though they are still triangle meshes and not primitive objects). For each of the links we have estimated the weight, center of

mass position and inertial properties. Specifically, we have not estimated or set any properties regarding the motors, friction, slippage or other dynamic properties.

Similarly, we put in the simulator a rough terrain representation that is created directly by triangulating the point cloud (either from the laser scanner or from the point map). Such map is in no means smooth, rigid or regular. It contains triangles with wrongly estimated normals or even corner positions and it is non-continuous. Creating a more sophisticated map is an option for improving the estimated safety function, but it is difficult and we want to show that this algorithm works well even with the cluttered map and simplistic robot model. Thus, the task environment can be considered unstructured.

The simulation is then done in the following manner: first we get the triangulated map and place the robot to the position corresponding to the real world. Then we shift it forward 30 cm, set the desired flipper angle and let the robot "fall" on the ground, adjusting the flipper angle according to the given policy. If the flipper policy is safe, then the robot only falls a few millimeters and remains in a stable state, and we can mark all passed state-action pairs as *safe*. The policy is considered unsafe if the robot touches the terrain with its fragile parts, if it turns over or if it ends up too far from the desired $[x, y]$ coordinate – then the simulator tags all the state-action pairs as *unsafe*. For a reference on how the robot looks visualized by Gazebo, refer to Figure 1.

Fortunately, physical simulations in this setting proved to satisfy the requirement on cautiousness of the simulator. To even more ensure cautiousness, we perturb each simulated state several times and return the ratio of safe simulations to all simulations as the final result (thus our simulator returns values from $\langle 0, 1 \rangle$). Here the great advantage of our algorithm showed up – the simple physical model, as well as the triangulated map, are matters of hours to create. If we should create a precise physical model (of both the robot and the terrain), it would still have cases where it fails, and it would have needed much more effort to be done. Moreover, there are properties of the terrain that cannot be modeled in advance, and our perturbation approach could overcome some of them.

It is important to notice that the simulations are performed in a space much larger than the feature space (which is 1-dimensional). The simulations are performed with full 3D models (triangle meshes) incorporating physical influences of forces. So what we do is simulate the problem in its full description, and then map the result of the simulation to the problem projected to a 2D subspace consisting of features and actions. If the projection is chosen wise, there should be no problem with this dimension shrinking.

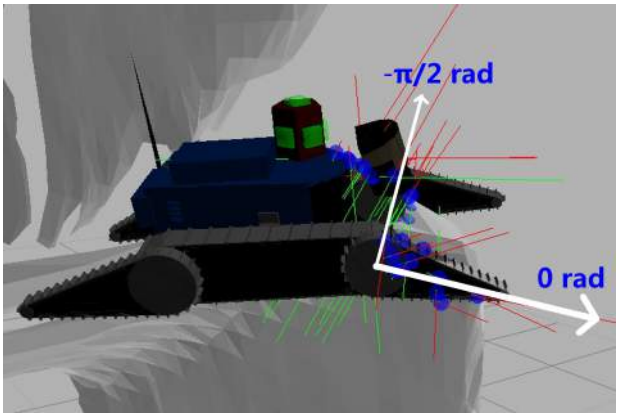


Figure 1. Robot simulation in the Gazebo simulator. Four articulated subtracks (flippers) can be seen in the image – the front ones on the right, and the rear ones on the left. All flippers are in a configuration corresponding to flipper angle 0 rad, and the white arrows symbolize some basic flipper configurations. So lifting up the flippers decreases the flipper angle. In the image there is also shown the triangulated terrain. The red and green segments denote detected robot-terrain collisions.

6.3. Realization of the experiment

To verify the safe exploration algorithm in practice, we drove the robot on several steps of different heights, running the algorithm after each trial. After each teleoperated trial there was an autonomous test of the generated policy. We always chose the policy that intersects the largest area of safe states.

6.4. Evaluation of the experiment

During the realization phase, the robot never tried to enter an unsafe state (both from the estimated unsafe set, and from the real unsafe states). It always managed to add new points to the safety function representation and enlarge the area of state space covered with the safe region. The safe and optimal policy did not change during the experiment, it was always a constant policy $\pi = 1.1 + 0x$.

The progress of the safety function, as well as its support vectors is shown in Fig. 3, note how the safety function's safe area growed gradually with each iteration.

After the final iteration, we compared the learned

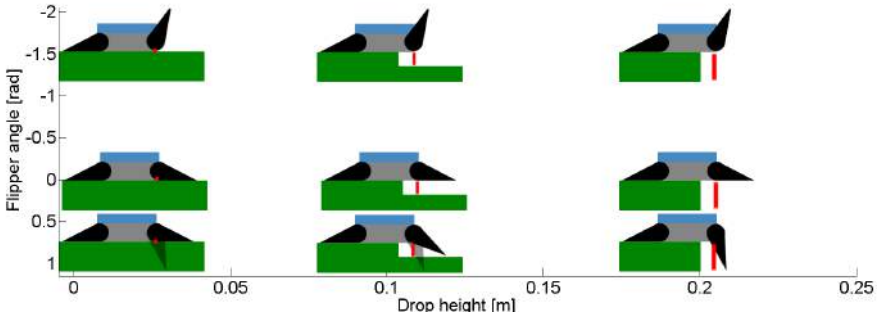


Figure 2. Poses of the robot to better illustrate the meaning of data points in Figure 3. The robot icons are placed approximately with their center on the data point (the left column represents drop height 0). The "ghost" flippers for angle 1 rad denote that the robot pushes to get to that angle, but the applied power is not sufficient (the flippers are compliant). The red bars illustrate the place where the drop height was measured.

safety function to the limits that an experienced operator would allow for the robot. Actually, in more complex instances of safe exploration, getting such limits is impractical. The comparison is shown in Fig. 3, and Fig. 2 provides a graphical understanding for the data points. It is evident from the figure that we have succeeded keeping the false negative (FN) rate at 0 (here FN denotes *unsafe* states classified as *safe*).

Using the classifier terminology, we can specify true negatives (TN) as the number of *safe* states classified *safe*, false positives (FP) the number of *safe* states classified *unsafe*, and true positives (TP) the number of *unsafe* states classified *unsafe*. Then we may define *accuracy* as $(TP + TN)/(TP + TN + FP + FN)$ and *precision* as $TP/(TP + FP)$. With this terms defined, we may say that the objective of the safe exploration algorithm is to achieve precision as close to 1 as possible, which means to minimize the difference between the estimated and real safety functions, while preserving $FN = 0$.

During the three model updates, the values of accuracy in the individual steps were $[0.70, 0.82, 0.81]$, and precision was $[0.42, 0.66, 0.69]$- Another interesting metric can be seen when we superimpose the last (best) SVM model S_2 over the set of visited points in previous model updates. This shows how the model gets gradually better – accuracy $[0.77, 0.82, 0.81]$, precision $[0.48, 0.66, 0.69]$. We note that compared to the first model S_0, the last model S_2 classifies several previously unsafe points as safe, increasing both accuracy and precision. On the second model S_1 there is no change if superimposed with S_2.

7. Conclusion and further work

In our work, we have presented a novel framework for achieving safe exploration in unstructured environments. Compared to other approaches, our method does not need to visit unsafe states, as well as it can guarantee that the robot doesn't visit unsafe states by accident (this holds only for the unsafe states we provide simulators for). It also allows to train the safety function(s) independently from the robot's other tasks, and such safety functions can be easily composed. The trained safety functions are then used to restrict reinforcement learning and other algorithms to only choose safe actions during exploration.

There are two main prerequisites for our safe exploration approach: having a cautious simulator and knowing how to represent the safety function. For the former, we have shown that creating such simple simulator can be easy at least for some problems. The latter can be circumvented by either analysis and modeling of prior knowledge, or by trial-and-error.

This algorithm can be advanced in several ways. Adjusting parameters of the simulator seems to be an interesting way of increasing performance. However, it is not clear how to do some kind of gradient descent with the whole simulator.

If we could safely visit critical states (those near the decision border), that could also help. This can be for example achieved by implementing a cautious exploration strategy (human operators also slow down in dangerous or unknown situations).

Further improvements can be done in the area of selecting which policy to execute. For example, if we could select a policy that would maximize the increase of the safe area, the exploration could be done faster.

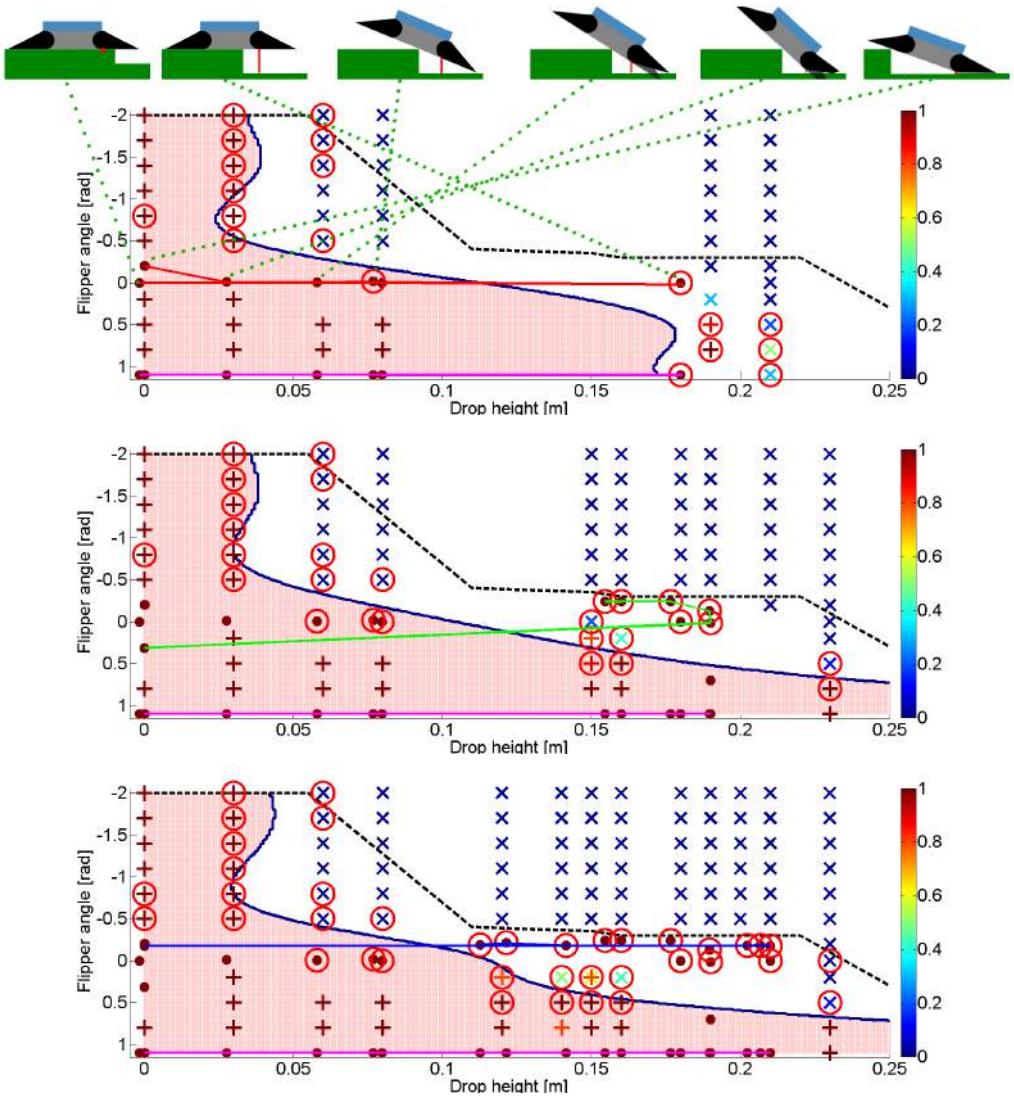


Figure 3. **The progress of learning the SVMs for safety model** (iterations 1, 2 and 3 from the top). The pink area is considered safe by the SVM (the blue solid line is its boundary). The dashed black line denotes the safety boundary estimated by an experienced operator (just for evaluation purposes). Data points from $\mathbf{X}_{real}$ are represented as brown dots, $\mathbf{X}^{sim}_{safe}$ as plus signs and $\mathbf{X}^{sim}_{unsafe}$ as crosses. Safety of $\mathbf{X}_{sim}$ data points is coded by color using the shown color scale (we used safety threshold $s_{min} = 0.7$). Encircled points are the Support Vectors. The thin red, green and blue lines represent the manually driven trajectories, and the magenta line at the bottom is the trajectory executed using π_i. To better understand the visualization of the trajectories, please refer to the robot poses depicted above the first iteration connected by green dotted lines to the corresponding data points (first, the drop height is 0, then it "jumps" to the maximum drop height, and as the robot climbs down, the drop height gets lower and lower). Note that manually visiting the green and azure points in the last step would greatly improve the safety function estimate.

A similar idea is to have an algorithm that would tell the operator which states classified as unsafe by simulator would be worth visiting in the real, if the operator considers them safe. Such approach could both minimize the number of needed human interventions and speed up the exploration process.

Acknowledgments

The 1st author was supported by CTU in Prague under Project SGS13/142/OHK3/2T/13, the 2nd author was supported by The Czech Grant Agency under Project GA14-13876S, the 3rd author was supported by EC project FP7-ICT-609763 TRADR. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

[1] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems 19*, volume 19, page 1, 2007. 1

[2] J. Bagnell. *Learning decisions: Robustness, uncertainty, and approximation*. PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2004. 2

[3] A. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998. 2

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1—-27:27, 2011. 5

[5] S. P. Coraluppi and S. I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 1999. 2

[6] M. Davenport. Controlling false alarms with support vector machines. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages V–V, 2006. 5

[7] P. Ertle, M. Tokic, R. Cubek, H. Voos, and D. Soffker. Towards learning of safety knowledge from human demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5394–5399. IEEE, Oct. 2012. 2

[8] J. Garcia and F. Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012. 2

[9] P. Geibel. Reinforcement learning with bounded risk. In *IEEE International Conference on Machine Learning*, pages 162–169, 2001. 2

[10] J. H. Gillula and C. J. Tomlin. Guaranteed safe online learning of a bounded system. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2979–2984. IEEE, Sept. 2011. 2

[11] A. Hans, D. Schneegaß, A. Schäfer, and S. Udluft. Safe exploration for reinforcement learning. In *Proceedings of European Symposium on Artificial Neural Networks*, number April, pages 23–25, 2008. 2

[12] M. Heger. Consideration of risk in reinforcement learning. In *11th International Machine Learning Conference*, 1994. 2

[13] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

[14] T. M. Moldovan and P. Abbeel. Safe Exploration in Markov Decision Processes. In *Proceedings of the 29th International Conference on Machine Learning*, May 2012. 1

[15] J. Neyman and E. Pearson. *Joint Statistical Papers*. Cambridge University Press, 1967. 5

[16] S. Ross and J. Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012. 1

[17] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac. Adaptive Traversability of Unknown Complex Terrain with Obstacles for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, pages 5177—-5182, 2014. 1

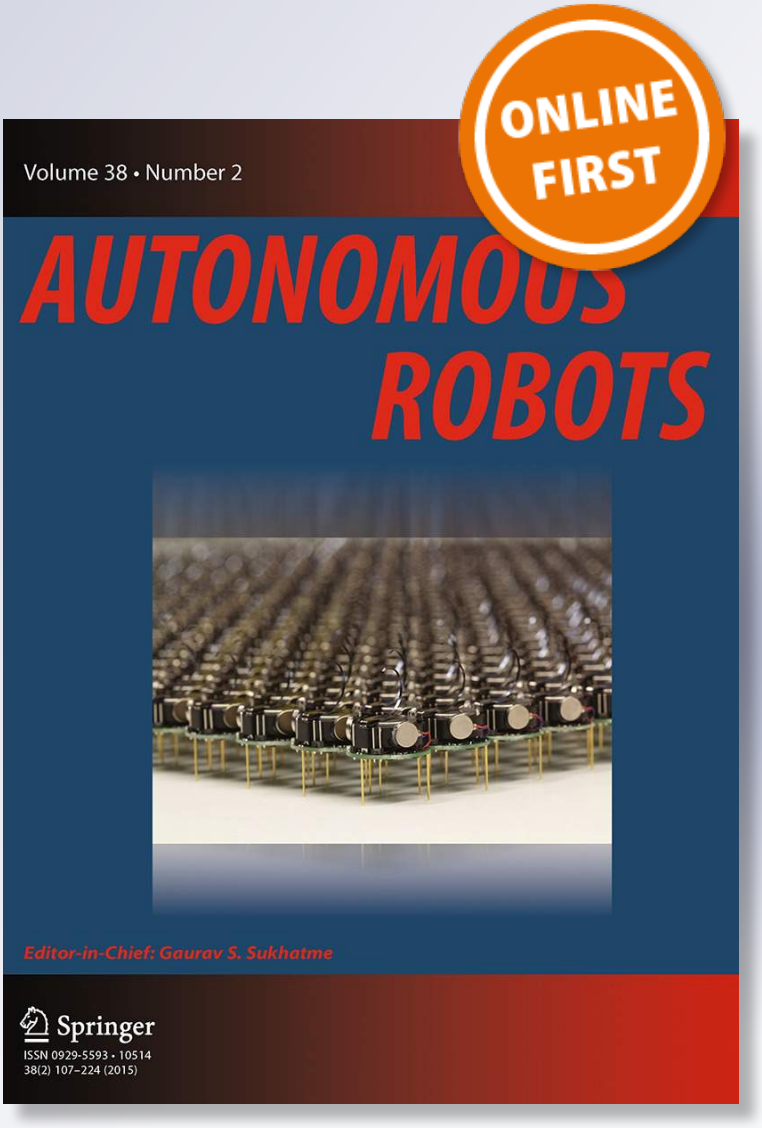
Improving multi-modal data fusion by anomaly detection

Jakub Simanek, Vladimir Kubelka & Michal Reinstein

Autonomous Robots

ISSN 0929-5593

Auton Robot
DOI 10.1007/s10514-015-9431-6



Springer

Auton Robot
DOI 10.1007/s10514-015-9431-6

Improving multi-modal data fusion by anomaly detection

Jakub Simanek · Vladimir Kubelka · Michal Reinstein

Received: 25 June 2014 / Accepted: 5 January 2015

Abstract If we aim for autonomous navigation of a mobile robot, it is crucial and essential to have proper state estimation of its position and orientation. We already designed a multi-modal data fusion algorithm that combines visual, laser-based, inertial, and odometric modalities in order to achieve robust solution to a general localization problem in challenging Urban Search and Rescue environment. Since different sensory modalities are prone to different nature of errors, and their reliability varies vastly as the environment changes dynamically, we investigated further means of improving the localization. The common practice related to the EKF-based solutions such as ours is a standard statistical test of the observations—or of its corresponding filter residuals—performed to reject anomalous data that deteriorate the filter performance. In this paper we show how important it is to treat well visual and laser anomalous residuals, especially in multi-modal data fusion systems where the frequency of incoming observations varies significantly across the modalities. In practice, the most complicated part is to correctly identify the actual anomalies, which are to be rejected, and therefore here lies our major contribution. We go beyond the standard statistical tests by exploring different state-of-the-art machine learning approaches and exploiting our rich dataset that we share with the robotics community. We demonstrate the implications of our research both indoor (with precise reference from a Vicon system) as well as in challenging outdoor environment. In the final, we prove that monitoring the health of the observations in Kalman filtering is something, that is often overlooked, however, it definitively should not be.

Keywords Localization · Kalman filter · Multi-modal data fusion · Anomaly detection · Mobile robots

1 Introduction

In this work, we seek for an improvement in localization of a mobile robot (see Fig. 1) by inspecting measurements provided to a state estimation algorithm. We wish to achieve more accurate and robust estimates of the robot pose (i.e., attitude, velocity, and position). We seek for a classifier that identifies an anomalous measurement and thus supports the pose estimation, which is in our case realized using an extended Kalman filter (EKF) combining proprioceptive (inertial measurements and velocities of the tracks) and exteroceptive (visual and laser measurements) modalities. In the context of state estimation, we define the anomalous measurements as those, which cause significant deviation of the state estimates from the true values. Consequently, anomaly detection is understood as a process that ensures identification and rejection of these defective exteroceptive pose data before they are processed by the estimation framework and spoil the localization. Since we use both statistical and machine learning methods, the key to successful evaluation of our anomaly detection is having rich datasets with precise and accurate external reference for attitude and position, such as captured using the Vicon system (for

Electronic supplementary material The online version of this article (doi:10.1007/s10514-015-9431-6) contains supplementary material, which is available to authorized users.

J. Simanek
Department of Measurement, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 16627 Prague, Czech Republic

V. Kubelka · M. Reinstein ( )
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo namesti 13, 12135 Prague, Czech Republic
e-mail: reinstein.michal@fel.cvut.cz

Published online: 21 January 2015

Fig. 1 The skid-steer mobile robot. See Sect. 3 for details about the sensory set and multi-modal data fusion

this purpose we exploit our publicly available datasets, see Sect. 4.1).

To avoid confusion, we would like to stress out that our multi-modal data fusion utilizes only relative observations (raw pose measurements or increments). Neither proprioceptive nor exteroceptive pose measurements exploit any kind of predefined map, loop closures or other prior knowledge that could correct the pose information before it is processed in the EKF update step. We primarily aim to improve the localization by mitigating the pose drift that accompanies any such data fusion without absolute corrections. Moreover, we consider the proprioceptive sensors—inertial measurement unit (IMU) and track odometry—as the primary sensory set for the mobile robot localization, which was previously proved to localize the robot up to a certain performance limit influenced especially by the yaw angle drift and track slippage (Reinstein et al. 2013; Simanek et al. 2015). To complement the proprioceptive set, we exploit the exteroceptive measurements, which provide corrections leading to more robust estimates. We claim that inspecting the Kalman filter residuals (i.e., difference between predicted and real measurements) generated by the exteroceptive sensory set and rejecting real-world disturbances is particularly essential for long-term navigation and autonomous behavior no matter which platform is used.

During our previous work in urban search and rescue (USAR) scenarios, we encountered frequent abnormal patterns in the visual odometry attitude (VO), and laser range finder attitude and position estimates. These usually occurred as a consequence of unexpected environmental effects or modality failures (e.g., dynamically changing conditions, terrain obstacles, limited range of view, low visibility). For this reason, we were motivated to investigate means of detection and subsequent rejection of such anomalous measurements and we extend our previous work regarding EKF-based mobile robot localization and adaptive odometry (Kubelka et al. 2014; Simanek et al. 2015; Reinstein et al. 2013; Reinstein and Hoffmann 2013; Kruijff et al. 2012). Furthermore, the aspect of inspecting EKF residuals is often overlooked in common practice or a simple threshold based approach is applied. In that case, threshold has to either be manually tuned for all of the EKF states or implemented as a statistical measure (e.g., chi-squared statistics test). In both cases, the detection performance was repeatedly referred as arguable (Ting et al. 2007; Ma et al. 2012).

Our major contribution lies in the in-depth analysis of the detection and resolution of anomalous measurements in EKF-based multi-modal data fusion framework. We propose several possible solutions using supervised machine learning: Gaussian Mixture Models (GMM) modified by a Mahalanobis distance-based decision, one-class Support Vector Machines (SVM), and logistic regression. We compare them to each other as well as to the common practice approaches—covariance and chi-squared threshold tests. We evaluate classifier performance by the means of true positive rate and true negative rate on the indoor datasets labeled with the use of a precise and accurate referential system. Finally we deploy the best classifier in several complex testing environments, such as urban park, disaster training site or dense forest, and evaluate the localization performance.

This paper is structured as follows. Related work is addressed in Sect. 2. Section 3 briefly introduces the data fusion used in our mobile robot and summarizes methods we use for anomaly detection. Section 4 provides details regarding experimental setup used for gathering data, as well as results achieved. The results are concluded in Sect. 5.

2 Related work

In this overview we focus on two areas: mobile robot localization—multi-modal data fusion without any prior knowledge about the surroundings or absolute measurements (e.g., GPS measurements, known map or landmarks, loop closures); and anomaly detection within the context of state estimation.

2.1 Multi-modal localization

Since our mobile robot is a ground vehicle, the most used and basic means of localization is the wheel or track 2D odometry. Odometry usually provides 2D pose estimates without excessive computational load at the cost of unbounded and significant error growth (Endo et al. 2007). Since the standalone odometry drifts rapidly (especially in the case of a skid-steered vehicle) and provides only planar localization, it is often complemented by inertial measurements (Yi et al. 2007), slippage estimation (Endo et al. 2007), or odometry derived constraints (Dissanayake et al. 2001). Generally, it is desired to aid the proprioceptive pose estimates by extero-

ceptive measurements to provide more accurate and reliable results, but still with unbounded error growth.

Visual odometry (Scaramuzza and Fraundorfer 2011) acts as one of the exteroceptive modalities and provides aiding for the IMU and odometry fusion (Shen et al. 2011); however, at the high computational costs that have to be investigated (Rodriguez et al. 2009). Another type of such sensor modality is the laser scanner, which can provide dense 3D point clouds used for mapping, or scan-to-scan translation and rotation estimates allowing robot pose estimation (Pomerleau et al. 2013; Yoshida et al. 2010; Suzuki et al. 2010).

The well established and popular (Görner and Stelzer 2013; Yi et al. 2007; Dissanayake et al. 2001) state estimation algorithm—the Kalman filter—excels in working with a uni-modal distribution parametrized by mean and covariance, providing state as well as its uncertainty estimates (BarShalom et al. 2001, pp. 381–394). Majority of the mobile robotics application use the extended or unscented Kalman filters (UKF), where the UKF compared to EKF does not require the derivation of Jacobians. The computational complexity of the UKF is higher (Chen 2003), which makes it less feasible for running the state estimation in real-time and on-board for large state space. Particle filters are a suitable alternative for localization especially for nonlinear systems with non-Gaussian noise. However, they require even higher computational power than the UKF (Gustafsson et al. 2002).

2.2 Anomaly detection in the context of state estimation

Anomaly detection addresses the task of finding patterns that do not conform to expected behavior (Chandola et al. 2009). There has been a lot of effort put into rejection of such disturbances in a broad range of different applications (e.g., network security, fraud alert, medical domain, etc. Dua and Du 2011).

Within the field of robotics the problem of system failures identification is often referred to as *fault detection and isolation* (FDI) (Pettersson 2005). Fault is defined as an anomaly in behavior of the monitored system and can be detected (indication that something is going as not expected), isolated (locating the faulty component) and identified (determination of magnitude of the fault) (Gertler 1998, p. 3). A reliable robotic system must deal with many uncertainties that can be handled by FDI, e.g., by quality inspection of the provided information (Brunner et al. 2013), comparing information providers (Sundvall and Jensfelt 2006), using information flow between the control and actuation (Christensen et al. 2008), monitoring the reliability of resources (Morales et al. 2008), or proper recognition and modeling of the sensor and mechanical failures (Goel et al. 2000).

Failures in the exteroceptive perception systems are one of many sources of uncertainty in mobile robot localization. In visual odometry or laser point cloud processing, outliers often occur in the frame-to-frame or scan-to-scan motion estimation process. It is crucial to implement outlier removal before the motion estimation step (Howard 2008) and monitor the image (point cloud) quality to prevent perceptual failures, which cause large localization errors (Brunner et al. 2013). An established standard for model estimation in the presence of outliers is represented by the RANSAC based algorithms (Fraundorfer and Scaramuzza 2012; Konolige et al. 2011). Despite the search for outliers is not the concern of our paper, a brief review has to be given to distinguish them from our exteroceptive-based pose anomalies we detect.

The most common fault detection approaches in robotics employ observer-based monitoring of the system and one of the most popular observers is represented by the Kalman filter (Pettersson 2005). Remedies for faulty measurements provided to the Kalman filter may include adjustment of the filter noise matrices (Sarkka and Nummenmaa 2009; Borges and Aldon 2003), smoothing algorithms (Agamennoni et al. 2011) or other modifications. These aim to make the Kalman filter estimates more robust by considering heavy-tailed non-Gaussian distributions, re-sampling or weighting techniques (Ting et al. 2007; Borges and Aldon 2003). Most of these modifications are still dependent on the undesired data detection, noise identification, or require modified versions of the Kalman filtering algorithms. Traditional statistical methods used for Kalman filter monitoring are based on the filter innovations (i.e., residual between the real and estimated measurement) (Hwang et al. 2010; Soule et al. 2005) and assume that the residuals are zero-mean Gaussian processes with given covariance. When the filter is optimal and consistent (BarShalom et al. 2001, p. 232), residuals can be checked if they are generated from a chi-squared distribution (Ali and Ushaq 2009; Caron et al. 2006; Sukkarieh et al. 1999). However, in practice one may struggle to tune the filter properly, since not all the filter assumptions are met, as reported for instance during fusion of VO and IMU attitude (Ma et al. 2012).

There exist machine learning alternatives to the standard inspection of the measurements. These are applied in many areas (Chandola et al. 2009), such as trajectory tracking and monitoring (Laxhammar et al. 2009), or human activity recognition monitored by classifier ensembles (Sagha et al. 2013). The idea of combining anomaly detection and state estimation appeared also in the area of filter self-tuning for automated CPU usage monitoring (Knorn and Leith 2008).

It was pointed out by Ndong and Salamatian (2011) that anomaly detection under the assumption about Kalman filter innovations (i.e., white and Gaussian process) produce large amount of false alarms and that innovations should be rather considered a mixture of Gaussians. In general, data driven anomaly detection exploits both supervised (utilizing data labeled as normal and anomalous) and unsupervised meth-

ods of learning (especially clustering and nearest neighbor methods) (Tsai et al. 2009).

3 Methodology

In the following methodology sections we, first, describe measurements provided by the robot sensory set. Second, we review the data fusion framework and provide a brief summary of the EKF equations and filter tuning (see Kubelka et al. 2014 for further explanation). Third, we present the anomaly detection concept along with both statistical and machine learning methods.

3.1 Proprioceptive measurements

Proprioceptive measurements are in our case provided by inertial measurement unit (IMU) and track odometry.

The IMU uses gyroscopes and accelerometers to provide angular rates and accelerations at 90 Hz in all three dimensions. Processing of the IMU data includes integration of angular rates to obtain attitude information and double integration of accelerations to obtain velocity and position, respectively. Both are known to be exposed to a time related drift, however gyroscope integration holds over longer periods of time and can be aided by accelerometers in the roll and pitch channels to suppress the drift (Kubelka and Reinstein 2012). In contrast, accelerometer integration is always prone to severe drift, therefore usage of the position estimates is restricted to very short time intervals and we primarily correct this drift by incorporating track odometry.

Second type of proprioceptive sensor, incremental rotary encoder, provides us the velocity of right and left tracks at 15 Hz. Since the odometry heading information degrades quickly due to skid-steer character of driving, we incorporate only the speed in the robot forward direction in our model. The slippage still affects both velocity and position, and if not corrected, these quantities are prone to unbounded error growth.

The main disadvantage of combining IMU and odometry lies in unbounded yaw and spatial drifts. Since we do not use any absolute pose measurements, we can only rely on other relative sources of aiding. Because the exteroceptive measurements are usually exposed to different sources of errors, we exploit them to aid the attitude and position estimates.

3.2 Exteroceptive measurements

Exteroceptive measurements are provided by panoramic camera and rotating laser scanner.

Camera images are processed by visual odometry (VO) (Scaramuzza and Fraundorfer 2011; Svoboda et al. 1998; Divis 2013), which provides us relative rotation between two consecutive panoramic images sampled at 2.5 Hz. Implementation uses `OpenCV`[1] Orb keypoint detector, 5-point RANSAC and sliding window bundle adjustment refinement (Kümmerle et al. 2011). In the case of omnidirectional camera, the most needed yaw angle aiding is well conditioned, since the image correspondences are significant in the majority of the omnidirectional image. However, there are many occasions where the VO attitude deteriorates: changing environment, camera occlusion, loss of image correspondences due to low-textured surfaces, insufficient or extensive illumination of the scene, or inadequate VO bandwidth with respect to robot dynamics (e.g., quick turns).

High density 3D point clouds produced by the rotating laser scanner are processed by the Iterative Closest Point (ICP) algorithm (Besl and McKay 1992; Pomerleau et al. 2013) (implementation uses `libpointmatcher`[2]). It provides us relative translation and rotation between two consecutive scans sampled at 0.3 Hz. As in the case of VO, there are situations, where the laser information may fail or deteriorate, especially in harsh environment, where an obstacle may cause the rotating laser to stop, or frequent indoor/outdoor transitions or large moving objects occur. The main problem is the low sampling rate of the laser pose information, which is in our case the only modality that can correct the error accumulated during the track slippage.

3.3 Data fusion scheme

In our multi-modal data fusion (originally published in Kubelka et al. (2014)), we exploit the EKF in feedback form (Farrell 2008, p. 209) to fuse several sensor modalities sampled at significantly different frequencies to estimate robot pose in a local coordinate frame (see Fig. 2, the modalities are highlighted in gray).

The robot is modeled as a rigid body moving through space with no dissipative forces. Following sensors are used in our data fusion: IMU Xsens MTi-G containing low-grade 3-axis accelerometer and 3-axis gyroscope; caterpillar track odometer; omni-directional Point Grey Ladybug3 camera; and continuously rotating SICK LMS-151 laser range finder capable of creating 3D scans of the environment in front of the robot.

3.3.1 Extended Kalman filter

The EKF in feedback form follows the error state implementation inspired by Weiss (2012) where the error state $\Delta\mathbf{x}$ is defined as the difference between true state and state

[1] Open-source computer vision library http://opencv.org/.

[2] Open-source ICP library https://github.com/ethz-asl/libpointmatcher.

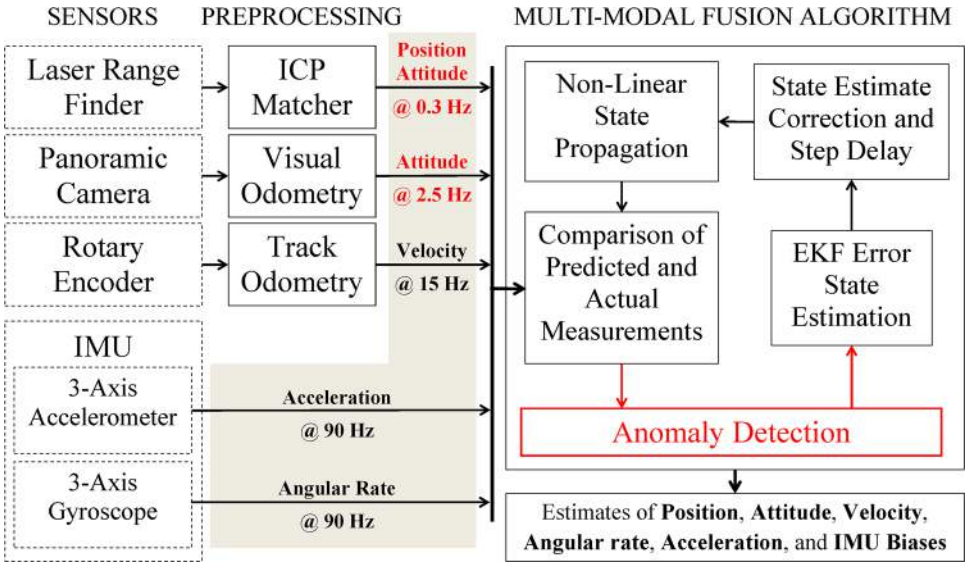


Fig. 2 EKF based multi-modal data fusion algorithm (Kubelka et al. 2014) with integrated anomaly detection (anomaly detection, VO and laser modalities marked in *red*) (Color figure online)

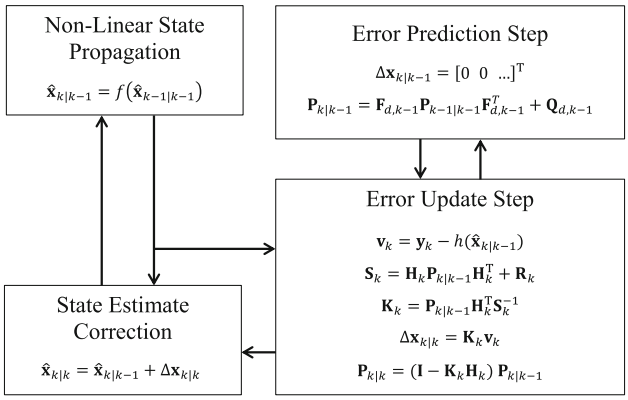


Fig. 3 Error state EKF flowchart (Kubelka et al. 2014)

estimate $\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$. A flowchart summary of the EKF equations is given in Fig. 3 and can be split into following steps:

- *Nonlinear state propagation* The a priori state estimate $\hat{\mathbf{x}}_{k|k-1}$ is obtained using a discretized non-linear system model f and previous state estimate $\hat{\mathbf{x}}_{k-1|k-1}$.
- *Error prediction step* The error state and error state covariance propagation in time are computed as follows: the a priori error state estimate $\Delta\mathbf{x}_{k|k-1}$ is initialized to zeroes and a priori error state covariance $\mathbf{P}_{k|k-1}$ is estimated using previous error state covariance, system matrix $\mathbf{F}_d$ and system noise covariance matrix $\mathbf{Q}_d$ [both discretized by the Van Loan method (Farrell 2008, p. 143)].
- *Error update step* Measurements $\mathbf{y}$ and nonlinear measurement model h (computes predicted measurements based on current a priori state estimates) are used to produce the measurement residuals $\mathbf{v}$ (innovations)[3]. Measurement residuals update the *a posteriori* error state estimate $\Delta\mathbf{x}_{k|k}$ via Kalman gain matrix $\mathbf{K}$. Kalman gain is computed using the residual matrix $\mathbf{S}$, measurement noise covariance matrix $\mathbf{R}$, measurement matrix $\mathbf{H}$, and error state covariance matrix $\mathbf{P}$. Final update step involves updating the error state covariance matrix to the a posteriori form $\mathbf{P}_{k|k}$.
- *State estimate correction* The EKF cycle is completed by correcting the a priori state with the updated error state to obtain a posteriori state $\hat{\mathbf{x}}_{k|k}$.

3.3.2 Filter tuning

The filter tuning (i.e., choosing the system and measurement noise covariances) was carried out in two phases ensuring proper filter performance. First, noise covariances were estimated according to the measured noise under static conditions or noise levels specified in the manufacturer's technical specification. Second, the noise covariances were iteratively tuned until the filter provided the best possible state estimates with respect to the precise and accurate reference obtained by Vicon motion capture system (see Sect. 4.1).

We are aware that the consistency criteria (zero-mean white innovations with given covariance (Simon 2006, pp. 298–301) may not be passed and the estimated covariance may end up too pessimistic or optimistic. However, such use of the referential data in the tuning process is considered a common practice and is an alternative approach to time-consuming manual approach (Lau and Lin 2011). It also ensures the best possible localization performance in the given environment.

[3] We exploit this difference between predicted and actual measurements in the anomaly detection.

3.4 Anomaly detection

3.4.1 *Detection and rejection of anomalous exteroceptive data*

It is highly advised to inspect the sensor data and reject the detected anomalous readings in the Kalman filter framework (Grewal and Andrews 2008, p. 271). The detection of anomalous data was in our case done by both statistical tests and machine learning approaches (see Sect. 3.4.2). All classifiers were deployed for the real-time detection in the EKF framework and when the anomaly was detected, the system ignored the measurement that caused the anomalous behavior.

To detect anomalies by statistical tests in real-time, we employed the information provided solely by the EKF. First, we tracked the error state estimates and looked for values larger than their estimated uncertainty. Second, we applied a statistical measure called chi-squared test, which monitors the innovation information provided by the EKF.

The machine learning approaches using supervised offline learning require the ground truth labels defining whether the exteroceptive measurements (i.e., VO attitude, laser attitude and position) are normal or anomalous. First, we ran the EKF framework with every experiment available in our datasets (see Sect. 4.1 for more details about the datasets). Second, referential pose from the Vicon motion capture system was compared with the EKF state estimates after the particular measurement was used for the EKF update. In other words, we artificially matched the estimate to the true value and tracked the progress of the precision of estimates. Finally, if the state estimate started to diverge rapidly from the true value in a short time interval (varying according to the modality sampling time), the measurement was labeled as anomalous; the other measurements were labeled as normal. Thus we ended up with the normal data within a specific tolerance around the true value during the fixed time interval.

Choosing the right input for the classifiers (i.e., features) is key in supervised learning. As our anomaly detection approach is purely based on EKF residuals, we propose features that consist of residuals in each dimension for particular quantity of interest (VO attitude, laser attitude and position) as well as their norm. Where whole residuals capture more the spatial dependency of the difference between actual and predicted measurements, the norm clearly indicates the distance of measurements from their prediction. For each classifier we performed a straightforward feature selection from the following set: residuals, norm of residuals, both norm and residuals. Best performing subset of the features was selected for the classification task (see Table 2).

Last but not least, there is a necessary prerequisite of having sufficiently large datasets for learning the hypothesis (i.e., the model that ensures detection of the anomaly). In our case, we divided the experiments (see Sect. 4.1 for details) as follows: 2/3 of the experiments were used as a training set (for learning) and 1/3 was used as a testing set (this part of the dataset served also for evaluation of the statistical methods).

3.4.2 *Classification methods*

This section introduces the methods we implemented for anomaly detection.

State estimates and variance are checked by monitoring diagonal terms of the covariance matrix $\mathbf{P}$, which correspond to estimation uncertainty of the particular error states (Farrell 2008, pp. 217–224). Since the covariance matrix, unlike the state estimates, is not dependent on the actual measurement data, the covariance analysis can be used for anomaly detection after measurement update of the filter. If the error state estimate is larger than a certain threshold (typically the $\pm 3\sigma$ boundary), the measurement is classified as anomalous.

Chi-squared gating test (referred also as a *filter consistency check*) is a standard procedure used in combination with the Kalman filter. It utilizes either the normalized estimation error squared (NEES), which however requires knowledge about the true states, or the normalized innovation squared (NIS) (BarShalom et al. 2001, pp. 232–244). In practice, there is a need for real-time consistency check, which is often realized as monitoring of the NIS $\boldsymbol{\epsilon} = \mathbf{v}^T \mathbf{S}^{-1} \mathbf{v}$, where $\mathbf{v}$ is the innovation (residual) vector, and $\mathbf{S}$ is the innovation (residual) covariance matrix. NIS has a χ^2 distribution with l degrees of freedom (i.e., dimension of $\mathbf{v}$) (BarShalom et al. 2001, p. 236) and the test is performed by comparing the NIS to a value from the χ^2 distribution table with given confidence level and degree of freedom.

Gaussian mixture models (GMM) is a probabilistic approach that models the data as a mixture of different Gaussian distributions. There are k Gaussian distributions parametrized by their centers μ, covariance matrices $\boldsymbol{\Sigma}$, and mixture proportions. The solution of finding the mixture of distributions with k components was implemented as maximum-likelihood parameter estimation and found by the Expectation-Maximization algorithm (Mitchell 1997, pp. 191–195).

In our approach, we estimated the GMM (i.e., learned their parameters) on the training data divided into anomalous and normal samples. Both were modeled by two separate mixtures and each mixture was modeled with a number of up to three components. For the classification, we computed the Mahalanobis distance of a incoming residual to every Gaussian in both anomalous and normal GMM. The Mahalanobis distance is defined as $d_M = \sqrt{(\mathbf{v}-\mu)^T \boldsymbol{\Sigma}^{-1} (\mathbf{v}-\mu)}$. The distances were then weighted by the mixture proportions of the GMM components (i.e., their estimated probability), averaged to obtain weighted distance measures (indicating

the distance of new residual to trained anomalous and normal GMM), and the final decision was made by comparing these two distance measures.[4]

One-class SVM algorithm was introduced by Schölkopf et al. (1999) as a support vector algorithm for *novelty detection*. The one-class SVM assumes that the available data are from one class, thus it can learn only on the normal data (majority of the training dataset) and distinguish the anomalies without knowing them. The problem is specified in following way: given a dataset with a probability distribution P, estimate a subset S of the feature space, such that a previously unseen point drawn from P lies outside the subset S with some a priori bounded probability. One-class SVM solves this problem by estimating a function that is positive on S (captures most of the normal data) and negative on the complement (anomalous data). The one-class version of SVM is characterized by the parameter ν which is an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. As the one-class SVM is sensitive to the chosen parameters (Sagha et al. 2013), we also compared the default settings of the selected kernels[5] with the parameters found by a grid search: fraction parameter ν (i.e., $\nu = 2^{-10}, 2^{-9.5}, \ldots, 2^{-1}$; for all approaches); RBF parameter ($\gamma = 2^{-10}, 2^{-9}, \ldots, 2^{0}$) and degree of the polynomial kernel ($d = 0, 1, \ldots, 6$). The grid search used a fivefold cross-validation (according to SVM guide for practice Hsu et al. 2003) and we included the anomalous samples in the validation data to obtain the training error (evaluated in the same way as defined in Sect. 4.1) on a dataset including both classes.[6]

Logistic regression can solve a binary classification problem (Murphy 2012, pp. 245–249), where model parameters are estimated by an iterative search for the minimum of the negative log-likelihood. In our case this was solved by the gradient descent algorithm.[7]

4 Experiments and results

Our experimental evaluation consists of two parts. First, we exploit indoor laboratory environment equipped with the Vicon motion capture system to gather datasets with precise ground truth[8] for the classifier learning and performance evaluation. Second, we show the impact of the best anomaly detection method on the accuracy of actual localization in four selected test cases from both indoor and outdoor unstructured environment.

Fig. 4 The environment simulating USAR conditions (slippery slopes, catwalks etc.)

Table 1 Learning datasets

Modality	Anomalies	Residuals	Anomaly ratio (%)
VO attitude	354	16871	2.1
Laser attitude	91	3477	2.6
Laser position	23	3477	0.7

Number of all anomalous vectors, all residual vectors, and anomaly ratio

4.1 Classification performance

For learning of the classifiers and evaluation of their classification performance, we tried to simulate USAR environment, including ramps, boxes, catwalks, small passages, etc. (see Fig. 4 that shows part of the setup). We recorded approximately 1.7 km of indoor data with ground truth[9]; 20 runs represent standard conditions (590 m in total), 25 runs represent failure cases of visual and laser modalities (e.g., partially blocked field of view, moving object in the field of view; 1120 m in total). The number of all learning data for VO and laser modalities is shown in Table 1, where the anomaly ratio is the ratio of anomalies to the normal data.

4.1.1 Classification performance metrics

As our approach is a binary classification problem, we evaluated the performance of all classifiers by means of a standard

[4] We used the *gmdistribution class* from Matlab Statistics toolbox and fitted the GMM for number of components equal to $k = 1, 2, 3$ with the standard parameters. Maximum of 3 mixtures was selected, because models with more than 3 mixtures usually resulted in negligible weights for some redundant mixtures.

[5] We took linear, polynomial and radial basis function (RBF) kernels in consideration.

[6] We used the LIBSVM tool (version 3.17) (Chang and Lin 2011).

[7] We used the *fminunc* from Matlab Optimization toolbox for the minimization.

[8] Data collected in a room monitored with twelve cameras covering more than 20 m^2 and giving a few millimeter accuracy at 100 Hz.

[9] The datasets are publicly available at https://sites.google.com/site/kubelvla/public-datasets.

Table 2 Classification results as obtained on the testing examples

	Features	TP	TN	FP	FN	TPR	TNR	G
VO attitude								
COV	$\Delta\mathbf{x}_{\mathtt{att}}, \boldsymbol{\sigma}_{\mathtt{att}}$	157	6773	971	12	0.93	0.87	0.901
CHI	$\mathbf{v}_{\mathtt{att}}, \mathbf{S}_{\mathtt{att}}$	130	6867	877	39	0.77	0.89	0.826
GMM	$\|\mathbf{v}_{\mathtt{att}}\|$	160	7702	42	9	0.95	0.99	**0.970**
SVM	$\mathbf{v}_{\mathtt{att}}$	160	7701	43	9	0.95	0.99	0.970
REG	$\|\mathbf{v}_{\mathtt{att}}\|$	47	7743	1	122	0.27	0.99	0.527
Laser attitude								
COV	$\Delta\mathbf{x}_{\mathtt{att}}, \boldsymbol{\sigma}_{\mathtt{att}}$	26	1170	407	1	0.96	0.74	0.845
CHI	$\mathbf{v}_{\mathtt{att}}, \mathbf{S}_{\mathtt{att}}$	14	1429	148	13	0.52	0.91	0.685
GMM	$\|\mathbf{v}_{\mathtt{att}}\|$	25	1499	78	2	0.93	0.95	0.938
SVM	$\mathbf{v}_{\mathtt{att}}$	26	1473	104	1	0.96	0.93	**0.948**
REG	$\|\mathbf{v}_{\mathtt{att}}\|$	4	1574	3	23	0.15	0.99	0.385
Laser position								
COV	$\Delta\mathbf{x}_{\mathtt{pos}}, \boldsymbol{\sigma}_{\mathtt{pos}}$	0	1596	0	8	0.00	1.00	0.000
CHI	$\mathbf{v}_{\mathtt{pos}}, \mathbf{S}_{\mathtt{pos}}$	0	1596	0	8	0.00	1.00	0.000
GMM	$\|\mathbf{v}_{\mathtt{pos}}\|$	8	1574	22	0	1.00	0.99	**0.993**
SVM	$\mathbf{v}_{\mathtt{pos}}$	8	1488	108	0	1.00	0.93	0.965
REG	$\|\mathbf{v}_{\mathtt{pos}}\|$	1	1595	1	7	0.12	0.99	0.353

Bold values indicate best result for each modality
$\Delta\mathbf{x}$ EKF error state vector, $\mathbf{v}$ EKF residual vector, $\mathbf{S}$ EKF residual covariance matrix, $\boldsymbol{\sigma}$ EKF standard deviation, $\|\mathbf{x}\|$ Euclidean norm of a vector $\mathbf{x}$, subscripts $\mathtt{att}$ and $\mathtt{pos}$ attitude and position elements in a vector or matrix, *COV* covariance analysis, *CHI* chi-squared test, *GMM* Gaussian mixture models with 2 mixtures (best of 1, 2 and 3 mixtures), *SVM* one-class SVM with RBF kernel (best of linear, polynomial and RBF kernels; learned using grid search), *REG* logistic regression

two-by-two confusion matrix[10]. The commonly used classification metrics (i.e., accuracy, error rate) are however not suitable for evaluating our imbalanced dataset (see the anomaly ratio in Table 1). We aimed for maximizing true positive rate $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, as well as true negative rate $\mathrm{TNR} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP})$. We used these two classification metrics to maximize the percentage of positive and negative examples correctly classified at the same time in the form of a geometric mean measure $\mathrm{G} = \sqrt{\mathrm{TPR} \cdot \mathrm{TNR}}$ (Kubat et al. 1997).

4.1.2 Classification results

Table 2 shows performance of all classifiers over the following modalities: VO attitude, laser attitude and position. The results were evaluated on the testing dataset, which was randomly selected from the whole dataset, thus the results show generalization of the methods to new data.

Performance of the statistical methods (covariance analysis and chi-squared test) exhibit relatively good results in the VO attitude modality, however with a high number of FP. This behavior is not such an issue, since the roll and pitch angles are primarily determined using the IMU measurements at high rate. The high number of FP causes covariance analysis to have the lowest TPR of all classifiers in the attitude modalities. Thus, it tends to incorrectly reject the normal data—the thresholds, defined implicitly by the EKF tuning of process and measurement noise covariances, seem to be set too low. However, as empirically verified on all the testing data, this behavior cannot be fixed by simply increasing the threshold, because then the TNR is improved at the expense of TPR. The lower TPR in the chi-squared test indicates that this classifier can miss real anomalies and let them propagate. This can be harmful especially in the yaw angle estimation, where no absolute corrections are available. In general, statistical methods perform poorly in the case of laser modalities and even fail in the laser position modality. The main reason behind it is the nature of the laser position measurements, which aim to correct the large error accumulated in track odometry due to slipping. When the laser measurement is processed, large values are introduced in the position residuals, thus any threshold-based method will classify such correction as anomalous data.

Two of the machine learning methods—GMM (learned with 2 mixtures) and one-class SVM (OCSVM, learned with RBF kernel and parameters determined via grid search)—outperform the other classifiers in all modalities tested. On top of that, the number of FP was lower in case of the GMM in the laser modalities. Low number of FP is in general more significant for the laser than visual modality, mainly due to the low sampling frequency of the laser measurements.

Logistic regression classifier does not achieve such performance, because there is only about 2 % of anomalies present in the datasets and regression should have reasonably balanced classes for proper learning. However, we wanted to

[10] True Positive (TP)—anomaly correctly classified as anomaly; False Negative (FN)—anomaly incorrectly classified as normal; False Positive (FP)—normal data incorrectly classified as anomaly; True Negative (TN)—normal data correctly classified as normal.

include these results to show, how the straightforward and popular learning approach will perform.

4.1.3 Discussion on classification performance

We can conclude that both the covariance analysis and the chi-squared threshold test were inferior to machine learning techniques we proposed and both are implicitly sensitive to the filter tuning. However, they were not so demanding when searching for the best model. The only parameter tuned was the sigma bound in case of covariance analysis. Also, there is actually a drawback when implementing the covariance test—it requires a posteriori error state estimate and hence the EKF measurement update needs to be recomputed when anomaly is detected. Since we deal with a multi-modal data fusion, the pose is usually influenced by other modalities and therefore we cannot simply skip the measurement step.

Both the GMM and the SVM methods performed in a very similar way and generalized for the indoor dataset (the performance on training and testing datasets was very similar). However, the number and range of parameters that needs to be tuned is larger in case of the SVM (i.e., bounding parameter ν and different kernels with associated parameters) than in case of the GMM, where we consider only the number of Gaussian mixtures. Thus the implementation of the GMM was more straightforward and computationally less demanding.

Finally, we would like to comment on the importance of fault isolation and identification—the two stages subsequent to detection. In our case, it is not necessary and often even not possible to identify the exact cause of the anomalous exteroceptive measurements since the measurements are already processed by the VO or the ICP algorithms. Therefore in general, inspection needs to be carried out at the lower level, where the images or point clouds are processed. Although isolation and identification may help the rescuers and operators to understand the situation, the pose estimates will benefit mainly from the detection part of the FDI process.

4.2 Localization performance

In this section, we demonstrate the performance in localization when solving the anomaly detection using the GMM (with 2 mixtures modified by a Mahalanobis distance based decision). The GMM was selected as best choice based on the results in Table 2 and advantages mentioned in Sect. 4.1.3. We evaluate the localization in four different scenarios that are most likely to appear in USAR missions. Three of the test cases take place in outdoor environment and are completely new and unseen in the learning process. Section 4.2.1 introduces the scenario of the robot operating in semi-structured indoor environment. Section 4.2.2 presents four experiments from an outdoor urban environment. Section 4.2.3 presents three experiments from a disaster training site with robot inspecting ruins of a building. Section 4.2.4 demonstrates the performance during an outdoor experiment carried out in a challenging forest environment. Table 3 summarizes the total distance traveled (1241 m in total), experiment duration, and maximum elevation difference for all four test cases.

Table 3 Summary of the test case experiments

	Test	Test[a]	Test[b]	Test
	Case 1	Case 2	Case 3	Case 4
Dist. traveled (m)	32	327	332	550
Exp. duration (min)	3	17	37	37
Elev. difference (m)	0	12	1	8

[a] Sum of 4 experiments
[b] Sum of 3 experiments

4.2.1 Test case 1: indoor with Vicon reference

The selected indoor experiment is a representative from the set of the learning experiments in semi-structured environment (see Fig. 4). We show the results with respect to ground truth measurements in both trajectory and attitude. The robot was exploring the room in an approximately square-shaped path, however the VO experienced failures due to lack of tracked features. As shown in red in Fig. 5, sudden loss and consequent lack of tracked visual features resulted in wrong VO attitude measurements (represented by roll, pitch, and yaw Euler angles). The data fusion framework with GMM as anomaly detection successfully detected and rejected the anomalous VO measurements (marked as gray vertical lines). This prevented deterioration in the state estimates and ensured improvement in the final position error normalized by the distance traveled (i.e., 6.2% for data fusion without anomaly detection and 0.8% for data fusion with anomaly detection). This metric is defined as $e_{final} = ||\mathbf{p}_l - \mathbf{p}_{ref,l}||/d$, where l denotes the last position sample and d is the distance traveled in meters. Accuracy improvement is apparent also in the average position error metric defined as $e_{avg}(k) = \sum_{i=1}^{k} ||\mathbf{p}_i - \mathbf{p}_{ref,i}||/k$, where $1 \leq k \leq$ *total number of samples*. Final e_{avg} for data fusion without anomaly detection is more than 3× larger than for data fusion with anomaly detection, see Fig. 5.

4.2.2 Test case 2: urban environment

Second test case includes outdoor experiments from an urban environment, namely one street experiment in a urban canyon and three experiments from a city park including dense vegetation[11], see Fig. 6. The ground truth position of the robot

[11] These experiments are publicly available as well at https://sites.google.com/site/kubelvla/public-datasets.

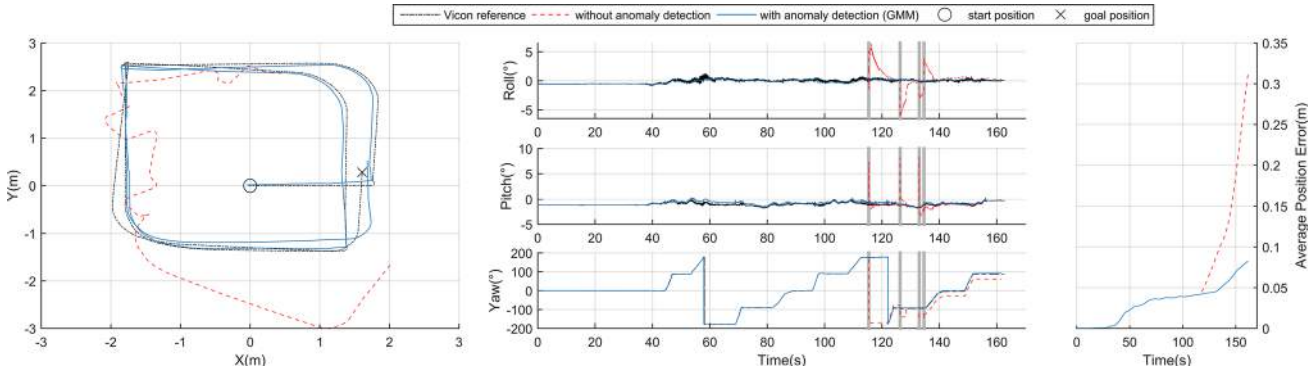


Fig. 5 Indoor experiment—VO attitude failure between 115 s and 140 s (*black dash-dot* Vicon reference; *red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection; *gray vertical lines* rejected VO attitude measurements). (*Left*) Shows 2D projection of the trajectories. (*Middle*) Euler angles. (*Right*) Average position error expressed as function of time (Color figure online)

Fig. 6 Pictures from the urban experiments. (*Left* to *Right*) Detail of the Leica reference system. Street environment. Park environment. Detail of a staircase in the park

in these experiments was tracked by an external reference system Total Station TS15 from Leica Geosystems (further refered to as Leica reference)[12]. Unfortunately, the Leica theodolite does not provide the orientation of the robot and the position ground truth is available only when the robot is in the direct line-of-sight with the theodolite. However, it still ensure precise position measurements even in urban areas, which are often not suitable for reliable GPS ground truth measurements due to limited sky view and multipath effects.

As in the previous case, we show the results from all urban experiments as a projection of the trajectory to the 2D plane accompanied by the average position error plots. The depicted urban experiments are as follows: a rectangular trajectory on the street, bigger and smaller lap in the park including stairs, and straight and slightly leveled path in the park. As shown in Fig. 7, there are improvements in terms of position error in all experiments except one, which did not include any severe anomalies and the position error is comparable with the standard results. In terms of final position error normalized by the distance traveled the results are the following: 5.9, 6.1, 0.9 and 17.9 % for standard data fusion, and 0.2, 1.9, 1.5 and 0.5 % for data fusion with anomaly detection.

[12] The distance measurements are taken in continuous mode at 7.5 Hz with measurement accuracy about 3 mm.

4.2.3 Test case 3: disaster training site

The third test case demonstrates the anomaly detection during USAR experiments carried out in a scenario where the robot was deployed to a training disaster site of fire and rescue service in Prato, Italy (see Figs. 8 and 11). The robot was teleoperated to assist firefighters in a training mission, where the goal was to find victims in areas inaccessible or too dangerous for people. Because the ground truth for these experiments was not available, we show particular instances where the applied anomaly detection contributed to the overall localization accuracy and robustness of the estimates. Namely, this test case includes two instances of disaster site exploration and one instance of robot inspecting an underground pipeline.

The state estimation results from the site exploration are shown in Figs. 9 and 10. The robot was teleoperated along the training site and returned back to initial position. There is a visible improvement in both trajectories when using the anomaly detection. The final position error normalized by

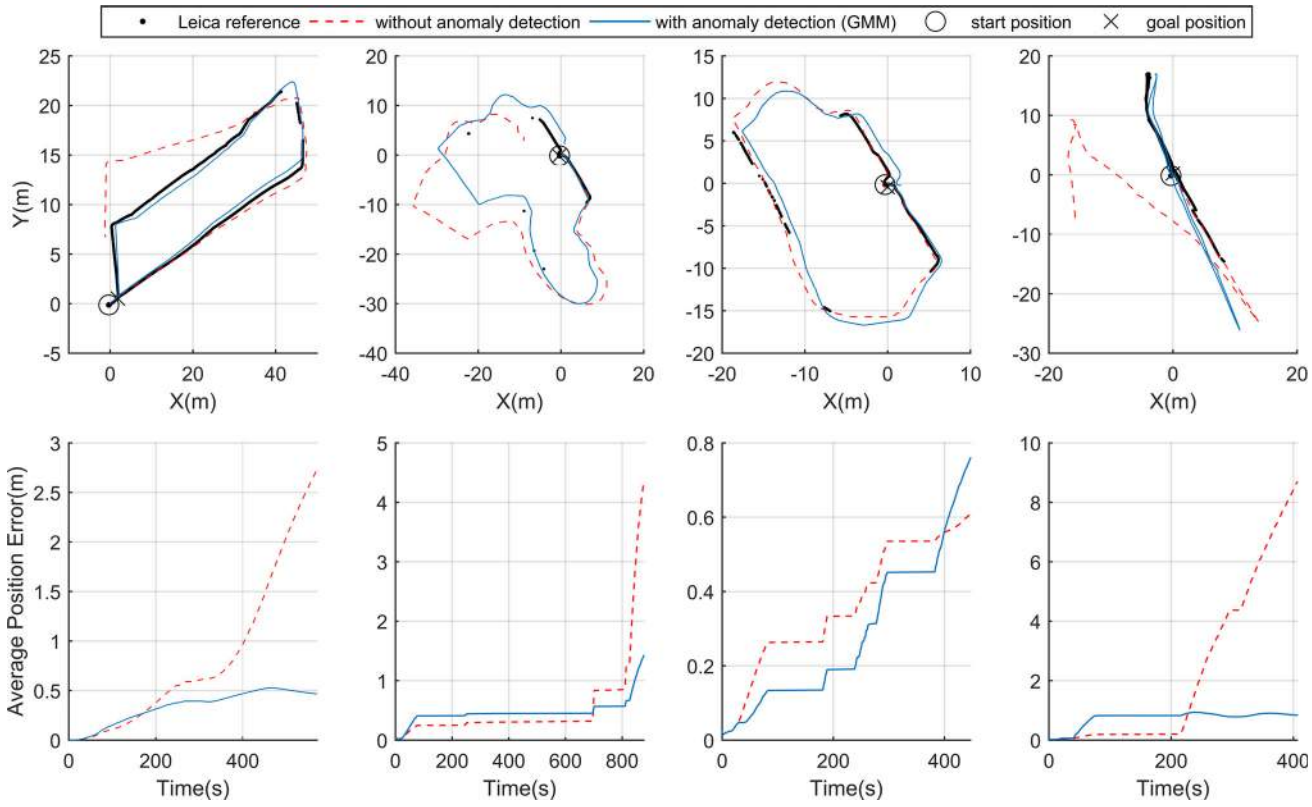


Fig. 7 Urban environment (*black dotted* Leica reference; *red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection). (*Top*) 2D projections of the trajectories. (*Bottom*) Average position errors expressed as function of time (Color figure online)

Fig. 8 Pictures from disaster training site experiment

the distance driven is for both the training site experiments as follows: 16.0 and 9.5 % for standard data fusion, and 1.2 and 2.8 % for data fusion with anomaly detection. The main reason of these improvements is due to rejecting anomalous attitude VO and laser corrections, especially in yaw angle, which otherwise spoil the trajectory. Details of such anomalous measurement rejections are shown in the Euler angles plots in Figs. 9 and 10.

The inspection experiment took place in a pipeline of diameter slightly larger than the robot, see Fig. 11e, f. As there was too dark for the VO modality, it was not used at all. Figure 12 shows the Euler angles during the whole experiment, as well as the Euler angles detail of a moment, when the robot drove out of the tunnel and the laser modality experienced sudden change from very tight area in the pipeline to a large area in front of the tunnel. There is a visible deterioration of the roll and yaw angles around the time sample of 181 s. This particular instance was handled well by the data fusion due to the anomaly detection.

4.2.4 Test case 4: outdoor forest

The last test case covers by far the most challenging environment from all our datasets. The experiment took place in a

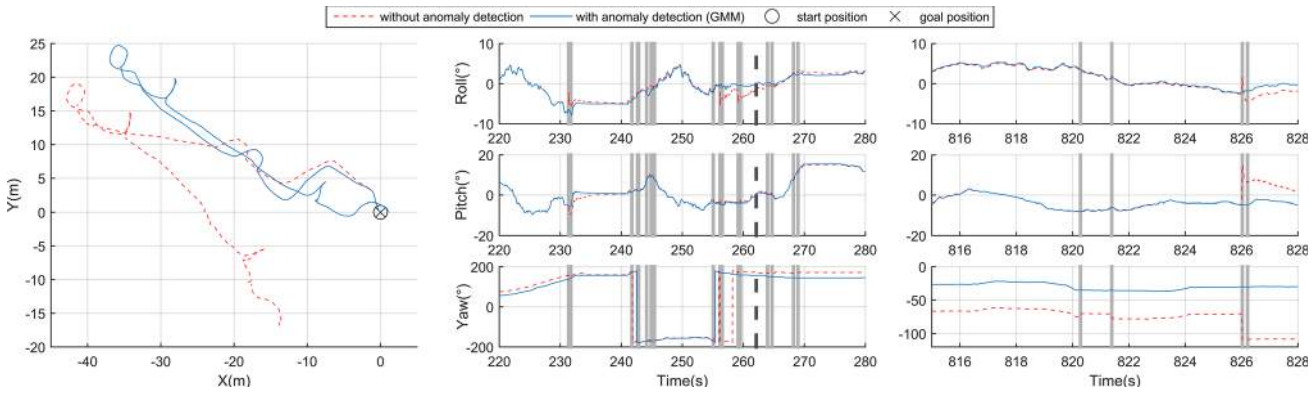


Fig. 9 Exploration of the disaster site 1 (*red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection; *light gray vertical lines* VO anomaly detected and rejected; *dark gray dashed vertical lines* laser anomaly detected and rejected). (*Left*) 2D projection of the trajectories. (*Middle*) Euler angles detail 1. (*Right*) Euler angles detail 2 (Color figure online)

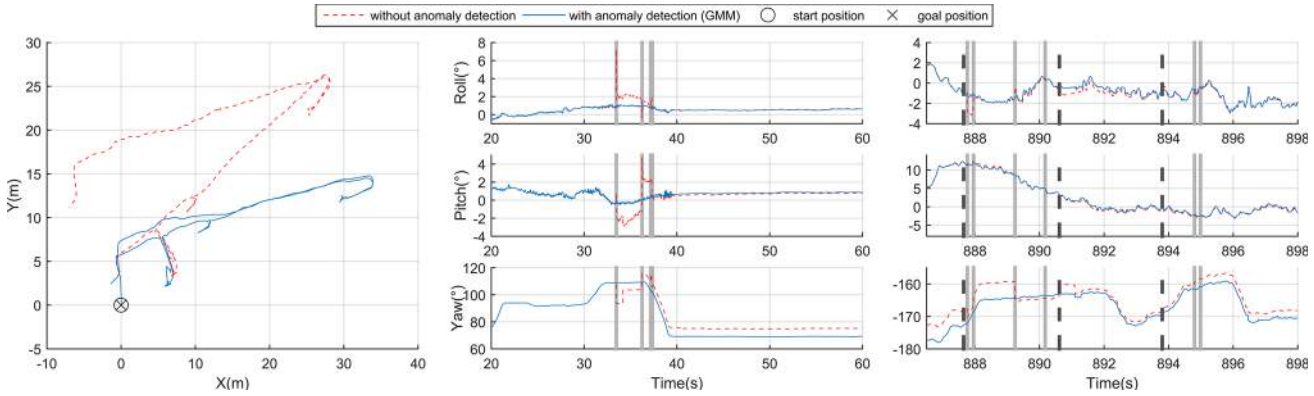


Fig. 10 Exploration of the disaster site 2 (*red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection; *light gray vertical lines* VO anomaly detected and rejected; *dark gray dashed vertical lines* laser anomaly detected and rejected). (*Left*) 2D projection of the trajectories. (*Middle*) Euler angles detail 1. (*Right*) Euler angles detail 2 (Color figure online)

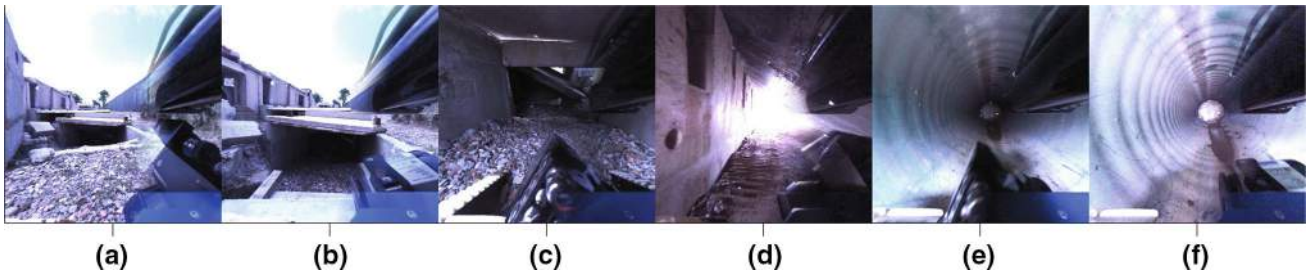


Fig. 11 Disaster site experiment from robot point of view. **a** USAR scenario. **b** Entering collapsed basement. **c**, **d** Inspecting building ruins at the basement level. **e**, **f** Inspecting a pipeline (scene artificially lightened up)

hilly forest area (see pictures in Fig. 13). The robot traversed uneven terrain covered by soil and foliage causing very high slippage. Moreover, both the vision and laser modalities were exposed to very severe conditions. These were mainly the vegetation blocking both camera and laser range of view, or the robot traversing deformable terrain introducing motions too fast to be properly sampled by relatively low frequency of the exteroceptive modalities. Furthermore, there were people simulating dynamic obstacles moving around the robot and partially blocking the camera and laser scanner range of view during the whole experiment.

As gathering reasonable ground truth data for such experiment is very difficult, we attached a GPS receiver on top of the robot, to get an approximate position information along the robot path. One sigma uncertainty of the GPS position was experimentally determined to be $\pm 5\,\mathrm{m}$ (mainly due to

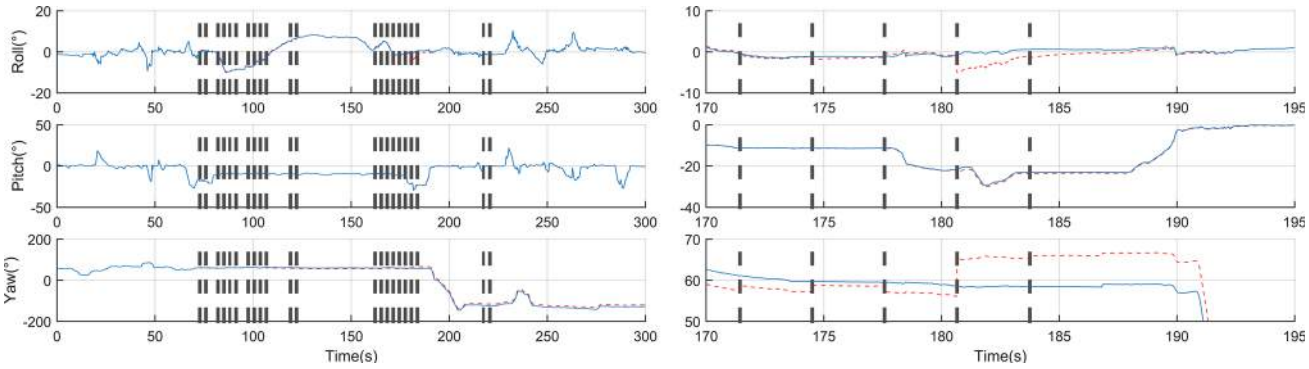


Fig. 12 Inspection of the pipeline (*red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection; *dark gray dashed vertical lines* laser anomaly detected and rejected). (*Left*) Euler angles. (*Right*) Detail of the Euler angles around 181s and anomalous laser attitude corrections casued by robot returning from the pipeline and entering large space area (Color figure online)

Fig. 13 Pictures from the forest experiment. (*Left*) Entering forest on the forest road. (*Middle*) Climbing a steep hill. (*Right*) Descending into a valley (model of the robot projected into a point cloud colored using camera)

dense vegetation and treetops covering most of the sky). This result indicates that the forest environment is almost GPS-denied case if we aim to achieve localization precision less than the body size of the robot.

A 2D projection of the original trajectory, trajectory obtained with anomaly detection, the referential GPS trajectory with its uncertainty, and the average position errors are shown in Fig. 14. Robot initial position was set to zero coordinates and robot proceeded as indicated by the black arrows. It can be clearly seen by visual inspection that the data fusion with anomaly detection outperformed the standard one. The average position error is computed with respect to the GPS trajectory and the final average error for data fusion without anomaly detection is $2\times$ larger than the error for data fusion with anomaly detection. On the other hand, both trajectories are influenced by a visible yaw estimate drift, which is however expected and common when no absolute yaw corrections are provided.

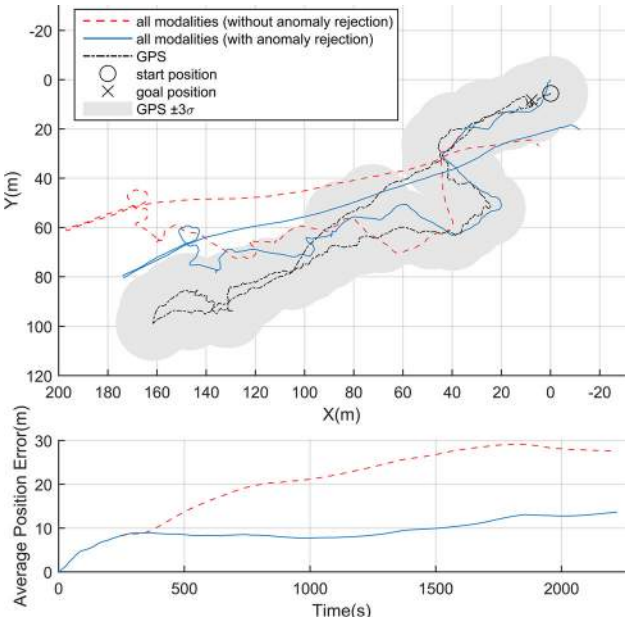


Fig. 14 Forest experiment (*black dash-dot* GPS; *red dashed* multi-modal data fusion without anomaly detection; *blue* multi-modal data fusion with anomaly detection; *gray* experimentally determined GPS uncertainty). (*Top*) 2D projection of the trajectories. (*Bottom*) Average position error expressed as function of time (Color figure online)

4.2.5 Discussion on localization performance

We would like to note that in our data fusion approach we do not have any absolute position or heading measurements available—the whole framework is based on relative mea-

Table 4 Performance summary of the test case experiments with and without anomaly detection (AD)

	$e_{final}(\%)$		$e_{avg}(m)$	
	Without AD	With AD	Without AD	With AD
Indoor	6.2	**0.8**	0.3	**0.1**
Street	5.9	**0.2**	2.7	**0.5**
Park 1	6.1	**1.9**	4.4	**1.4**
Park 2	**0.9**	1.5	**0.6**	0.7
Park 3	17.9	**0.5**	8.7	**0.8**
Train. site 1	16.0	**1.2**	n/a	n/a
Train. site 2	9.5	**2.8**	n/a	n/a
Forest	4.0	**3.4**	27.6	**13.6**

Summary includes all experiments for which the final position error e_{final} or average position error e_{avg} metric is available
Bold values indicate better result in the comparison 'Without AD vs With AD'

surements only—there is no loop-closure in the VO or the ICP mapping algorithm. Therefore, the results presented in Sect. 4.2 diverge from the reference in the same way as can be expected for a dead reckoning localization approach. Based on the analysis of our experiments, we identified that it is actually the yaw angle that contributes the most to the overall error accumulation in this relative pose estimation. The larger the error in yaw estimates, the larger the position error gets accumulated over time. This effect is dramatically magnified if anomalies propagate and influence the yaw angle estimates. From the results summarized in Table 4 as well as by visual inspection of all the projected trajectories, we can conclude that the data fusion supported by anomaly detection provides more accurate localization results. The reason is the anomaly detection handles most of the exteroceptive measurements that would otherwise introduce large errors.

In our previous work (Kubelka et al. 2014) we introduced a failure case methodology to test the actual limits of our proposed data fusion. Without the anomaly detection, the data fusion algorithm performance was limited as expected. Although it cannot be exactly quantified, we hope that by including the anomaly detection we are able to push these limits and even pass most of the failure cases. Although it cannot be proven, similar behavior can be expected during deployment to similar conditions.

5 Conclusion

In this paper, we extended our previous work regarding the multi-modal data fusion for localization of a mobile robot equipped with inertial sensors, track odometry, omnidirectional camera and laser range-finder. Even if the data fusion was well designed, there still occur real-world disturbances related to the environment, no matter which platform is used. These disturbances affect significantly the exteroceptive modalities (in our case the vision and the laser) and subsequently deteriorate the state estimates, causing degradation in precision and reliability of localization. We have shown that incorporating the anomaly detection to exteroceptive measurements is an important step towards a more robust and accurate state estimation.

We presented standard statistical tests for monitoring of the EKF observations and compared them to supervised machine learning approaches; namely: the covariance test, chi-squared test, modified Gaussian Mixture Models, one-class Support Vector Machines, and logistic regression. We demonstrated that the GMM modified by a Mahalanobis distance decision criteria performs the best out of the selected classifiers. This was experimentally verified in indoor tests with precise ground truth using a Vicon system, as well as in challenging outdoor environments. The anomaly detection handled serious local disturbances and vastly improved the overall performance, even in case of driving inside a pipe or through a dense forest. Despite the necessity of creating a labeled ground truth—which requires having a precise reference and is a time-costly procedure—for the supervised machine learning, we still think that the improvements are definitely worth the effort put into training the classifier.

Based on the results, we can conclude that anomaly detection is an important stage of any multi-modal state estimation framework and should not be overlooked or underestimated using a simplified approach. We are convinced that by having such reliable localization, we can now pursue our desired quest for long-term autonomy using our framework.

Acknowledgments The research presented here was supported as follows: Jakub Simanek was supported by the Grant Agency of the CTU in Prague (SGS13/144/OHK3/2T/13); Vladimir Kubelka was supported by the Czech Science Foundation (14-13876S); and Michal Reinstein was supported by the European Union FP7 under the TRADR Project No. FP7-ICT-609763. We would like to thank Francis Colas and Francois Pomerleau for their valuable advices concerning the optimal estimation, and Lorenz Oswald for the assistance during the experiments. The Vicon motion capture system was kindly provided by Autonomous Systems Lab, ETH Zurich, Switzerland.

References

Agamennoni, G., Nieto, J. I., & Nebot, E. M. (2011). An outlier-robust kalman filter. In *IEEE International Conference on Robotics and Automation (ICRA), 2011* (pp. 1551–1558). IEEE.

Ali, J., & Ushaq, M. (2009). A consistent and robust Kalman filter design for in-motion alignment of inertial navigation system. *Measurement*, *42*(4), 577–582.

BarShalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. New York: Wiley.

Besl, P., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.

Borges, G. A., & Aldon, M. J. (2003). Robustified estimation algorithms for mobile robot localization based on geometrical environment maps. *Robotics and Autonomous Systems*, *45*(3), 131–159.

Brunner, C., Peynot, T., Vidal-Calleja, T., & Underwood, J. (2013). Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *Journal of Field Robotics*, *30*(4), 641–666.

Caron, F., Duflos, E., Pomorski, D., & Vanheeghe, P. (2006). Gps/imu data fusion using multisensor Kalman filtering: Introduction of contextual aspects. *Information Fusion*, *7*(2), 221–230.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 1–27. http://www.csie.ntu.edu.tw/cjlin/libsvm.

Chen, Z. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, *182*(1), 1–69.

Christensen, A. L., OGrady, R., Birattari, M., & Dorigo, M. (2008). Fault detection in autonomous robots based on fault injection and learning. *Autonomous Robots*, *24*(1), 49–67.

Dissanayake, G., Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (2001). The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation*, *17*(5), 731–747.

Divis, J. (2013). *Visual odometry from omnidirectional camera.* Master thesism Charles University, Prague.

Dua, S., & Du, X. (2011). *Data mining and machine learning in cybersecurity*. Boca Raton: Taylor & Francis.

Endo, D., Okada, Y., Nagatani, K., & Yoshida, K. (2007). Path following control for tracked vehicles based on slip-compensating odometry. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007* (pp. 2871–2876).

Farrell, J. A. (2008). *Aided navigation: GPS with high rate sensors*. New York: McGraw-Hill.

Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, *19*(2), 78–90.

Gertler, J. (1998). *Fault detection and diagnosis in engineering systems*. Boca Raton: CRC Press.

Goel, P., Dedeoglu, G., Roumeliotis, S. I., & Sukhatme, G. (2000). Fault detection and identification in a mobile robot using multiple model estimation and neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2000, ICRA'00* (Vol. 3, pp. 2302–2309). IEEE.

Görner, M., & Stelzer, A. (2013). A leg proprioception based 6 DOF odometry for statically stable walking robots. *Autonomous Robots*, *34*(4), 311–326.

Grewal, M., & Andrews, A. (2008). Practical considerations. In *Kalman filtering: Theory and practice using MATLAB* (pp. 355–426). Wiley-IEEE Press.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., et al. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, *50*(2), 425–437.

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, IROS 2008* (pp. 3946–3952). IEEE.

Hsu, C. W., Chang, C. C., Lin, C. J., et al. (2003). A practical guide to support vector classification. Technical report.

Hwang, I., Kim, S., Kim, Y., & Seah, C. E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, *18*(3), 636–653.

Knorn, F., & Leith, D. J. (2008). Adaptive Kalman filtering for anomaly detection in software appliances. *INFOCOM Workshops 2008* (pp. 1–6). IEEE.

Konolige, K., Agrawal, M., & Sola, J. (2011). Large-scale visual odometry for rough terrain. In M. Kaneko & Y. Nakamura (Eds.), *Robotics research* (Vol. 66, pp. 201–212)., Springer tracts in advanced robotics Berlin: Springer.

Kruijff, G. J., Janicek, M., Keshavdas, S., Larochelle, B., Zender, H., Smets, N., et al. (2012). Experience in system design for human-robot teaming in urban search & rescue. In *Proceedings of 8th International Conference on Field and Service Robotics* (pp. 111–125). Springer, STAR.

Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. In *Machine learning: ECML-97* (pp. 146–153). Springer.

Kubelka, V., & Reinstein, M. (2012). Complementary filtering approach to orientation estimation using inertial sensors only. In *IEEE International Conference on Robotics and Automation (ICRA), 2012* (pp 599–605). IEEE.

Kubelka, V., Oswald, L., Pomerlau, F., Colas, F., Svoboda, T., & Reinstein, M. (2014). Robust data fusion of multi-modal sensory information for mobile robots. *Journal of Field Robotics*. doi:10.1002/rob.21535.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA), 2011* (pp. 3607–3613).

Lau, T. K., & Lin, K. (2011). Evolutionary tuning of sigma-point Kalman filters. In *IEEE International Conference on Robotics and Automation (ICRA), 2011* (pp. 771–776). IEEE.

Laxhammar, R., Falkman, G., & Sviestins, E. (2009). Anomaly detection in sea traffic—A comparison of the Gaussian mixture model and the kernel density estimator. In *12th International Conference on Information Fusion, 2009, FUSION'09* (pp. 756–763). IEEE.

Ma, J., Susca, S., Bajracharya, M., Matthies, L., Malchano, M., & Wooden, D. (2012). Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in GPS-denied environments. In *Proceedings of the IEEE International Robotics and Automation (ICRA) Conference* (pp. 619–626).

Mitchell, T. M. (1997). *Machine learning* (1st ed.). New York: McGraw-Hill Inc.

Morales, Y., Takeuchi, E., & Tsubouchi, T. (2008). Vehicle localization in outdoor woodland environments with sensor fault detection. In *IEEE International Conference on Robotics and Automation, 2008, ICRA 2008* (pp. 449–454). IEEE.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*., Adaptive computation and machine learning series Cambridge, MA: The MIT Press.

Ndong, J., & Salamatian, K. (2011). A robust anomaly detection technique using combined statistical methods. In *Ninth Annual Communication Networks and Services Research Conference (CNSR), 2011* (pp. 101–108). IEEE.

Pettersson, O. (2005). Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, *53*(2), 73–88.

Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, *34*(3), 133–148.

Reinstein, M., & Hoffmann, M. (2013). Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. *IEEE Transactions on Robotics*, *29*(2), 563–571.

Reinstein, M., Kubelka, V., & Zimmermann, K. (2013). Terrain adaptive odometry for mobile skid-steer robots. In *IEEE International Conference on Robotics and Automation (ICRA), 2013* (pp. 4706–4711).

Rodriguez, F. S. A., Fremont, V., & Bonnifait, P. (2009). An experiment of a 3d real-time robust visual odometry for intelligent vehicles. In *Proceedings of the 12th International IEEE Conference Intelligent Transportation Systems ITSC '09* (pp. 1–6).

Sagha, H., Bayati, H., Millan, Jd R, & Chavarriaga, R. (2013). On-line anomaly detection and resilience in classifier ensembles. *Pattern Recognition Letters*, *34*(15), 1916–1927.

Sarkka, S., & Nummenmaa, A. (2009). Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Transactions on Automatic Control*, *54*(3), 596–600.

Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry. *IEEE Robotics & Automation Magazine*, *18*(4), 80–92.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. C. (1999). Support vector method for novelty detection. *NIPS*, *12*, 582–588.

Shen, J., Tick, D., & Gans, N. (2011). Localization through fusion of discrete and continuous epipolar geometry with wheel and imu odometry. In *Proceedings of the American Control Conference (ACC)* (pp. 1292–1298).

Simanek, J., Reinstein, M., & Kubelka, V. (2015). Evaluation of the EKF-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Transactions on Mechatronics, 20(2)*, 985–990.

Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. www.wiley.com.

Soule, A., Salamatian, K., & Taft, N. (2005). Combining filtering and statistical methods for anomaly detection. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement* (pp. 31–31). USENIX Association.

Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, *15*(3), 572–578.

Sundvall, P., & Jensfelt, P. (2006). Fault detection for mobile robots using redundant positioning systems. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006, ICRA 2006* (pp. 3781–3786). IEEE.

Suzuki, T., Kitamura, M., Amano, Y., & Hashizume, T. (2010). 6-DOF localization for a mobile robot using outdoor 3d voxel maps. In *Proceedings of the IEEE/RSJ International Intelligent Robots and Systems (IROS) Conference* (pp. 5737–5743).

Svoboda, T., Pajdla, T., & Hlaváč, V. (1998). Motion estimation using central panoramic cameras. In S. Hahn (Ed.) *IEEE International Conference on Intelligent Vehicles* (pp. 335–340). Causal Productions.

Ting, J. A., Theodorou, E., & Schaal, S. (2007). A Kalman filter for robust outlier detection. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, IROS 2007* (pp. 1514–1519).

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11,994–12,000.

Weiss, S. M. (2012). *Vision based navigation for micro helicopters*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 20305, 2012.

Yi, J., Zhang, J., Song, D., & Jayasuriya, S. (2007). Imu-based localization and slip estimation for skid-steered mobile robots. In *Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems IROS 2007* (pp. 2845–2850).

Yoshida, T., Irie, K., Koyanagi, E., & Tomono, M. (2010). A sensor platform for outdoor navigation using gyro-assisted odometry and roundly-swinging 3d laser scanner. In *Proceedings of the IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conference* (pp. 1414–1420).

Jakub Simanek received the Ing. (M.Sc.) degree from the Faculty of Electrical Engineering, Czech Technical University in Prague (CTU), in 2011. He is currently a Ph.D. student at the Department of Measurement, CTU in Prague. His research interests include 3D localization, sensor fusion and machine learning.

Vladimir Kubelka received the Ing. (M.Sc.) degree in engineering of Air and Space Systems from the Faculty of Electrical Engineering, Czech Technical University in Prague (CTU), in 2013. He is currently a Ph.D. student at the Center for Machine Perception, Department of Cybernetics, CTU in Prague. His research interests are self-localization and navigation of unmanned ground and air vehicles in indoor/outdoor environments for deployment in Urban Search & Rescue scenarios.

Michal Reinstein received the Ing. (M.Sc.) and Ph.D. degrees in engineering of Aircraft Information and Control Systems from the Faculty of Electrical Engineering, Czech Technical University in Prague (CTU), in 2007 and 2011, respectively. He is currently working as researcher at the Center for Machine Perception, Department of Cybernetics, CTU in Prague. His research interests concern data fusion for mobile robotic systems, based on combined state estimation and machine learning techniques, intended for Urban Search & Rescue; currently: modeling using Gaussian processes and reinforcement learning for motion control. After the internship at the Artificial Intelligence Lab (Prof. R. Pfeifer), University of Zurich in 2009, he also focuses his research efforts towards exploring sensory-motor interaction and locomotion control in legged robots.

Robust Data Fusion of Multi-modal Sensory Information for Mobile Robots

Vladimír Kubelka
Center for Machine Perception, Dept. of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague
Technicka 2, 166 27, Prague 6, Czech Republic
kubelka.vladimir@fel.cvut.cz

Lorenz Oswald
ETH Zurich
Tannenstrasse 3, 8092 Zurich, Switzerland
loswald@student.ethz.ch

François Pomerleau
ETH Zurich
Tannenstrasse 3, 8092 Zurich, Switzerland
francois.pomerleau@mavt.ethz.ch

Francis Colas
ETH Zurich
Tannenstrasse 3, 8092 Zurich, Switzerland
francis.colas@mavt.ethz.ch

Tomáš Svoboda
Center for Machine Perception, Dept. of Cybernetics
Faculty of Elect. Eng., CTU in Prague
Technicka 2, 166 27, Prague 6, Czech Republic
svobodat@fel.cvut.cz

Michal Reinstein
Center for Machine Perception, Dept. of Cybernetics
Faculty of Elect. Eng., CTU in Prague
Technicka 2, 166 27, Prague 6, Czech Republic
reinstein.michal@fel.cvut.cz

Abstract

Urban Search and Rescue missions for mobile robots require reliable state estimation systems resilient to conditions given by the dynamically changing environment. We design and evaluate a data fusion system for localization of a mobile skid-steer robot intended for USAR missions. We exploit a rich sensor suite including both proprioceptive (inertial measurement unit and tracks odometry) and exteroceptive sensors (omnidirectional camera and rotating laser rangefinder). To cope with the specificities of each sensing modality, we introduce a novel fusion scheme based on Extended Kalman filter for 6DOF orientation and position estimation. We demonstrate the performance on field tests of more than 4.4 km driven under standard USAR conditions. Part of our datasets include ground truth positioning; indoor with a Vicon motion capture system and outdoor with a Leica theodolite tracker. The overall median accuracy of localization—achieved by combining all the four modalities—was 1.2 % and 1.4 % of the total distance traveled, for indoor and outdoor environments respectively. To identify the true limits of the proposed data fusion we propose and employ a novel experimental evaluation procedure based on failure case scenarios. This way we address the common issues like: slippage, reduced camera field of view, limited laser rangefinder range, together with moving obstacles spoiling the metric map. We believe such characterization of the failure cases is a first step towards identifying the behavior of state estimation under such conditions. We release all our datasets to the robotics community for possible benchmarking.

1 Introduction

Mobile robots are sought to be deployed for many tasks, from tour-guide robots to autonomous cars. With the rapid advance in sensor technology, it has been possible to embed richer sensor suites and extend the perception capabilities. Such sensor suites provide multi-modal information that naturally ensure perception robustness, allowing also better means of self-calibration, fault detection and recovery—given that appropriate data fusion methods are exploited. Independently from the application, a key issue of mobile robotics is state estimation. It is crucial for both perception, like mapping, and action, like avoiding obstacles or terrain adaptation.

In this paper, we address the problem of data fusion for localization of an Unmanned Ground Vehicle (UGV) intended for Urban Search and Rescue (USAR) missions. There has been a significant effort presented in the field of USAR for robot localization that mostly aims for a minimal suitable sensing setup; exploiting usually the inertial measurements aided with either vision or laser data. Having a sufficient on-board computational power, we therefore aim for a richer sensors suite and hence for better robustness and reliability. Therefore, our UGV used in this work (see Figure 1) embeds track encoders, an Inertial Measurement Unit (IMU), an omnidirectional camera, and a rotating laser range-finder.

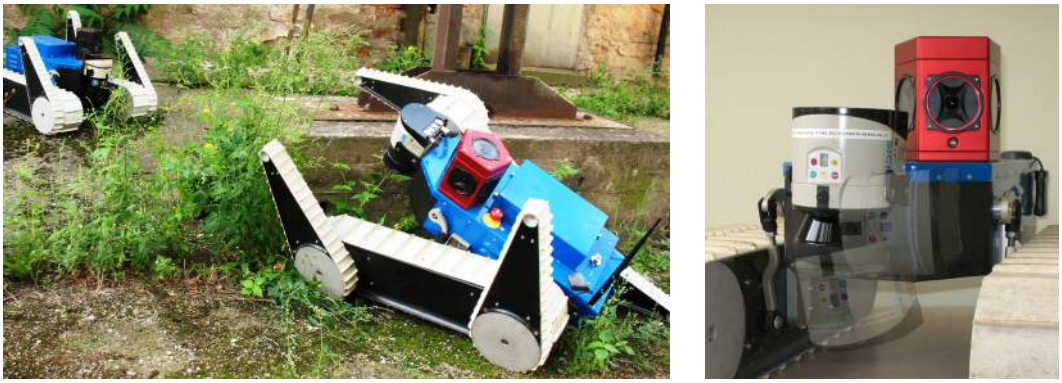


Figure 1: Picture of two USAR UGVs used for experimental evaluation (FP7-ICT-247870 NIFTi project) and a detail of the sensor setup (a PointGrey Ladybug 3 omnicamera and a rotating SICK LMS-151 laser range finder). See Section 3.1 for more details.

Our first contribution lies in the development of a model for such multi-modal data fusion using the Extended Kalman Filter (EKF), especially in the way we incorporate sensors with slow and fast measurement update rates. In order to cope with such significant difference in the update rates of various sensor modalities, we concentrated the model design on integrating the slow laser and visual odometry with the faster IMU and track odometry measurements. For this purpose, we propose and investigate three different possible methods—one of them, the *trajectory approach* (see Section 4.3.3 for further details), is our contribution that we compare it to the *velocity approach*, which is a common state-of-the-art practice. We show that a standard EKF designed with the *velocity approach* does not cope well with such significant differences in the frequency, whether our proposed *trajectory approach* does.

The context of USAR missions implicitly defines challenges and limitations of our application. The environment is often unstructured (collapsed buildings) and unstable (moving objects or other ongoing changes, deformable terrain causing high slippage). Robots need to cope with indoor-outdoor transitions (change from confined to open spaces), bad lighting conditions with rapid changes and sometimes decreased visibility (smoke and fire). These are essentially the main challenges that come with the sensor data we process. Therefore, our main contribution lies in the actual experimental evaluation and analysis of limits of the proposed filter. We review the different sensing modalities and their expected failure cases to assess the impact of possible data degradation (or outage) on the overall precision of localization. We believe that the field deployment of state estimation for multi-modal data fusion needs to be characterized both under standard

expected conditions as well as for case of partial or full failures of sensing modalities. Indeed, robustness to sensor data outage or degradation is a key element to the scaling up of a field robotics system. Therefore, we evaluate our filter using several hours and kilometers of experimental data validated by indoor or outdoor ground truth measurements. In order to share this contribution to the robotics community, we release all the captured datasets (including the ground truth measurements) to be used as benchmarks.[1]

The state of the art of sensor fusion for state estimation is elaborated in Section 2. In Section 3, we present the hardware and software used in this work before describing in details the design of our data fusion algorithm (Section 4). In Section 5, we explain our experimental evaluation including our fail-case methodology before a discussion and conclusion (Section 6).

2 Related work

In general, the information obtained from various sensors can be classified as either proprioceptive (inertial measurements, joint sensors, motor or wheel encoders, etc.) or exteroceptive (Global Positioning System (GPS), cameras, laser range finder, ultrasonic sensors, magnetic compass etc.). Exteroceptive sensors that acquire information from the environment can be also used to perceive external landmarks that are necessary for long-term precision in navigation tasks. In modern mobile robots, a popular solution lies usually in the combination of a proprioceptive component in the form of Inertial Navigation System (INS) (Titterton and Weston, 1997), that captures the body dynamics at high frequency, and an external source of aiding, using vision (Chowdhary et al., 2013) or range measurements (Bachrach et al., 2011). The key issue lies in the appropriate integration of the different characteristics of the different sensor modalities.

As it was repeatedly shown, the combination of an IMU with wheel odometry is a popular technique to localize a mobile robot in a dead reckoning manner. It generally allows very high sampling frequency as well as processing rate, usually without excessive computational load. Dead reckoning can be used for short term navigation without any necessity of perceiving surrounding environment via exteroceptive sensors. In real outdoor conditions, dynamically changing environment often causes signal degradation or even outage of exteroceptive sensors. However, proprioceptive sensing, in principle, is too prone to accumulating errors to be used as a standalone solution. Computational and environmental errors as well as errors caused by misalignment and instrumentation cause the dead reckoning system to drift quickly with time. Moreover, motor encoders do not reflect the true path, especially heading of the vehicle, in case of frequent wheel slip. In (Yi et al., 2007) and (Anousaki and Kyriakopoulos, 2004), an improvement through skid-steer model of a 4-wheel robot is presented, based on a Kalman filter estimating trajectory using velocity constraints and slip estimate. An alternative method appears in (Endo et al., 2007) where the IMU and odometry are used to improve tracked vehicle navigation via slippage estimates. We addressed this problem in (Reinstein et al., 2013). Substantial effort has also been made to investigate the odometry derived constraints (Dissanayake et al., 2001), or innovation of the motion models (Galben, 2011). Concerning all the references so far, localization of the navigated object via dead reckoning was performed only in 2D. There exist solutions providing real 3D odometry derived from the rover-type multi-wheel vehicle design (Lamon and Siegwart, 2004). Nevertheless, the error is still about one order of magnitude higher than what we aim to achieve (below 2% of the total distance traveled).

However, if long-term precision and reliability is to be guaranteed, dead-reckoning solutions require other exteroceptive aiding sensor systems. In the work of (Shen et al., 2011), it is shown that a very low-cost IMU and odometry dead-reckoning system can be realized and successfully combined with visual odometry (VO) (Sakai et al., 2009; Scaramuzza and Fraundorfer, 2011) to produce a reliable navigation system. With the increasing on-board computational power, visual odometry is becoming very popular even for large-scale outdoor environments. Most solutions are based on the Extended Kalman filter (EKF) (Oskiper et al., 2010; Civera et al., 2010; Konolige et al., 2011; Chowdhary et al., 2013) or a dimensional-bounded EKF with landmark classifier introduced in (Jesus and Ventura, 2012). However, in (Rodriguez F et al., 2009)

[1] The datasets are available as *bagfiles* for ROS at `https://sites.google.com/site/kubelvla/public-datasets`

it is pointed out that a trade-off between precision and execution time has to be examined. Moreover, VO degrades due to high rotational speed movements and it is susceptible to illumination changes and lack of sufficient scene texture (Scaramuzza and Fraundorfer, 2011).

Another typically used 6 DOF aiding source is a laser rangefinder, which is used for estimating vehicle motion by matching consecutive laser scans and creating a 3D metric map of the environment (Suzuki et al., 2010; Yoshida et al., 2010). Examples of successful application can be found for both indoor—without IMU but combined with vision (Ellekilde et al., 2007)—as well as outdoor—relying on the IMU (Bachrach et al., 2011). As in case of the visual odometry, solutions using EKF are often proposed (Morales et al., 2009; Bachrach et al., 2011). The most popular approach of scan matching is based on the Iterative Closest Point (ICP) algorithm first proposed by (Besl and McKay, 1992) and in parallel by (Chen and Medioni, 1991). More recently, (Nuchter et al., 2007) proposed a 6D Simultaneous Localization and Mapping (SLAM) system relying mainly on ICP. Closer to USAR applications, (Nagatani et al., 2011) demonstrated the use of ICP in exploration missions and used a pose graph minimization scheme to handle multi-robot mapping. (Kohlbrecher et al., 2011) proposed a localization system combining a 2D laser SLAM with a 3D IMU/odometry-based navigation subsystem. Combination of 3D-landmark-based SLAM and multiple proprioceptive sensors is also presented in (Chiu et al., 2013), their work aims mainly on low latency solution while estimating the navigation state by means of Sliding-Window Factor Graph. The problem of utilizing several sensors for localization that may provide contradictory measurements is discussed in (Sukumar et al., 2007). The authors use Bayes filters to estimate sensor measurement uncertainty and sensor validity to intelligently choose a subset of sensors that contribute to localization accuracy. As opposed to the later publications realized in the context of SLAM, we only consider the results of the ICP algorithm as a local pose measurement similarly to (Almeida and Santos, 2013) who use the ICP algorithm to extract steering angle and linear velocity of a car-like vehicle to update its non-holonomic model of motion. In our approach, the 3D reconstruction of the environment is considered locally coherent and neither loop detection nor error propagation is used.

As stated in (Kelly et al., 2012), it is the right time to address concerning issues of the state-of-the-art in long-term navigation and autonomy. In this respect, benefits and challenges of repeatable long-range driving were addressed in (Barfoot et al., 2012). In this context, we believe that bringing more insight into multi-modality state estimation algorithms is an important step for long-term stability of an USAR system evolving in a complex range of environments.

Regarding multi-modal data fusion, we built on our previous work concerning complementary filtering (Kubelka and Reinstein, 2012), odometry modeling (Reinstein et al., 2013), and design of EKF error models (Reinstein and Hoffmann, 2013), even though the later work applied to a legged robot.

3 System description

Our system aims at high state estimation accuracy while ensuring robust performance against rough terrain navigation and obstacle traversals. We selected four modalities to achieve this goal: the inertial measurements (IMU), odometry data (OD), visual odometry (VO) and laser rangefinder data (ICP) processed by the ICP algorithm. This section explains the motion capabilities of the Search & Rescue platform and the preprocessing computation applied to its sensors in order to extract meaningful inputs for the state estimation. These explanations provide a motivation for a list of states to be estimated by the EKF described in Section 4.

3.1 Mobile Robotic Platform

Figure 1 presents the UGV designed for USAR mission that we use in this paper. As described in (Kruijff et al., 2012), this platform was deployed multiple times in collaboration with various rescue services (Fire

Department of Dortmund/Germany, Vigili del Fuoco/Italy). It has two bogies linked by a differential that allows a passive adaptation to the terrain. On each of the tracks, there are two independent flippers that can be position-controlled in order to increase the mobility in difficult terrain. For example, they can be unfolded to increase the support polygon which helps in overcoming gaps and being more stable on slopes. They can also be raised to help with climbing over higher obstacles. Given that the robot was designed to operate in 3D unstructured environments, the state estimation system needs to provide a 6 DOF localization.

Encoders are placed on the differential, giving the angle between the two bogies and the body; on the tracks to give their current velocity; and on each flipper to give its position with respect to its bogies. Inside the body, vertical to the center of the robot, lies the Xsens MTi-G IMU providing angular velocities and linear acceleration along each of the three axes. The IMU data capture the body dynamics at high rate of 90 Hz. GPS is not taken into account due to the low availability of the signal indoors or in close proximity with building. Magnetic compass is also easily disturbed by metallic masses, pipes, and wires, which make it highly unreliable and hence we do not use it.

The exteroceptive sensors of the robot consist of an omnidirectional camera and a laser rangefinder. The omnidirectional camera is the PointGrey Ladybug 3 and produces a 12 megapixels stitched omni-directional images at 5-6 Hz. The omni-directionality of the sensor provides a stronger stability of rotation estimation at the expense of scale estimation, which would be better handled by a stereocamera. The laser rangefinder used is the Sick LMS-151 mounted on a rolling axis in front of the robot. The laser spins left and right alternately, taking a full 360° scan at approximately 0.3 Hz to create a point cloud of around 55,000 points.

3.2 Inertial data processing

Though the precision and reliability of the IMU measurements is sufficient in short term, in long term the information provided suffers from random drift that, together with integrated noise, cause unbounded error growth. To cope with these errors all the 6 sensor biases have to be estimated (see Section 4.1 for more details). Therefore, we have included sensor biases in the state space of the proposed EKF estimator. Furthermore, correct calibration of the IMU output and its alignment with respect to the robot's body frame has to be assured.

3.3 Odometry for skid-steer robots

Our platform is equipped with caterpillar tracks and therefore steering is realized by setting different velocities for each of the tracks (*skid-steering*). The encoders embedded in the tracks of the platform measure the left and right track velocities at approximatively 15 Hz. However, in contradistinction to differential robots, the odometry for skid-steering vehicles has significant uncertainties. Indeed, as soon as there is a rotation, the tracks must either deform or slip significantly. The slippage is affected by many parameters including the type and local properties of the terrain. To keep the computation complexity low, we assume only a simple odometry model and we do not model the slippage. Instead, we take advantage of the exteroceptive modalities in our data fusion to observe the true motion dynamics using different sources of information. Hence, the fusion compensates for cases when the tracks are slipping because the surface is slippery or because of an obstacle blocking the robot. Another advantage of using caterpillar tracks odometry lies in the opportunity to exploit nonholonomic constraints. Further explanations on those constraints are given in Section 4.3.

3.4 ICP-based localization

Using as *Input* the current 3D point cloud, a registration process is used to estimate the pose of the robot with respect to a global representation called *Map*. We used a derivation of the point-to-point ICP algorithm introduced by (Chen and Medioni, 1991) combined with the trimmed outlier rejection presented by

(Chetverikov et al., 2002).

The implementation uses `libpointmatcher`[2], an open-source library fast enough to handle real-time processing while offering modularity to cover multiple scenarios as demonstrated in (Pomerleau et al., 2013). The complete list of modules used with their main parameters can be found in Table 1. In more details, the configuration of the rotating laser produced a high density of points in front of the robot, which was desirable to predict collision but not beneficial to the registration minimization. Thus, we forced the maximal density to 100 points per m^3 after having randomly subsampled the point cloud in order to finish the registration and the map maintenance within 2 s. We expected the error on pre-alignment of the 3D scans to be less than 0.5 m based on the velocity of the platform and the number of ICP per second that was to be executed. So we used this value to limit the matching distance. We also removed paired points with an angle difference larger than 50° to avoid the reconstruction of both sides of walls to collapse when the robot was exploring different rooms. The surface normal vector used for the *outlier filtering* and for the *error minimization* are computed using 20 Nearest Neighbors (NN) of every point within a single point cloud. As for the global map, we maintained a density of 100 points per m^3 every time a new input scan was merged in it. A maximum of 600,000 points were kept in memory to avoid degradation of the computation time when exploring a larger environment than expected. However, the only output of the ICP algorithm we consider is the robot's localization, i.e. position and orientation relative to its inner 3D point-cloud map. We do not aim at creating a globally consistent map and we do not exploit the map in any other way than for analysis of the ICP performance (no map corrections or loop closures are performed).

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

	Step	*Module*	*Description*
Input	Read. filtering	`SimpleSensorNoise`	SickLMS
		`SamplingSurfaceNormal`	keep 80 %, surface normals based on 20 NN
		`ObservationDirection`	add vector pointing toward the laser
		`OrientNormals`	orient surface normals toward the obs. direction
		`MaxDensity`	subsample to keep point with density of 100 pts/m^3
Registration	Ref. filtering	-	processing from the rows Map
	Read. filtering	-	processing from the rows Input
	Data association	`KDTree`	kd-tree matching with 0.5 m max. distance, $\epsilon = 3.16$
	Outlier filtering	`TrimmedDist`	keep 80 % closest points
		`SurfaceNormal`	remove paired normals angle $> 50°$
	Error min.	`PointToPlane`	point-to-plane
	Trans. checking	`Differential`	min. error below 0.01 m and 0.001 rad
		`Counter`	iteration count reached 40
		`Bound`	transformation fails beyond 5.0 m and 0.8 rad
Map	Ref. filtering	`SurfaceNormal`	Update normal and density, 20 NN, $\epsilon = 3.16$
		`MaxDensity`	subsample to keep point with density of 100 pts/m^3
		`MaxPointCount`	subsample 70 % if more than 600,000 points

There is one ICP-related issue observed with our platform. Although the ICP creates a locally precise metric map, the map as whole tends to slightly twist or bend (we do not perform any loop-closure). This is the reason why the position and the attitude estimated by the ICP odometry collide with other position information sources. Another limitation is the refresh rate of the pose measurements limited to 0.3 Hz. This rate is far from our fastest measurement (i.e., the IMU at 90 Hz), which poses a linearization problem. For these reasons, we investigated three different types of measurement models; see Section 4.3.3 for details.

Furthermore, the true bottleneck of the ICP-based localization lies in the way it is realized on our platform and hence prone to mechanical issues. As the laser rangefinder has to be turning to provide full 3D point cloud, in environment with high vegetation such mechanism is easily struck, causing this modality to fail. Large open spaces, indoor / outdoor transitions, or significantly large moving obstacles can also cause the ICP to fail updating the metric map. Since this modality is very important, we analyzed these failure cases in Section 5.4.

[2] `https://github.com/ethz-asl/libpointmatcher`

3.5 Visual odometry

Our implementation of visual odometry generally follows the usual scheme (Tardif et al., 2008; Scaramuzza and Fraundorfer, 2011). The VO computation runs solely on the robot on-board computer and estimates the pose at the frame rate 2-3 Hz which, compared to the robot speed, is sufficient. It does search for correspondences (i.e., image matching) (Rublee et al., 2011), landmark reconstruction and sliding bundle adjustment (Kummerle et al., 2011; Fraundorfer and Scaramuzza, 2012), which refines the landmark 3D positions and the robot poses. The performance essentially depends on the visibility and variety of landmarks. The more variant landmarks are visible at more positions, the more stable and precise is the pose estimation. The process uses panoramic images constructed from spherical approximation of the Ladybug camera model. The Ladybug camera is approximated as one central camera. The error of the approximation is acceptable for landmarks which are few meters from the robot.

The visual odometry starts with detecting and matching features in two consecutive images. We use OpenCV implementation of the Orb keypoint detector and descriptor (Rublee et al., 2011). Only the matches, which are distinctive above certain threshold, survive. The initial matching is supported by a guided matching which uses an initial estimate of the robot movement. The robot movement is estimated by the 5-point solver (Li and Hartley, 2006) encapsulated in RANSAC iterations. As the error measure we use the angular deviation of points from epipolar planes. This is less precise than the usual distance from epipolar lines. However, as we work with spherical projection we have epipolar curves. Computing angular deviations is faster than computing distance to the epipolar curve. The movement estimate projects already known landmarks and we can actively search around the projection. The feature tracks are updated and associated with landmarks if the they pass an observation consistency test. The landmark 3D position is triangulated from all possible observations and the complete estimate of landmark and robot positions are refined by a bundle adjustment (Kummerle et al., 2011).

Using an almost omnidirectional camera for the robot motion estimation is geometrically advantageous (Brodsky et al., 1998; Svoboda et al., 1998). The scale estimation however, depends on the precision of 3D reconstruction where the omnidirectionality does not really help. It is also important to note the omnidirectional camera we use sits very low above the terrain (below 0.5 m) and directly on the robot body. This makes a huge difference compared to, e.g. (Tardif et al., 2008), where the camera is more than 2 m above the terrain and sees the ground plane much better than our camera. Estimation of the yaw angle is still well conditioned since it relies mostly on the side correspondences. The pitch estimation however, would sometimes need more landmarks on the ground plane. The pitch part of the motion induces largest disparity of the correspondences in the front and back cameras. Unfortunately the back view is significantly occluded by the battery cover. This is especially problem in the street scenes where the robot moves along the street, see e.g., Figure 11. The front cameras see the street level better however, the uniform texture of the tar surface often generates only a few reliable correspondences. The search for correspondences is further complicated by the tilting flippers which occlude the field of view and induce outliers. Second problem is the agility of the robot combined with relatively low frequency of the visual odometry. The robot can turn on spot very quickly, much quicker than an ordinary wheeled car. Even worse, the quick turn is the usual way how the movement direction is changed. This all makes correspondence search difficult. In the future versions of the visual odometry we want to improve the landmark management in order to resolve the problem of too few landmarks surviving the sudden turn. We also think about replacing the approximate spherical model by reformulating it in a multiview model.

4 Multi-modal data fusion

The core of the data fusion system is realized by an error state EKF inspired by the work of (Weiss, 2012). The description of the multi-modal data fusion solution we propose can be divided into two parts. First is the process error model for the EKF, that shows how we model the errors, which we aim to estimate and use for corrections. Second part is the measurement model, that couples the sensory data coming at different

rates.

The overall scheme of our proposed approach is shown in Figure 2. Raw sensor data are preprocessed and used as measurements in the error state EKF (the *FUSION* block). There is no measurement rejection implemented; based on the assumption that fusion of several sensor modalities should deal with anomalous data inherently—for details see Section 5 and Section 6—this however will be subjected to a future work. As apparent from the Figure 2, measurement rates significantly differ among the sensor modalities—main difference is especially between the IMU at 90Hz and the ICP output at 0.3Hz. Having the update rate of the EKF at 90Hz the experiments have proven that this issue is crucial and has to be resolved as part of the filter design to ensure reliable output from the fusion process (see Section 5.3.3). In our case, this problem concerns mainly the ICP-based localization that provides measurements at very low rate of 0.3Hz—too low to capture the motion dynamics as the IMU does (i.e. the motion dynamics spectrum gets sub-sampled). During these 3 seconds, real-world disturbances (which are often non-Gaussian and difficult to model and predict, e.g. tracks slippage) accumulate. This was the motivation to investigate various ways of fusing measurements at significantly different rates. Three proposed approaches that incorporate the ICP measurements are described in the Section 4.3.3.

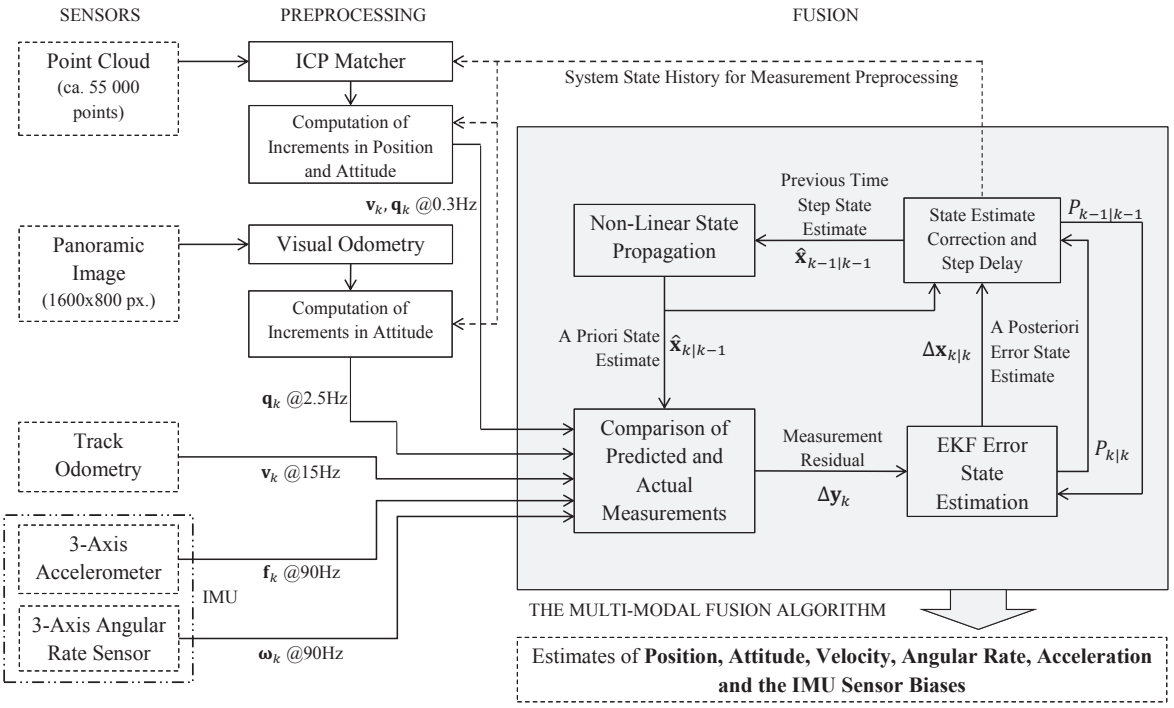


Figure 2: The scheme of the proposed multi-modal data fusion system (ω is angular velocity, $\mathbf{f}$ is specific force (Savage, 1998), $\mathbf{v}$ is velocity and $\mathbf{q}$ is quaternion representing attitude).

4.1 Process error model

For the purpose of localization, we model our robot as a rigid body with constant angular rate and constant rate of change of velocity ($\dot{\omega} = 0, \dot{v} = \text{const.}$). Presence of constant gravitational acceleration is expected and incorporated into system model; no dissipative forces are considered.

We define four coordinate frames: *R(obot)* frame coincides with center of the robot, *I(MU)* frame represents the Inertial Measurement Unit coordinate frame as defined by the manufacturer, *O(dometry)* frame represents

the tracked gear–frame, and *N(avigation)* frame represents the world frame. In all these frames, the North-West-Up axes convention is followed, with the x-axis pointing forwards (or to the North in the N–frame), the y-axis pointing to the left (or to the West), and the z-axis pointing upwards. Rotations about each axis follow the *right-hand rule*. The fundamental part of the system design are the differential equations describing development of the states in time. The state space with the corresponding errors is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_N \\ \mathbf{q}_N^R \\ \mathbf{v}_R \\ \omega_R \\ \mathbf{f}_R \\ \mathbf{b}_{\omega,I} \\ \mathbf{b}_{f,I} \end{bmatrix} \qquad \Delta\mathbf{x} = \begin{bmatrix} \Delta\mathbf{p}_N \\ \delta\theta \\ \Delta\mathbf{v}_R \\ \Delta\omega_R \\ \Delta\mathbf{f}_R \\ \Delta\mathbf{b}_{\omega,I} \\ \Delta\mathbf{b}_{f,I} \end{bmatrix} \tag{1}$$

where $\mathbf{p}_N$ is position of the robot in the N–frame, $\mathbf{q}_N^R$ is unit quaternion representing its attitude, $\mathbf{v}_R$ is velocity expressed in the R–frame, ω_R is angular rate, $\mathbf{f}_R$ is specific force (Savage, 1998), $\mathbf{b}_{\omega,I}$ and $\mathbf{b}_{f,I}$ are accelerometer and angular rate sensor IMU-specific biases expressed in the I–frame.

The error state $\Delta\mathbf{x}$ is defined—following the idea of (Weiss, 2012, eq. 3.25)—as difference between the system state and its estimate $\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$ except for attitude, where rotation error vector $\delta\theta$ is the vector part of the error quaternion $\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}$ multiplied by 2; $\otimes$ represents quaternion multiplication as defined in (Breckenridge, 1999).

The states and the error states of the robot, modeled as a rigid body movement, propagate in time according to the following equations:

$$\dot{\mathbf{p}}_N = C^T_{(\mathbf{q}_N^R)}\mathbf{v}_R \qquad \Delta\dot{\mathbf{p}}_N \approx C^T_{(\hat{\mathbf{q}}_N^R)}\Delta\mathbf{v}_R - C^T_{(\hat{\mathbf{q}}_N^R)}\delta\theta \tag{2}$$

$$\dot{\mathbf{q}}_N^R = \frac{1}{2}\Omega(\omega_R)\mathbf{q}_N^R \qquad \delta\dot{\theta} \approx -\lfloor\hat{\omega}_R\rfloor\delta\theta + \Delta\omega_R + \mathbf{n}_\theta \tag{3}$$

$$\dot{\mathbf{v}}_R = \mathbf{f}_R - C_{(\mathbf{q}_N^R)}\mathbf{g}_N + \lfloor\mathbf{v}_R\rfloor\omega_R \qquad \Delta\dot{\mathbf{v}}_R \approx \Delta\mathbf{f}_R - \lfloor C_{(\hat{\mathbf{q}}_N^R)}\mathbf{g}_N\rfloor\delta\theta + \lfloor\hat{\mathbf{v}}_R\rfloor\Delta\omega_R - \lfloor\hat{\omega}_R\rfloor\Delta\mathbf{v}_R + \mathbf{n}_v \tag{4}$$

$$\begin{array}{cccc} \dot{\omega}_R = 0 & \dot{\mathbf{f}}_R = 0 & \dot{\mathbf{b}}_{\omega,I} = 0 & \dot{\mathbf{b}}_{f,I} = 0 \\ \Delta\dot{\omega}_R = \mathbf{n}_\omega & \Delta\dot{\mathbf{f}}_R = \mathbf{n}_f & \Delta\dot{\mathbf{b}}_{\omega,I} = \mathbf{n}_{b,\omega} & \Delta\dot{\mathbf{b}}_{f,I} = \mathbf{n}_{b,f} \end{array} \tag{5}$$

where derivation of the left part of (3) can be found in (Trawny and Roumeliotis, 2005, eq. 110) and the left part of (4) is based on (Nemra and Aouf, 2010, eq. 5); the difference from the original is caused by different ways of expressing attitude. The right parts of (2-4) can be derived by neglecting higher-order error terms and by approximation of the error in attitude by the rotation error vector $\delta\theta$ following (Weiss, 2012, eq. 3.44). We define $\mathbf{g}_N = [0, 0, g]^T$, $\mathbf{n}_{(.)}$ are the system noise terms and $\Omega(\omega_R)$ in (3) is a matrix representing quaternion and vector product operation (Trawny and Roumeliotis, 2005, eq. 108). It is constructed as

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{6}$$

In (5), time derivations of angular rates and specific forces are equal to zero—usually, they are considered rather as input than state. However, we included them into the state vector to be updated by the EKF. The error model equations can be expressed in compact matrix form:

$$\Delta\dot{x} = F_c\Delta\mathbf{x} + G_c\mathbf{n} \tag{7}$$

where F_c is continuous time state transition matrix, G_c is noise coupling matrix and $\mathbf{n}$ is noise vector composed of all the $\mathbf{n}_{(.)}$ terms; the F_c matrix is as follows:

$$F_c = \left[\begin{array}{c|c|c|c|c|c|c} \emptyset_3 & -C^T_{(\hat{q}^R_N)} & C^T_{(\hat{q}^R_N)} & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & -\lfloor \hat{\omega}_R \rfloor & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & -\lfloor C_{(\hat{q}^R_N)} g_N \rfloor & -\lfloor \hat{\omega}_R \rfloor & \lfloor \hat{v}_R \rfloor & I_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \end{array}\right] \tag{8}$$

and the $G_c\mathbf{n}$ term is

$$G_c\mathbf{n} = \left[\begin{array}{c|c|c|c|c|c} \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 \\ \hline \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & I_3 \end{array}\right] \left[\begin{array}{c} \mathbf{n}_\theta \\ \mathbf{n}_v \\ \mathbf{n}_\omega \\ \mathbf{n}_f \\ \mathbf{n}_{b,\omega} \\ \mathbf{n}_{b,f} \end{array}\right] \tag{9}$$

The noise coupling matrix describes, how particular noise terms affect the system state. Each $\mathbf{n}_{(.)}$ term is a random variable with Normal probability distribution. Properties of these random variables are described by their covariances in the system noise matrix Q_c. Since they are assumed independent, the matrix Q_c is diagonal $Q_c = diag(\sigma^2_{\theta_x}, \sigma^2_{\theta_y}, \sigma^2_{\theta_z}, \sigma^2_{v_x}, \sigma^2_{v_y}, \cdots)$, where σ is standard deviation.

In order to implement the proposed model, we have to transform the continuous time equations to discrete time domain. We use the Van Loan discretization method (Loan, 1978) instead of explicitly expressing values of the discretized matrices. We substitute into matrix M defined by Van Loan

$$M = \begin{bmatrix} -F_c & GQ_cG^T \\ \emptyset & F_c^T \end{bmatrix} \Delta t \tag{10}$$

and evaluate the matrix exponential

$$\mathrm{e}^M = \begin{bmatrix} \ddots & F_d^{-1}Q_d \\ \emptyset & F_d^T \end{bmatrix} \tag{11}$$

The result of the matrix exponential contains the discretized system matrix F_d in the bottom-right part and the discretized system noise matrix Q_d left multiplied by the inversion of F_d in the top-right part. The discretized system matrix F_d can be easily extracted; Q_d can be obtained by left multiplying the upper right part of e^M by F_d.

4.2 State prediction and update using EKF

The Extended Kalman filter (Smith et al., 1962; McElhoe, 1966), is a modification of the Kalman filter (Kalman, 1960), i.e. optimal observer minimizing variances of the observed states. Since the error state EKF is used in our approach, the state of the system is expressed as sum of current best estimate ($\hat{\mathbf{x}}$) and some small error ($\Delta\mathbf{x}$). The only difference compared to a standard EKF is that the linearised system matrices F and Q describe only the error state and the error state covariance propagation in time, rather than the whole state and state covariance propagation in time. This is mainly beneficial from the computational point of view since it simplifies linearisation of the system equations. Flowchart describing the error state EKF computation is shown in Figure 3 and can be decomposed into a series of steps that describe the actual implementation. As new measurements arrive, state estimate ($\hat{\mathbf{x}}$) and its error covariance matrix (P) are available from the previous time-step (or as initialized during first iteration). This state estimate $\hat{\mathbf{x}}$ is propagated in time using the nonlinear system equations. The continuous time F_c and G_c matrices are evaluated based on the current

value of $\hat{\mathbf{x}}$. Van Loan discretization method is used to obtain discrete forms of F_d and Q_d. Then the error state covariance matrix P is propagated in time. Expected measurements are compared to the incoming ones and their difference is expressed in form of measurement residual $\Delta\mathbf{y}$. Innovation matrix H, expressing the measurement residual as a linear combination of the error state components, is evaluated. Using the a priori estimate of P, H and the variance of the sensors signals expressed as R, the Kalman gain matrix K is computed. The error state $\Delta\mathbf{x}$ is updated using the Kalman gain and the measurement residual; the a posteriori estimate of the error state covariance matrix P, is evaluated as well. Finally, the a priori state estimate $\hat{\mathbf{x}}$ is corrected using the estimated error $\Delta\mathbf{x}$.

Although this EKF cycle can be repeated each time measurements arrive, yet, for performance reasons, we have chosen to group the incoming measurements to the highest frequency measurement, i.e. the IMU data. Hence, each time any non-IMU measurement arrives, it is slightly delayed until the next IMU measurement is available. The maximum possible sampling error caused by this grouping approach is $1/(2 \cdot 90)$s and thus it can be neglected compared to the significantly longer sampling periods of the non-IMU data sources. The update rate of the EKF is then equal to the IMU sampling rate, i.e. 90 Hz.

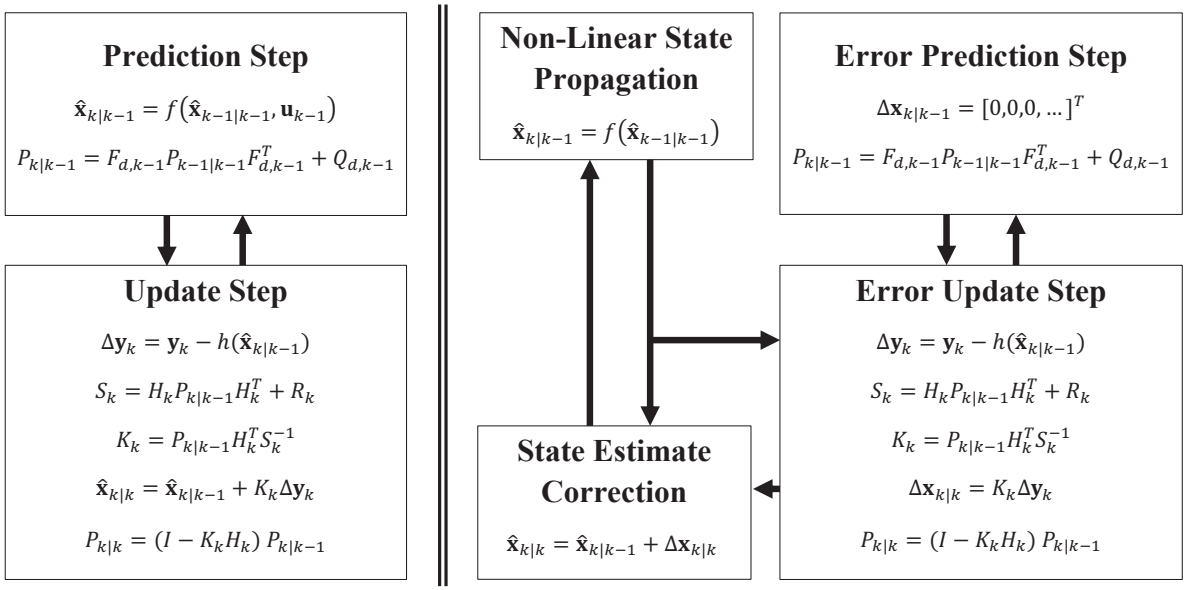


Figure 3: Standard EKF (left) computation flowchart compared to the error state EKF computation flowchart (right): in the error state EKF prediction step, the a priori state is estimated using the non-linear system equation $f()$ and the covariances are estimated using F_d (linearized matrix form of the error state propagation equations). In the update step, measurement residual $\Delta\mathbf{y}$ is obtained by comparing the incoming measurement $\mathbf{y}$ with its predicted counterpart. The residual covariance S and the Kalman gain K are evaluated and used to update the state and covariance matrix to obtain the a posteriori estimates. Note that in the case of the error state EKF, Q_d and H_k couple system noise and measurements with the error state $\Delta\mathbf{x}$ rather than $\hat{\mathbf{x}}$.

4.3 Measurement error model

In general, the measurement vector $\mathbf{y}$ can be described as sum of measurement function $h(\mathbf{x})$ of the state $\mathbf{x}$ and of some random noise $\mathbf{m}$ due to properties of the individual sensors:

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{m} \tag{12}$$

Using the function h, we can predict the measured value based on current knowledge about the system state:

$$\hat{\mathbf{y}} = h(\hat{\mathbf{x}}) \tag{13}$$

There is a difference $\Delta\mathbf{y} = \hat{\mathbf{y}} - \mathbf{y}$ caused by the modeling imperfections in the state estimate as well as by the sensor errors. This difference can be expressed in terms of the error state $\Delta\mathbf{x}$:

$$\begin{aligned} \Delta\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}} &= h(\mathbf{x}) - h(\hat{\mathbf{x}}) + \mathbf{m} \\ &= h(\hat{\mathbf{x}} + \Delta\mathbf{x}) - h(\hat{\mathbf{x}}) + \mathbf{m} \end{aligned} \tag{14}$$

If function h is linear, (14) becomes

$$\Delta\mathbf{y} = h(\Delta\mathbf{x}) + \mathbf{m} \tag{15}$$

Although the condition of linearity is not always met we still can approximate the behavior of h in some close proximity to the current state $\hat{\mathbf{x}}$ by a similar function h', which is linear in elements of $\hat{\mathbf{x}}$ such that

$$h(\hat{\mathbf{x}} + \Delta\mathbf{x}) - h(\hat{\mathbf{x}}) \approx h'(\Delta\mathbf{x})|_{\hat{\mathbf{x}}} = H_{\hat{\mathbf{x}}}\Delta\mathbf{x} \tag{16}$$

where $H_{\hat{\mathbf{x}}}$ is the innovation matrix projecting observed differences in measurements onto the error states.

4.3.1 IMU measurement model

The inertial measurement unit (IMU) is capable of measuring specific force (Savage, 1998) in all three dimensions as well as angular rates. The specific force measurement is a sum of acceleration and gravitational force, but it also contains biases—constant or slowly changing value independent of the actual acting forces—and sensor noise, which is expected to have zero mean normal probability. All the values are measured in the I–frame.

$$\mathbf{y}_{f,I} = \mathbf{f}_I + \mathbf{b}_{f,I} + \mathbf{m}_{f,I} \tag{17}$$

where $\mathbf{y}_{f,I}$ is the measurement, $\mathbf{f}_I$ is the true specific force, $\mathbf{b}_{f,I}$ is sensor bias and $\mathbf{m}_{f,I}$ is sensor noise.

Since the interesting value $\mathbf{y}_{f,I}$ is expressed in the I–frame, we define a constant rotation matrix C_R^I of R–frame to I–frame. Translation between the I– and R–frames does not affect the measured values directly; thus, it is not considered. Since the IMU is placed close to the R–frame origin, we neglect centrifugal force induced by rotation of R–frame and conditioned by non-zero translation between R– and I–frames. Using this rotation matrix, we express the measurement as:

$$\mathbf{y}_{f,I} = C_R^I\mathbf{f}_R + \mathbf{b}_{f,I} + \mathbf{m}_{f,I} \tag{18}$$

where both $\mathbf{f}_R$ and $\mathbf{b}_{f,I}$ are elements of the system state. If we compare the measured value and the expected measurement, we can express the h function, which is—in this case—equal to the h':

$$\begin{aligned} \mathbf{y}_{f,I} - \hat{\mathbf{y}}_{f,I} = \Delta\mathbf{y}_{f,I} &= C_R^I\mathbf{f}_R + \mathbf{b}_{f,I} - C_R^I\hat{\mathbf{f}}_R - \hat{\mathbf{b}}_{f,I} + \mathbf{m}_{f,I} \\ &= C_R^I\Delta\mathbf{f}_R + \Delta\mathbf{b}_{f,I} + \mathbf{m}_{f,I} \end{aligned} \tag{19}$$

and hence can be expressed in $H_{\hat{\mathbf{x}}}\Delta\mathbf{x}$ form as

$$\Delta\mathbf{y}_{f,I} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & C_R^I & \emptyset_3 & I \end{bmatrix}\Delta\mathbf{x} + \mathbf{m}_{f,I} \tag{20}$$

where the error state $\Delta\mathbf{x}$ was defined in (1).

The angular rate measurement is treated identically; the output of the sensor is

$$\mathbf{y}_{\omega,I} = \omega_I + \mathbf{b}_{\omega,I} + \mathbf{m}_{\omega,I} \tag{21}$$

where ω_I is angular rate, $\mathbf{b}_{\omega,I}$ is sensor bias and $\mathbf{m}_{\omega,I}$ is sensor noise.

Similarly, the measurement residual is obtained:

$$\mathbf{y}_{\omega,I} - \hat{\mathbf{y}}_{\omega,I} = \Delta\mathbf{y}_{\omega,I} = C_R^I\Delta\omega_R + \Delta\mathbf{b}_{\omega,I} + \mathbf{m}_{\omega,I} \tag{22}$$

which can be expressed in the matrix form

$$\Delta\mathbf{y}_{\omega,I} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & \emptyset_3 & C_R^I & \emptyset_3 & \emptyset_3 & I \end{bmatrix}\Delta\mathbf{x} + \mathbf{m}_{\omega,I} \tag{23}$$

4.3.2 Odometry measurement model

Our platform is equipped with caterpillar tracks and therefore, steering is realized by setting different velocities to each of the tracks (*skid-steering*). The velocities are measured by incremental optical angle sensors at 15 Hz. Originally, we implemented a complex model introduced in (Endo et al., 2007), which exploits angular rate measurements to model the slippage to further improve the odometry precision. However, with respect to our sensors, no improvement was observed. Moreover, since the slippage is inherently corrected via the proposed data fusion, we can neglect it in the odometry model, assuming only a very simple but sufficient model:

$$v_{O,x} = \frac{v_r + v_l}{2} \tag{24}$$

where $v_{O,x}$ is the forward velocity, v_l and v_r are track velocities measured by incremental optical sensors—the velocities in the lateral and vertical axes are set to zero. Since the robot position is obtained by integrating velocity expressed in R–frame, we define a rotation matrix C_R^O:

$$\mathbf{v}_O = C_R^O \mathbf{v}_R \tag{25}$$

which expresses the $\mathbf{v}_R$ in the O–frame.

During experimental evaluation, we observed a minor misalignment between these two frames, which can be described as rotation about the lateral axis by approximately 1 degree. Although relatively small, this rotation caused the position estimate in the vertical axis to grow at constant rate while the robot was moving forward. To compensate for this effect, we handle the C_R^O as constant—its value was obtained by means of calibration. The measurement equation is then as follows:

$$\mathbf{y}_{v,O} = C_R^O \mathbf{v}_R + \mathbf{m}_{v,O} \tag{26}$$

where $\mathbf{y}_{v,O}$ is linear velocity measured by the track odometry, expressed in O–frame. Since this relation is linear, the measurement innovation is

$$\begin{aligned} \mathbf{y}_{v,O} - \hat{\mathbf{y}}_{v,O} = \Delta\mathbf{y}_{v,O} = & \\ &= C_R^O \mathbf{v}_R - C_R^O \hat{\mathbf{v}}_R + \mathbf{m}_{v,O} \\ &= C_R^O \Delta\mathbf{v}_R + \mathbf{m}_{v,O} \end{aligned} \tag{27}$$

and expressed in the matrix form

$$\Delta\mathbf{y}_{v,O} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & C_R^O & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{v,O} \tag{28}$$

4.3.3 ICP-based localization measurement model

The ICP algorithm is used to estimate translation and rotation between each new incoming laser scan of the robot surroundings and a metric map created from the previously registered laser scans. In course of our work, three approaches processing the output of the ICP were proposed and tested. The first approach treats the ICP-based localization as movement in the R–frame in between two consecutive laser scans in form of a position increment (the *incremental position approach*). The idea of measurements expressed in a form of some $\Delta\mathbf{p}$ can be, for example, found in (Ma et al., 2012). In our case, the increment is obtained as:

$$\Delta\mathbf{p}_{R,ICP,i} = C_{(\mathbf{q}^R_{N,ICP,i-1})}(\mathbf{p}_{N,ICP,i} - \mathbf{p}_{N,ICP,i-1}) \tag{29}$$

where both the position $\mathbf{p}_{N,ICP}$ and attitude $\mathbf{q}^R_{N,ICP}$ are outputs of the ICP algorithm. The increment $\Delta\mathbf{p}_{R,ICP,i}$ is added to the position estimated by the whole fusion algorithm at time-step $i-1$ to be used as a direct measurement of position. The same idea is applied in the case of attitude (an increment in attitude is extracted by means of quaternion algebra). The purpose is to overcome the ICP world frame drift. However,

it is impossible to correctly discretize the system equations respecting the laser scan sampling frequency ($\frac{1}{3}$ Hz). Also, the assumption of measurements being independent is violated by utilizing a previously estimated state to create a new measurement. Thus, corrections that propagate to the system state from this measurement tend to be inaccurate.

The second approach treats the ICP output as velocity in the R–frame (the *velocity approach*). We consider it a state-of-the-art practice utilized, for example, by (Almeida and Santos, 2013). The velocity is expressed in the N–frame first:

$$\mathbf{v}_{N,ICP} = \frac{\mathbf{p}_{N,ICP,i} - \mathbf{p}_{N,ICP,i-1}}{t(i) - t(i-1)} \tag{30}$$

where $t()$ is time corresponding to a time-step i. To express the velocity in the R–frame:

$$\mathbf{v}_{R,ICP}(t) = C_{(\mathbf{q}^R_{R',ICP}(t) \otimes \mathbf{q}^{R'}_{N,ICP,i-1})} \mathbf{v}_{N,ICP} \tag{31}$$

it is necessary to interpolate the attitude between $\mathbf{q}^R_{N,ICP,i-1}$ and $\mathbf{q}^R_{N,ICP,i}$ in order to obtain the increment $\mathbf{q}^R_{R',ICP}(t)$. Angular velocity is assumed to be constant between the two laser scans. The velocity $\mathbf{v}_{R,ICP}$ and the constant angular velocity obtained from the interpolation can be directly used as measurements which are independent of the estimated state and because of the interpolation, they can be generated with arbitrary frequency and thus, there is no problem with discretization (compared to the previous approach). However, this approach expects the robot to move in a line between the two ICP scans. This is a too strong assumption and also a major drawback of this approach that results in incorrect trajectory estimates.

Therefore, we propose the third approach, the *trajectory approach*, which overcomes the assumption of the *velocity approach* by (sub-optimal) use of the estimated states in order to approximate possible behavior of the system between each two consecutive ICP scans. This *trajectory approach* proved to be the best for pre-processing the output of the ICP algorithm; for details see Section 5.4.5.

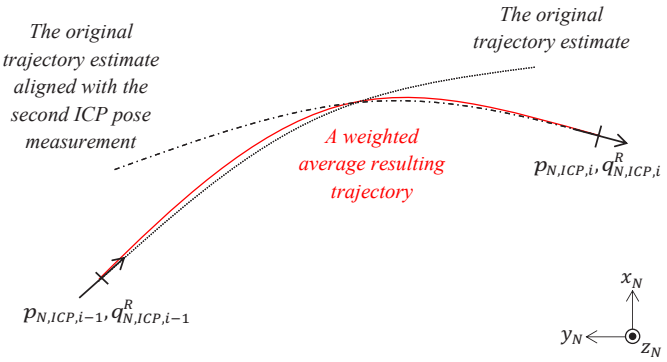


Figure 4: The principle of *trajectory approach*: when the new ICP measurement arrives (time-step i), trajectory estimate based on measurements other than ICP (black dotted line) is duplicated and aligned with the incoming ICP measurement (black dashed line) and weighted average (red solid line) of these two trajectories is computed.

The *trajectory approach* assumes that the first estimate of the trajectory (without the ICP measurement) is locally very similar to the true trajectory (up to the effects of drift). Thus, when a new ICP measurement arrives the trajectory estimated since the previous ICP measurement is stored to be used as the best guess around the previous ICP pose. The ICP poses at time-steps i and $i-1$ are aligned with the N–frame so the ICP pose at time-step $i-1$ coincides with the first pose of the stored trajectory. This way the ICP world frame drift is suppressed. Then, the stored trajectory is duplicated and aligned with the new ICP pose to serve as the best guess around the new ICP pose; see Figure 4. The resulting trajectory is obtained

as weighted average of the original and the duplicated trajectories:

$$\hat{\mathbf{p}}_{N,weighted,k} = \hat{\mathbf{p}}_{N,k} w_k + \hat{\mathbf{p}}'_{N,k} w'_k \tag{32}$$

where $\hat{\mathbf{p}}_{N,k}$ are points of the original trajectory (black dotted line in Figure 4), $\hat{\mathbf{p}}'_{N,k}$ are points of the realigned duplicated trajectory (black dashed line in Figure 4) and w_k, w'_k are weights—linear functions of time equal to 1 at time-step of associated ICP measurement and equal to 0 at time-step of the other ICP measurement. The resulting trajectory is used to generate the velocity measurements in the N–frame as follows:

$$\mathbf{v}_{N,weighted,k} = \frac{\mathbf{p}_{N,weighted,k} - \mathbf{p}_{N,weighted,k-1}}{t(k) - t(k-1)} \tag{33}$$

where $t(k)$ and $t(k-1)$ are time-steps of poses of the resulting weighted trajectory. The k denotes indexing of the fusion algorithm high-frequency samples. Velocities can be expressed in R–frame using the attitude estimates $\hat{\mathbf{q}}^R_{N,k}$:

$$\mathbf{v}_{R,weighted,k} = C_{(\hat{\mathbf{q}}^R_{N,k})} \mathbf{v}_{N,weighted,k} \tag{34}$$

and can be used directly as measurement, whose projection onto the error state vector yields:

$$\Delta\mathbf{y}_{v,weighted} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{v,weighted} \tag{35}$$

The velocity expressed in R–frame can be used this way as measurement, but its values for the time period between two consecutive ICP outputs are known only *after* the second ICP measurement arrives. Thus it is necessary to recompute state estimates for this whole time period (typically in length of 300 IMU samples), including the new velocity measurements.

To process the attitude information provided as the ICP output, we use a simple incremental approach such that the drift of the ICP world frame with respect to the N–frame is suppressed. To achieve this, we extract only the increment in attitude between two consecutive ICP poses:

$$\mathbf{q}^R_{N,ICP,i} = \mathbf{q}^R_{R',ICP} \otimes \mathbf{q}^{R'}_{N,ICP,i-1} \tag{36}$$

$$\mathbf{q}^R_{R',ICP} = \mathbf{q}^R_{N,ICP,i} \otimes \left(\mathbf{q}^{R'}_{N,ICP,i-1}\right)^{-1} \tag{37}$$

where $\mathbf{q}^R_{R',ICP}$ is rotation that occurred between two consecutive ICP measurements $\mathbf{q}^{R'}_{N,ICP,i-1}$ and $\mathbf{q}^R_{N,ICP,i}$. We apply this rotation to the attitude state estimated at time-step $k' \equiv i-1$:

$$\mathbf{y}_{q,ICP} = \mathbf{q}^R_{R',ICP} \otimes \hat{\mathbf{q}}^R_{N,k'} \tag{38}$$

To express the measurement residual, we define the following error quaternion:

$$\delta\mathbf{q}_{ICP,i} = \hat{\mathbf{q}}^R_{N,k} \otimes (\mathbf{y}_{q,ICP})^{-1} \tag{39}$$

where $\hat{\mathbf{q}}^R_{N,k}$ is the attitude estimated at time-step $k \equiv i$. We express this residual rotation by means of rotation vector $\delta\theta_{ICP,i}$

$$\delta\theta_{ICP,i} = 2\delta\vec{\mathbf{q}}_{ICP,i} \tag{40}$$

which can be projected onto the error state as

$$\Delta\mathbf{y}_{\delta\theta,ICP} = \begin{bmatrix} \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 \end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{\delta\theta,ICP} \tag{41}$$

Although the ICP is very accurate in measuring translation between consecutive measurements, the attitude measurement is not as precise. Noise introduced in the pitch angle can cause wrong velocity estimates expressed in R–frame, resulting in problem described as *climbing robot*—the system tends to slowly drift in the vertical axis. Since the output of the *trajectory approach* is velocity $\mathbf{v}_{R,weighted,i}$, applying a constraint assuming only planar motion in the R frame is fully justified, easy to implement and resolves this issue.

4.3.4 Visual odometry measurement model

As explained in Section 3.5, the visual odometry (VO) is an algorithm for estimating translation and rotation of a camera body based on images recorded by the camera. The current implementation of the data fusion utilizes only the rotation part of the motion estimated by the VO, since it is not affected by the scale. The set of 3D landmarks maintained by the VO is not in any way processed by the fusion algorithm—it is used by the VO to improve its attitude estimates internally. Similarly, the bundle adjustment ensures more consistent measurements, yet still, it does not enter the data fusion models.[3] The way we incorporate the VO measurements is equivalent to the ICP *trajectory approach*, however, reduced only to the incremental processing of the attitude measurements. This way, the whole VO processing block can easily be replaced by an alternative (for example by stereo vision based VO), provided the output—the estimated rotation—is available in the same way. The motivation is to have the VO measurement model independent on the VO internal implementation details. The implementation of the VO attitude aiding is identical to the ICP attitude aiding; the attitude increment is extracted and used to construct a new measurement $\mathbf{y}_{q,VO}$:

$$\mathbf{q}^R_{N,VO,i} = \mathbf{q}^R_{R',VO} \otimes \mathbf{q}^{R'}_{N,VO,i-1} \tag{42}$$

$$\mathbf{q}^R_{R',VO} = \mathbf{q}^R_{N,VO,i} \otimes \left(\mathbf{q}^{R'}_{N,VO,i-1}\right)^{-1} \tag{43}$$

where $\mathbf{q}^R_{R',VO}$ is rotation that occurred between two consecutive VO measurements $\mathbf{q}^{R'}_{N,VO,i-1}$ and $\mathbf{q}^R_{N,VO,i}$. We apply this rotation to the attitude state estimated at time-step $k' \equiv i-1$:

$$\mathbf{y}_{q,VO} = \mathbf{q}^R_{R',VO} \otimes \hat{\mathbf{q}}^R_{N,k'} \tag{44}$$

Then, the measurement residual is expressed as error quaternion:

$$\delta\mathbf{q}_{VO,i} = \hat{\mathbf{q}}^R_{N,k} \otimes \left(\mathbf{y}_{q,VO}\right)^{-1} \tag{45}$$

where $\hat{\mathbf{q}}^R_{N,k}$ is the attitude estimated at time-step $k \equiv i$. We express this residual rotation by means of rotation vector $\delta\theta_{VO,i}$

$$\delta\theta_{VO,i} = 2\delta\vec{\mathbf{q}}_{VO,i} \tag{46}$$

which can be projected onto the error state as

$$\Delta\mathbf{y}_{\delta\theta,VO} = \begin{bmatrix}\emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3\end{bmatrix}\Delta\mathbf{x} + \mathbf{m}_{\delta\theta,VO} \tag{47}$$

where $\mathbf{m}_{\delta\theta,VO}$ is the VO attitude measurement noise.

5 Experimental evaluation

Our evaluation procedure involves several different tests. First, we describe our evaluation methodology in Section 5.1. It covers obtaining ground truth positioning measurements for both in- and outdoors. Then we present and discuss our field experiments with the global behavior of our state estimation (Section 5.2). We also show two examples of typical behavior of the filter in order to give more insight on its general characteristics (Section 5.3). We take advantage of them to explain the importance of the *trajectory approach*, compared to more standard measurement models. Finally, we analyze the behavior of the filter under failure case scenarios involving partial or full outage of each sensory modality (Section 5.4).

[3] The same idea applies for the ICP-based localization: although it builds an internal map, this map is independent from our localization estimates. This would not be the case in a SLAM approach with integrated loop closures.

5.1 Evaluation metrics

In order to validate the results of our fusion system, we need accurate measurements of part of our system states to confront with the proposed filter. For indoor measurements, we use a Vicon motion capture system with nine cameras covering more than $20\,\mathrm{m}^2$ and giving a few millimeter accuracy at 100 Hz.

For external tracking, we use a theodolite from Leica Geosystems, namely the Total Station TS15; see Figure 5 (left). It can track a reflective prism to measure its position continuously at an average frequency of 7.5 Hz. The position precision of the theodolite is 3 mm in continuous mode. However, this system cannot measure the orientation of the robot. Moreover, the position measured is that of the prism and not directly of the robot, therefore we calibrated the position of the prism with respect to the robot body using the theodolite and precise blueprints. However, the position of the robot cannot be recovered from the position of the prism without the information about orientation. That explains why, in the validations below, we do not compare the position of the robot but the position of the prism from the theodolite and reconstructed from the states of our filter. With these ground-truth measurements, we use different metrics for evaluation. First, we simply plot the error as a function of time. More precisely, we consider *position error*, *velocity error*, and *attitude error* and we compute it by taking the norm of the difference between the prediction made by our filter and the reference value.

Since this metric shows how the errors evolve over time, a more condensed measure is needed to summarize and compare the results of different versions of the filter. Therefore, we use the *final position error* expressed as a percentage of the total trajectory length:

$$e_{rel} = \frac{||\mathbf{p}_l - \mathbf{p}_{ref,l}||}{\text{distance travelled}} \tag{48}$$

where l is the index of the last position sample $\mathbf{p}_l$ with the corresponding reference position $\mathbf{p}_{ref,l}$.

While this metric is convenient and widely used in the literature, it is however representative only of the end point error regardless of the intermediary results. This can be misleading for long trajectories in confined environment as the end-point might be close to the ground truth by chance. This is why we introduce, as a complement, the *average position error*:

$$e_{avg}(l) = \frac{\sum_{i=1}^{l} ||\mathbf{p}_i - \mathbf{p}_{ref,i}||}{l} \tag{49}$$

where $1 \leq l \leq$ *total number of samples*. To improve legibility of this metric in plots, we express the e_{avg} as a function of time

$$e'_{avg}(t) = e_{avg}(l(t)) \tag{50}$$

where $l(t)$ simply maps time t to the corresponding sample l.

5.2 Performance overview of the proposed data fusion

With these metrics, we can actually evaluate the performance of our system in a quantitative way. We divided the tests into indoor and outdoor experiments.

5.2.1 Indoor performance

For the indoor tests, we replicated semi-structured environment found in USAR environments, including ramps, boxes, a catwalk, a small passage, etc. Figure 5 (right) shows a picture of part of the environment. Due to limitations of our motion capture set-up, this testing environment is not as large as typical indoor USAR environments. Nevertheless, it features most of the complex characteristics that make state estimation challenging in such an environment.

Figure 5: The experimental setup with the Leica reference theodolite for obtaining ground truth trajectory (left). Part of the 3D semi-structured environment for indoor test with motion capture ground truth (right).

For this evaluation, we recorded approximately 2.4 km of indoor data with ground truth; 28 runs represent standard conditions (765 m in total), 36 runs represent failure cases of different sensory modalities induced artificially (1613 m in total). Table 2 presents the results of each combination of sensory modalities for the 28 standard conditions runs; the failure scenarios are analyzed in Section 5.4 separately.

The sensory modalities combinations can be divided into two groups by including or excluding the ICP modality; these two groups differ by the magnitude of the final position error. From this fact, we conclude that the main source of error is slippage of the caterpillar tracks—the VO modality in our fusion system corrects only the attitude of the robot. Also, the results confirmed sensitivity to erroneous attitude measurements originating from the sensory modalities. In this instance, VO has slightly worsen the median of the final position error—the indoor experiments are not long enough to make the difference between drift rates of the bare IMU+OD combination and possible VO errors that originate from incorrect pairing of image features. Nevertheless, the results are not significantly different.[4] A significant improvement is brought with the ICP modality, which compensates the tracks slippage and reduces the resulting median of the final position errors by 50% (approximately). As expected during the filter design, fusing all sensory modalities yields the best result (not significantly different that without VO), with a median of 1.2% final position error; the occasional VO attitude measurement errors are diminished by the ICP modality attitude measurement (and vice versa).

5.2.2 Outdoor performance

We ran outdoor tests in various different environments; namely a street canyon and a urban park with trees and stairs in Zurich. Figure 6 shows pictures of the environments.

In those environments we recorded in total approximately 2 km, with ground truth available for 1.6 km, the rest were returns from the experimental areas. These 1.6 km are split into 10 runs and, as for the indoor experiments, Table 3) presents the results of each combination of sensory modalities for each run.

Contrary to the indoor experiments, combining all four modalities does not improve precision of localization compared to ICP, IMU and odometry fusion (the fusion of all is significantly worse than ICP, IMU and odometry only). Although some runs show improvement while combining all the sensory modalities (runs 7, 9 and 10) or are at least comparable with the best result 0.4|**0.6**|1.2 (runs 4, 5 and 6), there are several experiments, where VO failed due to the specificities of the environments. Such failures result in erroneous attitude estimates significantly exceeding expected VO measurement noise and compromising localization accuracy of the fusion algorithm. The reasons for failures are described in the Section 5.4 together with other failure cases. Since we did not artificially induce these VO failures as we did in the case of the indoor experiments, we do not exclude these runs from the performance evaluation in Table 3—we consider

[4] All statistical signicance results are assessed using the Wilcoxon signed-rank test with $p < 0.05$ testing whether the median of correlated samples is different.

Exp.	Distance traveled [m]	Exp. duration [s]	Final position error in % of the distance travelled			
			OD, IMU	OD, IMU, VO	OD, IMU, ICP	OD, IMU, ICP, VO
1	47.42	254	2.17	2.30	1.71	0.79
2	36.52	186	1.99	2.21	0.36	0.14
3	48.74	244	3.15	2.63	0.50	0.18
4	29.40	237	2.22	2.06	0.42	0.45
5	82.10	585	2.51	2.24	0.90	0.71
6	74.64	452	2.05	3.64	0.98	1.24
7	74.65	387	1.70	1.72	2.28	0.58
8	30.57	194	1.98	3.42	1.59	2.29
9	26.58	287	2.67	2.23	1.90	1.19
10	26.57	236	1.53	3.94	0.77	2.11
11	26.96	208	1.25	1.20	0.95	0.66
12	29.13	211	1.27	1.29	0.88	0.87
13	26.35	180	1.37	1.25	0.94	0.77
14	40.23	240	6.58	6.70	0.88	0.99
15	21.01	167	5.26	5.27	0.61	0.57
16	19.04	209	5.94	5.95	0.55	0.60
17	10.95	405	3.44	2.89	2.15	2.05
18	8.65	238	2.87	2.77	1.36	1.38
19	9.36	284	4.14	3.91	1.83	1.85
20	9.02	282	2.90	3.36	2.73	2.65
21	10.82	308	3.79	3.23	1.43	1.41
22	9.45	237	5.36	5.45	2.66	2.68
23	12.75	204	2.65	2.84	2.66	1.79
24	7.81	179	1.58	1.83	2.82	3.06
25	10.85	165	3.85	4.14	3.25	2.17
26	10.83	163	2.36	1.84	0.62	0.68
27	12.79	237	15.42	14.95	2.48	2.53
28	12.07	239	28.42	27.07	2.89	2.98
Lower quartile|**Median**|Upper quartile			2.0|**2.7**|4.0	2.1|**2.9**|4.0	0.8|**1.4**|2.4	0.7|**1.2**|2.1

Table 2: Comparison of combinations of different modalities evaluated on indoor experiments performed under standard conditions with the Vicon system providing ground truth in position and attitude. Final position error expressed in percents of the total distances traveled was chosen as metric for each experiment; the total distance of the 28 experiments was 765 m, including traversing obstacles.

such environments standard for USAR. Moreover, we treat them as another proof of the fusion algorithm sensitivity to erroneous attitude measurement originating both from VO and ICP modalities and address them in the conclusions and future work.

5.3 In-depth analysis of the examples of performance

In order to have more insight on the characteristics of the filter, we selected some trajectories and show more information than just the final position error metric.

5.3.1 Example of data fusion performance in indoor environment

In this example we address the caterpillar tracks slippage when traversing an obstacle (Figure 7). Since we are looking forward to USAR missions, such environment with conditions inducing high slippage can be expected, e.g. collapsed buildings full of debris and dust that impairs traction on smooth surfaces such as exposed concrete walls or floors, mass traffic accidents with oil spills, etc. The Vicon system was used to

Figure 6: Pictures of the outdoor environments in Zurich. Left: street canyon, right: urban park.

Experiment	Distance traveled [m]	Exp. duration [s]	Final position error in % of the distance travelled			
			OD, IMU	OD, IMU, VO	OD, IMU, ICP	OD, IMU, ICP, VO
1: basement 1	120.62	825	2.08	26.61	1.83	17.84
2: basement 2	175.67	853	1.37	12.53	2.42	5.91
3: hallway straight	159.42	738	1.10	20.48	0.43	12.22
4: street 1	135.18	584	2.78	0.72	0.24	0.62
5: street 2	259.86	992	9.74	0.80	0.26	0.80
6: park big loop	145.31	918	2.65	2.66	1.03	1.76
7: park small loop	88.20	601	1.94	1.60	1.25	0.97
8: park straight	99.29	560	1.20	20.18	0.62	11.50
9: 2 floors	238.28	1010	9.10	0.62	0.58	0.43
10: 2 floors opposite	203.23	1107	3.23	6.79	0.51	0.42
Lower quartile|**Median**|Upper quartile			1.4|**2.4**|3.2	0.8|**4.7**|20.2	0.4|**0.6**|1.2	0.6|**1.4**|11.5

Table 3: Comparison of combinations of different modalities evaluated on outdoor experiments performed under standard conditions with the Leica system providing ground truth in position.

obtain precise position and orientation ground truth for computing the *average position error* development n time.

When traversing a slippery surface, any track odometry inevitably fails with the tracks moving with significantly diminishing traction. For this reason, trajectory and state estimates resulting from the IMU+OD fusion showed unacceptable error growth; see Figure 8. The robot was operated to attempt climbing up the yellow slippery board (Figure 7), which deteriorated the traction to the point the robot was sliding back down with each attempt to steer. Because of the slippage, it failed to reach the top. Then, it was driven around the structure and up, to further slowly slip down the slope backwards, with the tracks moving forward to spoil traction. The effect of the slippage on the OD is apparent from the purple line in Figure 8. The corresponding average position error of the bare combination of IMU+OD starts to build up as soon as the robot enters the slippery slope. At 75 seconds, the IMU+OD has already an error of 0.5 m and finishes at 200 sec at an error of 4.4 m (outside Figure 8). Without exteroceptive modalities this problem is unsolvable and as expected, including these modalities significantly improves the localization accuracy; the final average position error is only 0.14 m for the IMU+OD+VO+ICP combination. The resulting state estimates for combination of all modalities are shown in Figure 9 and 10. Figure 9 depicts position estimates (the upper left quarter) with the reference values. The difference between the estimate and the reference is plotted in the bottom left quarter; similarly, the right half of the figure displays the velocity estimate. In the left part of the Figure 10, the attitude estimate expressed in Euler angles is shown with its error compared to the Vicon reference. The right part of this figure demonstrates estimation of the sensor biases, which are part of

the system state. Note, that the biases in angular rates are initialized to values obtained as the mean of angular rate samples measured when the robot stays stationary before each experiment—short self-calibration. Concluded, adding the exteroceptive sensor modalities—as proposed in our filter design—compensated the effect caused by high slippage shown in this example as shown by the shape of trajectories and the average position error.

Figure 7: The 3D structure for testing of obstacle traversability shown a metric map created by ICP.

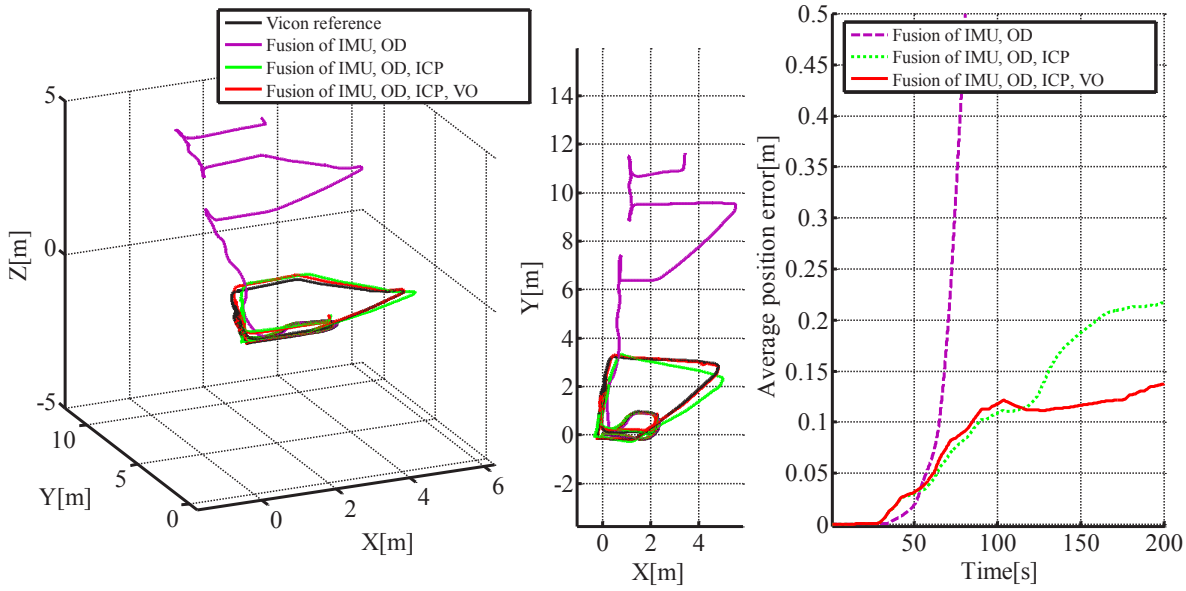


Figure 8: Trajectories obtained by fusing different combinations of modalities during the indoor experiment testing obstacle (depicted in Figure 7) traversability under high slippage (left, middle); development of the average position error (right).

5.3.2 Example of data fusion performance in outdoor environment

This outdoor experiment took place on the Clausiusstrasse street (nearby ETH in Zurich) (Figure 11) and the purpose was testing the exteroceptive modalities (the ICP and the VO) in open urban space. In this standard setting, both the ICP and the VO are expected to perform reasonably well, though the ICP—compared to a closed room—is missing a significant amount of spatial information (laser range is limited approximately to 50 meters, no ceiling etc.). The Leica theodolite was used to obtain the ground truth position during this experiment (Figure 5).

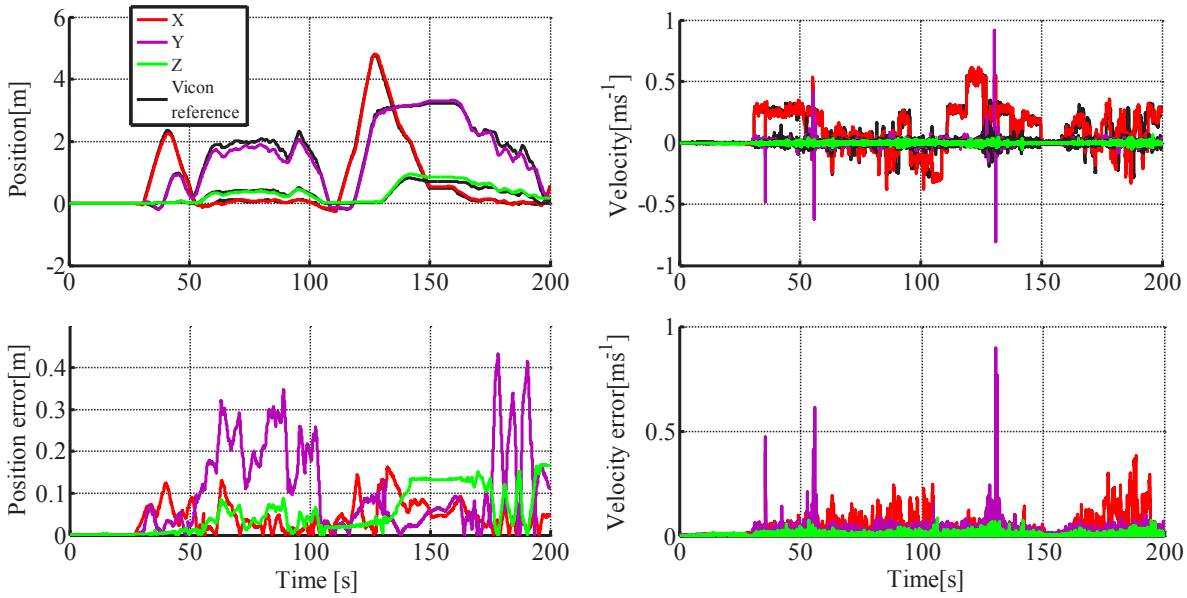


Figure 9: The corrected position (top left) and velocity estimates (top right) for the IMU+OD+ICP+VO combination corresponding to the trajectory in Figure 8 (testing obstacle traversability). Errors in position and velocity are obtained as norm of difference between the Vicon reference and the corresponding state at each time-step (bottom left, bottom right). The Vicon reference for both position and velocity is shown in black.

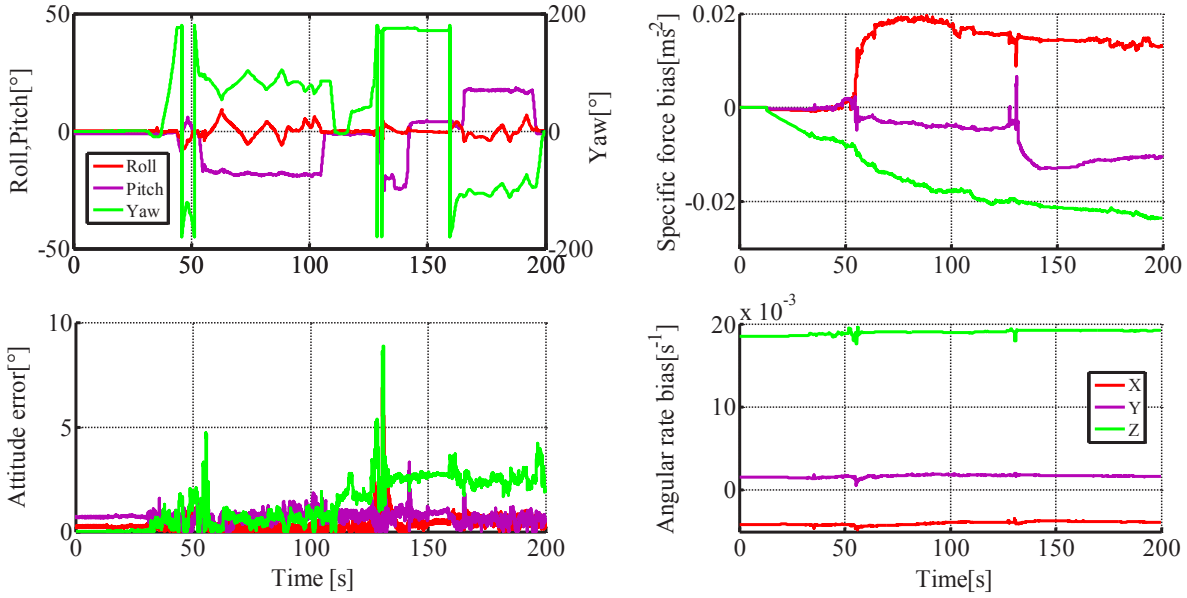


Figure 10: The corrected attitude estimates (top left) for the full multi-modal combination IMU+OD+ICP+VO corresponding to the trajectory shown in Figure 8 (testing obstacle traversability). Errors in attitude are obtained as the difference between the Vicon reference and the corresponding state at each time-step (bottom left). Estimated biases for the specific forces (top right) and angular rates (bottom right).

Figure 11: An example of trajectory driven by the robot over the Clausiusstrasse street.

The results are shown in Figures 12 and 13, demonstrating the improvement of performance when including more modalities up to the full setup. The basic dead-reckoning combination (IMU+OD) showed clearly drift in the yaw angle caused by accumulating error due to angular rate sensor noise integration (see the purple trajectory in the left part of Figure 12). By including the VO attitude measurements (resulting in IMU+OD+VO) the drift was compensated. Though the VO is not in fact completely drift-free, the performance is clearly better than the angular rate integration—it is rather the scale of the trajectory that matters. The IMU+OD+VO modality combination suffered from inaccurate track odometry velocity measurements (the green line in Figure 12), but this problem was resolved by including the ICP modality into the fusion scheme. The IMU+OD+ICP+VO combination proved to provide the best results; see the average position error plot in Figure 12 (right). The attitude estimates and estimates of the sensor biases are shown in Figure 14.

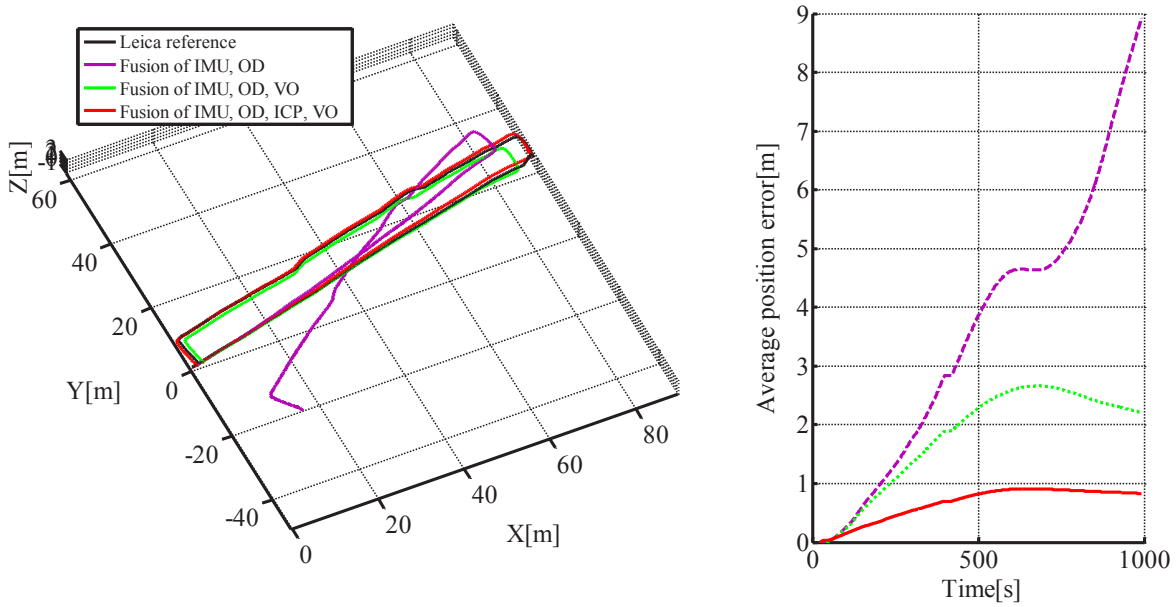


Figure 12: Trajectories obtained by fusing different combinations of modalities during the outdoor experiment with Leica reference system (left) and the corresponding average position error in time (right).

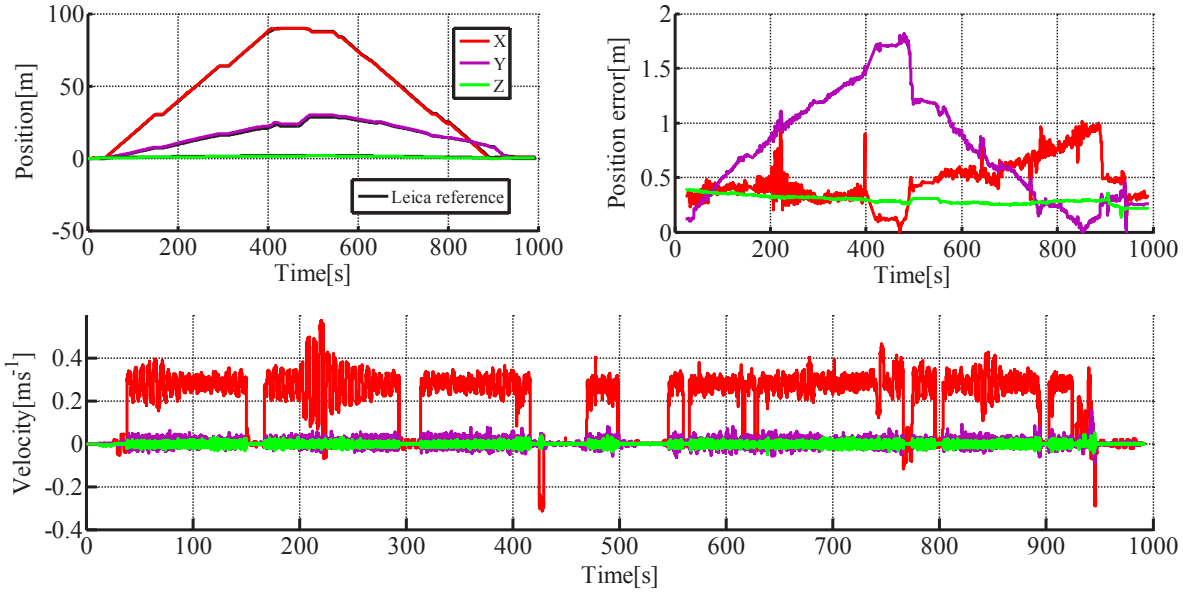


Figure 13: The position and velocity estimates (top left and bottom respectively) for the IMU+OD+ICP+VO combination corresponding to the outdoor trajectory in Figure 12; errors in position obtained as norm of differences between the Leica reference and the corresponding state at each time-step (top right).

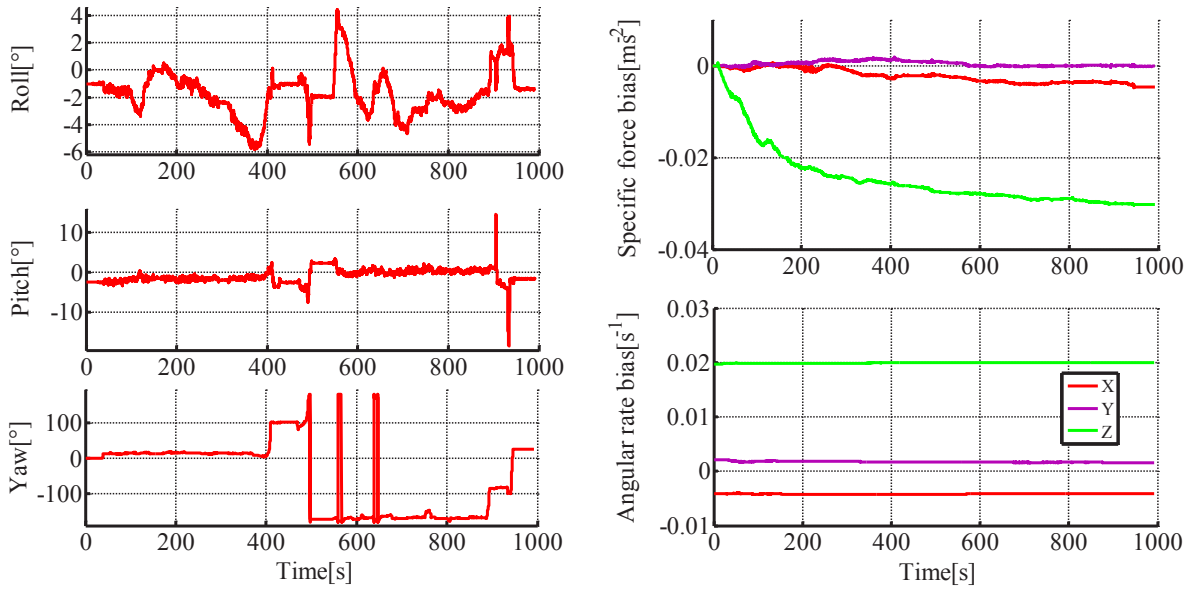


Figure 14: The attitude estimates (left) for the IMU+OD+ICP+VO combination corresponding to the outdoor trajectory in Figure 12; biases estimated for the specific forces (top right) and angular rates (bottom right).

5.3.3 Evaluation of the measurement model

We claim that a standard measurement model—as usually used for measurements coming at comparable frequency—is not well suitable for measurements with significant differences in sampling frequencies as well

Model	Indoor		Outdoor	
	e_{rel}	e_{avg}	e_{rel}	e_{avg}
incremental position	0.4|**0.7**|1.2	0.1|**0.1**|0.2	0.8|**1.5**|11.0	0.7|**2.4**|6.1
velocity	1.0|**1.3**|2.3	0.1|**0.1**|0.3	0.9|**1.8**|12.2	0.8|**2.5**|6.1
trajectory	0.7|**1.2**|2.1	0.0|**0.1**|0.2	0.6|**1.4**|11.5	0.6|**2.2**|6.1

Table 4: Comparison of the different measurement models; for each model we show the lower|**median**|higher quartile statistics of the relative and average metrics. The average metric e_{avg} is evaluated for the last sample of each experiment, see (49). We distinguish the in- and outdoor environments.

as in values, which correspond to the same state observed. This is crucial, when the difference in states obtained from the IMU or the OD at high frequency is very large compared to the measurements provided by the ICP or the VO sensory modalities at relatively low frequency—such as in case of high slippage.

Table 4) shows the overall comparison of the three measurement models we evaluated for fusing the ICP and the VO sensory modalities in the filter. Figure 15 presents a typical example of trajectory reconstructed by all the three measurement approaches we introduced in Section 4.3.3. The *velocity approach*—the state-of-the-art practice—that considers those information as relative measurements, is the least precise, with the highest average position error; see Figure 15 (right). This is due to the *corner cutting* behavior emphasized in Figure 15 (middle). The *incremental position approach* performs reasonably well in indoor environments, which are well -conditioned for the ICP and the VO sensory modalities—especially the ICP algorithm is really precise as there are enough features to unambiguously fix all degrees of freedom. On the other hand, in larger environments with less constraints (expected for USAR), the *trajectory approach* allows the IMU and the OD information to better correct the drift of the ICP and the VO sensory modalities.

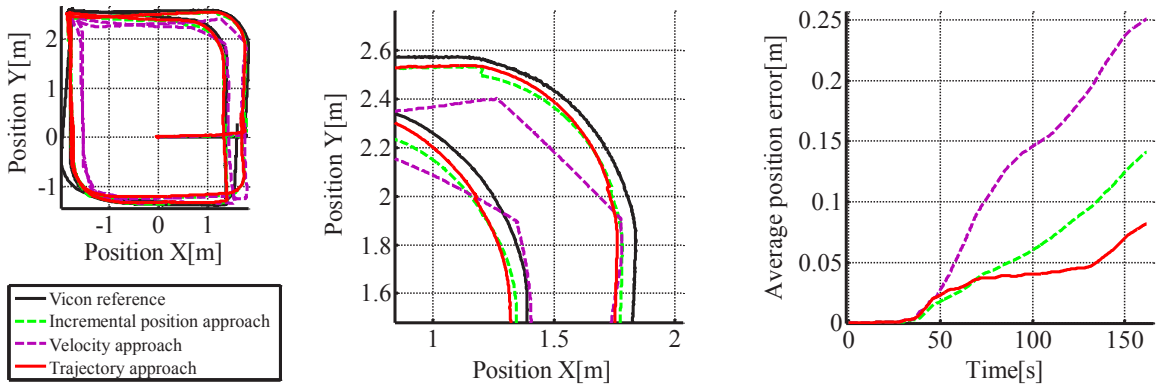


Figure 15: Comparison of effects of the three different ICP aiding approaches on the estimated trajectory (left, middle) and on the average position error (right). Note the *corner cutting* effect of the *velocity approach*.

5.4 Failure case analysis

As seen in the previous sections, there are plenty of occasions in USAR environments for which the generic assumptions of the EKF are not valid. The most frequent example is track slippage that violates the assumption of Gaussian observation centered on the actual value.

Our failure case analysis reviews each sensory modality involved in the filter to see how the resulting estimate degrades with partial outage of the modality. IMUs are not subject to much partial failure other than bias and noise, that are already accounted for in our filter.

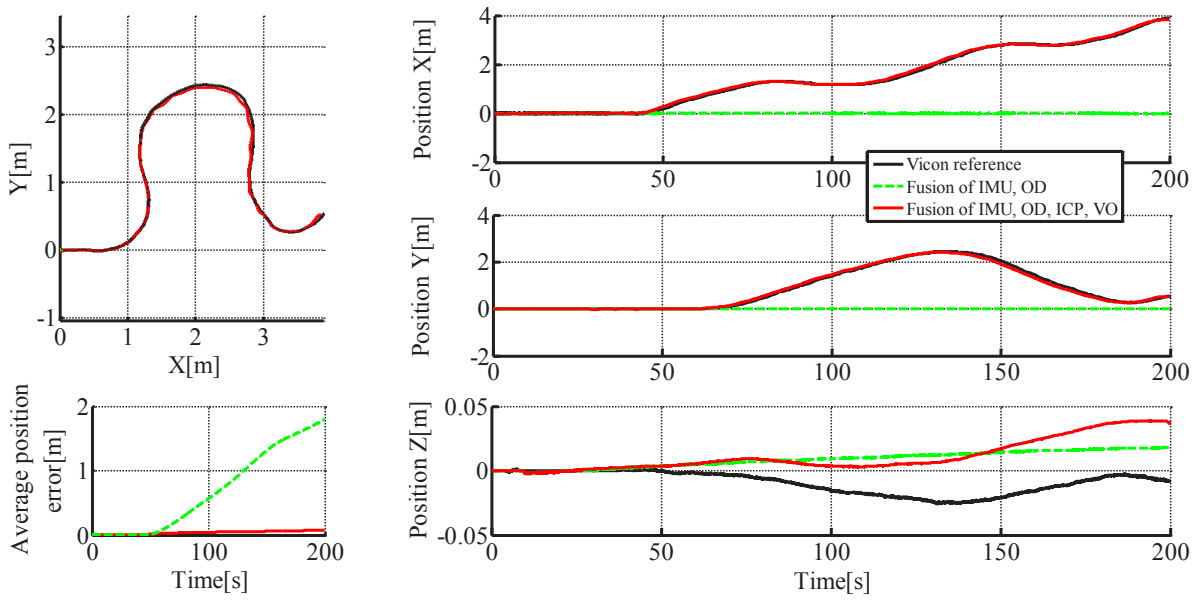


Figure 16: Test trajectory for robot slippage. Black line: ground-truth; red solid line: state estimate with all four modalities; green dashed line: IMU and odometry fusion. Top left: top view of the trajectory; bottom left: average error as a function of time; top, middle, bottom right: evolution of x, y, z coordinate.

5.4.1 Robot slippage and sliding

A typical failure case of the odometry modality is significant slippage. Small slippage occurs routinely when turning skid-steer robots and is usually accounted for by the uncertainty in the odometry model. However, on surfaces like ice, or inclined wet or smooth surfaces, stronger slippage can occur. Stronger slippages or sliding are outliers of the odometry observation model. IMU, ICP and VO sensory modalities are not affected in such a case. In order to simulate such a situation, we placed the robot on a trolley and moved it manually.

Figure 16 shows both the trajectory from the top (top-left plot) and the comparison between the fusion of all four sensory modalities and the fusion of only IMU+OD. We can see that the latter wrongly estimates no motion whereas the fusion of all modalities correctly estimates the trajectory. The failure of the partial filter can be explained by the low acceleration of the platform during the test. As the IMU acceleration signal is quite noisy, confidence on the IMU cannot compensate for the odometry modality asserting an absence of motion.

It should be noted that such a failure of the odometry modality does not lead to a failure of our complete filter.

5.4.2 Partial occlusion of visual field of view

Partial occlusion, overexposure, or projections of dirt on the camera could lead to faulty estimation of the motion by the VO. In order to test this situation, we occluded one of the cameras of the omnicamera (see Figure 17). Reduction of the field of view of the omnicamera causes in the vast majority of cases a reduction in the number of visual features being robustly detected by the VO. The insufficient number of features can then cause the VO to incorrectly estimate the attitude. This information then propagates into the state estimate and can make the fusion algorithm fail.

Figure 18 shows the result of the filter in such a case. We can see that during a first loop of the trajectory,

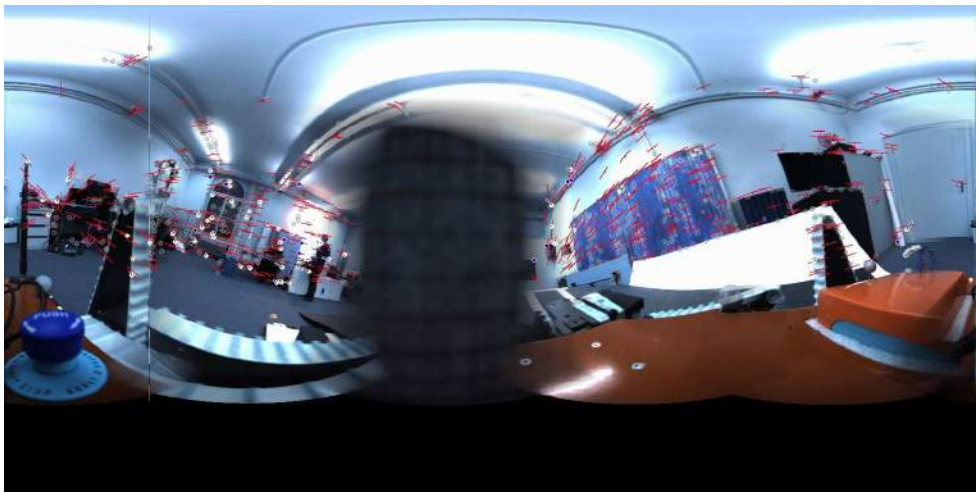


Figure 17: Picture from the partially occluded omnicamera. Notice the dark rectangle in the middle

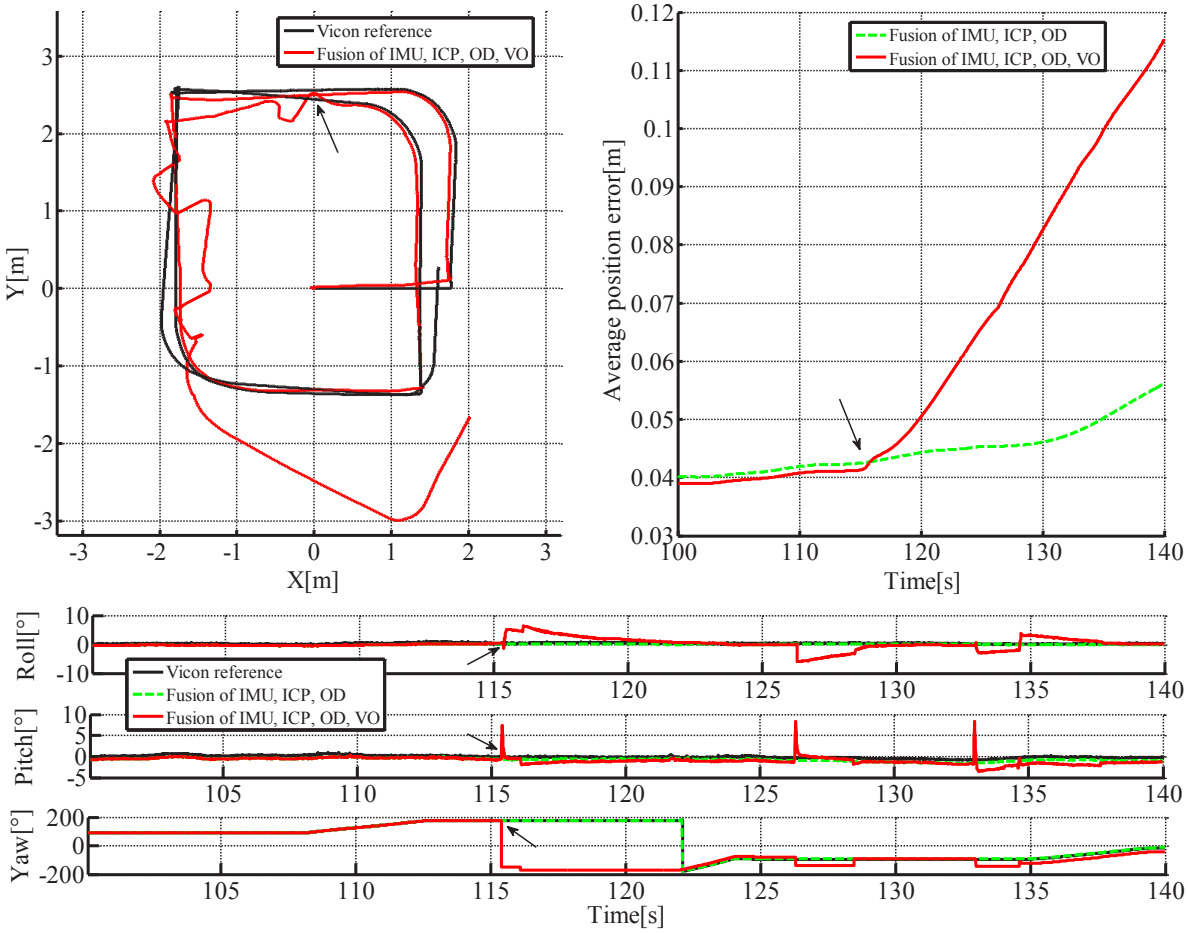


Figure 18: Trajectory reconstruction with partial omnicamera occlusion. Black line: ground truth; Solid red line: state estimate with all four modalities; Dashed green line: state estimate excluding visual odometry; Black arrow: visual odometry failure. Top left: top view of the trajectory; Top right: average position error around visual odometry failure; Bottom: attitude estimated along the trajectory.

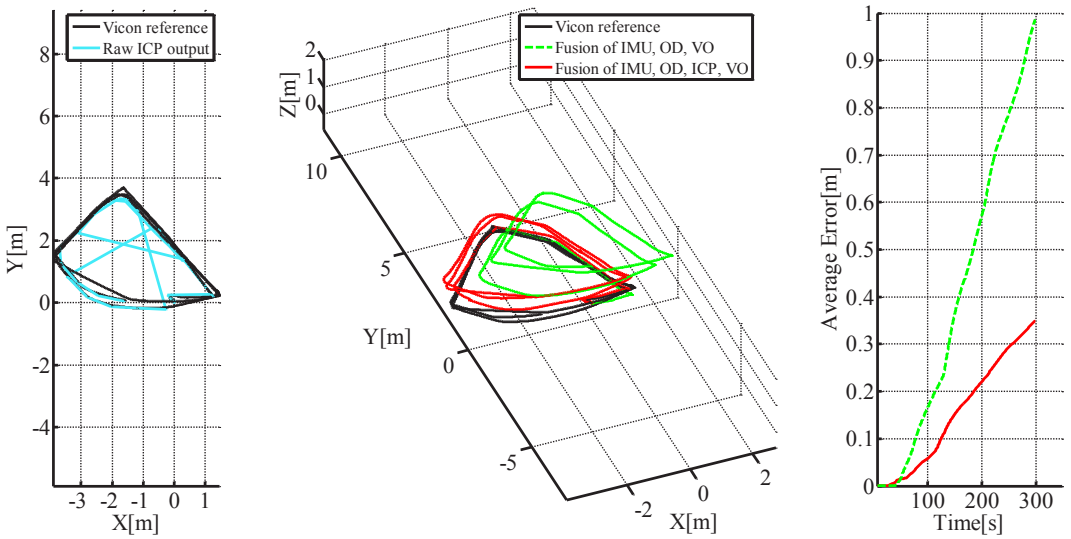


Figure 19: Trajectory estimates in case of low ICP frequency. Black line: ground truth; cyan line: positions estimated by ICP alone; red line: state estimate with all four modalities; green dashed line: state estimate excluding ICP measurements. Left: top view; middle: 3D view; right: average position error.

the state estimation is correct. Then, lacking a sufficient number of features, the VO computes an erroneous estimate and the final state estimate degenerates. On the contrary, by leaving out the visual odometry, the state estimation would continue to perform satisfactorily.

It is to be noted that more than the portion of the field of view occluded, it is the number, quality and distribution of features that matters. One typical way to prevent this issue is to monitor the number of features and eventually their distribution in the field of view—our VO tries to have corresponding features spread over the whole image. As Figure 18 shows, even with the partial occlusion of the field of view, the VO performed correctly during most of the trial. This observation holds also for too dark or over-exposed images.

5.4.3 Temporary laser scanner outage

As demonstrated above, our trajectory approach to fusion of ICP measurements is able to cope with the relatively low frequency of laser scanning. As the laser is moving, it can be blocked in case of collision or high vibration of the platform (safety precaution at the level of the motor controller). When this happens, it is necessary to initiate a recalibration procedure that can take around 30 s.

We simulated this situation by throttling the laser point clouds, which resulted in ICP measurement outages of up to 40 s. Figure 19 shows the trajectory estimates for this test. On the left the cyan polygon shows the position estimates of ICP linked by straight lines (no filtering). It is to be noted that in this case, the positions are accurate compared to the ground-truth but of very low sampling rate. We can see in the middle and right graphs that the filter estimates degrade gracefully. There is some drift, mostly along elevation due to slippage, but even with this low frequency, the ICP measurements help correcting the state estimates over just the IMU, odometry and visual odometry.

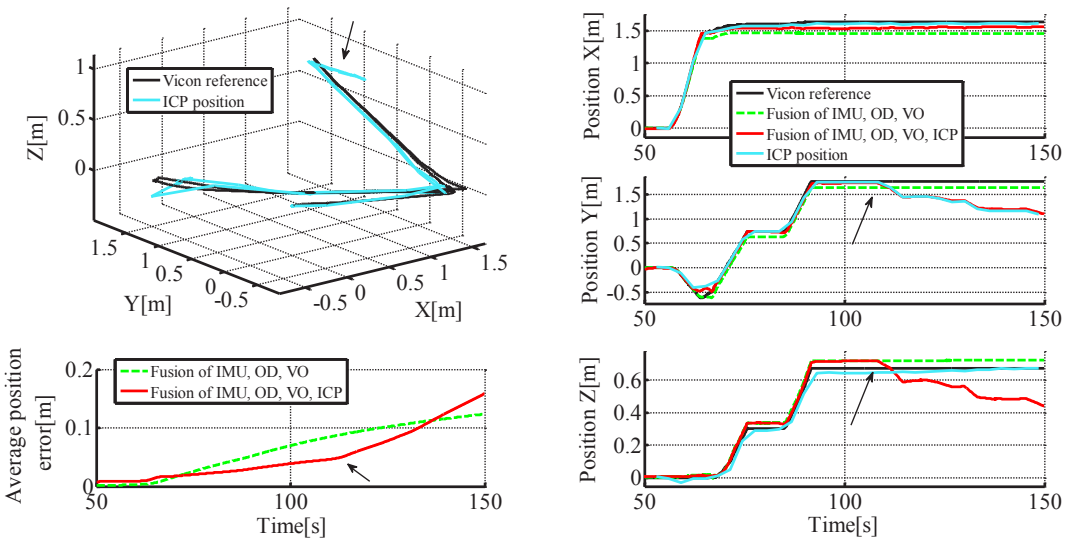


Figure 20: Trajectory estimates in case of moving obstacle in reduced field of view. Solid black line: ground truth; solid red line: state estimate with all four modalities; dashed green line: state estimate excluding ICP measurements; cyan line: position estimated by ICP alone; black arrow: start of moving obstacle. Top left: 3D view; bottom left: average error as a function of time; right: x, y, and z coordinates as a function of time.

5.4.4 Moving obstacle and limited laser range

Unlike the cameras, laser range sensors are not sensitive to illumination conditions. On the other hand, they have a limited sensor range which can induce a lack of points in large environments. Close range obstacles might then be the dominant cluster of points and hence the ICP registration might converge to a wrong local minimum, following the motion of the obstacles.

In order to test this situation, we artificially limited the range of the laser range sensor to 2 m. This is similar to heavy smoke or dust scenarios that can arise in USAR conditions. This prevents the laser to observe the walls and the ceiling that are, indoors, usually the strongest cues for correct point cloud registration.

Additionally, we used a large board to simulate a moving obstacle of significant size. This caused the ICP to drift, following the motion of the board.

Figure 20 shows the result of the filter compared to ground truth. We can see that when the large obstacle starts to move, the estimate of the ICP drifts with it. As a consequence the whole filter drifts as well. This is analogous to the slippage situation, in which the ICP modality compensates the combined estimate of the other three modalities. Using the omnicamera information not only as visual compass but also as a complete visual odometry modality would probably allow to make the difference between those two situations.

5.4.5 Map deformation

As explained above, the ICP map is not globally optimized. This means that the map might have some large scale deformations due to the accumulation of small errors. We were able to observe this particularly in a long corridor that we used to assess the impact of map deformation on the state estimate.

Figure 21 shows an instance of the deformed map. We drove along two superposed corridors over two floors. We can see that both ends of the corridor are not aligned: the ground plane of the blue end has a roll angle

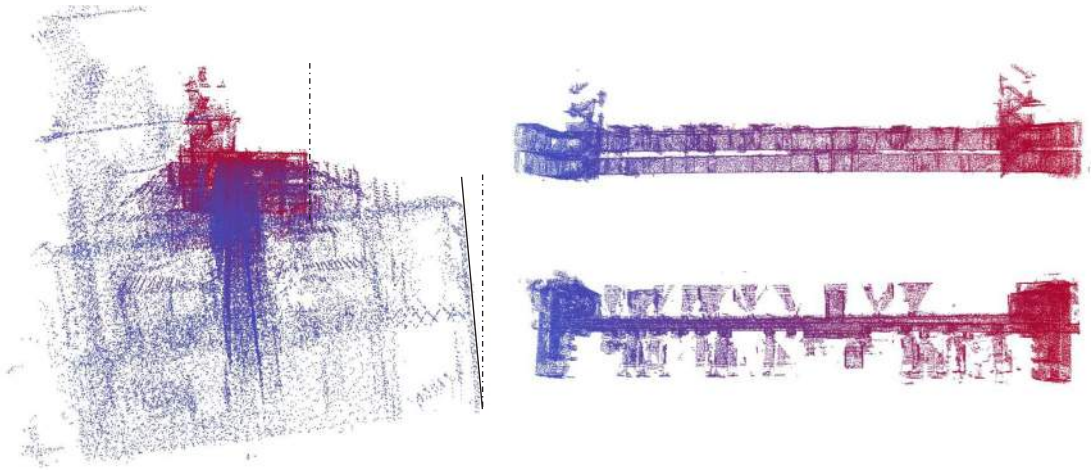


Figure 21: Deformed point cloud map created by ICP. The points are colored alongside the corridor from red (initial position) to blue. Left: front view; top right: side view; bottom right: top view.

of several degrees compared to the red end. We used the theodolite system to acquire ground truth on the upper floor.

Figure 22 shows the impact of map deformation on the state estimate. The top graph shows that even if the ICP estimate is erroneous, the full filter maintains a correct, drift-free estimate. The bottom graph compares the estimate of the roll angle between ICP only and the fusion. It clearly shows the drift in roll of the ICP estimate and the lack of impact it has on the fusion. The difference with previous failure case lies on the kind of drift. The drift of the roll angle can be compensated for by the IMU, especially the accelerometer. On the other hand, the drift in position of previous failure case is not observable by the other modalities.

6 Conclusion

We designed and evaluated a multi-modal data fusion system for state estimation of a mobile skid-steer robot intended for Urban Search and Rescue missions. USAR missions often involve in- and outdoor environments with challenging conditions like slippage, moving obstacles, bad or changing light conditions, etc. In order to cope with such environment, our robot is equipped with both proprioceptive (IMU, tracks odometry) and exteroceptive (laser rangefinder, omnidirectional camera) sensors. We designed such a data fusion scheme in order to adequately include measurements from all these four modalities with order of magnitude difference in update frequency from 90 Hz to $\frac{1}{3}$ Hz.

We tested our algorithm on approximately 4.4 km of field tests (over more than 9 hours of data) both in- and outdoors. In order to have precise quantitative analysis, we recorded ground truth using either a Vicon motion capture system (indoors) or a Leica theodolite tracker (outdoors). This way we proved that our scheme is a significant improvement upon standard approaches. Combining all four modalities: IMU, tracks odometry, visual odometry and ICP-based localization, we achieved precision in the total distance driven of 1.2 % error in the indoor environment and 1.4 % error in the outdoor environment. Moreover, we characterized the reliability of our data fusion scheme against sensor failures. We designed failure case scenarios according to potential failures of each sensory modality that are likely to occur during real USAR missions. In course of this testing, we evaluated robustness with respect to: heavy slippage (odometry failure case), reduction of field of view of the omnicamera (visual odometry failure case), and reduction of the laser rangefinder together with large moving obstacles spoiling the created metric map (ICP-based localization failure case).

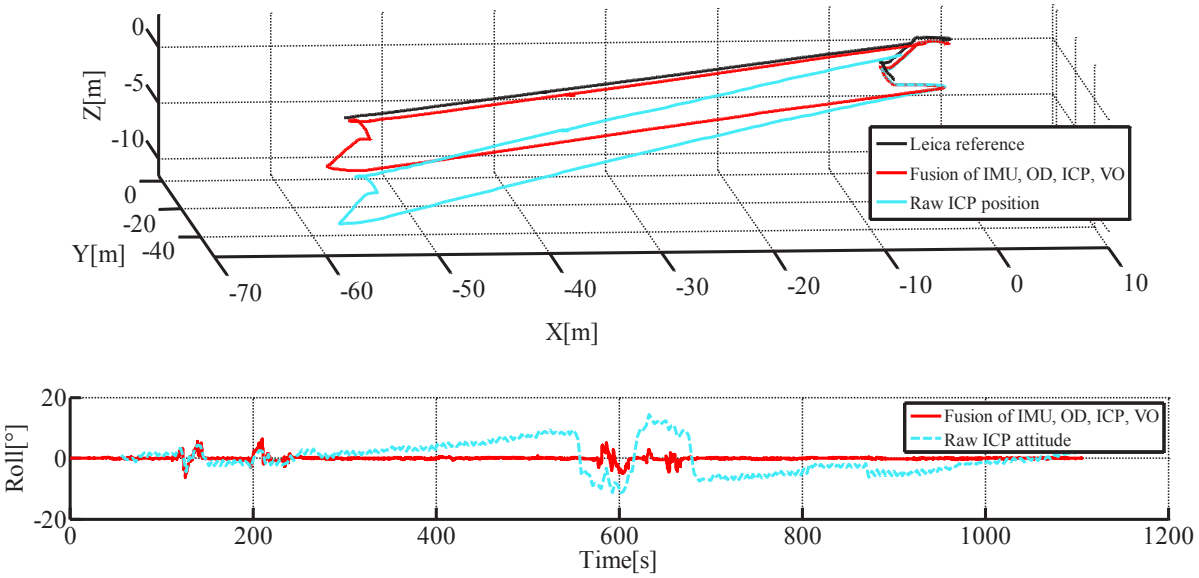


Figure 22: Trajectory estimates in case of map deformation. Solid black line: ground truth; solid red line: state estimate with all four modalities; cyan line: position estimated by ICP alone. Top: side view; bottom: roll angle along the trajectory.

While our filter demonstrates good accuracy during our field tests and is robust against some of the failures expected in USAR, there is still a space for improvement—the need for an automatic failure detection and resolution. Exploring different methods of detecting anomalous measurements and rejecting them in order to improve the overall performance is one of the ways, but it is currently left for future work. Furthermore, developing a visual odometry solution capable of providing also estimates of scaled translation is another topic for the future.

It is not surprising that combining more modalities yields more precision. But we were able to show that if such a rich multi-modal system is well designed, it will perform reasonably well even in cases, where other systems exploiting fewer modalities fail completely. We describe how to design such system using the commonly used EKF. In this way we contribute by proposing and comparing three different approaches to treat the ICP measurements; out of which the *trajectory approach* proved to perform best.

To contribute to the robotics community, we release our datasets used in this paper, including the ground truth measurements from Vicon and Leica systems.

Acknowledgments

The research presented here was funded by the EU FP7 IP projects NIFTI (ICT-247870; `http://www.nifti.eu`) and TRADR (EU-FP7-609763; `http://www.tradr-project.eu`). François Pomerleau was supported by a fellowship from the Fonds québécois de recherche sur la nature et les technologies (FQRNT). Vladimir Kubelka was supported by the Czech Science Foundation (Project Registration No. 14-13876S). We would like to thank the anonymous reviewers for their constructive suggestions that greatly improved the manuscript.

References

Almeida, J. and Santos, V. M. (2013). Real time egomotion of a nonholonomic vehicle using lidar measurements. *Journal of Field Robotics*, 30(1):129–141.

Anousaki, G. and Kyriakopoulos, K. J. (2004). A dead-reckoning scheme for skid-steered vehicles in outdoor environments. In *Robotics and Automation (ICRA), 2004. Proceedings of the IEEE International Conference on*, pages 580–585.

Bachrach, A., Prentice, S., He, R., and Roy, N. (2011). RANGE—Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666.

Barfoot, T., Stenning, B., Furgale, P., and McManus, C. (2012). Exploiting reusable paths in mobile robotics: Benefits and challenges for long-term autonomy. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 388–395.

Besl, P. and McKay, H. (1992). A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256.

Breckenridge, W. G. (1999). Quaternions - proposed standard conventions. Technical report, JPL.

Brodsky, T., Fermueller, C., and Aloimonos, Y. (1998). Directions of motion fields are hardly ever ambiguous. *International Journal of Computer Vision*, 26(1):5–24.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Robotics and Automation (ICRA), 1991. Proceedings of the IEEE International Conference on*, pages 2724–2729.

Chetverikov, D., Svirko, D., Stepanov, D., and Krsek, P. (2002). The Trimmed Iterative Closest Point algorithm. In *Pattern Recognition, 2002. Proceedings of the 16th International Conference on*, pages 545–548.

Chiu, H.-P., Williams, S., Dellaert, F., Samarasekera, S., and Kumar, R. (2013). Robust vision-aided navigation using sliding-window factor graphs. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 46–53.

Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *Journal of Field Robotics*, 30(3):415–438.

Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2010). 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631.

Dissanayake, G., Sukkarieh, S., Nebot, E., and Durrant-Whyte, H. (2001). The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation*, 17(5):731–747.

Ellekilde, L.-P., Huang, S., Miro, J. V., and Dissanayake, G. (2007). Dense 3D Map Construction for Indoor Search and Rescue. *J. Field Robotics*, 24(1-2):71–89.

Endo, D., Okada, Y., Nagatani, K., and , K. (2007). Path following control for tracked vehicles based on slip-compensating odometry. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, pages 2871–2876.

Fraundorfer, F. and Scaramuzza, D. (2012). Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications. *Robotics Automation Magazine, IEEE*, 19(2):78–90.

Galben, G. (2011). New Three-Dimensional Velocity Motion Model and Composite Odometry–Inertial Motion Model for Local Autonomous Navigation. *IEEE Transactions on Vehicular Technology*, 60(3):771–781.

Jesus, F. and Ventura, R. (2012). Combining monocular and stereo vision in 6d-slam for the localization of a tracked wheel robot. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–6.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):34–45.

Kelly, J., Sibley, G., Barfoot, T., and Newman, P. (2012). Taking the Long View: A Report on Two Recent Workshops on Long-Term Autonomy. *Robotics & Automation Magazine, IEEE*, 19(1):109–111.

Kohlbrecher, S., Stryk, O. V., Meyer, J., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160.

Konolige, K., Agrawal, M., and Sola, J. (2011). Large-scale visual odometry for rough terrain. In Kaneko, M. and Nakamura, Y., editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 201–212. Springer.

Kruijff, G. J. M., Janicek, M., Keshavdas, S., Larochelle, B., Zender, H., Smets, N. J. J. M., Mioch, T., Neerincx, M. A., van Diggelen, J., Colas, F., Liu, M., Pomerleau, F., Siegwart, R., Hlavac, V., Svoboda, T., Petricek, T., Reinstein, M., Zimmerman, K., Pirri, F., Gianni, M., Papadakis, P., Sinha, A., Balmer, P., Tomatis, N., Worst, R., Linder, T., Surmann, H., Tretyakov, V., Surmann, H., Corrao, S., Pratzler-Wanczura, S., and Sulk, M. (2012). Experience in System Design for Human-Robot Teaming in Urban Search and Rescue. In *Field and Service Robotics*, pages 1–14, Matsushima, Japan.

Kubelka, V. and Reinstein, M. (2012). Complementary filtering approach to orientation estimation using inertial sensors only. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 599–605.

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

Lamon, P. and Siegwart, R. (2004). Inertial and 3D-odometry fusion in rough terrain - towards real 3D navigation. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2004)*, pages 1716–1721.

Li, H. and Hartley, R. (2006). Five-point motion estimation made easy. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 630–633.

Loan, C. V. (1978). Computing integrals involving the matrix exponential. *Automatic Control, IEEE Transactions on*, 23(3):395–404.

Ma, J., Susca, S., Bajracharya, M., Matthies, L., Malchano, M., and Wooden, D. (2012). Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in gps-denied environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 619–626.

McElhoe, B. A. (1966). An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus. *Aerospace and Electronic Systems, IEEE Transactions on*, 2(4):613–623.

Morales, Y., Carballo, A., Takeuchi, E., Aburadani, A., and Tsubouchi, T. (2009). Autonomous robot navigation in outdoor cluttered pedestrian walkways. *Journal of Field Robotics*, 26(8):609–635.

Nagatani, K., Okada, Y., Tokunaga, N., Kiribayashi, S., Yoshida, K., Ohno, K., Takeuchi, E., Tadokoro, S., Akiyama, H., Noda, I., Yoshida, T., and Koyanagi, E. (2011). Multirobot exploration for search and rescue missions: A report on map building in robocuprescue 2009. *Journal of Field Robotics*, 28(3):373–387.

Nemra, A. and Aouf, N. (2010). Robust INS/GPS Sensor Fusion for UAV Localization Using SDRE Nonlinear Filtering. *IEEE Sensors Journal*, 10(4):789–798.

Nuchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2007). 6D SLAM - 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722.

Oskiper, T., Chiu, H.-P., Zhu, Z., Samarasekera, S., and Kumar, R. (2010). Multi-modal sensor fusion algorithm for ubiquitous infrastructure-free localization in vision-impaired environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1513–1519.

Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. (2013). Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots*, 34(3):133–148.

Reinstein, M. and Hoffmann, M. (2013). Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. *IEEE Transactions on Robotics*, 29(2):563–571.

Reinstein, M., Kubelka, V., and Zimmermann, K. (2013). Terrain adaptive odometry for mobile skid-steer robots. In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pages 4706–4711.

Rodriguez F, S. A., Fremont, V., and Bonnifait, P. (2009). An experiment of a 3D real-time robust visual odometry for intelligent vehicles. In *Proc. 12th Int. IEEE Conf. Intelligent Transportation Systems ITSC '09*, pages 1–6.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564 –2571.

Sakai, A., Tamura, Y., and Kuroda, Y. (2009). An efficient solution to 6DOF localization using Unscented Kalman Filter for planetary rovers. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2009*, pages 4154–4159.

Savage, P. G. (1998). Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics*, 21(No. 2):208 – 221.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. *IEEE Robot Autom Mag*, 18(4):80–92.

Shen, J., Tick, D., and Gans, N. (2011). Localization through fusion of discrete and continuous epipolar geometry with wheel and IMU odometry. In *Proc. American Control Conf. (ACC)*, pages 1292–1298.

Smith, G. L., Schmidt, S. F., and McGee, L. A. (1962). Optimal filtering and linear prediction applied to a midcource navigation system for the circumlunar mission. Technical report, U.S. Government Printing Office.

Sukumar, S. R., Bozdogan, H., Page, D. L., Koschan, A. F., and Abidi, M. A. (2007). Sensor selection using information complexity for multi-sensor mobile robot localization. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4158–4163.

Suzuki, T., Kitamura, M., Amano, Y., and Hashizume, T. (2010). 6-DOF localization for a mobile robot using outdoor 3D voxel maps. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 5737–5743.

Svoboda, T., Pajdla, T., and Hlaváč, V. (1998). Motion estimation using central panoramic cameras. In *IEEE International Conference on Intelligent Vehicles*, pages 335–340.

Tardif, J., Pavlidis, Y., and Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538.

Titterton, D. H. and Weston, J. L. (1997). *Strapdown Inertial Navigation Technology*. The Lavenham Press ltd, Lavenham, UK.

Trawny, N. and Roumeliotis, S. I. (2005). Indirect Kalman Filter for 3D Attitude Estimation - A Tutorial for Quaternion Algebra. Technical report, University of Minnesota.

Weiss, S. M. (2012). Vision based navigation for micro helicopters. Dissertation, ETH Zurich.

Yi, J., Zhang, J., Song, D., and Jayasuriya, S. (2007). Imu-based localization and slip estimation for skid-steered mobile robots. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, pages 2845–2850.

Yoshida, T., Irie, K., Koyanagi, E., and Tomono, M. (2010). A sensor platform for outdoor navigation using gyro-assisted odometry and roundly-swinging 3D laser scanner. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 1414–1420.