DR 2.4: Collaborative Mobile Manipulation (and Network Resilience)

Petter Ögren[*], Sergio Caccamo[*], Luigi Freda[†], Rainer Worst[‡], Abel Gawel[¶], Renaud Dubé[¶], Cesar Cadena[¶], Tomáš Svoboda[‖], and the TRADR Consortium

[*]*KTH - Royal Institute of Technology, Stockholm, Sweden*
[†]*Alcor Laboratory, Department of Computer, Control, and Management Engineering "Antonio Ruberti" - Sapienza University of Rome, Italy*
[‡]*Fraunhofer IAIS, Sankt Augustin, Germany*
[¶]*ETH Zürich - Autonomous Systems Lab, Zurich, Switzerland*
[‖]*CTU - Czech Technical University in Prague, Czech Republic*
⟨petter@kth.se⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	March 2018
Actual submission date:	March 2018
Lead partner:	KTH
Revision:	Final
Dissemination level:	PU

This document describes the work of providing the TRADR robots with motion capabilities. For the UGVs we have worked on formation obstacle avoidance as an enabling technology for collaborative manipulation. We have also investigated two different ways of using a manipulator to estimate traversabilty of surfaces. Furthermore, we have continued the work on network resilience in terms of both path planning and teleoperation.

For UAV-UGV collaboration we have studied ways of combining maps for very different viewpoints as well as sensors. Finally, for UAVs, we have investigated the possibility of picking and delivery using a magnetic actuator.

Executive Summary

This report describes work towards providing the TRADR robots, Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) with motion capabilities, in terms of coordinated motion, network resilient motion, motion enabled by manipulator actions, motion enabled by coordinated sensing and motion for aerial picking and delivery.

The coordinated motion capability was design to enable a group of UGV to travel in formation while lifting and transporting an object through a highly obstacle dense environment. The approach builds upon a combination of Constraint Based Programming (CBP) and Rapidly exploring Random Trees (RRTs). By using CBP when trying to find a joint motion between to sampled states, the probability of a successful motion is much higher, but the needed computation time rises as well. It was shown that for sufficiently cluttered environments, the benefit of CBP outweighs the drawback in computations time, compared to a standard RRT approach where the extra computation is used to sample more nodes in the tree.

The network resilient motion capability is a continuation of the preliminary work reported earlier, with additional publications and ongoing extensions. By measuring, mapping and predicting the Radio Signal Strengths (RSS) we are able to improve performance of both teleoperation and autonomous path planning. For teleoperation we have shown that the number of objects found before connection loss in a search scenario increases, and for path planning we have improved the reactivity relative to sudden losses of network Access Points (APs), or reductions of RSS due to other reasons.

The motion enabled by manipulator actions includes passing over areas with initially uncertain traversibility. Two algorithms were proposed using Gaussian Process Regression (GPR). First, investigating the terrain shape in occluded areas, and second estimating the deformability of visible areas that might look traversable but are not due to e.g. excessive amounts of mud or water.

The motion enabled by coordinated sensing is the capability to integrate sensing from very different viewpoints and sensors, such as data from a UAV and a UGV deployed in the same area. This is very difficult, but as the same time potentially very useful, as the UGV is dependent on traversability estimates that itself can only construct locally, while a UAV is free to move over a much larger space, providing data from areas well beyond the horizon of the UGV. The techniques investigated include both 3D-feature based registration and semantic graphs.

The motion for aerial picking and delivery is the capability of doing pick and place with a UAV. This was also explored in a collaborative fashion, where the UAV picked an item from the UGV and delivered it to a static target location nearby. This capability can also be used when the UGV does sampling of potentially toxic fluids. Then the UAV could quickly bring the

collected samples out of the disaster zone for further analysis, enabling the UGV to continue sampling without having to spend time on the arduous journey back and forth.

Finally, we have made a number of improvements to the navigation system, that is crucial to many of the above capabilities. These lie at the intersection of WP1, WP2 and WP4 and are therefore described briefly in Section 1.3, and in detail in the WP4 deliverable, Dr4.4.

Role of navigation, exploration and manipulation in TRADR

Mobility of the TRADR robots is of key importance to the successful execution of the disaster response scenarios. In many instances, the overall system performance is improved when both operator and robot can contribute with their key strengths.

Persistence

Persistence is present in WP2 in terms of maps of both terrain and network being created, re-used and shared across entities.

Contribution to the TRADR scenarios and prototypes

The motion capabilities of the TRADR UAVs and UGVs are essential for the use cases. In particular, the content of this deliverable relates to the following use cases.

- Generic use case 1: *UAV[x] detect/search for X, using method Y* Capabilities for this use case are described in Section 1.3.5 below.
- Generic use case 3: *UGV[x] go to location X (optionally via Y).* Capabilities for this use case are described in Section 1.3.2-4 below.
- Generic use case 4: *UGV[x] go to location X on (semi)autonomous mode.* Capabilities for this use case are described in Section 1.3.3-4 below.
- Generic use case 5:*UGV[x] detect/search for X, using method Y.* Capabilities for this use case are described in Section 1.3.2-4 below.
- Generic use case 6: *UGV[x] manipulates object X* Capabilities for this use case are are described in Section 1.3.1 below.

- Generic use case 7: *UGV[x] encounters obstacle X, takes action Y to overcome* Capabilities for this use case are described in Section 1.3.2 below.

1 Tasks, objectives, results

1.1 Planned work

The work described in this report (D2.4) was performed within the scope of Tasks T2.7 (Collaborative mobile manipulation) The objectives of this tasks was to develop cooperative mobile manipulation capabilities for the UGV.

However, in response to the reviewers' comments below, parts of the effort in this WP has been shifted from manipulation towards network resilience, hence the document title: *Collaborative mobile manipulation (and Network Resilience)*.

1.2 Addressing reviewers' comments

Below we collect the reviewer comments made in Year 3 regarding WP2, with corresponding answers.

1. Overall excellent progress has been achieved in this workpackage.

   Response: We thank the reviewers for this overall feedback.

2. A repeating recommendation is that the technologies developed by TRADR must be network resilient. Rather than focusing on improving the network robustness, and on methods for detecting and fixing network failures, the focus should be on developing technologies in which network failures and limitations are a given, and cannot be eliminated.

   Response: We have shifted resources from manipulation towards network resilience in response to these recommendations.

3. Unfortunately, still a large gap exists for the validation of the strong claims made for the FLC control mode for UGVs, that it would be better than other established control modes like third-person view in any cases.

   Response: There seems to be an unfortunate misunderstanding regarding the term control mode. By control mode we mean the way user control inputs are translated to track velocities and camera motions. Using this terminology gives us the two options of Tank Control and FLC. We have compared these in the paper. Another important part of the GUI is the camera view. Either a 1st person camera view is used (the raw video feed), of a virtual 3rd person view is used, rendered from fused sensor data, showing the robot from the outside in relation to a model of its surrounding. Thus the concepts of 1st or 3rd person view is orthogonal to the concepts of Tank Control and FLC, and can be combined freely. In fact, a number of computer games,

e.g. the Unchartered series, use FLC with a 3:rd person view. Thus it makes no sense to compare FLC to a 3:rd person view.

4. The claimed superiority of the FLC is especially questioned for cluttered environments which are typical for disaster areas.

   Response: There seems to be a misunderstanding on this issue as well. We completely agree that FLC is not suitable in very cluttered environments. This was stated in the Y3 slides of WP2, where the bottom of slide 7 read: "But FLC is not suitable for very cluttered environments".

   Furthermore, in [68], we write: "In extremely narrow passages, a given camera motion might cause the UGV chassis to collide with an object. In applications where this is a problem, allowing the operator to switch between Tank Control and FLC might be useful."

5. The conduction of a meaningful scientific study on the different modes of control of UGV robots in realistic scenarios is strongly recommended

   Response: During the work in TRADR, our experience has shown that this kind of studies are extremely time consuming to perform, and harder to publish than non-user studies, even though we have human factor specialist in the team. Given this fact, and the misunderstandings above, we have chosen not to perform user studies beyond the ones reported in papers [5, 70].

1.3 Results

In this section we will report the results of work done in WP2. However, we start by mentioning some important work on the navigation system, that lie in the intersection of WP1, WP2 and WP4, and are reported in detail in Deliverable DR4.4.

A safe and robust navigation plays a crucial role for many of the TRADR system capabilities. In order to improve the performance of the autonomous navigation system of our UGVs, we revised three different components. First, we revised the onboard path planner pipeline. In particular, we analysed and tested different point cloud segmentation methods in order to improve the underlying traversability analysis module and the recognition of stairs and ramps in the environment. Second, we integrated the adaptive traversal algorithm into the path planner, with the aim of pushing the autonomous UGV navigation capabilities towards more challenging and harsh terrains. Third, we developed an RGBD-SLAM method, PLVS (Points, Lines, Volumetric mapping and incremental Segmentation), which can be used to robustly build a denser point cloud map and enable a more advanced and accurate analysis of the terrain and surrounding objects.

We now describe the rest of the WP2 results in more detail.

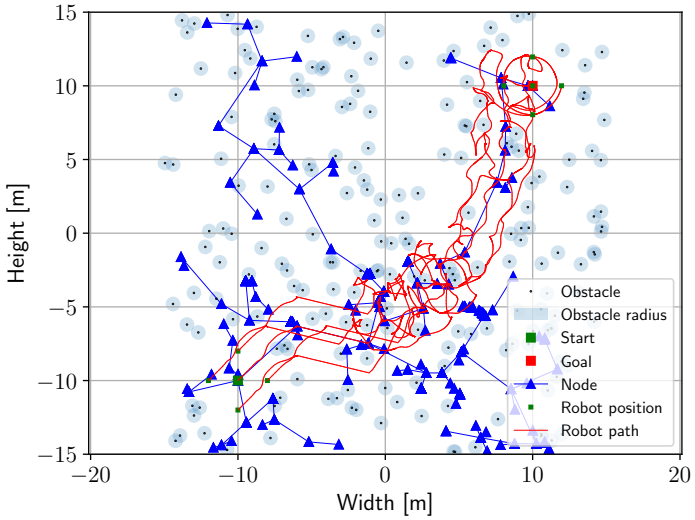


Figure 1: Example execution of CBP-RRT with 200 obstacles. Non-accessible areas belonging to obstacles are indicated by blue circles with a black dot at the center. The start position of the virtual structure is located at (-10, -10), while target position is at (10, 10). Smaller green squares indicates robot positions and blue triangles indicates nodes in the RRT.

1.3.1 Formations for Collaborative Mobile Manipulation

Collaborative mobile manipulation denotes the activity of multiple mobile manipulators lifting and moving an object in a collaborative fashion. This is a complex problem since it requires all motions to be synchronized, in particular the non-arm part of all mobile manipulators need to translate in a coordinated fashion, and this is the problem addressed in this paper.

We consider obstacles of arbitrary shape, modelled by the union of a large set of possibly overlapping circular discs, see Figure 1. The formation is given as a desired rigid shape, that can translate and rotate to get past obstacles and arrive at the goal, and the robots are modelled as kinematic points.

In this paper, we will use Constraint Based Programming (CBP) [67] for formation control, where the constraints will be given by both the formation keeping objective, and the desired obstacle clearance. The objective function will be the progress towards the next waypoint, and the waypoints are chosen using an Rapidly Exploring Random Tree (RRT) [48] algorithm. An example of the CBP local steering function can be found in Figure 2.

We will now compare the performance of the proposed approach (CBP-RRT) with a standard approach using linear interpolation for local steering in the RRT algorithm (LI-RRT). We look at the time required for calcu-

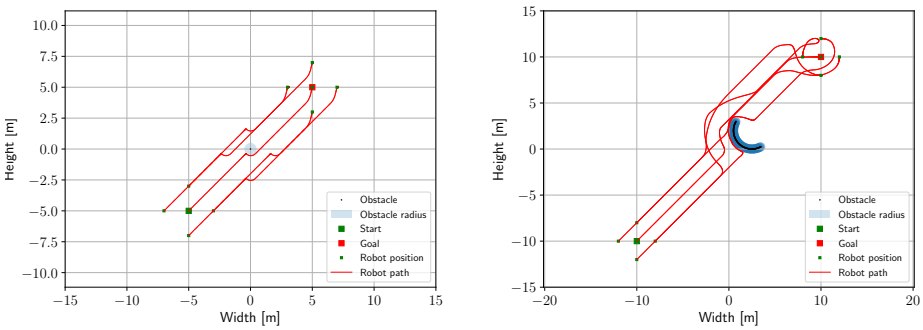


Figure 2: Example of the paths generated by CBP-RRT with a single node, with a single obstacle (a) and for a non-convex obstacle configuration (b). Sufficient time to reach the target with one node was provided. Note that for both cases, LI-RRT would not give a valid solution, due to the collision check.

lation, and the number of nodes (iterations) needed by CBP-RRT and LI-RRT respectively. Each run consists of generating an environment through randomly placing obstacles, and then running the two approaches for the generated environment. This is repeated, and the average values are presented. The results are presented in Table 1, where **Method** is either the proposed method CBP-RRT, or the baseline method LI-RRT. **Obstacles** is the number of obstacles randomly placed in the environment. **Nodes** is the number of nodes (iterations) needed until convergence, and **Time** is the time required, in seconds, to find a solution (lower is better). **Rep.** is the number of runs used to compute the average, with total number of runs in parenthesis.

Examples of executions with CBP and Linear Interpolation are shown in Figures 3 and 4. We also supply a link to a video[1] showing the execution of the plan obtained by CBP-RRT.

As can be seen the number of nodes are significantly lower for CBP-RRT than for LI-RRT, for any example with obstacles present. However, the linear interpolation is faster than the CBP, so if we look at execution times the picture is more complex. LI-RRT is faster for low obstacle densities, but for sufficiently cluttered environments, the computation used for the CBP is compensated for by the lower number of nodes.

This work was awarded as Finalist for Best Paper at the SSRR 2017 in Shanghai, China. For details, see the appended paper [4] (Annex Overview 2.1).

[1] Also available on `https://www.youtube.com/watch?v=OtvFIZFg68M`

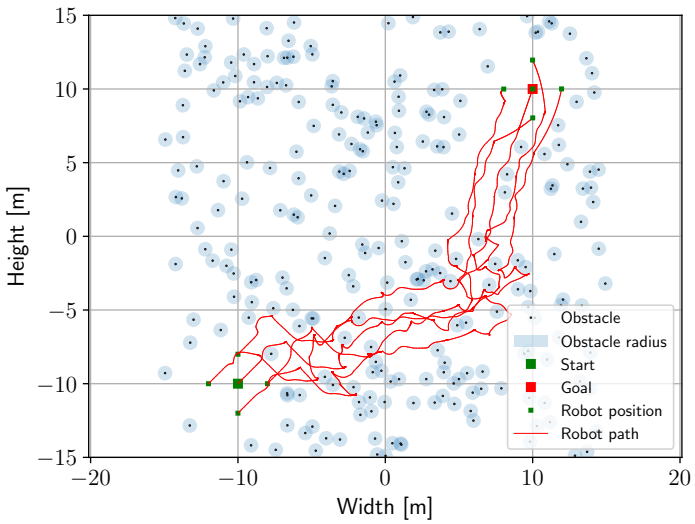


Figure 3: Execution of path generated by CBP-RRT for 250 obstacles.

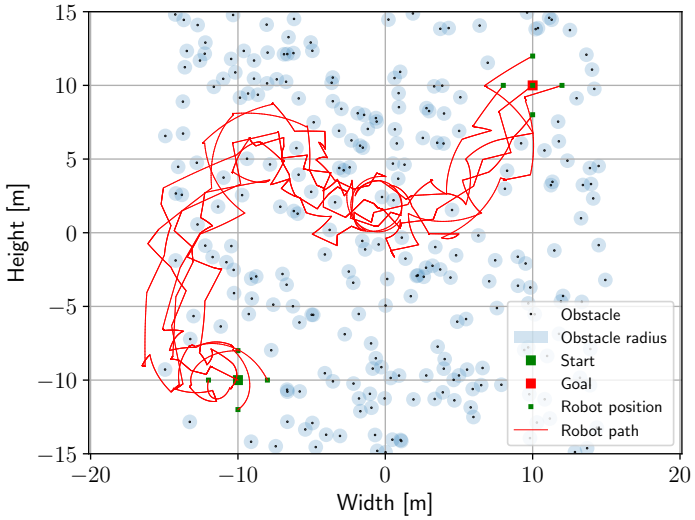


Figure 4: Execution of path generated by LI-RRT for 250 obstacles.

Table 1: Average results after running both algorithms, with 5 seconds for execution between each node, ordered by increasing number of obstacles. Cases where no solution could be found were excluded when calculating the average.

Method	**Obstacles [-]**	**Nodes [-]**	**Time [s]**	**Rep. [-]**
LI-RRT	0	33.6	0.06	100 (100)
CBP-RRT	0	36.0	0.06	100 (100)
LI-RRT	25	77	0.26	100 (100)
CBP-RRT	25	33.6	2.77	100 (100)
LI-RRT	50	146.4	0.45	100 (100)
CBP-RRT	50	38.0	5.26	100 (100)
LI-RRT	100	1018.8	2.64	100 (100)
CBP-RRT	100	51.8	13.49	100 (100)
LI-RRT	150	6027.8	15.35	100 (100)
CBP-RRT	150	106.6	43.22	100 (100)
LI-RRT	200	30539.6	95.33	100 (100)
CBP-RRT	200	183.0	87.74	100 (100)
LI-RRT	250	93852.1	311.67	91 (100)
CBP-RRT	250	388.4	212.89	100 (100)

1.3.2 Manipulation for Low Visibility Perception

Hostile environmental conditions, that frequently characterize urban search and rescue scenarios, represent a serious threat to modern vision perception system. The vast majority of Unmanned Ground Vehicles are equipped with sensors such as LiDARs, RGB cameras and thermo cameras, that the robots use to build their understanding of the disaster area. Platforms used in TRADR, for instance, make heavy use of LiDARs and RGB cameras to built a geometric (point cloud map) representation of the hot zone that is then used to plan actions and share knowledge with other team members. The presence of smoke, fire, fog and other environmental phenomena affects the performance of these sensors that can produce noisy data and lead the robot to make wrong guesses on the nature of the surrounding. For example, the overwhelming presence of layers of grey dust on a post-earthquake USAR scenario, hide object shapes and colors leading object detection algorithms to fail. For this reason, we investigate ways of enhancing vision perception trough the use of tactile sensor systems.

We propose two active perception algorithms, based on *Gaussian Process Regression* (GPR) for (I) investigating the terrain shape in occluded areas surrounding the UGV and (II) estimate the terrain deformability in the proximity of the UGV.

In the first work, an arm equipped UGV physically investigates the shape

of the terrain in regions where the point cloud representation is incomplete. In order to detect the areas of the terrain surface that require a physical interaction, the framework trains a *Gaussian Random Field* (2.5D GPR) on the 3D points of the map located around the robot and then computes a Delaney triangulation on the same region. Constraints on the triangles size and eccentricity along with the value of the variance of the GPR, enables highlighting sparser regions on the point clouds that present higher uncertainty (i.e. the probabilistic model is unable to estimate the surface shape). The robot then uses its manipulator to touch the surface on the selected areas in order to obtain a rich signal of tactile information which is fed into the probabilistic model. The mean of the Gaussian Process is then used to refine the point cloud representation and therefore the geometric map. With this approach we enable a UGV to obtain insights on the environment when its point cloud representation is limited by poor vision conditions.

A muddy or unstable terrain can hamper the robot motion and slow down the mission significantly. A robot able to estimate the terrain deformability can decide to plan trajectories or manipulate objects on rigid surfaces. This motivated the development of an interactive perception framework which uses tactile information to estimate the terrain deformability in the robot surrounding. Following the aforementioned mechanism, the proposed framework trains a Gaussian Random Field (2.5D GPR) on the 3D points of the map located around the robot. Differently from (I), the system observes the behavior of the terrain subjected to touch and uses a mathematical framework based on *Position Based Dynamics* (PBD) to estimate a parameter (β) that represents the local deformability of the terrain on the stressed area. The system uses the GPR to obtain compact representations of the region of the surface under analysis during the physical contact. If the terrain is rigid the estimated β value will be close to zero, and an elastic or deformable terrain will generate a β value close to 1. The system then initializes a second GPR to map the deformability of the terrain trained on the observed β values. As a result the robot is able to map the heterogeneous deformability of the terrain after a few physical interactions and observations. Contrarily to the computationally expensive and difficult to tune *Finite Element Method* (FEM), PBD is faster and requires only few visual clues to obtain feasible deformability models, which however are less accurate and do not describe the real physical properties of the shape under analysis. This allows faster environmental interactions which is a vital requirement of any active perception algorithm designed for USAR operation.

For details, see the appended papers [15] (Annex Overview 2.2), [14] (Annex Overview 2.3).

1.3.3 Network Aware Path Planning

Note: Parts of this work was reported in Y3 as a paper in submission, but is now published in the proceedings of IEEE IROS. We are currently working on a journal papers extension of the work with the following additions:

- Implementation and testing will be done on real scenarios with real robots
- A new self re-connection strategy that takes advantage of a redesigned utility function for our Communication Aware motion Planner. Specifically, we designed a "hierarchical optimization" scheme which suitably combines together the traversability cost with the estimated RSS in the underlying randomized A* expansion of the path planner. This technique automatically steers the robot along planned paths where traversability cost is minimized while, at the same time, a minimum RSS quality RSS_{min} is guaranteed. In this context, when connection is lost, the active re-connection strategy drives the robot towards the closest region where the estimated RSS is greater than RSS_{min}. Next, the robot is allowed to re-plan a new path towards the assigned destination.
- More robust Wireless Map Generation. The framework uses a new sampling strategy for building up the training set used by the Gaussian Process Regression model (GPR). Instead of a moving queue of sampling points located along the robot trajectory, the framework now collects RSS measurements depending on the spacial sparsity of the training set, on the quality of the model prediction and on the life time of the previously stored samples. The training set grows faster when the robot explores new locations and becomes sparser (training points are discarded) on areas explored several time stamps in the past. If the predicted RSS value on the map, obtained from the GPR, differs considerably from the measured value at the robot location, the training set is updated with the new measurement and other local training points (which are now outdated) are discarded.
- The system can now map multiple wireless distributions, generated by multiple Access Points (AP) at the same time in a multi layer wireless map. An important limitation of the previous implementation is that multiple signal distributions were flattened down into the same wireless map; now the robot can track the temporal behavior of several access points independently and plan trajectories accordingly.

An overall description of the work now follows.

In search and rescue robotics, the main task of a robot is often to gather information from a disaster area. Sometimes this information is delivered

at the end of the mission, but more often, the human rescue workers get the information online, during the mission. Then, a reliable and resilient wireless connection to the robot is vital to the mission success.

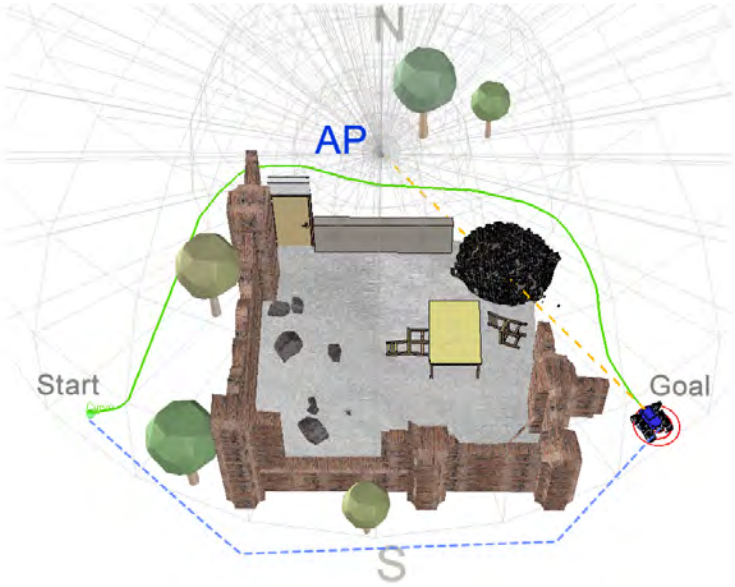


Figure 5: Experimental scenario 1. The UGV tries to reach the goal position avoiding connection drops. The blue dotted line represents the shortest path, that will cause a connection loss (going outside the AP range). The green line represents a path that reaches the goal position while keeping the robot connected to the AP.

Experience has shown that wireless connectivity in disaster areas is bound to be unreliable. Therefore, we propose a Resilient Communication-Aware Motion Planner (RCAMP). Previous solutions relied on detailed information on the network, or used back-tracking to improve connection quality. This work goes beyond that, allowing the system to plan a trajectory that improves connectivity, based on information that is gathered online, during the mission. The proposed solution has two key components, a Gaussian Random Field (GRF) based probabilistic model used to map the Radio Signal Strength (RSS), and traversability map created from a laser sensor. Based on these two sources of information, we device a strategy to regain connectivity, while moving towards the given goal. We will now illustrate the RCAMP in two scenarios.

Scenario 1: In the first scenario, see Fig. 5, the UGV is placed on the start position and must traverse an area containing a damaged building, to reach the goal position. An AP is placed on the northern part of the map (zone N in Fig. 5). The AP uses an omni-directional antenna covering a circular area that extends to half of the map, leaving the southern part (zone S in Fig. 5) uncovered. Start and goal positions are placed such that the shortest connecting path between the two points would traverse the poorly

connected part of the map (S). Thus, RCAMP must generate a trajectory that connects the start and goal positions while keeping the robot in the signal covered area avoiding communication drops. With this scenario we want to demonstrate the capability of our utility function in keeping the robot connected to the AP.

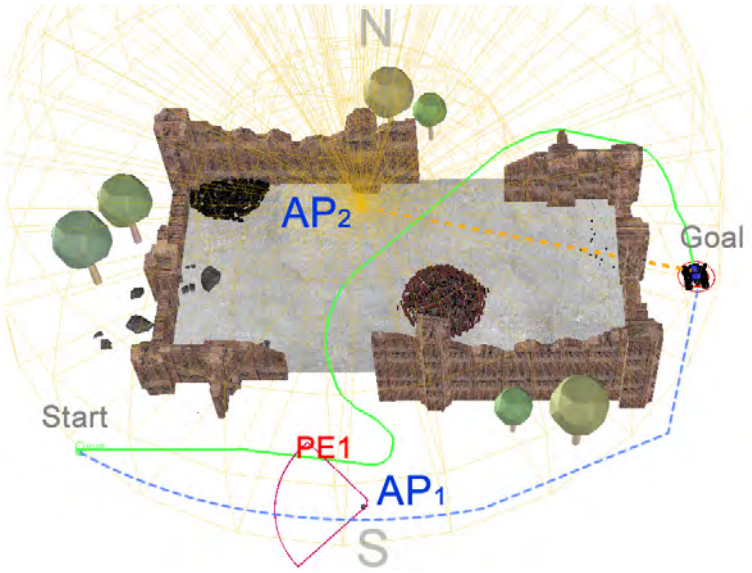


Figure 6: Experimental scenario 2. The UGV tries to reach the goal position avoiding connection drops. The blue dotted line represents the shortest path to the goal position. The UGV is connected to AP_1 in the first part of the path. PE1 indicates the location of the UGV when AP_1 shuts down after a simulated hardware failure. The green line represents a new path that reaches the goal position while keeping the UGV connected, after switching from AP_1 to AP_2.

Scenario 2: In the second scenario, see Fig. 6, two different APs cover the whole map. In this use case we want to test the promptness of the RCAMP to adapt to drastic changes in the wireless signal distribution. The robot starts the mission connected to AP_1. The RCAMP must generate a path from the start position to the goal position that ensures WiFi coverage. During the mission, AP_1 is switched off when the robot enters the region PE1, so to simulate a communication loss event. When the connection is lost, the robot connects to other APs (if available) in the same network, in a typical roaming behaviour. Once the robot connects to AP_2, the WMG must adapt its predictive model to the new signal distribution accordingly and reshape the RSS map. The RCAMP must then promptly update the path to the goal to ensure WiFi coverage.

For details, see the appended paper [13] (Annex Overview 2.4).

1.3.4 Network Aware Teleoperation, Design and User Study

Note: Most of this work was reported in Y3 as a tech report, but is now published in the IJHRI Journal.

Recent and current Urban Search and Rescue (USAR) missions show that the range and coverage of the wireless connection between the operator and the teleoperated Unmanned Ground Vehicle (UGV) presents a significant constraint on the mission execution. For continuing operation, the operator needs to continuously adapt to the dynamic network connectivity across the environment in addition to performing the primary navigation, observation and manipulation tasks.

In this work, a new teleoperation User Interface (UI) is presented that integrates information on the Direction of Arrival (DoA) of the radio signal. The proposed approach consists of (1) a method for estimating the DoA and (2) a color-bar representation surrounding the video feed that informs the operator which navigation directions of motion are safe, even when moving in regions close to the connectivity threshold.

The UI was evaluated in a user study with 24 participants who performed a search task under challenging wireless connectivity conditions. The results show that using the proposed interface resulted in more objects found, and less missions aborted due to connectivity problems, as compared to a standard interface.

Today, teleoperated UGVs play an increasingly important role in a number of high risk applications, including USAR and Explosive Ordinance Disposal (EOD). The successful completion of these missions depend on a reliable communication link between operator and UGV, but unfortunately experiences from Fukushima and the World Trade Center disaster show that cables can limit performance, or break [65], and wireless network connectivity can be lost [62].

Despite improvements in wireless technology, it is reasonable to believe that the very nature of USAR scenarios imply a high risk of damages to infrastructure, including electricity and network facilities. To avoid relying on wireless technology, one possible solution would be to enable the UGVs to operate autonomously, but for the foreseeable future, human operators will remain more versatile than autonomous systems when it comes to decision making, in particular in challenging and unpredictable USAR environments [89, 64]. Therefore, Connectivity awareness is viewed as a component of Situation Awareness (SA), determining where the robot can operate.

In this work, we address the problem of improving SA such that the operator is aware of dynamic network connectivity and adjust the UGV operation to it. This is done by extending the user interface (UI) with not only a measure of Radio Signal Strength (RSS), but also a notion of the motion direction (i.e. the DoA) that would increase this signal strength, and thereby the communication quality (delay, packet loss, etc.) which has

shown to affect teleoperation task performance [73].

Using the proposed solution, an operator close to the connectivity limit knows which way to go to improve the connection. An operator who, for example, would like to move the UGV a bit more to the left to inspect a cavity, knows if this move will improve, worsen or leave the RSS unchanged.

The proposed UI is composed of two parts, first the DoA is estimated, then it is presented to the operator in an efficient manner.

The estimation of the DoA is done by using spatially dispersed wireless receivers on the four edges of the UGV and applying the finite differences method to extract the RSS gradient. We then employ spatial and temporal filtering schemes to mitigate multipath fading effects and transient noises in the measurements. The estimation and filtering algorithms run online and dynamically adapts to the wireless environment such as a change in network connection (e.g. introduction of an intermediate relay robot as a signal repeater) or a mobile wireless access point connecting the robot to the base station.

The presentation of the DoA to the operator was chosen in view of the fact that gaining a good SA is very challenging in USAR missions [47]. In fact, it was shown in [11, 96] that as much as 49% of mission time is normally devoted to improving the operator SA. Further, it was recommended in [97] to use a large central part of the screen for the video feed. Therefore, we propose to add the DoA information in the form of a color bar surrounding the video feed to provide SA to the operator in terms of network connectivity and physical surroundings.

For the evaluation, we identified two important challenges associated with teleoperation of UGVs in USAR missions: (1) providing effective SA to the operator and (2) ensuring resilient wireless connectivity with the UGV. High SA can reduce mission time and improve operator decisions, while a resilient network connection will avoid losing control of the UGV. For details, see the appended paper, [70] (Annex Overview 2.5).

1.3.5 Collaborative Mobile Sensing

The task of collaborative mobile sensing focuses on the collaborative perceptual assessment and exploration of disaster scenes with multiple heterogeneous robots. On one hand this deals with the fusion of heterogeneous data between robots, e.g., different sensor modalities or view-points into common representation. On the other hand, this enables transfer of perceptual data between robots. We have developed a common registration architecture that relies on an abstraction layer which enables the registration of heterogeneous robot data, see Fig. 7. To this end, we have developed several integration strategies, both global and local.

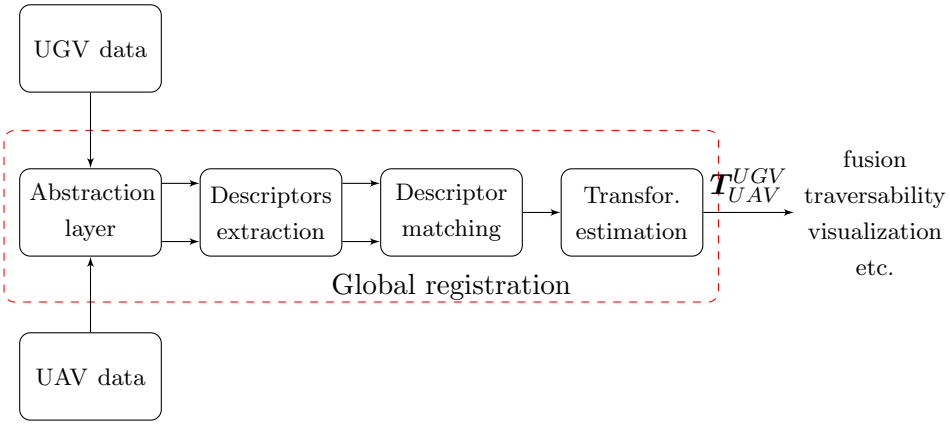
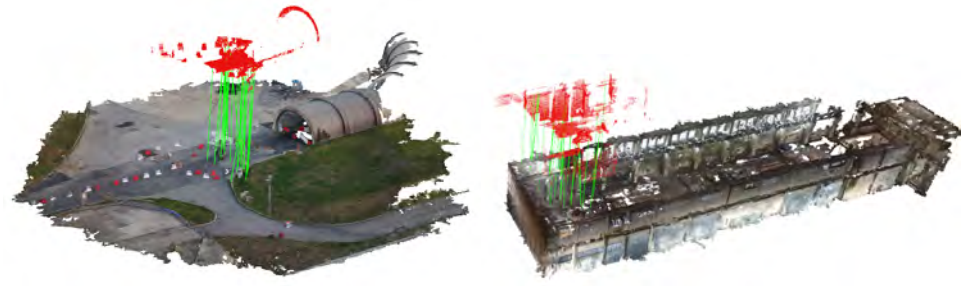


Figure 7: Global localization system overview. The inputs to the system are UAV and UGV data, e.g., from different modalities of drastically different view-points. These are transmitted into an abstraction layer that deals with the heterogeneity of the data, e.g., using 3D structure or semantics. On this data, we extract descriptors, match them and estimate the transformation between the data. Finally, the achieved registration can be used to fuse the data or enable further autonomy of the robots.

3D UAV-UGV Global Localization

Here, we consider the scenario in which a UAV can be quickly deployed to survey a large disaster site. While the image data can help first responders to assess a situation, it can also help the UGVs in their mission planning, e.g., by using 3D reconstructions of the environment for mission planning. In the considered disaster scenarios however, we cannot rely on external sensing for the robots and require techniques to register data from on-board sensing. Furthermore, we cannot directly apply the single-modality approaches used on the UGVs. Since the UGVs rely on 3D data for their traversability analysis of the terrain, a fusion in this space is desirable as the data can directly be used by other modules of the system. We therefore use 3D structure as abstraction layer. In the work of Annex 2.6, we extended our earlier work on sparse 3D data [32] and evaluated different techniques for registering dense reconstructions from UAVs with the UGV LiDAR data. Notably, we evaluated multiple 3D-feature-based registration techniques and devise insights into designing a global localization system based on heterogeneous 3D data. We evaluated the approaches on two dataset gathered in the TRADR project, one outdoor at a test location in Montelebretti, Italy and one indoor at a powerplant in Dortmund, Germany, see Fig. 8. The main benefit of these approaches is that drastically different view-points can globally be fused. We illustrate examples of the matching in Fig. 9.

Figure 8: Example images of the datasets used for 3D registration: (left) Firemen training site Montelebretti, Italy. (right) Gustav Knepper powerplant in Dortmund, Germany.



(a) (b)

Figure 9: Illustration of the 3D geometric registration approach, a LiDAR point cloud from the UGV (red) is matched against the UAV map (color). (a) Outdoor firemen training site Montelebretti, Italy. (b) Indoor Gustav Knepper powerplant in Dortmund, Germany.

Semantic Localization using data from UAVs and UGVs

For large scale scenarios on *km*-scale, registration methods based on 3D structure can become unreliable and computationally increasingly intense. Therefore, we developed *X-View*, a heterogeneous global localization system based on semantic graphs (see Annex 2.8). Instead of using 3D reconstructions, *X-View* leverages recent advances in semantic scene understanding for efficient localization from drastically different view-points, e.g., UAV to UGV. Here, graphs of semantic instances are extracted and matched using random walk descriptors while using a similar estimation back-end as the 3D registration approach presented in the previous section. This enables the system to outperform contemporary localization algorithms on real and simulated dynamic urban outdoor data, especially in the presence of drastically different view-points. Annex 2.8 demonstrates the system implemented for semantic data extracted from RGB images using CNNs, as illustrated in Fig. 10. However, the system is generic in the sense of the used input

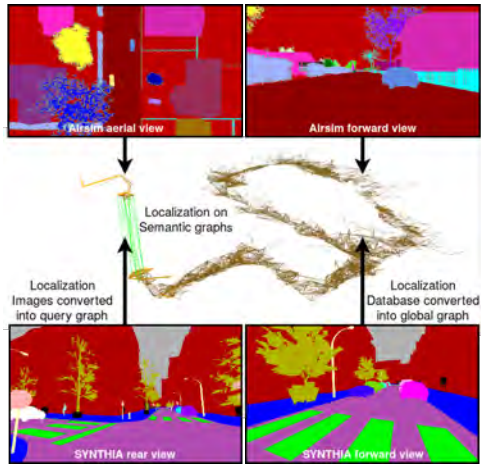


Figure 10: Exemplary approach of the *X-View* system: A database graph is constructed from data of one robot view. Then a small query graph is constructed from another view and localized against the database graph using semantic graph descriptors.

modality, and it can potentially be extended to fusing between modalities in the future.

Aerial following and transportation

While the registration techniques rely on global localization between the robot data, we have also worked on direct collaboration between UGVs and UAVs, by enabling UAVs to apply visual servoing to move above a target and transport samples. This can for example be useful to let the UAV automatically follow the UGV and extend its view point with a 3rd-person view. Furthermore, the outfit of a UAV with a gripper is an interesting feature as it can potentially pick up samples from an in-field UGV and rapidly deliver these outside the disaster area, while the UGV can continue its mission. In addition to contributing to the collaborative sensing, this is furthermore a feature towards the goal of collaborative manipulation, and an example is illustrated in Fig. 11.

1.4 Relation to the state-of-the-art

In this section we will describe how the results of D2.4 relate to the state-of-the-art.

(a) (b) (c) (d)

Figure 11: Illustration of the aerial gripping approach. (a) UAV visually servoes above moving UGV, (b) UAV approaches and picks sample from UGV, (c) UAV ascends to operation height, (d) UAV delivers sample to target location.

1.4.1 Formations for Collaborative Mobile Manipulation

Constraint Based Programming (CBP) is an approach for designing robot controllers that take a set of constraints, in terms of equalities and inequalities into consideration [67]. CBP has its roots in the concepts of the *additional tasks* of [79], the *user defined objective functions* of [71], and the *sub-tasks* of [85]. Similar ideas were used in the *Stack of Tasks* approach [53, 54], the *iTaSC* approach [23, 81], and in a variation using Quadratic programming that was proposed in [103] and [102].

In this paper, we will use CBP for formation control, where the constraints will be given by both the formation keeping objective, and the desired obstacle clearance. The objective function will be the progress towards the next waypoint, and the waypoints are chosen using an Rapidly Exploring Random Tree (RRT) algorithm.

RRTs are sampling based path planners [48],[49] that are known to outperform grid based planners when there are a large number of dimensions, and/or a need for very small grid cells.

The main contribution of this paper is that we combine CBP with RRTs to solve formation obstacle avoidance problems in very cluttered environments. In this way, the local reactive properties of CBP, allowing the formation to rotate and translate past the obstacles, is combined with the global properties of RRT, avoiding local minima problems and efficiently exploring the statespace. To the best of our knowledge, this has not been done before.

1.4.2 Manipulation for Low Visibility Perception

In recent years, a large variety of machine learning techniques have been developed to model and represent complex surfaces, as in [82] and [69]. In particular, probabilistic models based on Gaussian Processes (GP)[74] have proved to be especially suitable for modeling terrain shapes [88, 87, 66]. An application of 3D Gaussian Processes called Gaussian Process Implicit Surfaces (GPIS) [91] allows to extend implicit surfaces to include uncertainty,

a property needed when the model is the result of sensor data fusion [25]. Properly representing the morphology of the terrain is vital for navigation tasks in any UGV, as demostrated by the authors in [105] who developed an algorithm that allows a mobile robot to autonomously traverse natural complex obstacles in a forest. Occlusion and reflections, e.g. caused by a puddle of water or broken glass, can trick the vision system of the robot to generate incomplete point clouds, and lead the system to make wrong assumption on the nature of the terrain.

In our works we propose two *interactive perception*[9] frameworks based on Gaussian Processes, that merge visual and tactile information in order to identify and reconstruct incomplete regions in the point cloud map and estimate the deformability behavior of a portion of terrain.

When the visual perception system of the robot fails, we ask a robotic arm to strategically explore the environment around the UGV and collect tactile data.

In order to estimate the *heterogeneous* deformability of the terrain we take advantage of a fast Position-Based Dynamics (PBD) simulator as opposed to most of the existing methods which rely on computationally expensive force based simulators [30].

Additionally, our systems do not require a complex setup as the one in [6], but uses only a mobile manipulator equipped with a tactile force sensor.

The computational power on board the robot, the battery capacity and the available time are key constraints in many urban search and rescue mission. For this reasons, the perception frameworks proposed in our work aim to reduce the environmental interactions as much as possible while reconstructing the surface and estimating its deformability.

1.4.3 Network Aware Path Planning

To address this problem, several researchers have focused on Communication-Aware Motion Planning (CAMP) to simultaneously handle motion and communication constraints and finding and executing an optimal path towards a destination [101]. In particular, Mostofi et al. laid solid foundations in this research area [37, 52, 95]. It can be noted that most previous works consider either a binary or a disk based connectivity model, or an accurate communication model to optimize the robots motion and communication energy without focusing on resilience. Additionally, none of the previous works explicitly addresses the problem of efficiently re-establishing the communication in case of a connection loss, in ways beyond straightforward backtracking.

In this paper, we propose a Resilient Communication-Aware Motion Planner (RCAMP) that combines two key elements: 1) a Gaussian Random Field (GRF) based probabilistic model to map the Radio Signal Strength

(RSS) of an unknown environment and use it to predict the communication quality of the planned path; 2) a motion planning strategy that starting from sensory information (such as LIDAR), computes a traversability map for a given robot taking into account environmental constraints. Additionally we propose a strategy to autonomously repair a communication loss by steering the robot towards a communication-safe position using the proposed RCAMP.

Specifically, inspired by [27], we use GRFs for dynamically mapping the heterogeneous distribution of the RSS. We then merge this online framework with a motion planner

- to obtain a semi-optimal path considering both communication and motion constraints, and
- to quickly re-establish connection in case of signal loss.

We demonstrate the feasibility of our approach through extensive simulations on a variety of use cases that reproduce realistic wireless network changes (e.g. a sudden connection loss) in single and multi-channel set-ups. Contrarily to many existing methods [52, 24], we do not assume any prior knowledge about the positions and the number of transmission sources or the dielectric characteristics of the obstacles in the environment. The main advantages of our planner compared to others are the response to dynamic changes in the network configuration (e.g. disruptions or movement in Access Points) or in the environment (e.g. path planning in presence of dynamic obstacles) and the fact that we do not require prior knowledge of the network, such as the location of the Access Points. We propose a fully online, dynamic and reactive CAMP that adapts to recent sensory information.

1.4.4 Network Aware Teleoperation, Design and User Study

The main contributions of this work are three-fold. We first propose a new way of estimating DoA for teleoperated UGVs. We then propose a way of integrating this DoA information in a UGV teleoperation UI. Lastly, we perform a user study, showing that the proposed approach in fact increases the number of found objects during a search mission, and decreases the chances of losing the connection to the UGV. To the best of our knowledge, none of these items have been done in a UGV teleoperation context before.

The wireless network connectivity of USAR UGVs have often proved unreliable [63, 17], with examples including real incidents where robots were lost during disaster inspection operations [65, 62]. Casper et al. [18] investigated user confidence in remotely operated robots with intermittent communications, and found that these problems had a significant impact on the usability of the systems. They even suggested that because of communication dropout problems, wireless robots should be avoided. However, the

flexibility of wireless systems compared to tethered robots still make them an important alternative in many applications.

A natural way of avoiding loss of communications is to make the user aware of the connection quality. A decade ago, this information was usually not displayed in the Operator Control Unit (OCU) [28], but more recently, it is often added in the form of a "signal bar" (as in modern cell phones) or in form of a percentage. Typical examples of such representation can be seen in [46, 38] including the recent Quince 2 robot's OCU [99]. Furthermore, the Wayfarer OCU for Packbot robots [94] represent the radio signal level in a vertical bar manner, in addition to a numeric indicator.

The literature on robot interfaces also include examples where information about gradients and directions is made available to the user. In [39] two microphones on the left and right of the robot were used to estimate the direction of a sound source, which was displayed (overlaid on the video) in the form of a pointer floating on a horizontal line. A similar representation was used in [38] to show robot speed information. In [22], the authors proposed a tactile belt that vibrates in the direction of detected collisions to improve SA, while in [80] a study found that the use of a tactile vest did not improve SA significantly in navigation tasks.

An influential study in Human-Robot Interface (HRI) design [97] advocates the use of a large single interface with a significant percentage of the screen dedicated to video. The authors also recommend providing more spatial information about the environment to increase SA, and using fused sensor information to lower the cognitive load on user.

In this work we go beyond the related work described above by having the teleoperation interface include not only a scalar value to describe the network connectivity situation, but also the direction in which it is expected to improve, i.e. the DoA. Assessing the geographic distribution of network connectivity is a spatial task, for which the visual modality fits best with the human information processing (e.g., see the multi-resource model of Wickens [90]). Therefore we choose to present the DoA in the form of visual gradient bars surrounding the video feedback.

Carefully integrating the DoA information into the visual feedback is crucial. For this we use FLC (Free Look Control) [68] as the control layer. FLC is essentially a "navigate-by-camera" mode as envisioned in [98]. In the FLC mode, the operator controls the UGV in relation to the camera frame instead of the world frame, making it more intuitive than the traditional so-called *Tank Control* mode. Hence it is appropriate to use FLC for presenting the DoA information in direct reference to the camera frame, making the UGV control easier while simultaneously enhancing local SA. The proposed DoA interface integrated with FLC indeed has the potential to satisfy all the three levels of SA (perception, comprehension, and prediction/projection) mentioned in [26].

1.4.5 Heterogeneous global localization

Heterogeneous global localization is a difficult task for robotic systems. In contrast to homogeneous systems, the assumptions of same sensor modality or similar view-point does not hold anymore. This review is based on our literature reviews in [33], and [35].

A common approach to global localization is visual feature matching. A large amount of approaches have been proposed in the last decade, giving reliable performance under perceptually similar conditions [31, 50, 21]. Several extensions have been proposed to overcome perceptually difficult situations, such as seasonal changes [61, 20], daytime changes [12, 1], or varying view-points using CNN landmarks [84, 19]. However, drastic view-point invariance, e.g., between views from aerial and ground robots continues to be a challenging problem for appearance-based techniques.

The field of *2D* metrical map-merging based on overlapping map segments is well studied in the literature [7, 8, 77**?**]. However, the task is increasingly difficult when moving to *3D* environments [78], especially when dealing with heterogeneous robotic teams, where *3D* data is generated from different sensors and with different noise characteristics [16].

Michael et al. [60] demonstrate a system for collaborative UAV-UGV mapping. The authors propose a system where a UGV equipped with a LiDAR sensor performs *2.5D* mapping, using the flat ground assumption and consecutively merging scans using ICP. In dedicated locations, a UAV equipped with a 2D LiDAR is launched from the UGV and maps the environment using a pose-graph SLAM algorithm. Maps generated from the UAV are then fused online with the UGV map using ICP initialized at the UAV starting location.

Forster et al. [29] go a step further in fusing UAV-UGV map data from different sensors, i.e., RGB-D maps from the UGV and dense monocular reconstruction from the UAV. The registration between the maps is performed using a *2D* local height map fitting in x and y coordinates with an initial guess within a $3m$ search radius. The orientation is a priori recovered from the magnetic north direction as measured by the IMUs. In a related setting, Hinzmann et al. [40] evaluate different variants of ICP for registering dense *3D* LiDAR point-clouds and sparse *3D* vision point-clouds from SfM recorded with different UAVs into a common point-cloud map using an initial GPS prior for the map alignment.

Instead of using the generated *3D* data for localizing between RGB and *3D* LiDAR point-cloud data, Wolcott and Eustice [92] propose to generate *2D* views from the LiDAR point-clouds based on the surface reflectivity. However, this work focuses only on localization and it is demonstrated only on maps recorded from similar points of view.

In our previous work [32] we presented a global registration scheme between sparse *3D* LiDAR maps from UGVs and vision keypoint maps from

UAVs, exploiting the rough geometric structure of the environment. Here, registration is performed by clustering of geometric keypoint descriptors matches between map segments under the assumption of a known z-direction as determined by an IMU.

Zeng et al. [100] present geometric descriptor matching based on learning. However, this approach is infeasible in unknown SaR scenarios, as the descriptors do not generalize well to unknown environments.

Assuming good initialization of the global registration, Zhou et al. [104] perform a robust optimization. The work claims faster and more robust performance than ICP.

Other approaches to global localization are based on topological mapping [41, 56]. Here, maps are represented as graphs encoding relationships between vertices. While these works focus on graph merging by exhaustive vertex matching on small graphs, they do not consider graph extraction from sensory data or ambiguous vertices. Furthermore, the computationally expensive matching does not scale to larger graph comparisons.

With the recent advances in learning-based semantic extraction methods, using semantics for localization is a promising avenue [**?** 45, 10, 2]. In [10, 2] the authors focus on the *data association* problem for semantic localization using EM and the formulation of the pose estimation problem for semantic constraints as an error minimization. The semantic extraction is based on a standard object detector from visual key-points.

Stumm et al. [83] propose to use graph kernels for place recognition on visual key-point descriptors. Graph kernels are used to project image-wise covisibility graphs into a feature space. The authors show that graph descriptions can help localization performance as to efficiently cluster multiple descriptors meaningfully. However, the use of large densely connected graphs sets limitations to the choice of graph representation.

In summary, the community addresses the problem of heterogeneous global localization. However, there is a research gap in globally localizing from one sensor modality to the other in full *3D* without strong assumptions on view-point, terrain or initial guess. We have identified the use of semantic information [35] or *3D* structure [33] as promising research avenues to address these challenges.

1.4.6 Aerial transportation

We focus our review of related work on recent advances in aerial gripping and servo positioning techniques for reliably detecting and approaching objects using a MAV. The review is adopted from [34].

In [36] the authors propose an integrated object detection and gripping system for MAVs using IR diodes for detection and a mechanical gripper for gripping stationary objects. In contrast, our system aims to detect objects using a standard RGB camera and also grip moving objects with an partly

ferrous surface.

Transportation of objects using MAVs was reported in [59, 57, 75]. However, the authors mainly focus on the control of MAVs transporting objects. In contrast to our work they do not implement a grip and release mechanism which is an important aspect for fully autonomous delivery.

An aerial manipulation task using a quadrotor with a two DOF robotic arm was presented in [43]. The kinematic and dynamic models of the combined system were developed and an adaptive controller was designed in order to perform a pick and place task. Such system offers high manipulability, however, the shape of the objects to be picked is limited since the robotic arm is only able to pick thin objects in specific configurations, i.e., thin surfaces pointing upwards. Furthermore, this work assumes that the position of the object to be picked is known in advance.

A self-sealing suction technology for grasping was tested in [42]. A system capable of grasping multiple objects with various textures, curved and inclined surfaces, was demonstrated. Despite being able to achieve high holding forces, the gripping system requires a heavy compressor and an activation threshold force to pick up the objects. Also, all the tests were performed using a motion capture system with known object positions.

Another type of mechanical gripper was shown in [58]. The gripper uses servo motors to actuate the pins that penetrate the object and create a strong and secure connection. A similar design was also presented in [3]. The main limitation of such a gripper is its restriction to pick only objects with a penetrable surface. Furthermore, if the surface is not elastically deformable, the gripper might cause irreversible damage to the object.

In [93], a bio-inspired mechanical gripper was designed in order to allow quadcopters to carry objects with large flat or gently curved surfaces. In addition to being small and light, the gripper consists of groups of tiles coated with a controllable adhesive that allows for very easy attachment and detachment of the object. Nevertheless, the gripper is limited to smooth surfaces, requires tendon mechanism for attachment, and has a limited payload.

OpenGrab EPM[2] is a gripper developed using the principle of electropermanent magnets [44]. It is a low-weight, energy efficient and high-payload solution developed for robotic applications and because of its advantages, we have decided to use the same principle for our own gripper. Since OpenGrab EMP is only able to pick flat surfaces, we have developed a more sophisticated design which allows our gripper to pick objects with curved surfaces, while maintaining an equal load distribution on all contacts between object and gripper.

Visual Servoing (VS) is a well established technique where information extracted from images is used to control the robot motion [72, 51, 36]. There are many approaches to deal with VS, however some of the most popular

[2]http://nicadrone.com/

include image based VS and pose based VS. In the vision based approach, the control law is based entirely on the error in the image plane, no object pose estimation is performed. In [76] the authors employ this method to perform pole inspection with MAVs, while in [86] it is used to bring a MAV to a perching position, hanging from a pole.

In the pose based approach, the object pose is estimated from the image stream, then the robot is commanded to move towards the object to perform grasping or an inspection task for instance [55].

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[2] Nikolay Atanasov, Menglong Zhu, Kostas Daniilidis, and George J Pappas. Semantic localization via the matrix permanent. In *Robotics: Science and Systems*, 2014.

[3] Federico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W. Mueller, Jan Willmann, Fabio Gramazio, Matthias Kohler, and Raffaello DAndrea. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems*, 34(4):46–64, 2014.

[4] Fredrik Båberg and Petter Ögren. Formation obstacle avoidance using rrt and constraint based programming. In *Safety, Security and Rescue Robotics (SSRR), 2017 IEEE International Symposium on*, pages 1–6. IEEE, 2017.

[5] Fredrik Båberg, Sergio Caccamo, Nanja Smets, Mark Neerincx, and Petter Ögren. Free look ugv teleoperation control tested in game environment: Enhanced performance and reduced workload. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 312–319. IEEE, 2016.

[6] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. Capture and modeling of nonlinear heterogeneous soft tissue. *ACM Trans. Graph.*, 28(3):89:1–89:9, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531395. URL `http://doi.acm.org/10.1145/1531326.1531395`.

[7] Andreas Birk and Stefano Carpin. Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, pages 1384–1397, 2006.

[8] Jose-Luis Blanco, Javier González-Jiménez, and Juan-Antonio Fernández-Madrigal. A robust, multi-hypothesis approach to matching occupancy grid maps. *Robotica*, pages 687–701, 2013.

[9] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, Dec 2017. ISSN 1552-3098. doi: 10.1109/TRO.2017.2721939.

[10] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017.

[11] J. Burke, RR Murphy, M. Coovert, and D. Riddle. Moonlight in Miami: An ethnographic study of human-robot interaction in USAR. *Human-Computer Interaction, special issue on Human-Robot Interaction*, 19:1–2, 2004.

[12] Mathias Bürki, Igor Gilitschenski, Elena Stumm, Roland Siegwart, and Juan Nieto. Appearance-based landmark selection for efficient long-term visual localization. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4137–4143, 2016.

[13] S. Caccamo, R. Parasuraman, L. Freda, M. Gianni, and P. Ögren. Rcamp: A resilient communication-aware motion planner for mobile robots with autonomous repair of wireless connectivity. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2010–2017, Sept 2017. doi: 10.1109/IROS.2017.8206020.

[14] Sergio Caccamo, Yasemin Bekiroglu, Carl Henrik Ek, and Danica Kragic. Active exploration using gaussian random fields and gaussian process implicit surfaces. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 582–589. IEEE, 2016.

[15] Sergio Caccamo, Püren Güler, Hedvig Kjellström, and Danica Kragic. Active perception and modeling of deformable surfaces using gaussian processes and position-based dynamics. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 530–537. IEEE, 2016.

[16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous

localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[17] J. Carlson and Robin R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437, 2005.

[18] Jennifer Casper and Robin R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(3):367–385, 2003.

[19] Zetao Chen, Adam Jacobson, Niko Sunderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[20] Titus Cieslewski, Elena Stumm, Abel Gawel, Mike Bosse, Simon Lynen, and Roland Siegwart. Point cloud descriptors for place recognition using sparse visual information. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4830–4836, 2016.

[21] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[22] Paulo G De Barros, Robert W Lindeman, and Matthew O Ward. Enhancing robot teleoperator situation awareness and performance using vibro-tactile and graphical feedback. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 47–54. IEEE, 2011.

[23] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes, and Herman Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455, 2007.

[24] Anna Derbakova, Nikolaus Correll, and Daniela Rus. Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3863–3868. IEEE, 2011.

[25] Mohammed A El-Beltagy and W Andy Wright. Gaussian processes for model fusion. In *Artificial Neural NetworksICANN 2001*, pages 376–383. Springer, 2001.

[26] Mica R Endsley et al. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, pages 3–32, 2000.

[27] A. Fink, H. Beikirch, M. Voss, and C. Schroder. RSSI-based indoor positioning using diversity and Inertial Navigation. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2010.

[28] Terrence Fong and Charles Thorpe. Vehicle teleoperation interfaces. *Autonomous robots*, 11(1):9–18, 2001.

[29] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *IROS*, pages 3971–3978, 2013.

[30] Barbara Frank, Cyrill Stachniss, Rdiger Schmedding, Matthias Teschner, and Wolfram Burgard. Learning object deformation models for robot motion planning. *Robotics and Autonomous Systems*, 62(8): 1153 – 1174, 2014. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2014.04.005. URL `http://www.sciencedirect.com/science/article/pii/S0921889014000797`.

[31] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[32] Abel Gawel, Titus Cieslewski, Renaud Dubé, Mike Bosse, Roland Siegwart, and Juan Nieto. Structure-based vision-laser matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 182–188, 2016.

[33] Abel Gawel, Renaud Dubé, Hartmut Surmann, Juan Nieto, Roland Siegwart, and Cesar Cadena. 3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation. In *IEEE International Symposium on Safety, Security*, 2017.

[34] Abel Gawel, Mina Kamel, Tonci Novkovic, Jakob Widauer, Benjamin Pfyffer von Altishofen, Roland Siegwart, and Juan Nieto. Aerial picking and delivery of magnetic objects with mavs. In *IEEE International Conference on Robotics and Automation*, 2017.

[35] Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieta, and Cesar Cadena. X-view: Graph-based semantic multi-view localization. In *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[36] Vaibhav Ghadiok, Jeremy Goldin, and Wei Ren. Autonomous indoor aerial gripping using a quadrotor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[37] Aa Ghaffarkhah and Ya Mostofi. Channel learning and communication-aware motion planning in mobile networks. In *American Control Conference (ACC), 2010*, pages 5413–5420, June 2010.

[38] Andreas Hedström, Henrik I Christensen, and Carl Lundberg. A wearable gui for field robots. In *Field and Service Robotics*, pages 367–376. Springer, 2006.

[39] D. Hestand and H.A. Yanco. Layered sensor modalities for improved human-robot interaction. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2966–2970 vol.3, Oct 2004. doi: 10.1109/ICSMC.2004.1400784.

[40] Timo Hinzmann, Thomas Stastny, Gianpaolo Conte, Patrick Doherty, Piotr Rudol, Marius Wzorek, Enric Galceran, Roland Siegwart, and Igor Gilitschenski. Collaborative 3d reconstruction using heterogeneous uavs: System and experiments. In *International Symposium on Experimental Robotics*, pages 43–56. Springer, 2016.

[41] Wesley H Huang and Kristopher R Beevers. Topological map merging. *The International Journal of Robotics Research*, 24(8):601–613, 2005.

[42] Chad C Kessens, Justin Thomas, Jaydev P Desai, and Vijay Kumar. Versatile aerial grasping using self-sealing suction. In *IEEE International Conference on Robotics and Automation*, 2016.

[43] Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[44] Ara Nerses Knaian. *Electropermanent Magnetic Connectors and Actuators: Devices and Their Application in Programmable Matter*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2010.

[] Philipp Koch, Stefan May, Michael Schmidpeter, Markus Kühn, Christian Pfitzner, Christian Merkl, Rainer Koch, Martin Fees, Jon Martin, Daniel Ammon, et al. Multi-robot localization and mapping based on signed distance functions. *Journal of Intelligent & Robotic Systems*, 83(3-4):409–428, 2016.

[45] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015.

[46] B Larochelle, Geert-Jan Kruijff, N Smets, T Mioch, and P Groenewegen. Establishing human situation awareness using a multi-modal op-

erator control unit in an urban search & rescue human-robot team. *RO-MAN, 2011 IEEE*, pages 229–234, 2011.

[47] Benoit Larochelle and GM Kruijff. Multi-view operator control unit to improve situation awareness in usar missions. In *RO-MAN, 2012 IEEE*, pages 1103–1108. IEEE, 2012.

[48] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[49] Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[50] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[51] Ezio Malis and Patrick Rives. Robustness of image-based visual servoing with respect to depth distribution errors. In *IEEE International Conference on Robotics and Automation*, 2003.

[52] Ma Malmirchegini and Ya Mostofi. On the spatial predictability of communication channels. *IEEE Transactions on Wireless Communications*, 11(3):964–978, March 2012.

[53] Nicolas Mansard and Franois Chaumette. Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics*, 23(1):60–72, 2007.

[54] Nicolas Mansard, Oussama Khatib, and Abderrahmane Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.

[55] Eric Marchand, Patrick Bouthemy, Francois Chaumette, and Valerie Moreau. Robust visual tracking by coupling 2d motion and 3d pose estimation. In *IEEE International Conference on Image Processing*, 1999.

[56] Dimitri Marinakis and Gregory Dudek. Pure topological mapping in mobile robotics. *IEEE Transactions on Robotics*, 26(6):1051–1064, 2010.

[57] Ivan Maza, Konstantin Kondak, Markus Bernard, and Aníbal Ollero. Multi-uav cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, 57(1-4):417–449, 2010.

[58] Daniel Mellinger, Quentin Lindsey, Michael Shomin, and Vijay Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[59] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[60] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *JFR*, pages 832–841, 2012.

[61] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649, 2012.

[62] Robin R Murphy. *Disaster Robotics*. MIT Press, 2014.

[63] R.R. Murphy. Human–Robot Interaction in Rescue Robotics. *IEEE Transactions on systems, Man, and Cybernetics, Part C: Application and Reviews*, 34(2), 2004.

[64] Sebastian Muszynski, Jorg Stuckler, and Sven Behnke. Adjustable autonomy for mobile teleoperation of personal service robots. *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 933–940, September 2012. doi: 10.1109/ROMAN.2012.6343870.

[65] 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. ISSN 1556-4967. doi: 10.1002/rob.21439.

[] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[66] S. O'Callaghan, F.T. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using gaussian processes. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1054–1060, May 2009. doi: 10.1109/ROBOT.2009.5152754.

[67] Petter Ögren and John WC Robinson. A model based approach to modular multi-objective robot control. *Journal of Intelligent & Robotic Systems*, 63(2):257–282, 2011.

[68] Petter Ögren, Peter Svenmarck, Patrik Lif, Martin Norberg, and Nils Emil Söderbäck. Design and Implementation of a New Teleoperation Control Mode for Differential Drive UGVs. *Autonomous Robots*, 2014.

[69] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882293. URL `http://doi.acm.org/10.1145/882262.882293`.

[70] Ramviyas Parasuraman, Sergio Caccamo, Fredrik Båberg, Mark Neerincx, and Petter Ögren. A new ugv teleoperation interface for improved awareness of network connectivity and physical surroundings. *International Journal of Human Robot Interaction*, 2017.

[71] Z-X Peng and Norihiko Adachi. Compliant motion control of kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 9(6):831–836, 1993.

[72] Roger Pissard-Gibollet and Patrick Rives. Applying visual servoing techniques to control a mobile hand-rye system. In *IEEE International Conference onRobotics and Automation*, 1995.

[73] M. Rank, Z. Shi, H. J. Mller, and S. Hirche. Predictive communication quality control in haptic teleoperation with time delay and packet loss. *IEEE Transactions on Human-Machine Systems*, 46(4):581–592, Aug 2016. ISSN 2168-2291. doi: 10.1109/THMS.2016.2519608.

[74] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 13978-0-262-18253-9. URL `http://www.gaussianprocess.org/gpml/chapters/`.

[75] Robin Ritz and Raffaello D'Andrea. Carrying a flexible payload with multiple flying vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[76] Inkyu Sa, Stefan Hrabar, and Peter Corke. Inspection of pole-like structures using a vision-controlled vtol uav and shared autonomy. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[77] Sajad Saeedi, Liam Paull, Michael Trentini, and Howard Li. Multiple robot simultaneous localization and mapping. In *IROS*, pages 853–858, 2011.

[78] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *JFR*, pages 3–46, 2016.

[79] Homayoun Seraji. Configuration control of redundant manipulators: Theory and implementation. *IEEE Transactions on Robotics and Automation*, 5(4):472–490, 1989.

[80] Nanja JJM Smets, Guido M te Brake, Mark A Neerincx, and Jasper Lindenberg. Effects of mobile map orientation and tactile feedback on navigation speed and situation awareness. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 73–80. ACM, 2008.

[81] Ruben Smits, Tinne De Laet, Kasper Claes, Herman Bruyninckx, and Joris De Schutter. itasc: A tool for multi-sensor integration in robot manipulation. In *Multisensor Fusion and Integration for Intelligent Systems*, pages 235–254. Springer, 2009.

[82] Florian Steinke, Bernhard Schlkopf, and Volker Blanz. Support vector machines for 3d shape processing. *Computer Graphics Forum*, 24(3): 285–294, 2005. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2005.00853.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2005.00853.x`.

[83] Elena Stumm, Christopher Mei, Simon Lacroix, Juan Nieto, Marco Hutter, and Roland Siegwart. Robust visual place recognition with graph kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4535–4544, 2016.

[84] Niko Sunderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and Systems*, 2015.

[85] Enver Tatlicioglu, David Braganza, Timothy C Burg, and Darren M Dawson. Adaptive control of redundant robot manipulators with sub-task objectives. In *American Control Conference, 2008*, pages 856–861. IEEE, 2008.

[86] Justin Thomas, Giuseppe Loianno, Kostas Daniilidis, and Vijay Kumar. Visual servoing of quadrotors for perching by hanging from cylindrical objects. *IEEE Robotics and Automation Letters*, 1(1):57–64, 2016.

[87] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair. Gaussian process modeling of large scale terrain. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1047–1053, May 2009. doi: 10.1109/ROBOT.2009.5152677.

[88] Shrihari Vasudevan. Data fusion with gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528 – 1544, 2012. ISSN 0921-8890. doi: 10.1016/j.robot.2012.08.006. URL `http://www.sciencedirect.com/science/article/pii/S0921889012001388`.

[89] R. Wegner and J. Anderson. Agent-Based Support for Balancing Teleoperation and Autonomy in Urban Search and Rescue. *International Journal of Robotics and Automation*, 21(2):1–19, 2006. ISSN 1925-7090. doi: 10.2316/Journal.206.2006.2.206-2796.

[90] Christopher D Wickens. Multiple resources and mental workload. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):449–455, 2008.

[91] O. Williams and A. Fitzgibbon. Gaussian process implicit surfaces. *Gaussian Proc. in Practice*, 2007.

[92] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *IROS*, pages 176–183, 2014.

[93] Elliot Wright Hawkes, Hao Jiang, and Mark R. Cutkosky. Three-dimensional dynamic surface grasping with dry adhesion. *International Journal of Robotics Research*, 35(8):943–958, 2016.

[94] Brian M Yamauchi. Packbot: a versatile platform for military robotics. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics, 2004.

[95] Yuan Yan and Yasamin Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Transactions on Wireless Communications*, 12(4):1562–1572, 2013.

[96] H.A. Yanco and J. Drury. Where Am I? Acquiring Situation Awareness Using a Remote Robot Platform. *IEEE Conference on Systems, Man and Cybernetics*, 2004.

[97] Holly A Yanco and Jill L Drury. Rescuing interfaces: A multi-year study of human-robot interaction at the aaai robot rescue competition. *Autonomous Robots*, 22(4):333–352, 2007.

[98] Holly A Yanco, Michael Baker, Robert Casey, Brenden Keyes, Philip Thoren, Jill L Drury, Douglas Few, Curtis Nielsen, and David Bruemmer. Analysis of human-robot interaction for urban search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 22–24, 2006.

[99] Tomoaki Yoshida, Keiji Nagatani, Satoshi Tadokoro, Takeshi Nishimura, and Eiji Koyanagi. Improvements to the rescue robot quince toward future indoor surveillance missions in the fukushima daiichi nuclear power plant. In *Field and Service Robotics*, pages 19–32. Springer, 2014.

[100] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[101] Bo Zhang, Yunlong Wu, Xiaodong Yi, and Xuejun Yang. Joint communication-motion planning in wireless-connected robotic networks: Overview and design guidelines. *arXiv preprint arXiv:1511.02299*, 2015.

[102] Yunong Zhang and Shugen Ma. Minimum-energy redundancy resolution of robot manipulators unified by quadratic programming and its online solution. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 3232–3237. IEEE, 2007.

[103] Yunong Zhang, Shuzhi Sam Ge, and Tong Heng Lee. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2126–2132, 2004.

[104] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *ECCV*, pages 766–782, 2016.

[105] Karsten Zimmermann, Petr Zuzanek, Michal Reinstein, and Vaclav Hlavac. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5177–5182. IEEE, 2014.

2 Annexes

2.1 Baberg (2017), "Formation Obstacle Avoidance using RRT and Constraint Based Programming"

Bibliography Baberg and Ögren "Formation Obstacle Avoidance using RRT and Constraint Based Programming" In *Proceedings of the IEEE SSRR 2017*, Shainghai, China, 2017.

Abstract In this paper, we propose a new way of doing formation obstacle avoidance using a combination of Constraint Based Programming (CBP) and Rapidly Exploring Random Trees (RRTs). RRT is used to select waypoint nodes, and CBP is used to move the formation between those nodes, reactively rotating and translating the formation to pass the obstacles on the way. Thus, the CBP includes constraints for both formation keeping and obstacle avoidance, while striving to move the formation towards the next waypoint. The proposed approach is compared to a pure RRT approach where the motion between the RRT waypoints is done following linear interpolation trajectories, which are less computationally expensive then the CBP ones. The results of a number of challenging simulations show that the proposed approach is more efficient in scenarios with high obstacle densities.

Relation to WP Formation obstacle avoidance is an enabling technology for doing collaborative manipulation, in particular for collaboratively lifting and moving large objects.

Availability Unrestricted.

2.2 Caccamo (2016), "Active Perception and Modeling of Deformable Surfaces using Gaussian Processes and Position-based Dynamics"

Bibliography Caccamo, Guler, Kjellstrom, and Kragic. "Active Perception and Modeling of Deformable Surfaces using Gaussian Processes and Position-based Dynamics" In *Proceedings of the IEEE Humanoids 2016*.

Abstract Exploring and modeling heterogeneous elastic surfaces requires multiple interactions with the environment and a complex selection of physical material parameters. The most common approaches model deformable properties from sets of offline observations using computationally expensive force-based simulators. In this work we present an online probabilistic framework for autonomous estimation of a deformability distribution map of heterogeneous elastic surfaces from few physical interactions. The method

takes advantage of Gaussian Processes for constructing a model of the environment geometry surrounding a robot. A fast Position-based Dynamics simulator uses focused environmental observations in order to model the elastic behavior of portions of the environment. Gaussian Process Regression maps the local deformability on the whole environment in order to generate a deformability distribution map. We show experimental results using a PrimeSense camera, a Kinova Jaco2 robotic arm and an Optoforce sensor on different deformable surfaces.

Relation to WP This paper describes a mobile manipulation approach applicable in low visibility situations.

Availability Unrestricted.

2.3 Caccamo (2016), "Active Exploration Using Gaussian Random Fields and Gaussian Process Implicit Surfaces"

Bibliography Caccamo, Bekiroglu, Ek, and Kragic "Active Exploration Using Gaussian Random Fields and Gaussian Process Implicit Surfaces". In *Proceedings of the IEEE IROS 2016*.

Abstract In this work we study the problem of exploring surfaces and building compact 3D representations of the environment surrounding a robot through active perception. We propose an online probabilistic framework that merges visual and tactile measurements using Gaussian Random Field and Gaussian Process Implicit Surfaces. The system investigates incomplete point clouds in order to find a small set of regions of interest which are then physically explored with a robotic arm equipped with tactile sensors. We show experimental results obtained using a PrimeSense camera, a Kinova Jaco2 robotic arm and Optoforce sensors on different scenarios. We then demostrate how to use the online framework for object detection and terrain classification.

Relation to WP This paper describes a mobile manipulation approach applicable in low visibility situations.

Availability Unrestricted.

2.4 Caccamo (2017), "RCAMP: A Resilient Communication-Aware Motion Planner for Mobile Robots with Autonomous Repair of Wireless Connectivity"

Bibliography Caccamo, Parasuraman, Freda, Gianni and Ögren "RCAMP: A Resilient Communication-Aware Motion Planner for Mobile Robots with

Autonomous Repair of Wireless Connectivity" In *Proceedings of the IEEE IROS 2017.*

Abstract Mobile robots, be it autonomous or teleoperated, require stable communication with the base station to exchange valuable information. Given the stochastic elements in radio signal propagation, such as shadowing and fading, and the possibilities of unpredictable events or hardware failures, communication loss often presents a significant mission risk, both in terms of probability and impact, especially in Urban Search and Rescue (USAR) operations. Depending on the circumstances, disconnected robots are either abandoned, or attempt to autonomously back-trace their way to the base station. Although recent results in Communication-Aware Motion Planning can be used to effectively manage connectivity with robots, there are no results focusing on autonomously re-establishing the wireless connectivity of a mobile robot without back-tracing or using detailed a priori information of the network. In this paper, we present a robust and online radio signal mapping method using Gaussian Random Fields, and propose a Resilient Communication-Aware Motion Planner (RCAMP) that integrates the above signal mapping framework with a motion planner. RCAMP considers both the environment and the physical constraints of the robot, based on the available sensory information. We also propose a self-repair strategy using RCMAP, that takes both connectivity and the goal position into account when driving to a connection-safe position in the event of a communication loss. We demonstrate the proposed planner in a set of realistic simulations of an exploration task in single or multi-channel communication scenarios.

Relation to WP This paper describes network aware UGV mission planning.

Availability Unrestricted.

2.5 Parasuraman (2017), "A New UGV Teleoperation Interface for Improved Awareness of Network Connectivity and Physical Surroundings"

Bibliography Ramviyas Parasuraman, Sergio Caccamo, Fredrik Båberg, Mark Neerincx, Petter Ögren. "A New UGV Teleoperation Interface for Improved Awareness of Network Connectivity and Physical Surroundings." In *Journal of Human Robot Interaction 2017.*

Abstract A reliable wireless connection between the operator and the teleoperated unmanned ground vehicle (UGV) is critical in many urban search and rescue (USAR) missions. Unfortunately, as was seen in, for example,

the Fukushima nuclear disaster, the networks available in areas where USAR missions take place are often severely limited in range and coverage. Therefore, during mission execution, the operator needs to keep track of not only the physical parts of the mission, such as navigating through an area or searching for victims, but also the variations in network connectivity across the environment. In this paper, we propose and evaluate a new teleoperation user interface (UI) that includes a way of estimating the direction of arrival (DoA) of the radio signal strength (RSS) and integrating the DoA information in the interface. The evaluation shows that using the interface results in more objects found, and less aborted missions due to connectivity problems, as compared to a standard interface. The proposed interface is an extension to an existing interface centered on the video stream captured by the UGV. But instead of just showing the network signal strength in terms of percent and a set of bars, the additional information of DoA is added in terms of a color bar surrounding the video feed. With this information, the operator knows what movement directions are safe, even when moving in regions close to the connectivity threshold.

Relation to WP This paper describes network aware UGV teleoperation.

Availability Unrestricted.

2.6 Gawel (2017a), "3D registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation"

Bibliography Abel Gawel, Renaud Dube, Hartmut Surmann, Juan Nieto, Roland Siegwart, Cesar Cadena. "3D registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation." In *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), 2017.*

Abstract Global registration of heterogeneous ground and aerial mapping data is a challenging task. This is especially difficult in disaster response scenarios when we have no prior information on the environment and cannot assume the regular order of man-made environments or meaningful semantic cues. In this work we extensively evaluate different approaches to globally register UGV generated 3D point-cloud data from LiDAR sensors with UAV generated point-cloud maps from vision sensors. The approaches are realizations of different selections for: a) local features: key-points or segments; b) descriptors: FPFH, SHOT, or ESF; and c) transformation estimations: RANSAC or FGR. Additionally, we compare the results against standard approaches like applying ICP after a good prior transformation has been given. The evaluation criteria include the distance which a UGV needs to

travel to successfully localize, the registration error, and the computational cost. In this context, we report our findings on effectively performing the task on two new Search and Rescue datasets. Our results have the potential to help the community take informed decisions when registering point-cloud maps from ground robots to those from aerial robots.

Relation to WP This paper describes 3D registration between UAV and UGV.

Availability Unrestricted.

2.7 Gawel (2017b), "Aerial picking and delivery of magnetic objects with MAVs"

Bibliography Abel Gawel, Mina Kamel, Tonci Novkovic, Jakob Widauer, Dominik Schindler, Benjamin Pfyffer von Altishofen, Roland Siegwart, Juan Nieto. "Aerial picking and delivery of magnetic objects with MAVs." In *IEEE International Conference on Robotics and Automation (ICRA) 2017.*

Abstract Autonomous delivery of goods using a Micro Air Vehicle (MAV) is a difficult problem, as it poses high demand on the MAVs control, perception and manipulation capabilities. This problem is especially challenging if the exact shape, location and configuration of the objects are unknown. In this paper, we report our findings during the development and evaluation of a fully integrated system that is energy efficient and enables MAVs to pick up and deliver objects with partly ferrous surface of varying shapes and weights. This is achieved by using a novel combination of an electro-permanent magnetic gripper with a passively compliant structure and integration with detection, control and servo positioning algorithms. The systems ability to grasp stationary and moving objects was tested, as well as its ability to cope with different shapes of the object and external disturbances. We show that such a system can be successfully deployed in scenarios where an object with partly ferrous parts needs to be gripped and placed in a predetermined location.

Relation to WP This paper describes visual servoing over moving targets, collection of samples, and transportation using UAVs.

Availability Unrestricted.

2.8 Gawel (2018), "X-View: Graph-Based Semantic Multi-View Localization"

Bibliography Abel Gawel, Carlo Del Don, Roland Siegwart, Juan Nieto, Cesar Cadena. "X-View: Graph-Based Semantic Multi-View Localization." In *IEEE Robotics and Automation Letters (RA-L) 2018*.

Abstract Global registration of multi-view robot data is a challenging task. Appearance-based global localization approaches often fail under drastic view-point changes, as representations have limited view-point invariance. This work is based on the idea that human-made environments contain rich semantics which can be used to disambiguate global localization. Here, we present X-View, a Multi-View Semantic Global Localization system. X-View leverages semantic graph descriptor matching for global localization, enabling localization under drastically different view-points. While the approach is general in terms of the semantic input data, we present and evaluate an implementation on visual data. We demonstrate the system in experiments on the publicly available SYNTHIA dataset, on a realistic urban dataset recorded with a simulator, and on real-world StreetView data. Our findings show that X-View is able to globally localize aerial-to-ground, and ground-to-ground robot data of drastically different view-points. Our approach achieves an accuracy of up to 85 % on global localizations in the multi-view case, while the benchmarked baseline appearance-based methods reach up to 75 %.

Relation to WP This paper describes semantic localization between heterogeneous robots.

Availability Unrestricted.

2017 IEEE International Symposium on
Safety, Security and Rescue Robotics (SSRR)
Shanghai, China, October 11-13, 2017

Formation Obstacle Avoidance using RRT and Constraint Based Programming

Fredrik Båberg[1] and Petter Ögren[1]

***Abstract*— In this paper, we propose a new way of doing formation obstacle avoidance using a combination of Constraint Based Programming (CBP) and Rapidly Exploring Random Trees (RRTs). RRT is used to select waypoint nodes, and CBP is used to move the formation between those nodes, reactively rotating and translating the formation to pass the obstacles on the way. Thus, the CBP includes constraints for both formation keeping and obstacle avoidance, while striving to move the formation towards the next waypoint.**

The proposed approach is compared to a pure RRT approach where the motion between the RRT waypoints is done following linear interpolation trajectories, which are less computationally expensive than the CBP ones. The results of a number of challenging simulations show that the proposed approach is more efficient for scenarios with high obstacle densities.

I. Introduction

In this paper, we study the problem of formation obstacle avoidance, i.e. moving a formation through an environment with obstacles, as seen in Figure 1. This problem is important for a number of reasons. In general, it is believed that having a multi agent system, i.e. a group of agents, perform a task can improve robustness, flexibility and performance, as compared to having a few more complex agents. Robustness is improved by reducing the impact of single failures. Flexibility is improved by the possibility of moving agents to different locations to perform simultaneous work, and performance can be improved by e.g. providing sensing capabilities across a larger area in a search task.

Formation obstacle avoidance is important when e.g. a group of robots are jointly carrying an object, when they need to cover an area, using a formation that is optimized relative to their sensors, or when they need to maintain relative distances to enable the best use of some communication channel.

All of the tasks above can be applicable to, for instance, search and rescue applications. A formation which optimizes the use of sensors could be used, or the robots could transport a larger object on top of the formation.

In a bit more technical detail, the obstacles we consider can be of arbitrary shape, modelled by the union of a large set of possibly overlapping circular discs. The formation is given as a desired rigid shape, that can translate and rotate to get past obstacles and arrive at the goal. The robots are modelled as kinematic points.

The work on formation obstacle avoidance can be structured based on how strict the formation keeping objective is, relative to the objectives of reaching the goals and avoiding the obstacles.

[1]Both authors are with the Robotics, Perception and Learning Lab, Centre for Autonomous Systems, Royal Institute of Technology (KTH), SE-100 44 Stockholm, {fbaberg,petter}@kth.se

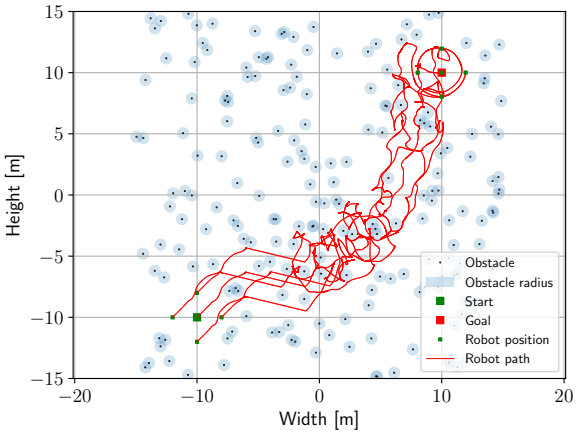


Fig. 1: Example execution of CBP-RRT with 200 obstacles. Non-accessible areas belonging to obstacles are indicated by blue circles with a black dot at the center. The start position of the virtual structure is located at (-10, -10), while target position is at (10, 10). Smaller green squares indicates robot positions.

In some approaches, the formation requirement is quite loose, corresponding to animal flocking. In [1] the robots need to avoid coming closer than a given minimum separation to both obstacles and other robots, but there is no ideal formation shape to be achieved.

In other approaches, the system itself is allowed to make a tradeoff between an ideal formation and avoiding obstacles, this includes behavior based approaches [2] and other potential field methods [3].

Work on navigation functions represent a formal extension of the potential field ideas, but require that the obstacles are enough separated so that robots can pass between them [4]. This assumption is also present in work focussing on the control of the actual vehicles [5].

With loose formation requirements you can also allow the robots to break formation in order to pass obstacles, formally creating a safe mode, including formation keeping, and a danger mode disregarding the formation objective [6].

In cases where the formation keeping, up to translation, is a strict requirement we have approaches building on grid based path planning [7], [8]. There the formation shape, with a possible margin for control errors, is treated as a fixed

978-1-5386-3923-8/17/$31.00 ©2017 IEEE

super robot used to create configuration space obstacles for all formation positions that imply a collision between a robot-obstacle pair. These configuration obstacles are then captured in a grid map in which the motion planning is done.

Constraint Based Programming (CBP) is an approach for designing robot controllers that take a set of constraints, in terms of equalities and inequalities into consideration [9]. CBP has its roots in the concepts of the *additional tasks* of [10], the *user defined objective functions* of [11], and the *sub-tasks* of [12]. Similar ideas were used in the *Stack of Tasks* approach [13], [14], the *iTaSC* approach [15], [16], and in a variation using Quadratic programming that was proposed in [17] and [18].

In this paper, we will use CBP for formation control, where the constraints will be given by both the formation keeping objective, and the desired obstacle clearance. The objective function will be the progress towards the next waypoint, and the waypoints are chosen using an Rapidly Exploring Random Tree (RRT) algorithm.

RRTs are sampling based path planners [19], [20] that are known to outperform grid based planners when there are a large number of dimensions, and/or a need for very small grid cells.

The main contribution of this paper is that we combine CBP with RRTs to solve formation obstacle avoidance problems in very cluttered environments. In this way, the local reactive properties of CBP, allowing the formation to rotate and translate past the obstacles, is combined with the global properties of RRT, avoiding local minima problems and efficiently exploring the statespace. To the best of our knowledge, this has not been done before.

The outline of this paper is as follows. We will first give a brief introduction to CBP and RRT in Section II before formulating a combined problem, and suggesting our solution, in Section III. In Section IV we present an experiment for validating the approach, followed by results in Section V. Discussion and conclusions are provided in Section VI and VII.

II. Background

We will now give a brief introduction to RRTs, followed by CBP.

A. RRTs

RRT is a sampling based algorithm used for path planning, where a tree is built up by sampling states and connecting new states to the tree. The psuedo-code for RRT [19] is shown in Algorithm 1. The function names are fairly self-explaining, with state denoted by x, control sequence by u and the tree denoted by T. Note that collision detection can be embedded in step 5, where a control sequence is sought bringing the state from x_{near} to x_{rand}, or as close as possible if x_{rand} cannot be reached.

B. Constraint Based Programming

Consider the following control problem with constraints:

Algorithm 1 *GENERATE_RRT*$(x_{init}, K, \Delta t)$

1: $T.init(x_{init})$;
2: **for** k=1 to K **do**
3: $x_{rand} \leftarrow$ RANDOM_STATE();
4: $x_{near} \leftarrow$ NEAREST_NEIGHBOR(x_{rand}, T);
5: $u \leftarrow$ SELECT_INPUT(x_{rand}, x_{near});
6: $x_{new} \leftarrow$ NEW_STATE$(x_{near}, u, \Delta t)$;
7: T.add_vertex(x_{new});
8: T.add_edge(x_{near}, x_{new}, u);
return T

Problem 2.1: Given a time interval $[t_0, t_f]$, initial state $q(t_0) = q_0$ and a control system

$$\dot{q} = h(q,u),$$

where $q \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, solve the following optimal control problem

$$\min_{u(\cdot)} \quad f_j(q(t_f),t_f),\ j \in I \tag{1}$$

$$\text{(s.t.)} \quad f_i(q(t),t) \leq 0,\ \forall i \in I_{ie},\ t > t_0 \tag{2}$$

$$f_i(q(t),t) = 0,\ \forall i \in I_e,\ t > t_0, \tag{3}$$

where $f_i : \mathbb{R}^n \to \mathbb{R},\ i \in I \subset \mathbb{N}$ and we assume that the constraints are satisfied at t_0, i.e. $f_i(q_0,t_0) \leq 0$ for all $i \in I_{ie}$ and $f_i(q_0,t_0) = 0$ for all $i \in I_e$ and $I_{ie}, I_e \subset I$, where I_e, I_{ie} is the set of indices for equality and inequality constraints.

Being an optimal control problem, this is in general very hard to solve to optimality. CBP instead applies a reactive local approach by converting the problem above to a series of convex Quadratic Programming Problems (QPs) to be solved at every time step. In detail we have

Problem 2.2:

$$\min_{u} \quad \dot{f}_j(q(t),u,t) + u^T Q u,\ j \in I \tag{4}$$

$$\text{(s.t.)} \quad \dot{f}_i(q,u,t) \leq -k_i f_i(q,t),\ \forall i \in I_{ie}, \tag{5}$$

$$\dot{f}_i(q,u,t) = -k_i f_i(q,t),\ \forall i \in I_e, \tag{6}$$

where k_i are positive scalars and Q is a positive definite matrix.

To verify that a solution to Problem 2.2 is also a feasible solution to Problem 2.1, we first compare the inequality constraints. As long as $t > t_0$ we can see that if (5) is satisfied then (2) is satisfied. In the case of equality in (5), then the bounds of (2) will be approched, but not violated, with time constant $1/k_i$. For the equalities we have, in a similar way, that as long as (6) is satisfied then (3) will also be satisfied, for $t > t_0$. For the objective function, if we keep the derivative minimized then the objective $f_j(q(t_f),t_f)$ will be kept small. In order to avoid sudden large changes in the output we add a quadratic cost $u^\top Q u$, where Q is a positive-definite matrix assigning relative weights to the elements in u.

Now, in order to express this in coordinate velocities, we use the relation

$$\dot{f} = \frac{\partial f}{\partial q}\dot{q} \tag{7}$$

to obtain the following:

Problem 2.3:

$$\min_{u} \quad \frac{\partial f_j(q(t),u,t)}{\partial q}\dot{q} + u^T Q u, \ j \in I \tag{8}$$

$$\text{(s.t.)} \quad \frac{\partial f_i(q,u,t)}{\partial q}\dot{q} \leq -k_i f_i(q,t), \ \forall i \in I_{ie}, \tag{9}$$

$$\frac{\partial f_i(q,u,t)}{\partial q}\dot{q} = -k_i f_i(q,t), \ \forall i \in I_e. \tag{10}$$

This means that we need to find the derivative with respect to coordinates q, for the constraints and the objective.

If the system is a kinematic point, $\dot{q} = u$, we can write Problem 2.3 in matrix notation as follows

Problem 2.4:

$$\min_{u} \quad u^\top Q u + p^\top u \tag{11}$$

$$\text{(s.t.)} \quad Gu \leq h \tag{12}$$

$$Au = b \tag{13}$$

which is a classical QP. With this background, we will continue with the problem formulation.

III. Problem Formulation and Solution

In this section we will state the problem we address and the solution we propose.

Problem 3.1: Control the robots such that they

A) Reach the target position
B) Stay in a given formation
C) Stay clear of obstacles

Our proposed solution is a combination of CBP and RRT, denoted CBP-RRT, where CBP implements the reactive local control (`SELECT_INPUT`) in Line 5 of Algorithm 1, see Section II.

Remark 3.1: Note that the proposed approach can be applied to many different extensions of RRT, such as RRT* [21] and bidirectional RRT [20], but the focus in this paper is to investigate the implications of combining CBP with methods of the RRT family. Thus our comparison focuses on how standard RRT compares to CBP-RRT. One could also imagine comparing bidirectional RRT with bidirectional CBP-RRT, but we believe that the results would be similar.

In each execution of `SELECT_INPUT` we run CBP for a fixed number of timesteps. Note that as CBP is a reactive approach, there will be instances where non-convex obstacles prevent the algorithm from reaching the waypoint. Then, `SELECT_INPUT` will return the closest node it can find. The new nodes spawned by the RRT iterations will however sooner or later enable a safe passage of the non-convex obstacle. The advantages of using CBP is the ability of moving past some obstacles, significantly reducing the number of RRT nodes needed to find the goal. As will be shown below, this results in reduced overall computation times in highly cluttered environments. To quickly find solutions in open areas, we first try linear interpolation, and if that does not return a valid solution we apply CBP. For the formation keeping we will use a standard *virtual structure* approach, as described in [22].

A. RRT formulation

The standard RRT described in Algorithm 1 is used, with the addition that the algorithm terminates when the new state is within a given distance of the target state.Thus if a solution is found we will not add more nodes to the tree. Furthermore, since we know which state we want to reach, we sample the target location directly at every N:th sample.

As described above, in the RRT, the CBP implements the `SELECT_INPUT` step, with x_{near} as a starting state, x_{rand} as a target state, and returning a state that is as close to x_{rand} as possible. Thus, in some sense the RRT approach is addressing Problem 3.1 by solving a set of simpler instances of the same problem (i.e. Problem 3.1). In standard RRT, these need to be collision free, whereas in CBP-RRT it is enough if they are solvable using CBP.

B. CBP formulation

Now, we need to apply the approach described in Problem 2.3 i.e., Equations (8)-(10), to solve Problem 3.1, that is reach the target position, stay in a given formation, and stay clear of obstacles.

To formalize this problem we first need to introduce some notation. Let $q = [p_{vs}^T, \theta_{vs}, p_1^T, ..., p_M^T]^T \in \mathbb{R}^{3+2M}$ denote position and orientation of the virtual structure (vs), and position of agents and $u = [u_{vs}^T, \omega_{vs}, u_1^T, ...u_M^T]^T \in \mathbb{R}^{3+2M}$ denote velocity and angular velocity of the virtual structure and the velocities of agents, with, $\dot{q} = u$, for M agents. Finally, the goal is given by a position and orientation p_{goal}, θ_{goal}, the obstacles are given by positions p_{obs-k} and radii r_k, and the formation is given by offsets $d_i \in \mathbb{R}^2$ relative to p_{vs}.

In order to use Equations (8)-(10) we must first formalize the three objectives above, i.e. target position (A), formation keeping (B) and obstacle avoidance (C). We do this by creating quadratic functions as follows

$$f_A(q) = (p_{vs} - p_{goal})^T (p_{vs} - p_{goal}) + (\theta_{vs} - \theta_{goal})^2 \tag{14}$$

$$f_{Bi}(q) = (p_i - p_{vs} - R(\theta)d_i)^T (p_i - p_{vs} - R(\theta)d_i) = 0 \tag{15}$$

$$f_{Cik}(q) = (p_i - p_{obs-k})^T (p_i - p_{obs-k}) - r_k^2 \geq 0 \tag{16}$$

where $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. Differentiating the functions above with respect to time, using $\dot{R}(\theta) = R(\theta + \pi/2)\omega$ we get

$$\dot{f}_A(q) = 2(p_{vs} - p_{goal})^T u_{vs} + 2(\theta_{vs} - \theta_{goal})\omega_{vs} \tag{17}$$

$$\dot{f}_{Bi}(q) = 2(p_i - p_{vs} - R(\theta)d_i)^T (u_i - u_{vs} - R(\theta + \frac{\pi}{2})d_i\omega_{vs}) \tag{18}$$

$$\dot{f}_{Cik}(q) = 2(p_i - p_{obs-k})^T u_i \tag{19}$$

Now, looking at equations (8)-(10) we let the function f_i be given by f_A for $i = j$, f_B for $i \in I_{ie}$, and f_C for $i \in I_e$, i.e., the objective function in (8) is given by f_A, the equalities in (10) are given by f_{Bi}, for all agents i, and the inequalities in (9) are given by f_{Cik}, for all combination of agents i and obstacles k.

IV. Experiments

In order to evaluate the approach we perform a number of simulations. We will first introduce the software used.

The algorithms have been implemented in Python, with the optimization solver CVXOPT[1].The simulation engine is implemented in Python.

We investigate the approach using a setting with five robots modelled as kinematic points, with two degrees of freedom, position p_i. The goal is to navigate between two formation states given by position and orientation (start, target), while satisfying the constraints stated in Problem 3.1. The virtual structure has three degrees of freedom, position p_{vs}, and rotation θ_{vs}. A number of obstacles, that the formation has to avoid, are placed in the environment. An example map can be seen in Figure 1, with the start and target state.

For benchmarking we use LI-RRT with linear interpolation between waypoints of the RRT. Examples of execution of CBP-RRT for a single node with a single small obstacle and a larger obstacle are shown in Figures 2 and 3, respectively.

For the experiments we sample the target position every 20th time ($N = 20$). 5 seconds are allowed for movement, a maximum of 200.000 nodes can be sampled for the tree. Obstacle clearance is set to 0.5m, and the resolution of the solver is 0.2s. A solution is considered found by the RRT if the norm of the difference of the target state and the virtual structure state is less than 0.01.

V. Results

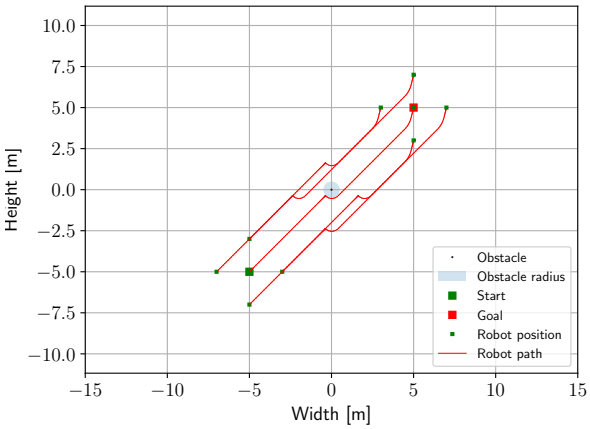


Fig. 2: Example of the path generated by CBP-RRT with a single node (only step 5 in Algorithm 1), with a single obstacle. Sufficient time to reach the target with one node was provided. Note that LI-RRT would not give a valid solution, due to the collision check.

We will now present the results of the experiments. For comparison we look at the time required for calculation, and the number of nodes (iterations) needed by CBP-RRT and LI-RRT respectively. Each run consists of generating an environment through randomly placing obstacles, and then running the two approaches for the generated environment. This is repeated, and the average values are presented. The results are presented in Table I, where **Method** is either the proposed method CBP-RRT, or the baseline method LI-RRT. **Obstacles** is the number of obstacles randomly placed in the environment. **Nodes** is the number of nodes (iterations) needed until convergence, and **Time** is the time required, in seconds, to find a solution (lower is better). **Rep.** is the number of runs used to compute the average, with total number of runs in parenthesis. When calculating the average, only cases where a solution could be found were included. Examples of executions with CBP-RRT and LI-RRT are shown in Figures 4 to 9. In the video[2] execution of the plan obtained by CBP-RRT is shown.

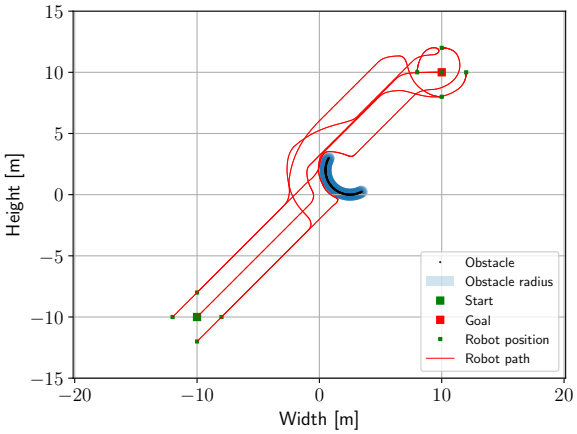


Fig. 3: Same setup as in Figure 2, for a non-convex obstacle configuration instead of a single obstacle. Also in this case, for a single node, LI-RRT would not give a valid solution.

TABLE I: Average results after running both algorithms, with 5 seconds for execution between each node, ordered by increasing number of obstacles. Cases where no solution could be found were excluded when calculating the average.

Method	Obstacles [-]	Nodes [-]	Time [s]	Rep. [-]
LI-RRT	0	33.6	0.06	100 (100)
CBP-RRT	0	36.0	0.06	100 (100)
LI-RRT	25	77	0.26	100 (100)
CBP-RRT	25	33.6	2.77	100 (100)
LI-RRT	50	146.4	0.45	100 (100)
CBP-RRT	50	38.0	5.26	100 (100)
LI-RRT	100	1018.8	2.64	100 (100)
CBP-RRT	100	51.8	13.49	100 (100)
LI-RRT	150	6027.8	15.35	100 (100)
CBP-RRT	150	106.6	43.22	100 (100)
LI-RRT	200	30539.6	95.33	100 (100)
CBP-RRT	200	183.0	87.74	100 (100)
LI-RRT	250	93852.1	311.67	91 (100)
CBP-RRT	250	388.4	212.89	100 (100)

For 200 obstacles, LI-RRT required more than 100.000

[1]http://cvxopt.org/

[2]Also available on `https://www.youtube.com/watch?v=OtvFIZFg68M`

nodes in three of the runs.

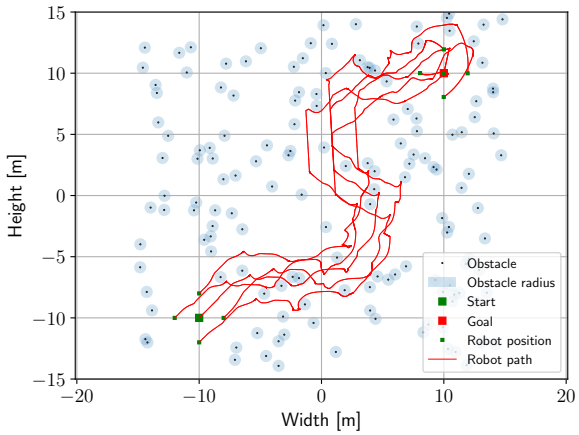


Fig. 4: Execution of path generated by CBP-RRT for 150 obstacles.

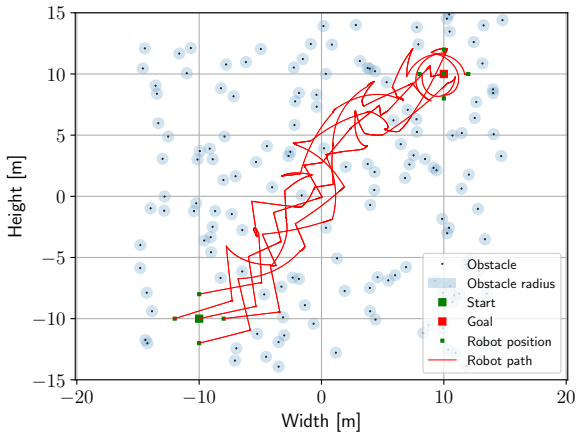


Fig. 5: Execution of path generated by LI-RRT for 150 obstacles.

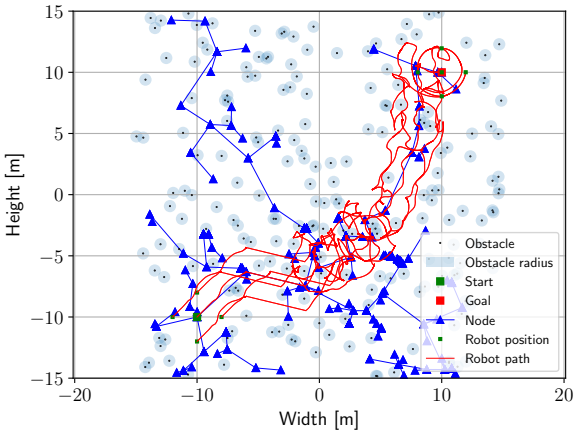


Fig. 6: Example of the path generated by CBP-RRT for 200 obstacles. Blue triangles indicates nodes in the RRT.

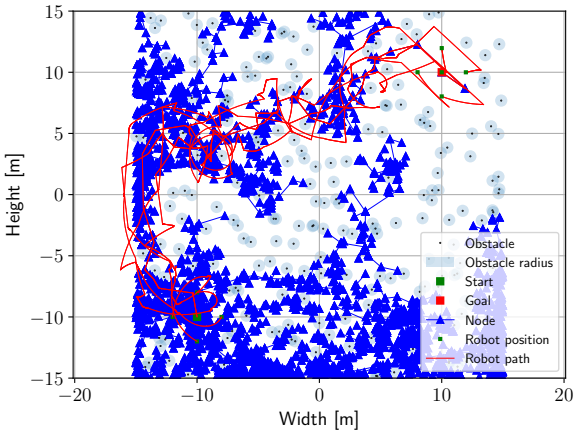


Fig. 7: Example of the path generated by LI-RRT for 200 obstacles. The plot also includes the nodes of the RRT.

VI. Discussion

We will now give a brief discussion on the results presented above.

In the comparison presented in Table I, we use 5 robots. In all scenarios, on average, CBP-RRT has fewer nodes in the RRT than LI-RRT. The execution times are however smaller for LI-RRT in the top of the table (scenarios with fewer obstacles). This is due to the fact that the execution of Line 5 in Algorithm 1 is fairly fast for LI-RRT, including linear interpolation and collision checks, whereas for CBP-RRT it includes the actual simulation of the CBP algorithm over a number of timesteps.

As the environments gets more cluttered, from 200 obstacles and above, the ability of CBP to negotiate the obstacles and actually reaching the waypoints chosen by the RRT starts paying off, with overall shorter execution times and at 250 obstacles a higher success rate. With 250 obstacles, 200.000 nodes was not enough for LI-RRT to find a solution in all of the runs. LI-RRT finds a solution in 91 of the 100 runs, whereas CBP-RRT succeeds in all of the 100 runs.

Example executions of the scenarios are shown in Figures 4-9. In Figures 4 and 5 a scenario with 150 obstacles is shown. Note that the LI-RRT has more straight robot trajectories between obstacles than the CBP-RRT.

In Figure 6 and 7 a scenario with 200 obstacles is shown, together with the nodes of the RRT in both cases. The huge number of nodes needed to find a solution in the LI-RRT is evident, as compared to the CBP-RRT.

In Figure 8 and 9 a scenario with 250 obstacles is shown. Note that getting the formation from start to goal without colliding is quite challenging.

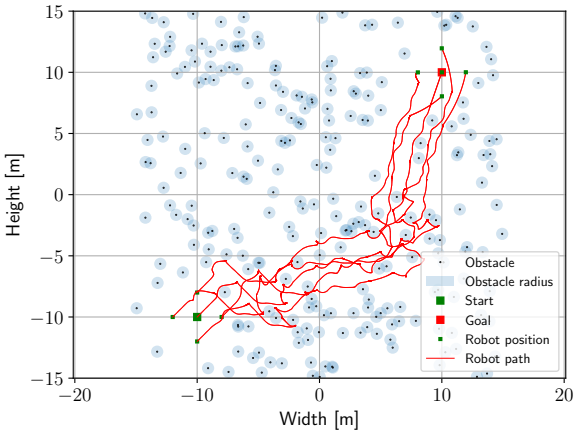


Fig. 8: Execution of path generated by CBP-RRT for 250 obstacles.

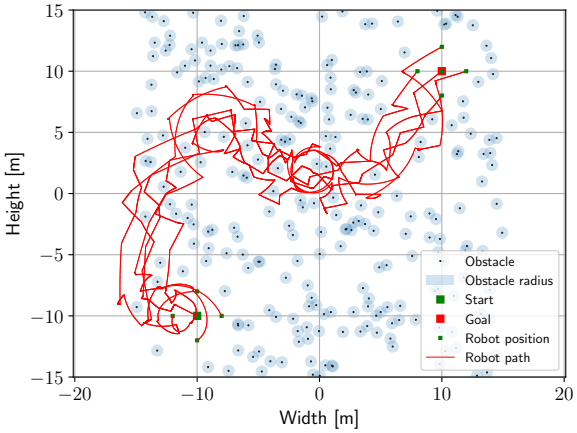


Fig. 9: Execution of path generated by LI-RRT for 250 obstacles.

VII. Conclusions

In this paper we propose to use CBP-RRT, a combination of contraint based programming and RRT, for doing formation obstacle avoidance in very cluttered environments. The proposed CBP-RRT approach was compared to LI-RRT, the classical approach using linear interpolation collision checking. The results show that CBP-RRT outperforms LI-RRT when the obstacle density is high enough.

Acknowledgment

The authors gratefully acknowledge funding under the European Union's seventh framework program, under grant agreements FP7-ICT-609763 TRADR.

References

[1] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on automatic control*, vol. 51, no. 3, pp. 401–420, 2006.

[2] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on robotics and automation*, vol. 14, no. 6, pp. 926–939, 1998.

[3] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 347–354, 2014.

[4] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1827–1832, 2012.

[5] C. De La Cruz and R. Carelli, "Dynamic model based formation control and obstacle avoidance of multi-robot systems," *Robotica*, vol. 26, no. 3, pp. 345–356, 2008.

[6] X. Wang, V. Yadav, and S. Balakrishnan, "Cooperative uav formation flying with obstacle/collision avoidance," *IEEE Transactions on control systems technology*, vol. 15, no. 4, pp. 672–679, 2007.

[7] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2492–2497.

[8] Y. Shapira and N. Agmon, "Path planning for optimizing survivability of multi-robot formation in adversarial environments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4544–4549.

[9] P. Ögren and J. W. Robinson, "A model based approach to modular multi-objective robot control," *Journal of Intelligent & Robotic Systems*, vol. 63, no. 2, pp. 257–282, 2011.

[10] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 472–490, 1989.

[11] Z.-X. Peng and N. Adachi, "Compliant motion control of kinematically redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 831–836, 1993.

[12] E. Tatlicioglu, D. Braganza, T. C. Burg, and D. M. Dawson, "Adaptive control of redundant robot manipulators with sub-task objectives," in *American Control Conference, 2008*. IEEE, 2008, pp. 856–861.

[13] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.

[14] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[15] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[16] R. Smits, T. De Laet, K. Claes, H. Bruyninckx, and J. De Schutter, "itasc: A tool for multi-sensor integration in robot manipulation," in *Multisensor Fusion and Integration for Intelligent Systems*. Springer, 2009, pp. 235–254.

[17] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2126–2132, 2004.

[18] Y. Zhang and S. Ma, "Minimum-energy redundancy resolution of robot manipulators unified by quadratic programming and its online solution," in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. IEEE, 2007, pp. 3232–3237.

[19] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[20] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[21] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[22] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

RCAMP: A Resilient Communication-Aware Motion Planner for Mobile Robots with Autonomous Repair of Wireless Connectivity

Sergio Caccamo, Ramviyas Parasuraman, Luigi Freda, Mario Gianni, Petter Ögren

***Abstract*—Mobile robots, be it autonomous or teleoperated, require stable communication with the base station to exchange valuable information. Given the stochastic elements in radio signal propagation, such as shadowing and fading, and the possibilities of unpredictable events or hardware failures, communication loss often presents a significant mission risk, both in terms of probability and impact, especially in Urban Search and Rescue (USAR) operations. Depending on the circumstances, disconnected robots are either abandoned, or attempt to autonomously back-trace their way to the base station. Although recent results in Communication-Aware Motion Planning can be used to effectively manage connectivity with robots, there are no results focusing on autonomously re-establishing the wireless connectivity of a mobile robot without back-tracing or using detailed a priori information of the network.**

In this paper, we present a robust and online radio signal mapping method using Gaussian Random Fields, and propose a Resilient Communication-Aware Motion Planner (RCAMP) that integrates the above signal mapping framework with a motion planner. RCAMP considers both the environment and the physical constraints of the robot, based on the available sensory information. We also propose a self-repair strategy using RCMAP, that takes both connectivity and the goal position into account when driving to a connection-safe position in the event of a communication loss. We demonstrate the proposed planner in a set of realistic simulations of an exploration task in single or multi-channel communication scenarios.

Index Terms*—*Mobile Robots, Self-Repair, Wireless Communication, Communication-Aware Motion Planning.

I. Introduction

Recent years have witnessed an increased development of wireless technologies and significant improvements in communication performance and quality. As wireless networks possess many advantages over a tethered connection, such as the ease of deployment and fewer physical constraints, it has become the 'de facto' means of communication in mobile robots. However, this development has not come without problems. A 2004 study [1] found a drastic increase in communication-related failures in robots compared to its prior in 2002.

These problems are important under normal circumstances, but become even more significant in USAR scenarios, where electromagnetic infrastructure is often damaged. Furthermore, USAR missions often rely more on bi-directional communication channels than other robotic applications, since the performance of a combined human-robot team is still superior compared to purely autonomous solutions in tasks such as inspecting or assessing potentially hazardous areas [2], [3].

The authors S.Caccamo, P.Ögren are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, KTH Royal Institute of Technology, Sweden. R.Parasuraman is with Purdue University, West Lafayette, USA. L.Freda and M.Gianni are with ALCOR Laboratory, DIAG, Sapienza University of Rome, Italy. e-mail: {caccamo|petter}@kth.se, ramviyas@purdue.edu, {freda|gianni}@dis.uniroma1.it

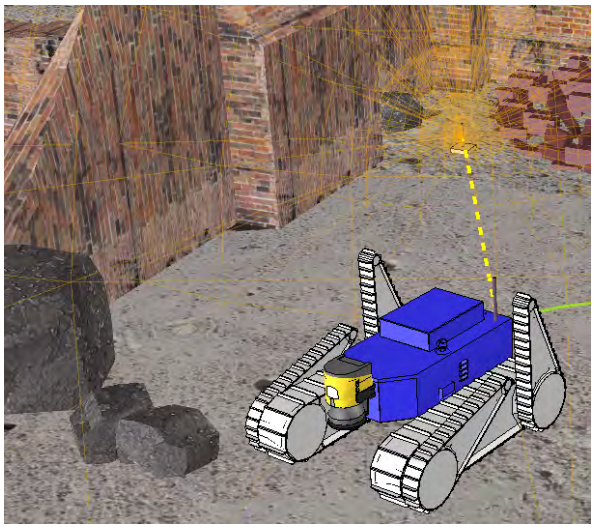


Fig. 1: The simulated mobile robot (UGV) with its receiver and an omnidirectional transmitter on a urban search and rescue scenario.

To address this problem, several researchers have focused on Communication-Aware Motion (or path) Planning (CAMP) to simultaneously optimize motion and communication constraints and finding and executing an optimal path towards a destination [4]. In particular, Mostofi et al. laid solid foundations in this research area [5]–[7]. It can be noted that most previous works consider either a binary or a disk based connectivity model, or an accurate communication model to optimize the robots motion and communication energy without focusing on resilience. Additionally, none of the previous works explicitly addresses the problem of efficiently re-establishing the communication in case of a connection loss.

In this paper, we propose a Resilient Communication-Aware Motion Planner (RCAMP) that combines two key elements: 1) a Gaussian Random Field (GRF) based probabilistic model to map the Radio Signal Strength (RSS) of an unknown environment and use it to predict the communication quality of the planned path; 2) a motion planning strategy that starting from sensory information (such as LIDAR), computes a traversability map for a given robot taking into account environmental constraints. Additionally we propose a strategy to autonomously repair a communication loss by steering the robot towards a communication-safe position using the proposed RCAMP.

Specifically, inspired by [8], we use GRFs for dynamically mapping the heterogeneous distribution of the RSS. We then merge this online framework with a motion planner

- to obtain a semi-optimal path considering both communication and motion constraints, and
- to quickly re-establish connection in case of signal loss.

We demonstrate the feasibility of our approach through extensive simulations on a variety of use cases that reproduce realistic wireless network changes (e.g. a sudden connection loss) in single and multi-channel set-ups. The main advantages of our planner compared to others are the response to dynamic changes in the network configuration (e.g. disruptions or movement in Access Points) or in the environment (e.g. path planning in presence of dynamic obstacles) and the fact that we do not require prior knowledge of the network, such as the location of the Access Points. We propose a fully online, dynamic and reactive CAMP that adapts to the recent sensory information.

II. Related Work

Considerable efforts have been made to address the problem of maintaining robust wireless communication between mobile robot(s) and a base station [3], [9], [10]. Many solutions focus on using mobile repeater (relay) robots to establish and/or repair an end-to-end communication link [11]–[13]. Other solutions focus on means to provide situation awareness of wireless connectivity to the robot or the teleoperator [14].

An overview of the CAMP problem is presented in [4]. Several works rely upon an oversimplified model in which the connectivity is modelled as a binary function. In this case, the predicted Signal to Noise Ratio (SNR) and the estimated distance from the robot (aerial or ground) to the radio source are empirically thresholded in order to identify regions with high probability of communication coverage [15].

In [16], the authors propose an optimization strategy to compute a path along which the predicted communication quality is maximized. They make use of supervised learning techniques (Support Vector Regression) to predict the link quality such as the Packet Reception Ratio. It is worth noting that in this case the learning mechanism is offline and hence can only be applied to a static environment.

A communication aware path planner is proposed in [17] for an aerial robot. Here, the authors present a probability function which is based on the SNR between two nodes. The SNR model is learned from the measurements online using an Unscented Kalman Filter (UKF) model.

Works that combine communication and motion planning are strongly influenced by Mostofi et al. In [6], the authors developed a mathematical framework to predict the communication quality (mainly the SNR) in unvisited locations by learning the wireless channels online. This prediction model is then used to define a motion planner either to improve the channel assessment [5] or to optimize for communication and motion energy to reach a given target [7]. This framework is further extended in [18] to include online channel learning for co-optimization of communication transmission energy and motion energy costs. Here, the transmission power is modelled as a function of SNR, whereas the motion power is a function of the robot's velocity and acceleration.

Recovering from a communication failure is a topic that has not been given much attention in the community. A simplistic solution is to back-track the robot along the path it has already travelled, until it regains communication. Alternatively, the robot can predict positions where the connection has high quality and move towards those locations in case of connection loss. In [19], a decentralized algorithm is proposed to move the disconnected robot towards the known position of the gateway (radio signal source or relay) by taking into account obstacles along the way. In [10], the authors demonstrated a behaviour to drive the disconnected robot towards the closest robot node (assuming a multi-robot network) and repeat this until connection is restored. Note that in the above mentioned works, the wireless channel parameters are not estimated, but instead perfect knowledge on the network topology is assumed (e.g, the positions of the gateway nodes, base station, etc.).

In the Wireless Sensor Networks (WSN) community, where it is commonly assumed that ample amounts of hopping nodes are available, the problem of repairing a connectivity failure is viewed differently. In this case, mobile robots can be used as sensor nodes which can be repositioned or added to replace failed nodes [20], [21].

It can be seen that predicting the communication quality in regions not explored by a mobile robot is a challenging problem. As pointed out above, probabilistic approaches such as maximum likelihood and UKF have been used to model the path loss and shadowing components of the RSS. Yet these models perform efficiently only when there is at least some prior information available regarding the network, such as source or relay node positions, which is difficult to know in field robotics applications such as the emergency deployment of robots to help in disaster response operations. In [8], a Gaussian Process based method is proposed to estimate the channel parameters and map the RSS in real-time using a few sample measurements. Taking inspirations from this work, in this paper, we propose a truly online Gaussian Random Field model to assess the RSS by continuously learning from the field measurements.

We make use of this probabilistic model to obtain the communication cost of a given path. We then co-optimize this cost along with the motion costs (ensuring feasibility of traversal by taking into account environment obstacles and constraints) to compute a path to a given destination. The motion planner then executes this path by actively re-planning. In case of a connection loss and if no destination is defined, the motion planner makes use of the online GRF model to quickly drive to a position that has the highest probability to restore connectivity, by setting the robot's starting position as the goal.

III. Methodology

In this section, we first define the RSS model, and then discuss how to apply Gaussian Random Fields (GRF[1]) to generate an online prediction map of the RSS distribution which will be used in both motion planning and reconnection planning. We conclude this section with a description of the Communication-Aware Motion Planner and its utility function. Note that the method can be extended to 3D and hence be applied to aerial robots as well.

A. Radio Signal Strength Model

When a radio signal propagates from a source to a destination, its strength attenuation depends on environmental factors such as distance (path loss), objects in the environment (shadowing) and spatio-temporal dynamics (multipath fading) [22]. A frequently used model to represent the RSS is given by [23]:

$$RSS_{(d,t)} = \underbrace{RSS_{d_0} - 10\eta \log_{10}(\frac{d}{d_0})}_{path\ loss} - \underbrace{\Psi_{(d)}}_{shadowing} - \underbrace{\Omega_{(d,t)}}_{multipath} . \quad (1)$$

Here, RSS_{d_0} is the RSS at a reference distance d_0 (usually 1m), which depends on the transmit power, antenna gain, and the radio frequency used. η is the path loss exponent which is a propagation constant of a given environment. $d = ||x - x_0||$ is the distance of the receiver (at position x) from the radio source (at position x_0). $\Psi \sim \mathcal{N}(0, \sigma)$ is a Gaussian random variable typically used to represent shadowing while Ω is a Nakagami-distributed variable representing multipath fading.

Usually, the RSS measurements (in dBm) coming from wireless adapters are prone to noise and temporal fluctuations in addition to multipath fading. This noise can be mitigated by applying an exponentially weighted moving average (EWMA) filter [12]:

$$RSS^f(i) = RSS^f(i-1) + \alpha(RSS(i) - RSS^f(i-1)), \quad (2)$$

where $RSS(i)$ is the RSS value measured at the i^{th} instant, RSS^f is the filtered RSS value and α is an empirical smoothing parameter.

We use Gaussian Processes for regression (GPR) [24] for modeling the radio signal distribution as demonstrated in [8], [25], [26]. A key difference compared to the previous approaches is that we employ online learning with dynamic training size that adapts to the changes in the environment (e.g. change from line of sight to non-line of sight of the source, switching between access points, losing/regaining a connection, etc.). Below we briefly describe how the GPR is performed.

[1]GRF is a term for the Gaussian Process Regression with 2.5 dimensional datasets where each $x-y$ coordinate has a single value v.

B. Gaussian Random Fields

The RSS distribution can be described with a function $f : \mathbb{R}^2 \to \mathbb{R}$ where each vector of xy-coordinates generates a single RSS. Such a function can be efficiently modeled by a GRF which places a multivariate Gaussian distribution over the space of $f(\mathbf{x})$. The GRF allows us to probabilistically handle noisy meausurements of a dynamic and unknow process and predict the behaviour of such a process at unknown and unexplored states. GRF have been widely used on a broad range of robotics problems such as haptic and visual perception [27], geometric shape description and planning [28]. As shown in [26], environmental observation of RSS can condition a GRF so that its posterior mean defines the signal distribution of interest. The GRF is in fact shaped by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x_i}, \mathbf{x_j})$.

To properly describe the probabilistic model we define the set $R_V = \{\mathbf{r_1}, \mathbf{r_2} \dots \mathbf{r_N}\}$, with $\mathbf{r_i} \in \mathbb{R}^3$, of measurements of robot xy-positions and corresponding RSS. $D_{RF} = \{\mathbf{x_i}, y_i\}_{i=1}^N$ is a training set where $\mathbf{x_i} \in \mathbf{X} \subset \mathbb{R}^2$ are the xy-coordinates of the points in R_V and y_i the RSS readings from the mobile robots wireless adapters. $\mathbf{X}_* \equiv \mathbf{X}_{\mathbf{rf}_*} \subset \mathbb{R}^2$ represents a set of M test points where $\mathbf{x}_{\mathbf{rf_i}} \in \mathbb{R}^2$ is a xy-coordinate of the environment.

The joint Gaussian distribution on the test set $\mathbf{X}_*$, assuming noisy observation $\mathbf{y} = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, assumes the following form

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(m(\mathbf{x}), \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^\mathbf{T} & \mathbf{k}_{**} \end{bmatrix} \right) \quad (3)$$

where $\mathbf{K}$ is the covariance matrix between the training points $[\mathbf{K}]_{i,j=1\dots N} = k(\mathbf{x_i}, \mathbf{x_j})$, $\mathbf{k}_*$ the covariance matrix between training and test points $[\mathbf{k}_*]_{i=1\dots N, j=1\dots M} = k(\mathbf{x_i}, \mathbf{x}_{*\mathbf{j}})$ and $\mathbf{k}_{**}$ the covariance matrix between the only test points $[\mathbf{k}_{**}]_{i,j=1\dots M} = k(\mathbf{x}_{*\mathbf{i}}, \mathbf{x}_{*\mathbf{j}})$.

We use the popular squared-exponential kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = \sigma_e^2 \exp\left(-\frac{(\mathbf{x_i} - \mathbf{x_j})^T (\mathbf{x_i} - \mathbf{x_j})}{\sigma_w^2} \right). \quad (4)$$

as it better represent the variance in RSS [8], [26].

Following the example of [8], we could define a model-based potential prior based on the path loss eq. (1) to improve the accuracy of prediction

$$m(\mathbf{x}) = RSS_0 - 10\eta \log_{10}(\|x - x^s\|), \quad (5)$$

where x^s is the source location which is an unknown parameter in the mean function. One could potentially optimize the mean hyper-parameters ($\theta_m = [RSS_0, \eta, x^s]$) by training the model with the measured data. In [8], [25], [26], they either assumed the knowledge of the source location or estimated it in a dedicated control/training phase with the measured data.

However, given the unbounded nature of the source location x^s and the fact that only sparse measurements in a limited explored area is available in a practical robotic application, optimizing these hyper-parameters will result in extensive computation and low accuracy.

Moreover, this model can be applied only to a fixed radio source (Access point). Therefore, considering a practical USAR scenario, where the Access Points can be mobile or is frequently moved, trying to optimize the source location in eq. (5) with the measured data will not only be inaccurate, but also result in poor prediction performance of the GPR model.

Finally, more complex potential priors can be used or interchanged in order to incorporate propagation phenomenas (e.g. attenuation due to walls, floors, etc.) or environmental knowledge and improve the prediction on those regions of the map far from the measured data [29]. However, such approaches require a larger amount of information and increase the number of hyperparameters to be optimized.

Thus in our work, we consider a constant mean function,

$$m(\mathbf{x}) = C, \tag{6}$$

for practical and computational aspects. Note that this mean function has shown low prediction errors in [30] when compared to a linear mean function.

The predictions are obtained from the GPR conditioning the model on the training set [24] :

$$p(f_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\overline{f_*}, \mathbb{V}[f_*]\right) \tag{7}$$

$$\overline{f_*} = m(\mathbf{x}) + \mathbf{k}_*^{\mathbf{T}}\left(\mathbf{K} + \sigma_n^2\mathbf{I}\right)^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{8}$$

$$V[f_*] = \mathbf{k}_{**} - \mathbf{k}_*^{\mathbf{T}}\left(\mathbf{K} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}_* \tag{9}$$

The predictive variance of the GRF highlights regions of low density or highly noisy data. The hyper-parameters of the mean and the kernel $\theta = [C, \sigma_c, \sigma_w]$ are periodically optimized while the mobile robot moves and collects measurements. The optimization (hyperparameter estimation) is done by maximizing the marginal logarithmic likelihood of the distribution on the measured data.

For online optimization purposes, we efficiently train the GPR after each measurement by dynamically adjusting the training set size based on the magnitude of the changes in the measurements. We optimize the GPR and start with the RSS prediction after the robot has moved enough to acquire the minimum amount of training samples (around 5 meters of displacement). The GPR model is continuously re-trained with every new collected sample. When the connection status is active, we keep increasing the training set size up to a certain maximum limit. If the connection is lost, we keep decreasing the training size until the minimum limit. The hyper-parameters are re-optimized with current measurements whenever the training size reaches a certain minimum.

Next we describe how to define a utility function that takes into account the prediction of the RSS and its uncertainty to generate trajectories that guarantee communication or to re-establish a connection in case of signal loss. We use the term "Wireless Map Generator (WMG)" to refer the above described Gaussian random field that generalizes over robot positions and RSS measurements to generate wireless distribution maps.

C. Communication-Aware Motion Planner

We use the RSS predictions from the GPR along with the traversibility cost in the RCAMP to plan and execute a path to a given destination. As the planner is dynamic, it keeps track of both RSS predictions and the traversibility based on the incoming sensory information. We detail the basic steps below.

1) Mapping and Point Cloud Segmentation: As a necessary prerequisite for path planning, a *map* representation $\mathcal{M}$ of the environment is incrementally built in the form of a point cloud. An ICP-based SLAM algorithm is used in order to register the different 3D laser scans collected by the robot. At each new scan, both the map and a structure interpretation of it are updated. In particular, the point cloud map $\mathcal{M}$ is segmented in order to estimate the traversability of the terrain.

In a first step, $\mathcal{M}$ is filtered using an efficient occupancy voxel-map representation [31]: recursive binary Bayes filtering and suitable clamping policies ensure adaptability to possible dynamic changes in the environment.

Next, geometric features such as surface normals and principal curvatures are computed and organized in histogram distributions. Clustering is applied on 3D-coordinates of points, mean surface curvatures and normal directions [32]. As a result, a classification of the map $\mathcal{M}$ in regions such as *walls*, *terrain*, *surmountable obstacles* and *stairs/ramps* is obtained.

2) Traversability Cost: Traversability is then computed on the map $\mathcal{M}$ as a cost function taking into account point cloud classification and local geometric features [33]. In particular, the traversability cost function $trav : \mathbb{R}^3 \longmapsto \mathbb{R}$ is defined as

$$trav(\mathbf{p}) = w_L(\mathbf{p})(w_{Cl}(\mathbf{p}) + w_{Dn}(\mathbf{p}) + w_{Rg}(\mathbf{p})) \tag{10}$$

where $\mathbf{p} \in \mathbb{R}^3$ is a map point, the weight $w_L(\mathbf{p})$ depends on the point classification, $w_{Cl}(\mathbf{p})$ is a function of the robot obstacle clearance, $w_{Dn}(\mathbf{p})$ depends on the local point cloud density and $w_{Rg}(\mathbf{p})$ measures the terrain roughness (average distance of outlier points from a local fitting plane). A *traversable map* $\mathcal{M}_t$ is obtained from $\mathcal{M}$ by suitably thresholding the obstacle clearance $w_{Cl}(\cdot)$ and collecting the resulting points along with their traversability cost.

3) Global and Local Path Planners: Path planning is performed both on global and local scales. Given a set of waypoints as input, the *global* path planner is in charge of (1) checking the existence of a traversable path joining them and (2) minimizing a combined RSS-traversability cost along the computed path. Once a solution is found, the *local* path planner safely drives the robot towards the closest waypoint by continuously replanning a feasible path in a local neighbourhood of the current robot position. This allows us to take into account possible dynamic changes in the environment and local RSS reconfigurations.

Both the global and the local path planners capture the connectivity of the traversable terrain by using a sampling-based approach. A tree is directly expanded on the traversability map $\mathcal{M}_t$ by using a randomized A* approach along the lines of [33]. The tree is rooted at the starting robot position.

Visited nodes are efficiently stored in a kd-tree. The current node $\boldsymbol{n}$ is expanded as follows: first, the robot clearance $\delta(\boldsymbol{n})$ is computed at n; second, a neighbourhood $\mathcal{N}(\boldsymbol{n})$ of points is built by collecting all the points of $\mathcal{M}_t$ which falls in a ball of radius $\delta(\boldsymbol{n})$ centred at $\boldsymbol{n}$. Then, new children nodes are extracted with a probability inversely proportional to the traversability cost. This biases the tree expansion towards more traversable and safe regions. The total traversal cost of each generated child is evaluated by using eqn. (12) and pushed in a priority queue $\mathcal{Q}$. The child in $\mathcal{Q}$ with the least cost is selected as next node to expand.

4) Cost Function: The randomized A* algorithm computes a sub-optimal path $\{\boldsymbol{n}_i\}_{i=0}^{N}$ in the configuration space $\mathcal{C}$ by minimizing the total cost

$$J = \sum_{i=0}^{N} c(\boldsymbol{n}_{i-1}, \boldsymbol{n}_i)\,, \tag{11}$$

where $\boldsymbol{n}_0$ and $\boldsymbol{n}_N$ are respectively the start and the goal configurations, and $\boldsymbol{n}_i \in \mathcal{C}$. In this paper we define the cost function $c : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}$ so as to combine traversability and RSS predictions. In particular

$$\begin{aligned} c(\boldsymbol{n}_i, \boldsymbol{n}_{i+1}) &= \big(d(\boldsymbol{n}_i, \boldsymbol{n}_{i+1}) + \\ &\quad h(\boldsymbol{n}_{i+1}, \boldsymbol{n}_N)\big)\pi_1(\boldsymbol{n}_{i+1})\pi_2(\boldsymbol{n}_{i+1}) \\ \pi_1(\boldsymbol{n}) &= \lambda_t \frac{trav(\boldsymbol{n}) - trav_{min}}{trav_{max} - trav_{min} + \varepsilon} + 1 \\ \pi_2(\boldsymbol{n}) &= \lambda_r \alpha_r e^{-t/\tau} \frac{rss_{max} - rss(\boldsymbol{n})}{rss_{max} - rss_{min} + \varepsilon} + 1 \end{aligned} \tag{12}$$

where $d : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}^+$ is a distance metric, $h : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}^+$ is a goal heuristic, $\lambda_t, \lambda_r \in \mathbb{R}^+$ are scalar positive weights, $rss : \mathcal{C} \times \mathcal{C} \longmapsto \mathbb{R}$ is the estimated RSS, $\alpha_r \in [0, 1]$ is a confidence which can be obtained by normalizing the variance of the RSS prediction (as returned by the GPR), ε is a small quantity which prevents division by zero and τ is an exponential decay constant (determines the amount of time after which π_2 goes to its minimum value 1). In particular, with abuse of notation we use $trav(\boldsymbol{n})$ to denote the traversability of the the point corresponding to $\boldsymbol{n}$. The first factor in eq. (12) sums together the distance metric and the heuristic function (which depends on the distance to the goal). The other two factors π_1 and π_2 respectively represent a normalized traversability cost and a normalized RSS cost, whose strengths can be increased by using the weights λ_t and λ_r respectively ($\pi_i \geq 1$). The exponential decay is used to decrease the effect of the RSS cost after a certain time (e.g. before the path planner is stopped by a timeout in case a path solution is difficult to find).

Note, instead of jointly optimizing the motion and communication energy for a given path as in [7], we plan an optimized trajectory to a given goal position using a cost function that represents a balanced optimization between communication and traversibility costs, includes normalization of the used metrics, and allows setting different priorities using the parameters λ_t and λ_r.

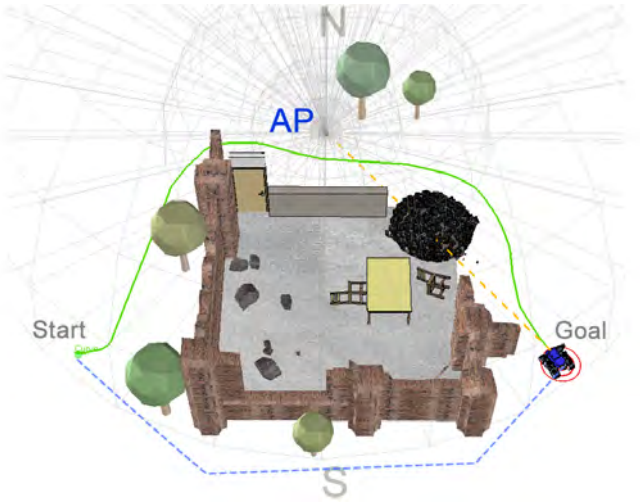


Fig. 2: Experimental scenario 1. The UGV tries to reach the goal position avoiding connection drops. The blue dotted line represents the shortest path, that will cause a connection loss (going outside the AP range). The green line represents a path that reaches the goal position while keeping the robot connected to the AP.

Self-recovery: The cost function in eq. (12) gives us the leverage in generating a trajectory that recovers from communication loss. In the case of a connection loss, we define the goal position as the robot's initial position or the AP position (if known), so as to bound the search and to guarantee the re-establishment of connectivity.

IV. Experimental Evaluation

We evaluated the performance of the proposed method through a series of experiments made on simulations using V-REP. Using the 3D model of the real UGV used in [34] we created 3 different simulation environments, reproducing typical USAR use cases, containing several obstacles and sources of signal (APs). The AP is simulated following eq. (1) with typical parameters such as $\eta = 3$, $\sigma = 2$ [23] considering a 2.4 GHz Wi-Fi communication. For each environment, we changed the positions of the robot and APs and repeated the experiments in several trials. All the software components including the RCAMP ran under the Robot Operating System (ROS).

Note we do not evaluate the GRF model separately. Nevertheless, the GRF with mean functions in eq. (5) and (6) have shown to perform well in signal source prediction and location estimations [25], [26], [30].

A. Experimental scenarios

Scenario 1: In the first scenario, see Fig. 2, the UGV is placed on the start position and must traverse an area containing a damaged building, to reach the goal position. An AP is placed on the northern part of the map (zone N in Fig. 2). The AP uses an omni-directional antenna covering a circular area that extends to half of the map, leaving the southern part (zone S in Fig. 2) uncovered. Start and goal positions are

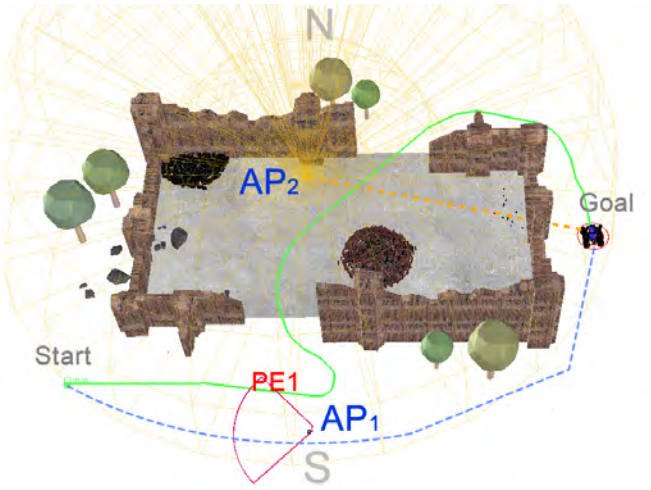


Fig. 3: Experimental scenario 2. The UGV tries to reach the goal position avoiding connection drops. The blue dotted line represents the shortest path to the goal position. The UGV is connected to AP_1 in the first part of the path. PE1 indicates the location of the UGV when AP_1 shuts down after a simulated hardware failure. The green line represents a new path that reaches the goal position while keeping the UGV connected, after switching from AP_1 to AP_2.

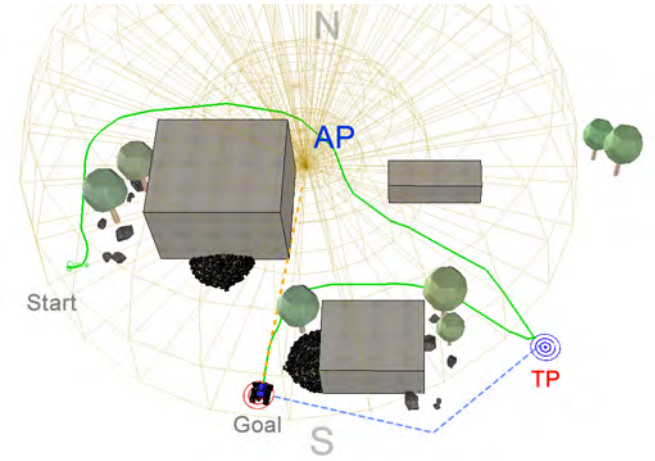


Fig. 4: Experimental scenario 3. The UGV is teleoperated in a USAR mission. The operator drives the robot outside the WiFi coverage area (at point TP) and the connection is lost. The system autonomously re-establishes the connection driving the UGV to a location with high RSS and then continues to reach the goal.

placed such that the shortest connecting path between the two points would traverse the poorly connected part of the map (S). Thus, RCAMP must generate a trajectory that connects the start and goal positions while keeping the robot in the signal covered area avoiding communication drops. With this scenario we want to demonstrate the capability of our utility function in keeping the robot connected to the AP.

Scenario 2: In the second scenario, see Fig. 3, two different APs cover the whole map. In this use case we want to test the promptness of the RCAMP to adapt to drastic changes in the wireless signal distribution. The robot starts the mission connected to AP_1. The RCAMP must generate a path from the start position to the goal position that ensures WiFi coverage. During the mission, AP_1 is switched off when the robot enters the region PE1, so to simulate a communication loss event. When the connection is lost, the robot connects to other APs (if available) in the same network, in a typical roaming behaviour. Once the robot connects to AP_2, the WMG must adapt its predictive model to the new signal distribution accordingly and reshape the RSS map. The RCAMP must then promptly update the path to the goal to ensure WiFi coverage.

Scenario 3: Finally, in the last scenario, see Fig. 4, we test our self-repair strategy in case of a complete connection loss event. The UGV is tele-operated until the connection drops (blue circle, outside the WiFi coverage area). The goal position (red circle) cannot be reached with teleoperation because of the missing communication channel. In this scenario, the UGV must autonomously re-establish the connection while moving to the goal position. If the goal position was not specified (e.g. during an exploration task) the UGV must move to the closest location in the map where the RSS is high enough to ensure re-connection to the AP.

V. Results

In the following we discuss the results of the experiments described in Sec. IV. Fig. 6 shows the recorded RSS and the path taken for the three scenarios. We present a comparison between the proposed RCAMP and a common path planner (PP). In the the first column we report the RSS values sensed by the antenna on-board the mobile robot.

In the first row (first experimental scenario) the PP leads the robot to lose connection whereas the RCAMP defines a trajectory that maintains the robot inside the operative range of the radio transmitter as it is possible to see in the second column of the same row. The second row of Fig. 6 shows that the RCAMP adapts to the drastic variation of the radio signal distribution (due to the simulated hardware failure and consecutive connection loss) and modifies the trajectory accordingly maintaining the robot inside the operative range of the new AP. The PP leads the robot to lose connection again. This demonstrates how the WMG promptly reacts to a connection loss in case a new source of signal is present.

Finally, in the last row we present the results for the third scenario where the mobile robot, after a brief exploration step, is tele-operated outside the wireless range. The RCAMP first brings the robot back to a position where the connection can be reestablished and then moves the robot to the goal position. The RSS value of the robot using the RCAMP, red signal in the third row, increases after the connection loss.

Fig. 5 shows the predicted radio signal distribution (WMG) for experiments 1 and 2. A red color indicates low or missing signal whereas a blue-purple color indicates high signal strength. As described in Sec. III-B, the training set consists of the last visited points in the environment along with the measured RSS. The size of the training set depends on the quality of the sensed signal. The first row (A1-5) shows the predicted radio signal distribution during the first experiment.

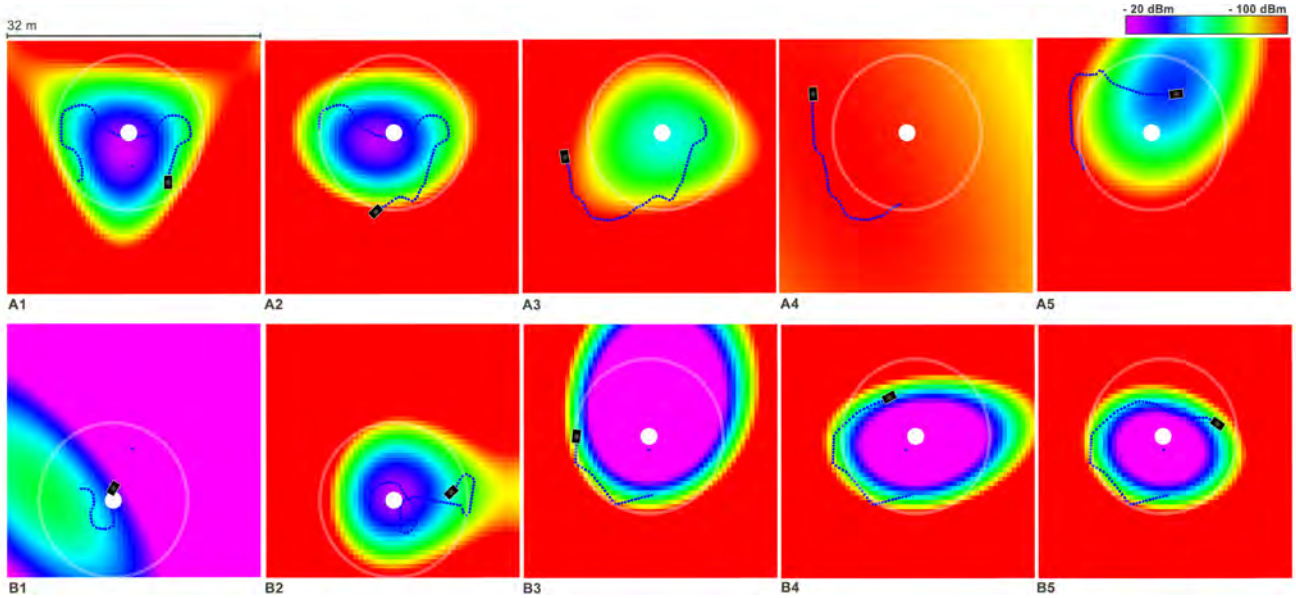


Fig. 5: Radio signal distributions for various trajectories in the maps of scenario 1 (A1-A5) and 2 (B1-B5). The white points represent the APs positions along with their operative ranges. The blue trajectories represent the training samples for the WMG. We can observe the changes in the RSS map generated by the WMG as the robot explores the region (without RCAMP). Note in A4 the robot is initially connected but is in a disconnected region the moat of the trajectory.

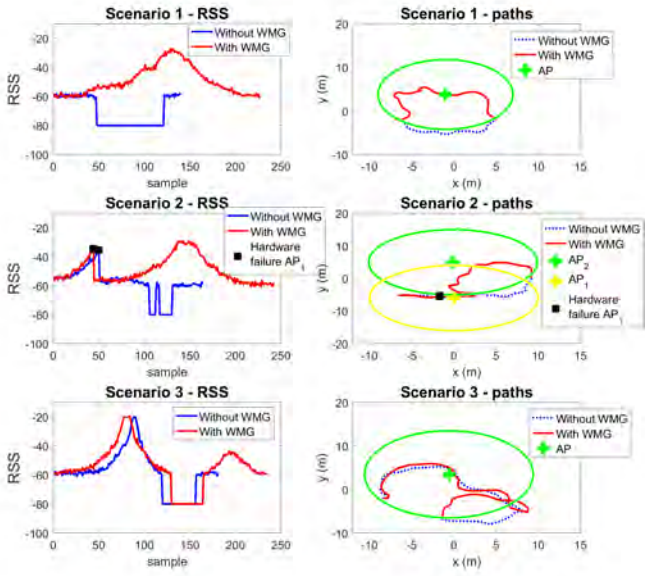


Fig. 6: Comparison between our RCAMP versus a normal path planner. The first column shows the RSS values measured using the onboard antenna during the three experimental scenarios. The RCAMP enables the robot to maintain an higher RSS value throughout the whole exploration. The second column shows the trajectories from start position to goal position for the three scenarios.

When the robot drives inside the operational range of the AP the training set increases and the model predicts correctly the position and the shape of the radio signal distribution (A1,2,5). Viceversa, when the mobile robot moves outside the operational range the communication with the AP drops and the training set shrinks as there is less useful information. This strategy allows the system to promptly adapt to a new source of signal as show in the last row. Initially the system adapts to the first source of signal (AP_1) as is visible in B1-2. When the first AP is shut down, the systems quickly re-sizes the training set size and the WMG converges to the new signal distribution allowing to identify the position of the second AP.

VI. Conclusions

Robots have a major potential in aiding first responders in USAR missions. In recent robot deployments, wireless networks were used in order to support mobile robot communication. This mean of communication poses several challenges, such as sudden network breakdowns and limited communication bandwidth. Based on our own experience in helping the Italian Firefighters with our UGVs and drones (under the EU-FP7 project TRADR [34]) to assess the damages in historical buildings after the recent earthquake in Amatrice, we concluded that the inherent limitations of a wireless network can compromise the outcome of a USAR mission. Most notably, the Access Points supporting robot communication had to be regularly relocated in order to let the robot re-estabilish communication.

To address some of these challenges, we proposed a Resilient Communication-Aware Motion Planner (RCAMP). Given a goal point, the RCAMP computes a trajectory by taking into account traveled distance, communication quality and environmental constraints. We used an online Gaussian Random Field to estimate the Radio Signal Strength requested by the motion planner in order to find a feasible path that

takes both traversability and connectivity into account. We also proposed an efficient strategy to autonomously repairing a communication loss by steering the robot towards a communication-safe position computed using the RCAMP. Alternatively, if a specific destination is available, the robot plans a path that combines the objectives of reaching the destination, and re-establishing the connection.

We demonstrated the proposed framework through simulations in V-REP under realistic conditions and assumptions. In future work, we plan to test the presented framework on real UGVs and further evaluate and analyze the performance and limits of the algorithms through more extensive field experiments.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding from the European Union's seventh framework program (FP7), under grant agreements FP7-ICT-609763 TRADR.

REFERENCES

[1] J. Carlson, R. Murphy, and A. Nelson, "Follow-up analysis of mobile robot failures," in *Proceedings. IEEE International Conference on Robotics and Automation (ICRA), 2004*, vol. 5, 2004, pp. 4987–4994 Vol.5.

[2] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013. [Online]. Available: http://dx.doi.org/10.1002/rob.21439

[3] R. Nattanmai Parasuraman, A. Masi, and M. Ferre, "Wireless Communication Enhancement Methods for Mobile Robots in Radiation Environments. Radio communication for robotic application at CERN," Ph.D. dissertation, Madrid, Polytechnic U., Oct 2014, presented 17 Oct 2014. [Online]. Available: https://cds.cern.ch/record/1956445

[4] B. Zhang, Y. Wu, X. Yi, and X. Yang, "Joint communication-motion planning in wireless-connected robotic networks: Overview and design guidelines," *arXiv preprint arXiv:1511.02299*, 2015.

[5] A. Ghaffarkhah and Y. Mostofi, "Channel learning and communication-aware motion planning in mobile networks," in *American Control Conference (ACC), 2010*, Jun. 2010, pp. 5413–5420.

[6] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, Mar. 2012.

[7] Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1562–1572, 2013.

[8] A. Fink, H. Beikirch, M. Voss, and C. Schroder, "RSSI-based indoor positioning using diversity and Inertial Navigation," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2010.

[9] C. Rizzo, D. Sicignano, L. Riazuelo, D. Tardioli, F. Lera, J. L. Villarroel, and L. Montano, *Guaranteeing Communication for Robotic Intervention in Long Tunnel Scenarios*. Cham: Springer International Publishing, 2016, pp. 691–703.

[10] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[11] B.-C. Min, Y. Kim, S. Lee, J.-W. Jung, and E. T. Matson, "Finding the optimal location and allocation of relay robots for building a rapid end-to-end wireless communication," *Ad Hoc Networks*, vol. 39, pp. 23 – 44, 2016.

[12] R. Parasuraman, T. Fabry, L. Molinari, K. Kershaw, M. D. Castro, A. Masi, and M. Ferre, "A Multi-Sensor RSS Spatial Sensing-Based Robust Stochastic Optimization Algorithm for Enhanced Wireless Tethering," *Sensors*, vol. 14, no. 12, pp. 23 970–24 003, 2014.

[13] K.-H. Kim, K. G. Shin, and D. Niculescu, "Mobile autonomous router system for dynamic (re) formation of wireless relay networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1828–1841, 2013.

[14] S. Caccamo, R. Parasuraman, F. Bberg, and P. gren, "Extending a ugv teleoperation flc interface with wireless network connectivity information," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 4305–4312.

[15] T. A. Johansen *et al.*, "Task assignment for cooperating uavs under radio propagation path loss constraints," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 3278–3283.

[16] G. A. Di Caro, E. F. Flushing, and L. M. Gambardella, "Use of time-dependent spatial maps of communication quality for multi-robot path planning," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2014, pp. 217–231.

[17] M. Stachura and E. W. Frew, "Cooperative target localization with a communication-aware unmanned aircraft system," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1352–1362, 2011.

[18] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, "Motion and communication co-optimization with path planning and online channel estimation," *arXiv preprint arXiv:1603.01672*, 2016.

[19] A. Derbakova, N. Correll, and D. Rus, "Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3863–3868.

[20] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, 2011.

[21] T. T. Truong, K. N. Brown, and C. J. Sreenan, "An online approach for wireless network repair in partially-known environments," *Ad Hoc Networks*, vol. 45, pp. 47 – 64, 2016.

[22] M. Lindhè, K. H. Johansson, and A. Bicchi, "An experimental study of exploiting multipath fading for robot communications," in *Proceedings of Robotics: Science and Systems*, June 2007.

[23] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006. [Online]. Available: http://www.gaussianprocess.org/gpml/chapters/

[25] L. S. Muppirisetty, T. Svensson, and H. Wymeersch, "Spatial wireless channel prediction under location uncertainty," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1031–1044, 2016.

[26] B. Ferris, D. Hahnel, and D. Fox, *Gaussian processes for signal strength-based location estimation*. MIT Press Journals, 2007, vol. 2, pp. 303–310.

[27] S. Caccamo, Y. Bekiroglu, C. H. Ek, and D. Kragic, "Active exploration using gaussian random fields and gaussian implicit surfaces," Nov 2016.

[28] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," pp. 2845–2850, May 2011.

[29] J. Strom and E. Olson, "Multi-sensor attenuation estimation (matte): Signal-strength prediction for teams of robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4730–4736.

[30] P. Richter and M. Toledano-Ayala, "Revisiting gaussian process regression modeling for localization in wireless sensor networks," *Sensors*, vol. 15, no. 9, pp. 22 587–22 615, 2015.

[31] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[32] M. Menna, M. Gianni, F. Ferri, and F. Pirri, "Real-time autonomous 3d navigation for tracked vehicles in rescue environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.

[33] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Point cloud segmentation and 3d path planning for tracked vehicles in cluttered and dynamic environments," in *Proc. of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, 2014.

[34] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI - Knstliche Intelligenz*, vol. 29, no. 2, pp. 193–201, 2015. [Online]. Available: http://dx.doi.org/10.1007/s13218-015-0352-5

A New UGV Teleoperation Interface for Improved Awareness of Network Connectivity and Physical Surroundings

Ramviyas Parasuraman
Purdue University, USA
Sergio Caccamo, Fredrik Båberg, Petter Ögren
KTH Royal Institute of Technology, Sweden
and
Mark Neerincx
TNO, Netherlands

A reliable wireless connection between the operator and the teleoperated unmanned ground vehicle (UGV) is critical in many urban search and rescue (USAR) missions. Unfortunately, as was seen in, for example, the Fukushima nuclear disaster, the networks available in areas where USAR missions take place are often severely limited in range and coverage. Therefore, during mission execution, the operator needs to keep track of not only the physical parts of the mission, such as navigating through an area or searching for victims, but also the variations in network connectivity across the environment.

In this paper, we propose and evaluate a new teleoperation user interface (UI) that includes a way of estimating the direction of arrival (DoA) of the radio signal strength (RSS) and integrating the DoA information in the interface. The evaluation shows that using the interface results in more objects found, and less aborted missions due to connectivity problems, as compared to a standard interface.

The proposed interface is an extension to an existing interface centered on the video stream captured by the UGV. But instead of just showing the network signal strength in terms of percent and a set of bars, the additional information of DoA is added in terms of a color bar surrounding the video feed. With this information, the operator knows what movement directions are safe, even when moving in regions close to the connectivity threshold.

Keywords: teleoperation, UGV, search and rescue, FLC, network connectivity, user interface

1. Introduction

Today, teleoperated UGVs play an increasingly important role in a number of high risk applications, including urban search and rescue (USAR) and explosive ordinance disposal (EOD). The successful

Journal of Human-Robot Interaction, Vol. 6, No. 3, 2017, Pages 48–70. DOI 10.5898/JHRI.6.3.Parasuraman

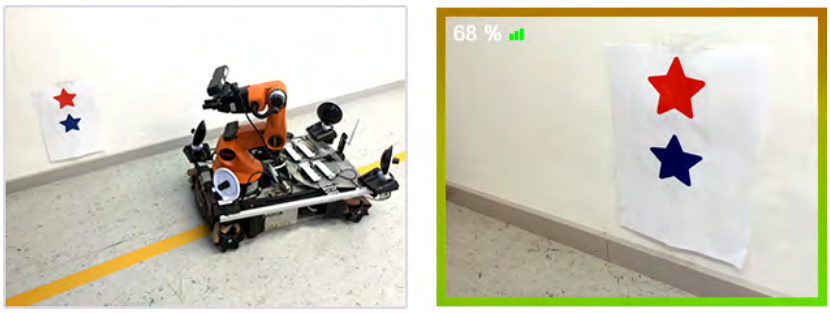


Figure 1. The youBot mobile robot equipped with wireless network hardware used in the experiments (left) shown along with the user interface (UI) displaying the RSS DoA as a color bar around the video feed from the robot.

completion of these missions depend on a reliable communication link between operator and UGV, but unfortunately, experiences from Fukushima and the World Trade Center disasters show that cables can limit performance, or break (Nagatani et al., 2013), and wireless network connectivity can be lost (Murphy, 2014).

It is reasonable to believe that the very nature of USAR scenarios imply a high risk of damages to infrastructure, including to electricity and wireless network facilities. To avoid relying on a stable network connection, one possible solution would be to enable the UGVs to operate autonomously, but for the foreseeable future, human operators will remain more versatile than autonomous systems when it comes to decision making, particularly in challenging and unpredictable USAR environments (Muszynski, Stuckler, & Behnke, 2012; Wegner & Anderson, 2006).

Yanco and Drury (2004) defines the situation awareness (SA) in the context of human-robot interaction as follows: *the perception of the robots' location, surroundings, and status; the comprehension of their meaning; and the projection of how the robot will behave in the near future*. Thus, the connectivity awareness is viewed as a component of SA (network status), determining where the robot can operate.

In this paper, we address the problem of improving SA such that the operator is aware of dynamic network connectivity status and adjust the UGV operation to it. This is done by extending the user interface (UI) with not only a measure of radio signal strength (RSS), but also a notion of the motion direction (i.e. the DoA) that would increase this signal strength, and thereby communication quality (delay, packet loss, etc.), which is known to affect teleoperation task performance (Owen-Hill, Parasuraman, & Ferre, 2013; Rank, Shi, Mller, & Hirche, 2016).

Using the proposed solution, an operator close to the connectivity limit knows which way to go to improve the connection. An operator who, for example, would like to move the UGV a bit more to the left to inspect a cavity knows if this move will improve, worsen, or leave the RSS unchanged.

The proposed UI is composed of two parts; first the DoA is estimated, then it is presented to the operator in an efficient manner. The estimation of the DoA is done by using spatially dispersed wireless receivers on the four edges of the UGV (as can be seen in Fig. 1) and applying the finite differences method to extract the RSS gradient. We then employ spatial and temporal filtering schemes to mitigate multipath fading effects and transient noises in the measurements. The estimation and filtering algorithms run online and dynamically adapt to changes in the wireless environment, such as a change in network connection (e.g., introduction of an intermediate relay robot as a signal repeater) or movement of a mobile wireless access point connecting the robot to the base station.

The presentation of the DoA to the operator was chosen in view of the fact that gaining a good

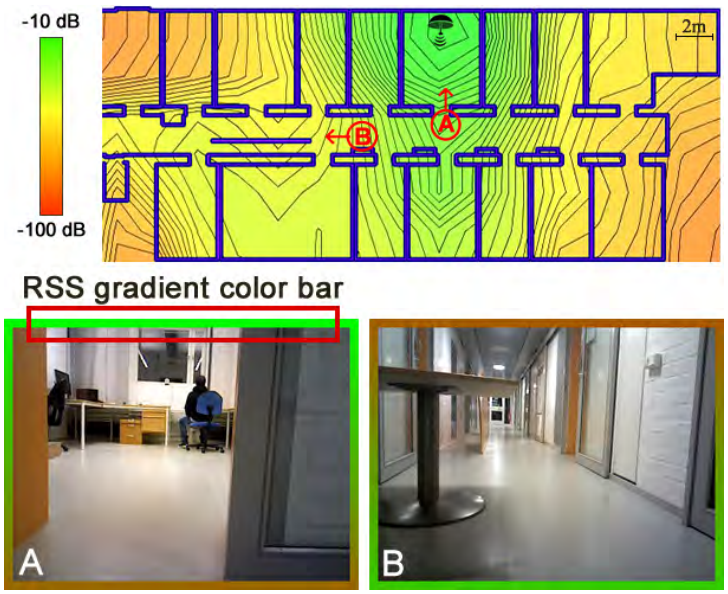


Figure 2. A map of the RSS in an office environment and two examples of the UI with the UGV at positions A and B. Note the green and red gradient, indicating higher and lower signal directions (DoA) in the color bars surrounding the videos.

SA is very challenging in USAR missions (Larochelle & Kruijff, 2012). In fact, it was shown in Burke, Murphy, Coovert, and Riddle (2004) and Yanco and Drury (2004) that as much as 49% of mission time is normally devoted to improving the operator SA. Further, it was recommended in Yanco and Drury (2007) to use a large central part of the screen for the video feed. Therefore, we propose to represent the DoA information in the form of a color bar surrounding the video feed (see Fig. 2) to provide SA to the operator in terms of connectivity status and physical surroundings. Note that there are many possible variations on the proposed idea of graphically illustrating the DoA, including arrows of different forms and placements. However, we focus the investigation on the potential benefits of providing such information. Comparing different variations on the theme is beyond the scope of this study.

For the evaluation, we identified two important challenges associated with teleoperation of UGVs in USAR missions: (1) providing effective SA to the operator and (2) ensuring resilient wireless connectivity with the UGV. High SA can reduce mission time and improve operator decisions, while a resilient network connection will avoid losing control of the UGV.

The main contributions of this paper are threefold. We first propose a new way of estimating DoA for teleoperated UGVs. We then propose a way of integrating this DoA information in a UGV teleoperation UI. Lastly, we perform a user study, showing that the proposed approach, in fact, increases the number of found objects during a search mission and decreases the chances of losing the connection to the UGV. To the best of our knowledge, none of these items have been done in a UGV teleoperation context before. This paper extends our previous work (Caccamo, Parasuraman, Båberg, & Ögren, 2015), with an improved design and a thorough evaluation of the proposed interface.

The paper is organized as follows. First, Section 2 reviews the literature on this topic and Section 3 describes the proposed approach. Then, Section 4 describes the human in the loop experi-

ments, with results in Section 5 and the discussion in Section 6. Finally, we conclude in Section 7 and provide suggestions for further work.

2. Related Work

The wireless network connectivity of USAR UGVs has often proved unreliable (Carlson & Murphy, 2005; Murphy, 2004), with examples including real incidents where robots were lost during disaster inspection operations (Murphy, 2014; Nagatani et al., 2013). Casper and Murphy (2003) investigated user confidence in remotely operated robots with intermittent communications, and found that these problems had a significant impact on the usability of the systems. They even suggested that because of communication dropout problems, wireless robots should be avoided. However, the flexibility of wireless systems compared to tethered robots still makes them an important alternative in many applications.

A natural way of avoiding loss of communications is to make the user aware of the connection quality. A decade ago, this information was usually not displayed in the operator control unit (OCU) (Fong & Thorpe, 2001), but more recently, it is often added in the form of a "signal bar" (as in modern cell phones) or in form of a percentage. Typical examples of such representation can be seen in Larochelle, Kruijff, Smets, Mioch, and Groenewegen (2011) and Hedström, Christensen, and Lundberg (2006), including the recent Quince 2 robot's OCU (Yoshida, Nagatani, Tadokoro, Nishimura, & Koyanagi, 2014). Furthermore, the Wayfarer OCU for Packbot robots (Yamauchi, 2004) represents the radio signal level in a vertical bar manner, in addition to providing a numeric indicator.

The literature on robot interfaces also includes examples where information about gradients and directions is made available to the user. In Hestand and Yanco (2004) and Baker, Casey, Keyes, and Yanco (2004), two microphones on the left and right of the robot were used to estimate the direction of a sound source, which was displayed (overlaid on the video) in the form of a pointer floating on horizontal and vertical lines. A similar representation was used in Hedström, Christensen, and Lundberg (2006) to show robot speed information. In Barros, Lindeman, and Ward (2011), the authors proposed a tactile belt that vibrates in the direction of detected collisions to improve SA, while in Smets, Brake, Neerincx, and Lindenberg (2008) a study found that the use of a tactile vest did not improve SA significantly in navigation tasks.

An influential study in human-robot interface (HRI) design (Yanco & Drury, 2007) advocates the use of a large single interface with a significant percentage of the screen dedicated to video. The authors also recommend providing more spatial information about the environment to increase SA and using fused sensor information to lower the cognitive load on the user. Moreover, multi-sensory interfaces had also been advocated in the literature (Barros & Lindeman, 2014).

In this paper, we go beyond the related work described above by having the teleoperation interface include not only a scalar value to describe the network connectivity situation, but also the direction in which it is expected to improve (i.e., the DoA). Assessing the geographic distribution of network connectivity is a spatial task, for which the visual modality fits best with the human information processing (see, e.g., the multi-resource model of Wickens (2008)). Therefore, we choose to present the DoA in the form of visual gradient bars surrounding the video feedback.

Carefully integrating the DoA information into the visual feedback is crucial. For this, we use FLC (Free Look Control) (Ögren, Svenmarck, Lif, Norberg, & Söderbäck, 2014) as the control layer. FLC is essentially a "navigate-by-camera" mode as envisioned in Yanco et al. (2006). In the FLC mode, the operator controls the UGV in relation to the camera frame instead of the world frame, making it more intuitive than the traditional so-called *Tank Control* mode. Hence, it is appropriate to use FLC for presenting the DoA information in direct reference to the camera frame, making the UGV control easier while simultaneously enhancing local SA.

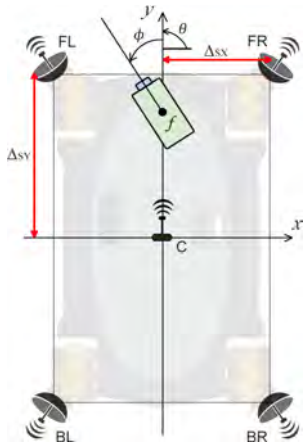


Figure 3. UGV equipped with a camera, one wireless adapter at the center, and four wireless adapters with directional antennas at the corners.

3. The proposed approach

The new user interface, as for most robot teleoperation UIs, is composed of two parts: receiving control commands from the operator and providing feedback to the operator. In the former, we use a gamepad controller and the FLC interface (Ögren, Svenmarck, Lif, Norberg, & Söderbäck, 2014) and in the latter, we present the DoA as an extra sensory feedback in addition to the video stream. Below, we present the hardware configuration and associated signal processing required to realize the new interface.

3.1 FLC interface

FLC, a UGV control interface inspired by the first person shooter (FPS) video games genre, combines camera and platform control, thus permitting the operator to completely focus on commands for moving the UGV camera (the UGV adjusts its heading accordingly) through the remote environment (world frame) without worrying about the orientation of the UGV chassis. This is in contrast with the standard interface, *Tank Control*, which is used in most of the teleoperated UGVs today, where the operator is required to mentally keep track of at least two orientations while teleoperating an UGV: the camera angle relative to the UGV, and the platform orientation with respect to the world frame. The advantages of FLC compared to Tank Control were investigated in Båberg, Caccamo, Smets, Neerincx, and Ögren (2016), and more details about implementing FLC can be found in Ögren, Svenmarck, Lif, Norberg, and Söderbäck (2014).

3.2 DoA estimation

A good estimate of the DoA forms a core part of the new interface. It has been shown that the DoA can be estimated by the direction of the RSS gradients (Han, Andersen, Kaminsky, Papagiannaki, & Seshan, 2009). For calculating RSS gradients, we use four wireless adapters connected to external directional antennas placed on the corners of the UGV,[1] as shown in Figs. 1 and 3. We also use a fifth wireless adapter connected to an omnidirectional antenna. The former four adapters are used for

[1] The squared planar arrangement of antennas is suggested in Parasuraman, Kershaw, and Ferre (2013), due to its robust nature. Moreover, the directional antennas are chosen due to high stability and accuracy in the measurement and higher link throughput (Min, Matson, & Khaday, 2013).

DoA estimation, whereas the latter is used for actual communication for the teleoperation between the UGV and the control station.

We measure the RSS (in dBm) from all the wireless adapters using the received signal strength indicator (RSSI[2]) metric, which is usually prone to noise and temporal variations (due to environmental dynamics) (Lindhè, Johansson, & Bicchi, 2007; Rappaport, 2001). These noise fluctuations are mitigated by applying an exponentially weighted moving average (EWMA) filter on the measured RSS from each wireless adapter using the following model (Parasuraman et al., 2014):

$$R^f(i) = R^f(i-1) + \alpha(R(i) - R^f(i-1)), \tag{1}$$

where $R(i)$ is the RSS value measured at the i^{th} instant, R^f is the filtered RSS value and α is an empirical smoothing parameter. In addition to the EWMA filter, a moving average filter (MAF) is also applied to mitigate spatial multipath fading, with a window size equal to about 10λ (λ being the wavelength) as suggested in Valenzuela, Landron, and Jacobs (1997). The MAF window depends on the UGV velocity, RSS sampling frequency, and the wavelength of the radio signal.[3]

Modeling the RSS as a two-dimensional scalar field[4] it is possible to obtain the gradient of the RSS field ($\overrightarrow{g} = [g_x, g_y]$) with respect to the center of the UGV using the central finite difference method (Bezzo et al., 2014; Parasuraman, Fabry, Kershaw, & Ferre, 2013):

$$g_x = \frac{(R_{FR} - R_{FL})}{2\Delta_{SX}} + \frac{(R_{BR} - R_{BL})}{2\Delta_{SX}}, \;\; g_y = \frac{(R_{FR} - R_{BR})}{2\Delta_{SY}} + \frac{(R_{FL} - R_{BL})}{2\Delta_{SY}}, \tag{2}$$

where Δ_{SX}, Δ_{SY} are the corresponding spatial separations between the antennas, R_{FR}, R_{FL}, R_{BR} and R_{BL} are the filtered RSS values of the front-right (FR), front-left (FL), bottom-right (BR), bottom-left (w.r.t the center of the UGV) receivers respectively, as can be seen in Fig. 3. The orientation of each antenna is aligned with its placement. It is possible to employ redundant gradient estimation methods to tackle device failures or misreadings, as discussed in (Parasuraman et al., 2014).

From the RSS gradient ($\overrightarrow{g}$), the DoA of the radio signal is obtained as,

$$\text{DoA}_\theta = \tan^{-1}(\frac{g_y}{g_x}). \tag{3}$$

3.3 User Interface

A large video feed on the OCU is used for teleoperating the UGV. This visual interface is extended to include wireless connectivity information by adding a colored gradient bar surrounding the real time video feed from the UGV. The added color bar illustrates the DoA relative to the camera view. This setup was inspired by computer game interfaces, where the direction of threats causing health-level changes is communicated using flashing colors in the appropriate part of the screen. Consequently, the new interface can be categorized as Type 3.2.2.1 (additional visual input: type - communication level) in the framework for analyzing human robot interaction (Richer & Drury, 2006).

In the UI, we create a rectangular border around the video, as illustrated in Fig 2. As the DoA computed with Eq. (3) is first given in the UGV frame, it is converted to the camera frame (to provide a first person view of the DoA to the operator). Then we translate the DoA to a color gradient bar around the camera view by scaling the color intensity according to a linear interpolation of the

[2] RSSI is a vendor-specific metric and therefore reports different values (or quantities) in different devices. The wireless adapters used in this paper reported reliable values of absolute signal power (dBm) as RSSI.

[3] For instance, if the UGV velocity is $0.2\,\text{m/s}$, RSS sampling frequency is $5\,\text{Hz}$, and using $2.4\,\text{GHz}$ signal (wavelength $\lambda = 12.5\,\text{cm}$), the MAF window size should be ≈ 30 to filter samples within $1.25\,\text{m}$ (10λ) displacement of the UGV.

[4] Being a ground vehicle, the UGV moves locally in a plane.

measured RSS values around the corners. A green color in the color bar indicates the higher signal strength direction, whereas a red color indicates a lower signal strength direction. Thus, the interface not only represents DoA but also gives a sense of the absolute RSS.

3.4 Experimental verification

In Caccamo, Parasuraman, Båberg, and Ögren (2015), we investigated the accuracy of the proposed DoA estimate. Specifically, we performed experiments to verify that the variation of the RSS along a robot path is indeed predicted by the DoA estimates. We summarize the key findings in this section.

Firstly, it was found that the DoA estimation had high accuracy in both line-of-sight and non-line-of-sight conditions, the absolute mean DoA error was within 0.2 rad (12 degrees), an accuracy that will turn out to be good enough for our purposes.

Secondly, a set of experiments were performed to evaluate the sensitivity, specificity, accuracy and precision of the DoA feedback provided by the interface. To gather data, the robot was tele-operated by a human operator, simulating short missions following different paths. Eight different trials of this kind were conducted. During each trial, we logged the robot position data obtained from the dead-reckoning of the wheel odometers, the RSS data, the estimated DoA and the streamed video. The dead-reckoning of the wheel odometers was not accurate, but this was not a problem as both motion directions and estimated DoA are given in the same local coordinate system. A video illustrating the proposed method with an example trial is available online.[5] As seen in the video illustration, the estimated DoA sometimes pointed towards the corridor or the doorways (instead of the true source location). This is expected because of substantial exposure of radio signals from these regions.

In a noise-free world, the following equality would hold:

$$\frac{dR}{dt} = \frac{dR}{dx}\frac{dx}{dt}\,, \tag{4}$$

where $x \in \mathbb{R}^2$ is the spatial dimension. The real world is however far from noise free, and we had to experimentally verify that our estimates provide useful information to the human operator. For the estimates to be useful, the measured RSS should increase when the UGV is moved in the direction of the DoA (i.e., the two sides of Eq. 4 should have the same sign). We used temporal differences in the measured RSS at the central receiver (R_C) to estimate $\frac{dR}{dt}$, $\overrightarrow{g}$ as the estimate of $\frac{dR}{dx}$, and the odometer robot velocity $\overrightarrow{\nu}$ to estimate $\frac{dx}{dt}$. The scalar (dot) product between the robot velocity and the computed RSS gradient at each instant is given by:

$$p(t) = \langle \overrightarrow{g}(t), \overrightarrow{\nu}(t) \rangle. \tag{5}$$

By comparing the scalar product $p(t)$ with the change in the RSS at the central receiver $\nabla_t R_C = \frac{dR_C}{dt}$, we evaluated the efficacy of the proposed system. We expected a steep increase in R_C when $p(t)$ is positive and close to 1 (i.e. when the user is moving towards the DoA). Similarly, we expected a sharp decrease in R_c when the p(t) is negative and close to -1 (i.e. when the user moves the robot away from the DoA).

Fig. 4 shows the variations of RSS at the central receiver (R_C) and the scalar product $p(t)$ with time for a sample trial. To quantify the system performance, we measured the number of true/false (T/F) positives/negatives (P/N) in the outcome. Using these measures, we computed sensitivity ($\frac{TP}{TP+FN}$), specificity ($\frac{TN}{FP+TN}$), precision ($\frac{TP}{TP+FP}$), and accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$) metrics.

In Table 1, we present the key results obtained from the eight experiments with an average mission time of 9.2 minutes each. The proposed system delivered high accuracy and precision in

[5] https://youtu.be/YcbPi1c7eaQ

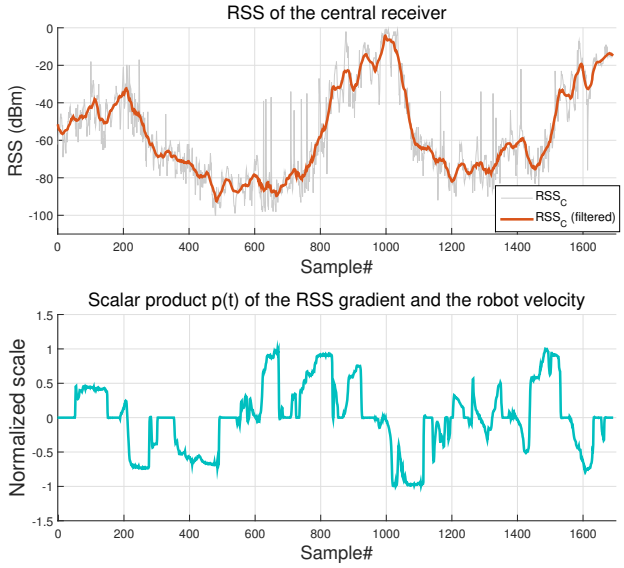


Figure 4. Quantitative evaluation of the new UGV teleoperation interface with the RSS DoA feedback. The estimate is useful when the scalar product $p(t)$ has the same sign as the derivative of the RSS. i.e. the changes in the RSS values follow the directions indicated by $p(t)$.

Table 1: Evaluation of the DoA feedback for sensitivity, specificity, precision and accuracy.

	Sensitivity	Specificity	Precision	Accuracy
Mean	0.74	0.83	0.82	0.78

guiding the teleoperator with network connectivity feedback in an indoor environment. As the analysis depended on the UGV's velocity from the odometer, we note that odometry errors could have impacted the analysis of the accuracy of the proposed system. Thus, a better localization technique would have improved the overall system analysis. Note that the system has shown reasonable sensitivity in directing the operator into high wireless signal regions (towards DoA) while maintaining high specificity in pointing out low-wireless signal regions.

Although the above quantitative results are fairly promising, a qualitative evaluation with user studies is required to investigate the effectiveness of the overall system. This will be done in the following sections.

4. User evaluation

To evaluate the actual system performance of the new interface, we conducted experiments with human subjects. The experimental setup consisted of an exploration task (search for symbols) with

Figure 5. Visualization of the proposed VDOA (A) and the standard BAR (B) interfaces to represent connectivity information.

a remotely operated UGV platform in an unfamiliar, maze-like environment. The objective of the experiments was to evaluate the new visual DoA interface (denoted *VDOA* hereafter) against the (state-of-the-art) standard OCU interface that displays radio connectivity using a signal bar and percentage value (denoted *BAR*). The two interfaces are shown in Fig. 5. To allow a fair comparison, both interfaces used FLC as control layer. Note that we are interested in the evaluation at the first two levels of SA (perception, comprehension) (Endsley, 2000) because the proposed DoA interface does not predict the future connectivity status (Level 3 - prediction). Nevertheless, *VDOA* allows the operator to infer the present and future network availability in different travel directions.

4.1 Evaluation framework

When designing the experiments, we followed the situated cognitive engineering (sCE) method (Neerincx & Lindenberg, 2008) in which we first identified the two *core functions* that we want to compare (see items 1 and 2 below). Then, we formulated a number of *claims*, that is, hypothesis connected to the *core functions*, listing a number of possible *upsides* (benefits) and *downsides* (drawbacks) of each hypothesis. These upsides and downsides are then used to define *what to measure* in the experiments. Finally, having performed the experiments, we can then see which of the possible upsides/downsides are confirmed by data and, hence, draw conclusions about the *claims* and when the different *core functions* can be beneficial.

The following core functions describe what the corresponding systems do.

1. *VDOA provides a graphical indication of the DoA and the RSS value in the periphery of the teleoperation display (see Fig. 5, left). It also shows the RSS value in the same way as BAR.*

2. *BAR shows the RSS value in the form of both a percentage and a set of signal bars (see Fig. 5, right).*

Listed below are the claims that we make on the core functions, with corresponding upsides/downsides (U/D) and what to measure in parenthesis.

Claim 1: VDOA leads to UGV trajectories in higher signal strength regions.

- U11: Less error in the estimated DoA (*radio source localization*)
- U12: Increased connectivity and less connection loss (*signal strength, loss of connectivity*)

- U13: More useful area covered and more time spent during exploration due to less connection loss (*coverage, execution time*)
- D11: Less concentration on the surroundings, such as objects and obstacles in the robot proximity (e.g., while the user follows the VDOA guidance to change direction) (*collisions, ease of finding symbols*)

Claim 2: VDOA provides better situation awareness.

- U21: Better SA on the search task and connectivity status (*situation awareness*)
- U22: More symbols found (*symbols found*)
- U23: More spatial understanding (*symbols mapping accuracy, radio source localization*)
- D21: Higher mental effort due to additional information (*mental effort*[6])

Claim 3: VDOA improves user experience.

- U31: Better usability and satisfaction of the useful DoA information on the interface (*usability, preference*)
- U32: Higher time utilization or longer time spent on actual tasks due to higher connectivity awareness and less connection loss (*execution time, symbols found*)
- U33: Better understanding of the network connectivity across various regions (*radio source localization, loss of connectivity*)

Claim 4: BAR increases focus and concentration on the actual task.

- U41: Less collisions during exploration (*collisions*)
- U42: Lower mental effort (*mental effort*)
- U43: Easier to operate the robot and use the interface due to its simplicity (*preference, usability*)
- D41: More connection loss, since it is more difficult to understand spatial (2D) wireless connectivity using BAR (*loss of connectivity, signal strength*)
- D42: Less situation awareness and less spatial understanding in terms of network status (*situation awareness, coverage, execution time, symbols mapping accuracy*)

These claims are used to argue which interface is better suited for UGV teleoperation, especially in a USAR scenario. The upside or downside of a claim can be confirmed if it is supported by at least one of its measures. The validation of these claims help us to determine the interface that is effective (maintains better connectivity), productive (higher task utilization, higher coverage areas), and is more appreciated by the operators (preference and better usability).

4.2 Method

4.2.1 Experimental design Considering that the radio propagation in a given environment is unique to specific settings, we conducted *"between-subjects"* trials instead of *"within-subjects"* trials for comparing the two interfaces. This is due to the very high probability of carryover effects associated with the memory of radio signal coverage if a *"within-subjects"* design is performed. Therefore, following a *"between-subjects"* design, N participants in each interface group (VDOA

[6] Rating Scale of Mental Effort (RSME) (Zijlstra & Doorn, 1985)

Table 2: Measurement variables used in the user evaluation.

Measurement	**How?**	**Claim**
Subjective		
Reported overall usability	Q	U31, U43
Mental effort	Q	D21, U42
Preference	Q	U31,U43
Ease of finding symbols	Q	D11
Situation Awareness (exploration, network)	Q	U21, D42
Objective		
Number of symbols found	Obs	U22, U32, D42
Situation Awareness (spatial - symbols mapping)	Q+Datalog	U23, D42
Number of collisions	Obs	D11, U41
Execution time	Obs+Datalog	U13, U32
Localization of radio source	Q+Obs	U11, U23, U33
Coverage (area/distance)	Datalog	U13, D42
Number of connection losses	Datalog	U12, U33, D41
Radio Signal Strength (RSS)	Datalog	U12, D41

and BAR) are recruited for executing tasks based on a set of instructions (explained below). The nature of the opponent group/experiment is revealed to the participants only at the end of the experiment (and survey) to avoid biasing effects.

4.2.2 Participants Based on statistical expectations on the outcome and the characteristics of the measured variables, the results of sample size and power calculations reveal that at least eight[7] participants are required for each group. Thus, a total of at least 16 participants were required. However, we recruited a total of 24 participants for this study to increase the power. The participants (15 male and 9 female) were all university students and staff in the same age group (mean age: 27.9). Most of the participants did not have prior experience with robots or UGVs (mean experience: 2.04 out of 5). Although we conducted the user study with 24 participants, the data of 4 participants were not useful because of technical issues such as motor drive fault, operator fault, etc. faced during the experiment. Therefore, we used the data of 20 participants (12 male, 8 female) with ten (N = 10) in each control group in our analysis.

4.3 Variables

In accordance with the claims to be tested, Table 2 lists the variables measured in the experiments. In the *How* column, the way of collecting the measurements is indicated as data logging in the real robot (Datalog), through manual observations (Obs), or through a questionnaire (Q). The *Claim* column shows the associated claims (upsides/downsides) of each measurement.

4.4 Test environment

4.4.1 Procedure Written and verbal instructions were given to the participant at the beginning. Participants then had to answer (fill in) general questions on their experiences with robots and games. They were then informed about the experiment, as per the instructions. This was followed by a training session where the users were asked to drive the UGV in a rectangular path in a small room without colliding. The user was also given the real position of the radio transmitter (used for training) in order to assess the connectivity information in the UI. The training session lasted until the users

[7] This number was derived using the standard power tests (Dell, Holleran, & Ramakrishnan, 2002) assuming a power level of 80% and false positive rate of 5% with at least 20% difference in the expected means of the two groups (with a standard deviation of 20%).

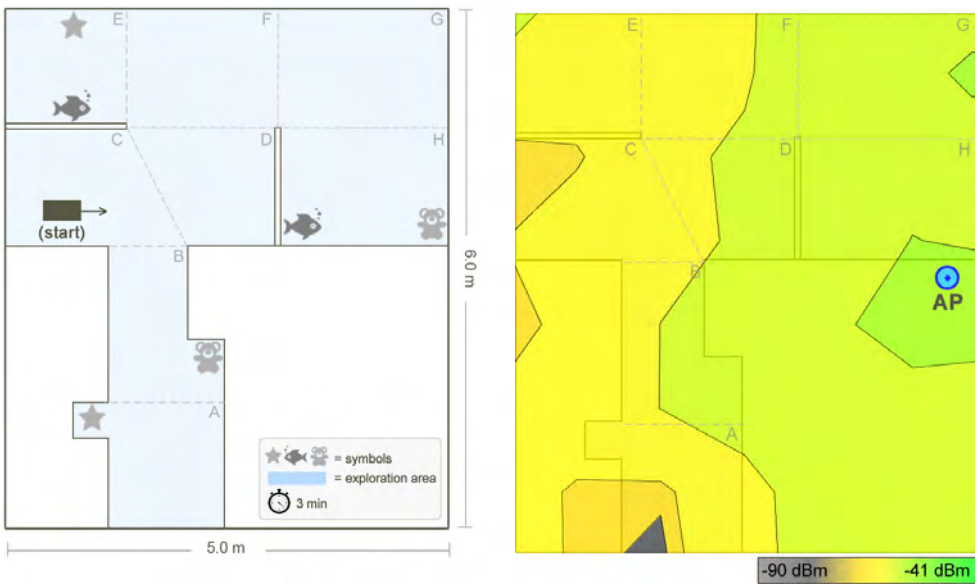


Figure 6. (Left) The map of the task scenario where the participant is asked to find symbols by exploring a given maze area. (Right) Snapshot of RSS map captured using a commercial wireless site survey tool from Ekahau. The radio source (WiFi router) location is marked as AP.

expressed comfort in using both the FLC for control and the UI for perception. [8] The real evaluation experiments commenced after the training session. The evaluation *task* is explained below. Note that we used two wireless routers placed in different positions, one for training and the other for the actual task.

After completing the experiments, the participants were asked to complete questionnaires on their experiences, situation awareness,[9] metal effort, and various other factors that are listed in Table 2 with the label "Q". The participants were also asked to indicate on a map similar to the one in Fig. 6, the location of symbols they found, the estimated radio source location, and the path taken by the UGV including the end position and orientation.

4.4.2 Hardware We used the same hardware and experimental setup as in Caccamo, Parasuraman, Båberg, and Ögren (2015).

4.5 Task

Participants were asked to drive around an indoor environment (as they are more challenging for wireless signals) to search for known symbols as depicted in Fig. 6. For this, a specially built maze was used. The maze is virtually split into eight regions as indicated with dotted lines. A time limit

[8] All training sessions lasted between 2–5 min.

[9] The key questions related to SA (on a scale of 1(No/Hard) to 5(Yes/Easy)) are the following: I have found all the symbols; I had enough time for exploring the area; How difficult was it to find the objects in the environment?; I think I have drawn the positions of the source correctly into the map; I think I have drawn the end position and orientation correctly; How difficult was it to find the source in the environment?

(3 minutes) was given to find symbols within the maze. The symbols shown in the figure have an area of approximately $40\,\mathrm{cm}^2$ and are placed on the walls of the maze with full visibility when the camera is aimed at them.

The goal of the operator was to find as many symbols as possible without losing connectivity to the UGV. The experiment was stopped when either a timeout period was reached or when the user had lost connectivity. The participant had no direct line of sight with the UGV, and the only source of SA was the UI.

During the task, data such as odometry, RSS, and loss of connectivity were recorded in a datalog. Execution time and the number of collisions were observed by a supervisor. The actual location of the radio source is indicated as AP on the map in Fig. 6 (which was not revealed to the participants). Often, the regions A and B experienced poor connectivity with high probability to lose connection, whereas the regions C and E experienced average connectivity but had a lower probability to lose connection. Finally, D, F, G and H experienced high connectivity levels. Thus, as will be seen, how and when the regions A and B are approached turned out to be crucial to mission performance. Note that, as can be seen in Fig. 6, the symbols are placed in a manner such that they are equally distributed in different connectivity zones (poor, medium, high) of the exploration area.

5. Results

Fig. 7 presents a boxplot result of the important variables. A summary of the user evaluation results can be found in Table 3. Below we describe the results in more detail, first in general, then specifically for the exploration task, and finally, we present the results related to the wireless network.

5.1 General results

5.1.1 Usability To measure the interface usability, we used a questionnaire that required seven responses,[10] each of which is scored between 1 (disagree) to 5 (agree), resulting in an overall value between 0 and 28 (5×7-7), where 0 is the most difficult to use and 28 is the easiest to use. The resultant value is obtained by summing the scores of four positive statements, subtracting the scores of three negative statements, and adding an offset of 11 to obtain a positive scale of 0–28.

A two-tailed[11] independent samples t-test was conducted to compare the usability of the VDOA and BAR. There was a significant difference in the reported usability with VDOA (M = 21.3, SD = 3.88) and BAR (M = 10.4, SD = 2.91) conditions ($t(18) = 7.09$, $p < 0.01$). This shows that the participants found the VDOA interface significantly easier to use than the BAR interface.

5.1.2 Mental Effort To rate mental effort we used the RSME scale (0 - absolutely no effort to 150 - extreme mental effort), which is essentially a one-dimensional version of the NASA-TLX scale. The resulting RSME scores of VDOA participants are M = 54.9 and SD = 26.25, whereas in the BAR group, the scores are M = 49.1 and SD = 20.2, respectively. It is interesting to note that there is no significant difference between the RSME scores of the two groups ($t(18) = 0.55$, $p = 0.58$). This means that the users of the VDOA interface experienced slightly but not significantly higher mental effort than the BAR group. Thus, the addition of the DOA interface did not have much impact on the cognitive load of the participants.

5.1.3 Preference As a between-subject study where each participant is assigned only to one group, the evaluation of the users' preference is handled as follows. After the whole experiment and at the end of the questionnaires, we briefly explained the alternative interface (VDOA in case of BAR participants and vice versa) and asked the participant to answer the question if they would choose

[10] Sample statements: I thought the interface was intuitive; I found the various functions in this interface well integrated; The interface response was slow; I thought the interface was easy to use; I enjoyed the experiment.

[11] All the analyses made in this paper are of two-tailed nature. M indicates mean and SD indicates standard deviation.

Table 3: Summary of the user evaluation results. A (+) sign indicates relatively better value.

Measurement	Predicted	Best in Eval.	VDOA *M*	VDOA *SD*	BAR *M*	BAR *SD*	*t*(18)	*p*
General measures								
Reported usability	-	VDOA	21.3(+)	3.88	10.4	2.91	7.09	<0.01
Mental effort	BAR	No sign. res.	54.9	26.25	49.1(+)	20.2	-	-
Preference	-	VDOA	1.8(+)	1.23	4.1	1.45	-3.82	<0.01
Task-related measures								
No. of Symbols Found	VDOA	VDOA	4.5(+)	0.97	2.6	2.07	2.63	<0.05
Ease of finding symbols	BAR	No sign. res.	4(+)	1.25	3.3	1.42	-	-
Execution Time (s)	VDOA	VDOA	167.8(+)	20.53	120.5	67.2	2.13	<0.05
No. of collisions	BAR	No sign. res.	0.8	0.92	0.4(+)	0.7	-	-
Coverage (m)	VDOA	VDOA	8.78(+)	2.84	5.28	3.3	2.55	<0.05
Situation awareness (explore)	VDOA	VDOA	4.2(+)	1.32	2.5	1.78	2.42	<0.05
Situation awareness (spatial)	VDOA	No sign. res.	1.63(+)	0.81	1.65	0.99	-	-
Network-related measures								
Localization of Router/AP (m)	VDOA	VDOA	1.39(+)	1.02	2.47	1.56	-1.83	<0.1
Situation awareness (network)	VDOA	VDOA	4.5(+)	0.97	3.6	1.26	1.78	<0.1
Connection loss	VDOA	VDOA	4/10(+)	-	6/10	-	-	-
Connection quality (RSS, dBm)	VDOA	VDOA	2.83(+)	1.02	-0.54	3.45	2.96	<0.01

the alternative interface if they were given another chance. The user could answer between 1 (No) to 5 (Yes). Note the measure used is *preference to the alternative interface* and not the *absolute preference to the used interface*. Participants that used the VDOA interface were less likely to switch to the BAR interface (i.e., not to keep using VDOA) with an average score of 1.8 (SD = 1.23), while significantly more participants in the BAR group preferred to switch to VDOA interface with mean score of 4.1 (SD = 1.45). The significance conditions are $t(18)$ = -3.82 and $p < 0.01$. Note that this measure could be biased due to the general notion that humans tend to think more information is better.

5.2 Results for the exploration scenario

5.2.1 Finding symbols Here we analyze how participants explored the maze in terms of the main exploration task, which is to find as many symbols as possible.

Number of symbols found - An independent samples t-test was conducted to compare the number of symbols found in the explore task. There was a significant difference in the number of objects found in VDOA (M = 4.5, SD = 0.97) and BAR (M = 2.6, SD = 2.07) conditions ($t(18) = 2.63$, $p < 0.05$). More symbols were found with VDOA than with BAR in the actual exploration task which means the participants were able to focus on the task more productively.

Ease of finding symbols - In terms of finding symbols with ease, we asked the participants to indicate how difficult it was to find symbols during the task. The participants rated the difficulty between 1 (hard) and 5 (easy). We expected the participants that used VDOA to have found it harder to find symbols as they had to share their focus between both video and the DOA interface. However, the results suggests otherwise. The VDOA (M = 4, SD = 1.25) respondents reported more ease in finding symbols than the BAR ones (M = 3.3, SD = 1.42), but the difference is not statistically significant.

We may conclude that adding the DOA interface did not affect the operators' ability to understand the spatial surroundings.

5.2.2 Execution time Recall that we provided 180 seconds (3 minutes) for each participant to explore the maze. The only reason for termination before the given time limit is when the robot loses connectivity with the control station, which will be displayed in the UI as a "SIGNAL LOST" message on front of the video feed. We manually observed with a stopwatch and logged the execution

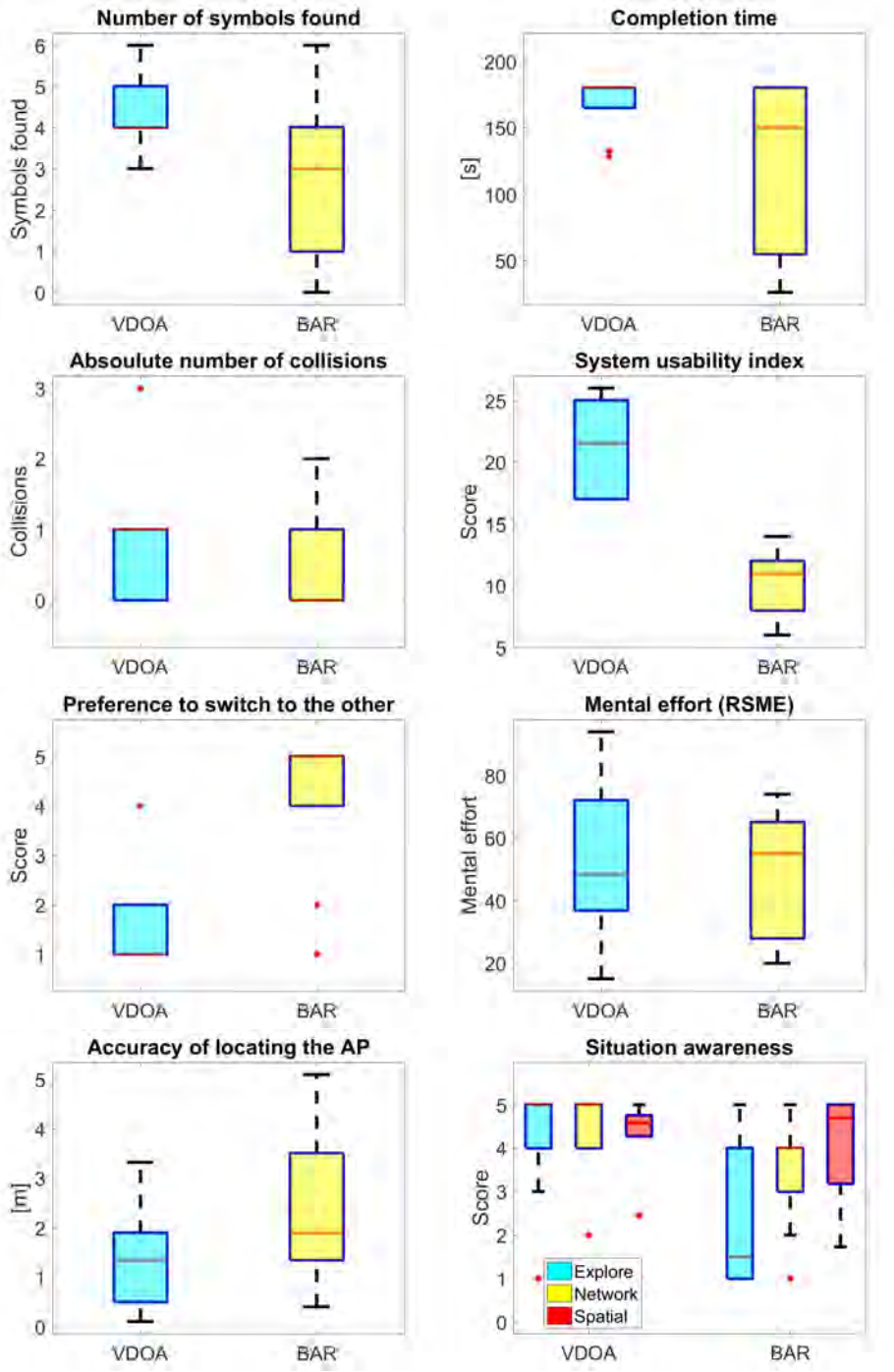


Figure 7. Resultant scores of important variables shown as boxplots. The red lines indicate the median values. A higher value indicates better results, except in the following variables: *absolute number of collisions; preference to switch to the other; accuracy of locating the AP.*

times. As the participants did not know in advance how many symbols were there, they would normally spend all of the 180 seconds searching for symbols. The hypothesis is that VDOA users are less likely to lose connection and hence end up with longer execution times. We found a significant difference in the execution time for VDOA (M = 167.8 s, SD = 20.53 s) and BAR (M = 120.5 s, SD = 67.2 s) conditions ($t(18)$= 2.13, p = 0.047). See the plot on "Completion Time" in Fig. 7. In a typical USAR mission, being able to use the robot for searching for the maximal amount of available time (as decided by, e.g., time between battery changes) is of high importance, and the VDOA interface has shown to achieve this.

5.2.3 Collisions During the task, collisions may happen between the robot and the walls (usually when turning). This is because there was no active collision avoidance system running, and participants may misinterpret the distances and sizes to obstacles in the video stream.We observed the number of collisions (shown in Fig. 7) with the walls of the maze in the exploration task of each participant. The average number of collisions in VDOA was 0.8 (SD = 0.92) whereas in the BAR group, the mean was 0.4 (SD = 0.7). Although the absolute number of collisions in VDOA was higher than the BAR group, the difference in means was not statistically significantly ($t(18)$ = 1.09, p = 0.29) given the population size.

We believe the reason for this was twofold. First, the VDOA users ran longer missions, as they were able to stay connected longer (see Section 5.2.2 below). Second, they explored more difficult parts of the map, in particular the upper part, where a u-turn was needed after covering the upper right corner (see Fig. 8, and as noted above most collisions occurred when turning).

Perhaps, a better measure is the number of collisions per path length as used in Barros and Lindeman (2013). However, since there were many participants that had no collisions in both groups, it would not be possible to have a fair comparison with the collisions/meter metric. For instance, the means of the collisions per path length in the participants that had at least one collisions is VDOA (M = 0.15, SD = 0.01), and BAR (M = 0.17, SD = 0.01). On the other hand, the sum of distance traveled by all the participants that had zero collisions in VDOA is 37.7 m (4 participants), whereas it is 28.63 m in BAR (7 participants).

5.2.4 Localization of radio source Participants were asked to guess the radio source (a concealed wireless router) location and mark it on the map. We manually calculated the distance of the marked location from the actual location of the router on the map from each participant answer sheets. Following a simple rule to measure the distance, we used the Euclidean measure (shortest distance) if the marked location was within the line of sight (LOS) from the router and the Manhattan measure (shortest ray distance) when the marked location was in a non-line of sight (NLOS) from the router. From Fig. 6, we can clearly observe that the regions A and B are NLOS and all other regions are LOS. We followed this strategy not to exaggerate markings in the NLOS regions but to represent reality based on the RF propagation principles.

The scores of localization error in each participant range from 0 to 6 meters. The VDOA group mean was 1.39 m (SD = 1.02 m) and the BAR group mean was 2.47 m (SD = 1.56 m). The difference in means are statistically significant under conditions $t(18)$ = -1.83 and $p < 0.1$. This means that the DoA information in the UI enabled the VDOA participants to better understand the connectivity situation in real time, which is particularly helpful in increasing search and rescue mission capabilities without losing control over the robot.

5.2.5 Coverage To measure the explored areas, we discretized the maze area in 15×15 cm squares and accumulated the number of visits the robot made in each square. A graphical representation of the coverage map is shown in Fig. 8, where a lighter color indicates unexplored regions. It can be seen from the map that the VDOA users spent more time exploring the regions with higher signal

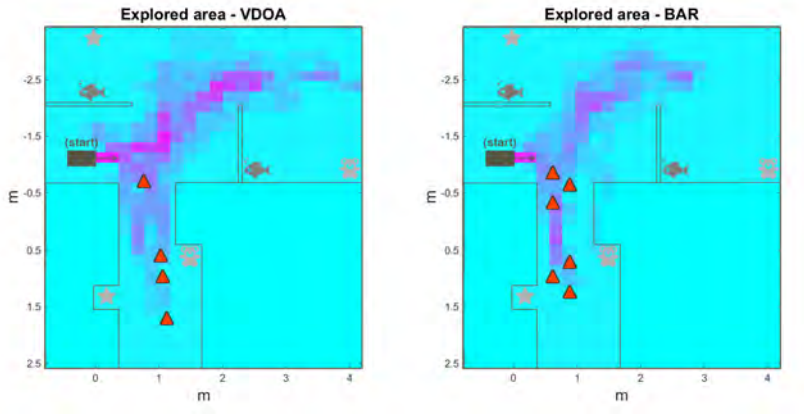


Figure 8. Two color maps showing the covered area in both groups. A lighter color represents the least explored region while a darker color represents the most covered region. Red triangles represent points of connection loss. Note how VDOA users spent more time searching the upper area, where no connection losses occurred, compared to the BAR users.

coverage than the BAR users. Specifically, the BAR group went more often into the low signal regions and also lost the connections much more often than the VDOA group.

We also calculated the total traveled distance by summing all the Euclidean displacements. The scores obtained are 8.78 m (M) and 2.84 m (SD) for VDOA and 5.28 m (M) and 3.33 m (SD) for BAR groups. A significant difference of the coverage area between both groups is noted (conditions: $t(18) = 2.55$, $p < 0.05$). Group using VDOA traveled farther in the area and covered a larger area than did their BAR counterparts.

5.2.6 Situation awareness In this study, we measured SA using a form, including both subjective (self-ratings) and objective (estimating positions in a map) components. In the experiments, the participants were given a fairly short time period (up to 3 minutes) to explore a fairly small experimental area. The reason for this is to provide a well-controlled experiment. All participants faced the same intersection, symbol placements, connectivity variations, user interface quality, and so on, without having decisions regarding search strategies influencing the data. However, the short mission times made it difficult to measure SA using methods such as SAGAT (Endsley, 1988) (which requires questionnaire interventions during a task). Hence, we partly evaluated the SA using self-ratings from the participants, methods that, according to Gatsoulis, Virk, and Dehghani-Sanij (2010), perform equally well compared to objective methods in evaluating SA.

The participants rated their confidence level in SA on a scale of 1 (lowest) to 5 (highest). There was a significant difference in how confident the participants felt that they had found all available symbols in the entire area using VDOA ($M = 4.2$, $SD = 1.32$) and BAR ($M = 2.5$, $SD = 1.78$) conditions ($t(18) = 2.42$, $p = 0.026$). With VDOA, the participants were more confident that they had explored the entire area. Note that the participants were not informed on the number of symbols existing in the environment.

Spatial SA (Symbols mapping) - We assessed the users' spatial awareness by asking the participants to mark the symbols they found during the task. Using the ground truth, we calculated the offset in the reported and actual positions in a discrete grid map of resolution 50 cm. The offset

measure was in grid spacing with 0 meaning the same grid and 9 meaning an offset of 9 grid cells. The offsets of all the found symbols were averaged to arrive at the score of each participant. A lower offset value means a better spatial awareness. Two participants in the BAR group did not find any symbols and therefore they are not considered in this analysis. We found no significant difference between the symbol mapping accuracy (spatial SA) of VDOA (M = 1.63, SD = 0.9) and BAR (M = 1.65, SD = 0.99) groups under conditions: $t(16)$ = -0.04; p = 0.48. In Fig. 7, the presented boxplot of this measure is a normalized[12] version to correspond with the scale of other SA measures.

On a different question, we asked participants if they felt that they had drawn the position of the radio source correctly on the map. The results (VDOA: M = 4.5, SD = 0.97; BAR: M = 3.6, SD = 1.26) reveal a significant difference between the groups ($t(18)$ = 1.78, $p < 0.1$). The VDOA group felt more aware of the network situation than the BAR group.

Finally, we asked participants to draw the path taken by the robot along with the final orientation after they finished the task and asked a question how confident they felt in marking the path. Although the VDOA group (M = 4.5, SD = 0.85) had higher confidence than the BAR group (M = 4, SD = 1.25) in general, there was no significant difference. This may attribute to the fact that both group used the same FLC control and may mean that having an additional indicator for directional wireless connectivity does not inhibit operator awareness of the robots position and orientation.

5.3 Network parameters

5.3.1 Connection loss The connection loss measure is directly related to the execution time, as the exploration task was terminated before the timeout only when the participant lost the connection. Therefore, one might expect that the analysis of the execution time holds also for the connection loss measure. However, a t-test on how many participants lost connection (4 out of 10 for VDOA and 6 out of 10 for BAR) during the study showed no significant difference between the means of VDOA (M = 0.4, SD = 0.52) and BAR (M = 0.6, SD = 0.52). One reason for this might be that when mission time grows to infinity, the chances of losing connectivity at some point tends to one, regardless of what interface is used. Furthermore, after detecting a certain number of symbols, VDOA users tended to adopt a riskier strategy, pushing the robot to explore the edges of the poorly connected area and causing a connection loss (4 out of 10; see Fig. 8).

5.3.2 Overall connection quality We used the RSS from the wireless adapter as a measure of overall connection quality. As we are interested in the improvement in connection quality from the starting position, we calculated the difference in the RSS from the initial RSS values and calculated the RSS gain averaged over the entire duration of the exploration task by each participant. In this, way we mitigated the influences of temporal variations and effects of influences due to network traffic conditions during the day. Positive values indicate that there is an improvement in the RSS values and negative values indicates the opposite.

We found significant difference in RSS gain between VDOA (M = 2.83 dBm, SD = 1.02 dBm) and BAR (M = -0.54 dBm, SD = 3.45 dBm) under conditions $t(18)$ = 2.96 and $p < 0.01$. These values are for the RSS of the central receiver, which is used to transfer data to and from the robot. The results are the same regardless of which receiver we consider, including the mean of all the receiver RSS values.

Fig. 9 shows the boxplots of both absolute RSS values and the RSS gains for both groups. It can be seen that in general, the VDOA group maintained higher RSS than the BAR group. Recall from Sec. 4.5 that the users had three options at the beginning of the task: go straight (high connectivity region - D, F, G, H), turn left (medium connectivity region - C and E), and turn right (poor connec-

[12] We normalized the $SA_{spatial}$ score by first negating the actual score and then normalizing to the range [1,5], where higher score represents better SA.

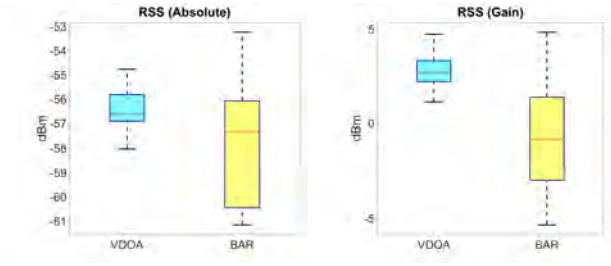


Figure 9. Boxplots of the average RSS values and the average RSS gains in both groups.

tivity region - A and B). Two symbols were placed in each of these regions, as can be seen in Fig. 8. The VDOA users mostly preferred going straight and left than the BAR users, due to the additional 2D connectivity information. However, after exploring those regions, they proceeded to explore the poor connectivity region with caution.

6. Discussion

In this section, we first discuss the results in relation to the claims we made in Section 4. The more upsides and downsides we can confirm, the stronger support we have for the corresponding claim. We then discuss the results in more detail and, finally, make some general remarks.

Regarding Claim 1, *'VDOA leads to UGV trajectories in higher signal strength regions,'* the upsides U11, U12, and U13 were confirmed. As there is no significant difference in the number of collisions and the ease of finding symbols, we cannot confirm the downside D11. These results support Claim 1.

Claim 2, *'VDOA provides better situation awareness,'* is more complex. U21 is partially confirmed, with participants being equally aware of the path traversed, but VDOA users being more confident to have explored the area. U22 is confirmed, with more symbols found by the VDOA users. U23 is partially confirmed, with the same accuracy of mapping found symbols, but better accuracy of radio source localization. D21 is not confirmed. To conclude, the results partially support Claim 2.

In Claim 3, *'VDOA improves user experience,'* all the upsides U31, U32, and U33 were confirmed. This strongly supports Claim 3.

Regarding Claim 4, *'BAR increases focus and concentration on the actual task,'* we are unable to confirm the upsides U41 and U42 because we did not find significant difference in the measures. Also, the upside U43 was refuted because the BAR was neither a preferred system nor rated higher in usability. On the other hand, we can confirm the downsides D41 and D42. Consequently, we only have a weak support for Claim 4.

Looking at the support for all claims, and in particular the fact that all upsides and no downsides of VDOA (Claims 1–3) were confirmed, and all downsides but no upsides of BAR (Claim 4) were confirmed, we can conclude that VDOA is preferable to BAR in wireless teleoperation of UGVs.

Regarding the general results, as described in Section 5.1, we note that VDOA was considered easier to use, similar in terms of mental effort required, and preferred by a majority of the operators. We believe that these advantages are due to the fact that the DoA information is added in the periphery of the video feed in a way that can be accurately and easily processed.

Regarding the exploration results, as described in Section 5.2, we note that VDOA resulted in more symbols found, a longer travelled distance and time, improved accuracy in locating the

radio source, similar accuracy in marking found symbols on a map, and a slight increase in number of collisions. We believe that these advantages are due to the fact that DoA information is very important when making decisions close to the connectivity threshold. Manually estimating the DoA using the information provided in the BAR interface is possible, but probably associated with a significant cost in terms of mental load and mission time, and impossible to do with an accuracy similar to the one observed in Section 3.4 (< 12 degrees). The reason for users of the BAR interface losing connection with the UGV was probably that they were not able to manually estimate the DoA accurately enough. Without a reliable estimate, a natural reaction when running into a low RSS area is to move back to the area just visited, but that strategy has a negative impact on the exploration objective.

Regarding the network results reported in Section 5.1, VDOA resulted in a higher overall connection quality. As noted above, having access to a DoA estimate enables the operator to choose paths that takes both the connectivity and exploration objectives into account. Thus, the DoA information is not guiding the robot; instead, it is enabling robot operations in low connectivity regions and in the regions close to the connectivity threshold. The operator chooses where to go to perform the search. With the VDOA information, the operator can predict the risk better. If entering a room presents a high risk of losing connectivity, the operator can still enter if the potential information gain is worth it. With BAR, the operator might lose connectivity without understanding that the risk was there, as shown in the user study (see Fig. 8). In USAR missions, where staying connected with the robot is critical for saving lives, this VDOA interface could play a vital role.

Finally, from a scientific point of view, we would like to note that this study provides a slight elaboration of the identifying, measuring and analyzing SA variables relevant to the context. As shown in this application, there can be an interaction of dynamic environmental conditions (e.g., network coverage) and robot capabilities (e.g., tele-operation) that affect task performance. So, SA support should not only focus on the perception, comprehension, and prediction of events and states that directly relate to the primary task (e.g., obstacles when navigating) but also focus on the availability and dependencies of the required resources for the task execution. Furthermore, the peripheral color bar in the display provides a general UI pattern for the corresponding SA-support, hardly interfering with the primary task and easily extendable to other forms of scalar field measurements, such as temperature, gas density, or sound volume.

7. Conclusions

In this paper, we proposed a way of estimating DoA of the radio signal and a way of including this information in a UGV teleoperation interface. We also investigated the quality of the estimates and conducted a user study showing that the new interface resulted in improved performance in an exploration scenario.

In the technical tests, we showed that the DoA estimates had a mean error of less than 12 degrees and were useful for predicting changes in RSS values over a typical mission trajectory.

In the user study, the benefits of the new interface, which incorporates directional wireless connectivity information in the free look control interface, were compared to the standard “signal bar” representation of the wireless connection used in modern UGV user interfaces for teleoperation.

We conducted a between-subjects user evaluation with 24 participants and were able to analyze 20 of them with 10 in each group (VDOA and BAR) and found that the new interface (VDOA) partially improves users’ situation awareness and significantly reduces connection loss with the robot. This is especially useful in robot-aided USAR situations where connection loss has a huge impact on mission performance.

A possible extension of this research is to integrate the proposed interface in an augmented reality display system (Krückel, Nolden, Ferrein, & Scholl, 2015) to represent the wireless connectivity in

a 3D fashion as some participants suggested in their feedback. Additionally, the directional antennas used in this study can also be exploited for communication redundancy, offering advantages such as increased coverage, stable connections, and coverage in elevated regions (Hada & Takizawa, 2011). Finally, we believe that the VDOA interface can be easily adapted to both teleoperated maritime and unmanned aerial vehicles.

Acknowledgments

The authors gratefully acknowledge funding under the European Union's seventh framework program (FP7), under grant agreements FP7-ICT-609763 TRADR.

References

Båberg, F., Caccamo, S., Smets, N., Neerincx, M., & Ögren, P. (2016, Oct). Free look UGV teleoperation control tested in game environment: Enhanced performance and reduced workload. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (pp. 312–319).

Baker, M., Casey, R., Keyes, B., & Yanco, H. A. (2004). Improved interfaces for human-robot interaction in urban search and rescue. In *IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, pp. 2960–2965).

Barros, P. G. de, & Lindeman, R. W. (2013). Performance effects of multi-sensory displays in virtual teleoperation environments. In *Proceedings of the 1st Symposium on Spatial User Interaction* (pp. 41–48).

Barros, P. G. de, & Lindeman, R. W. (2014). Multi-sensory urban search-and-rescue robotics: Improving the operators omni-directional perception. *Frontiers in Robotics and AI*, *1*, 14.

Barros, P. G. de, Lindeman, R. W., & Ward, M. O. (2011, March). Enhancing robot teleoperator situation awareness and performance using vibro-tactile and graphical feedback. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 47–54).

Bezzo, N., Griffin, B., Cruz, P., Donahue, J., Fierro, R., & Wood, J. (2014, February). A cooperative heterogeneous mobile wireless mechatronic system. *IEEE/ASME Transactions on Mechatronics*, *19*(1), 20–31.

Burke, J., Murphy, R., Coovert, M., & Riddle, D. (2004). Moonlight in Miami: An ethnographic study of human-robot interaction in USAR. *Human-Computer Interaction, Special Issue on Human-Robot Interaction*, *19*, 1–2.

Caccamo, S., Parasuraman, R., Båberg, F., & Ögren, P. (2015, Sept). Extending a UGV teleoperation FLC interface with wireless network connectivity information. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4305–4312).

Carlson, J., & Murphy, R. R. (2005). How UGVs physically fail in the field. *IEEE Transactions on Robotics*, *21*(3), 423–437.

Casper, J., & Murphy, R. R. (2003, June). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *33*(3), 367–385.

Dell, R. B., Holleran, S., & Ramakrishnan, R. (2002). Sample size determination. *ILAR Journal*, *43*(4), 207–213.

Endsley, M. R. (1988, May). Situation awareness global assessment technique (SAGAT). In *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference (NAECON),* (p. 789-795 vol.3).

Endsley, M. R. (2000, Jul). Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 3–32.

Fong, T., & Thorpe, C. (2001). Vehicle teleoperation interfaces. *Autonomous robots*, *11*(1), 9–18.

Gatsoulis, Y., Virk, G. S., & Dehghani-Sanij, A. A. (2010). On the measurement of situation awareness

for effective human-robot interaction in teleoperated systems. *Journal of Cognitive Engineering and Decision Making*, *4*(1), 69–98.

Hada, Y., & Takizawa, O. (2011). Development of communication technology for search and rescue robots. *Journal of the National Institute of Information and Communications Technology*, *58*(1/2), 132.

Han, D., Andersen, D., Kaminsky, M., Papagiannaki, K., & Seshan, S. (2009). Access point localization using local signal strength gradient. In S. Moon, R. Teixeira, & S. Uhlig (Eds.), *Passive and active network measurement* (Vol. 5448, pp. 99–108). Berlin Heidelberg: Springer.

Hedström, A., Christensen, H. I., & Lundberg, C. (2006). A wearable GUI for field robots. In P. Corke & S. Sukkariah (Eds.), *Field and service robotics* (pp. 367–376). Berlin Heidelberg: Springer.

Hestand, D., & Yanco, H. (2004, Oct). Layered sensor modalities for improved human-robot interaction. In *IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, pp. 2966–2970).

Krückel, K., Nolden, F., Ferrein, A., & Scholl, I. (2015, May). Intuitive visual teleoperation for ugvs using free-look augmented reality displays. In *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4412–4417).

Larochelle, B., & Kruijff, G. (2012, Sept). Multi-view operator control unit to improve situation awareness in USAR missions. In *Proceedings of 2012 IEEE RO-MAN: IEEE International Symposium on Robot and Human Interactive Communication* (pp. 1103–1108).

Larochelle, B., Kruijff, G.-J., Smets, N., Mioch, T., & Groenewegen, P. (2011, July). Establishing human situation awareness using a multi-modal operator control unit in an urban search & rescue human-robot team. In *Proceedings of 2011 IEEE RO-MAN: IEEE International Symposium on Robot and Human Interactive Communication* (pp. 229–234).

Lindhè, M., Johansson, K. H., & Bicchi, A. (2007, June). An experimental study of exploiting multipath fading for robot communications. In *Proceedings of 3rd International Conference on Robotics Science and Systems, RSS 2007, Atlanta, GA, USA* (pp. 289–296).

Min, B.-C., Matson, E., & Khaday, B. (2013, Oct). Design of a networked robotic system capable of enhancing wireless communication capabilities. In *Proceedings of 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).*

Murphy, R. (2004). Human-robot interaction in rescue robotics. *IEEE Transactions on systems, Man, and Cybernetics, Part C: Application and Reviews*, *34*(2).

Murphy, R. R. (2014). *Disaster robotics*. Cambridge, MA: MIT Press.

Muszynski, S., Stuckler, J., & Behnke, S. (2012, September). Adjustable autonomy for mobile teleoperation of personal service robots. In *Proceedings of 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication* (pp. 933–940).

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., et al. (2013). 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.

Neerincx, M. A., & Lindenberg, J. (2008). *Situated cognitive engineering for complex task environments*. Aldershot: Ashgate Publishing Limited.

Ögren, P., Svenmarck, P., Lif, P., Norberg, M., & Söderbäck, N. E. (2014, June). Design and implementation of a new teleoperation control mode for differential drive ugvs. *Autonomous Robots*, *37*(1), 71–79.

Owen-Hill, A., Parasuraman, R., & Ferre, M. (2013, Oct). Haptic teleoperation of mobile robots for augmentation of operator perception in environments with low-wireless signal. In *Proceedings of 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (pp. 1–7).

Parasuraman, R., Fabry, T., Kershaw, K., & Ferre, M. (2013, Dec). Spatial sampling methods for improved communication for wireless relay robots. In *Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE)* (pp. 874–880).

Parasuraman, R., Fabry, T., Molinari, L., Kershaw, K., Castro, M. D., Masi, A., et al. (2014). A multi-sensor rss spatial sensing-based robust stochastic optimization algorithm for enhanced wireless tethering. *Sensors*, *14*(12), 23970–24003.

Parasuraman, R., Kershaw, K., & Ferre, M. (2013). Experimental investigation of radio signal propagation in scientific facilities for telerobotic applications. *International Journal of Advanced Robotic Systems*, *10*(364).

Rank, M., Shi, Z., Mller, H. J., & Hirche, S. (2016, Aug). Predictive communication quality control in haptic teleoperation with time delay and packet loss. *IEEE Transactions on Human-Machine Systems*, *46*(4), 581–592.

Rappaport, T. (2001). *Wireless communications: Principles and practice* (2nd ed.). Upper Saddle River, NJ: Prentice Hall PTR.

Richer, J., & Drury, J. L. (2006). A video game-based framework for analyzing human-robot interaction, characterizing interface design in real-time interactive multimedia applications. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on human-robot interaction* (pp. 266–273).

Smets, N. J., Brake, G. M. te, Neerincx, M. A., & Lindenberg, J. (2008). Effects of mobile map orientation and tactile feedback on navigation speed and situation awareness. In *Proceedings of the 10th International Conference on Human-computer Interaction With Mobile Devices and Services* (pp. 73–80).

Valenzuela, R., Landron, O., & Jacobs, D. (1997, Feb). Estimating local mean signal strength of indoor multipath propagation. *IEEE Transactions on Vehicular Technology*, *46*(1), 203–212.

Wegner, R., & Anderson, J. (2006). Agent-based support for balancing teleoperation and autonomy in urban search and rescue. *International Journal of Robotics and Automation*, *21*(2), 1–19.

Wickens, C. D. (2008). Multiple resources and mental workload. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(3), 449–455.

Yamauchi, B. M. (2004, September). Packbot: a versatile platform for military robotics. In *Proceedings SPIE 5422, Unmanned Ground Vehicle Technology VI* (Vol. 5422, pp. 228–238).

Yanco, H. A., Baker, M., Casey, R., Keyes, B., Thoren, P., Drury, J. L., et al. (2006, August). Analysis of human-robot interaction for urban search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics, Gaithersburg, MD, USA* (pp. 22–24).

Yanco, H. A., & Drury, J. (2004, Oct). "Where Am I?" Acquiring situation awareness using a remote robot platform. In *Proceedings of IEEE's International Conference on Systems, Man and Cybernetics* (Vol. 3, pp. 2835–2840).

Yanco, H. A., & Drury, J. L. (2007). Rescuing interfaces: A multi-year study of human-robot interaction at the AAAI robot rescue competition. *Autonomous Robots*, *22*(4), 333–352.

Yoshida, T., Nagatani, K., Tadokoro, S., Nishimura, T., & Koyanagi, E. (2014). Improvements to the rescue robot quince toward future indoor surveillance missions in the Fukushima Daiichi nuclear power plant. In *Field and Service Robotics: Results of the 8th International Conference* (pp. 19–32). Berlin Heidelberg: Springer.

Zijlstra, F., & Doorn, L. van. (1985). *The construction of a scale to measure subjective effort* (Tech. Rep.). Delft University of Technology.

Ramviyas Parasuraman (corresponding author), Department of Computer and Information Technology, Purdue University, West Lafayette 47906, USA. Email: ramviyas@purdue.edu; Sergio Caccamo, Fredrik Båberg, and Petter Ögren, Centre for Autonomous Systems, School of Computer Science and Communication, KTH - Royal Institute of Technology, Stockholm 10044, Sweden. Emails, respectively: caccamo@kth.se, fbaberg@kth.se, and petter@kth.se; Mark Neerincx, TNO, Netherlands. Email: mark.neerincx@tno.nl

Active Exploration Using Gaussian Random Fields and Gaussian Process Implicit Surfaces

Sergio Caccamo, Yasemin Bekiroglu, Carl Henrik Ek, Danica Kragic

***Abstract*— In this work we study the problem of exploring surfaces and building compact 3D representations of the environment surrounding a robot through active perception. We propose an online probabilistic framework that merges visual and tactile measurements using Gaussian Random Field and Gaussian Process Implicit Surfaces. The system investigates incomplete point clouds in order to find a small set of regions of interest which are then physically explored with a robotic arm equipped with tactile sensors. We show experimental results obtained using a PrimeSense camera, a Kinova Jaco2 robotic arm and Optoforce sensors on different scenarios. We then demostrate how to use the online framework for object detection and terrain classification.**

***Index Terms*— Active perception, Surface reconstruction, Gaussian process, Implicit surface, Random field, Tactile exploration.**

I. Introduction

Acquiring a high quality 3D model of the environment is critical for many autonomous robotics problems such as grasping, segmentation, traversability or mapping. Mere vision perception does not often exhaustively describe the shape of the environment since volumetric data generated from modern vision sensors are prone to errors due to limited field of view, photometric effects, occlusions and noise.

Passive observation of a scene leads to incomplete shapes of objects and terrains facing the camera. Heuristic or symmetry assumptions [1] can be used to deal with lack of data in the observations, leading to errors in the representation.

Surface exploration through vision and haptic interactions is the task of purposefully touch and inspect a portion of environment so to reveal occluded information. It can be considered as a case of either interactive or active perception depending whether the physical interaction strategically modify the environment under analysis or not respectively.

Haptic exploration helps improving observations adding a new layer of information into the world model. Meier et al. [2] showed that tactile information alone can be used to adequately describe objects properties. A robotic manipulator equipped with tactile sensors can be used to encode properties of surfaces and objects [3] and enhance visual perception of shapes [4]. Studies [5] show that combining tactile and visual representations of object brings more reliable and robust shape estimation than either the visual or tactile alone. Even humans build their world representation using different sensory information and actively examine the environment to enhance their perception [6].

Caccamo, Ek and Kragic are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology (KTH), SE-100 44 Stockholm, SE. e-mail: {caccamo|chek|dani}@kth.se

Bekiroglu is with the School of Mechanical Engineering, University of Birmingham, UK e-mail: {Y.Bekiroglu}@bham.ac.uk

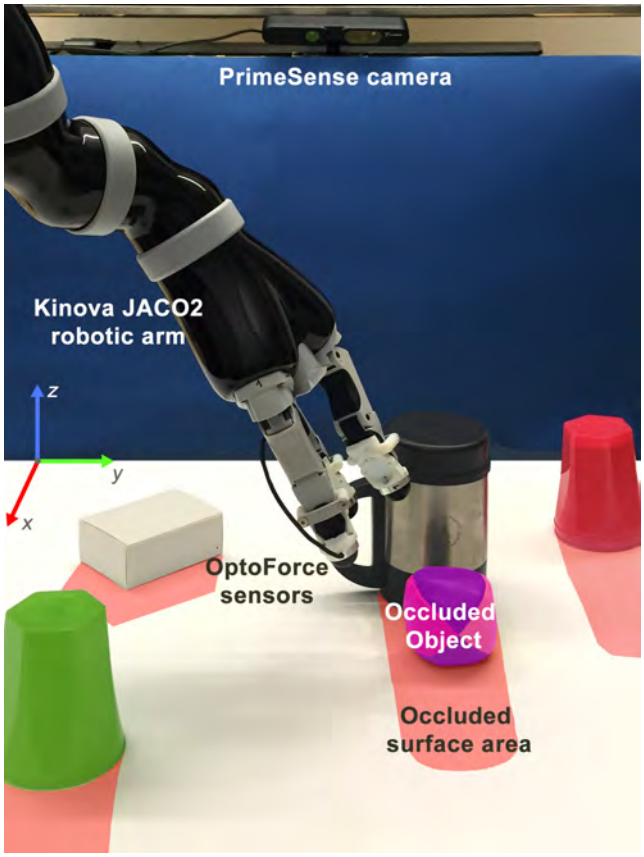


Fig. 1: Experimental setup. A kinova Jaco2 arm equipped with Optoforce sensor explores occluded areas of an environment.

In this paper we use the term VRS (visible and reachable surface) to refer to all the surfaces contained in the portion of space consisting in the intersection of the field of view of the camera and the reachable space of the robot. The VRS can contain several occluded regions and objects (i.e. occluded surfaces, see Fig. 1).
Examples of VRS are:

- The surface of a table and objects placed in front of a robot.
- The portion of map in front of an arm equipped Unmanned Ground Vehicle (UGV) as shown in Fig. 2.

Inspired by the work in [4] and field applications described in [7] we build 3D models of a VRS by merging vision and haptic data into a probabilistic framework. We study how to

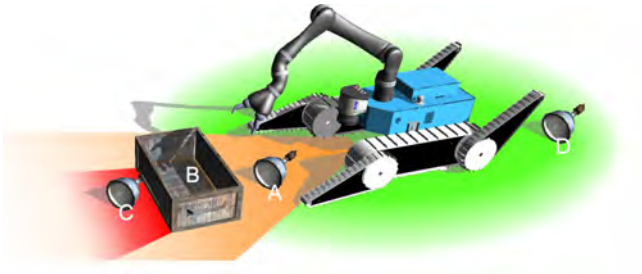


Fig. 2: Visible and reachable surface of an UGV. Objects A and B are inside the VRS, B is partially occluded. C and D are outside the VRS.

properly model shape of environments that contain different occluded objects and incomplete areas taking advantage of uncertainties in the sensory system. We also show how to reduce the exploration time and the amount of physical interactions needed. In summary, the contributions of this paper can be listed as follow:

1) We propose a new probabilistic framework for surface exploration that merges haptic and visual sensory information for building a local 3D map of the environment.
2) We show how to exploit Gaussian Processes and Delaunay triangulation for reducing the amount of interactions and the computational power/time needed.
3) We demonstrate the feasibility of the approach through hardware experiments letting a robotic arm explore different scenarios and show how few interactions can add useful information to a partially visible surface.
4) We show how to generically exploit the autonomous framework for problems such as object identification and terrain classification.

II. Related Work

Gaussian processes (GP) [8] have been used for terrain mapping and modeling [9], [10] in a wide range of applications including geophysics, aeronautics and robotics. In those studies any given point in a 2D Euclidean space is associated with a single elevation value, generating therefore a 2.5D surface called digital elevation model (DEM) also known as heightmap [11]. Another example of Gaussian processes applied to digital terrain modeling can be seen in [12]. Such models may fit terrain shapes but are not suitable for more complex surfaces or in applications such as object manipulation or segmentation.

Implicit surfaces [13] have been widely used to represent object shapes since their first appearances in [14]. Generic reconstruction of implicit surfaces from data points have been presented in [15]. Machine learning techniques have been progressively developed to represent complex surfaces as in [16] and [17]. More recently, Gaussian Process Implicit Surfaces (GPIS) [18] have become very popular allowing to extend implicit surface to uncertainty, a property needed when the model is the result of sensory data fusion [19]. Environmental observations can condition a GP so that its posterior mean define the implicit surface defining the terrain (including objects). Authors in [20] applied GPIS for building 3D representation of the environment by fusing laser and mm-wave radar data. Results in [21] show how GPIS as object representation can even improve complex tasks as grasping.

One disadvantage of these approaches is that during inference, Gaussian Process Regression (GPR) is computationally demanding. The major cost takes place from the inversion of large covariance matrices that, in the simplest implementation, have complexity $O\left(n^3\right)$. Mathematical tools as Cholesky decomposition or sparse kernels [22] can considerably reduce the computational effort.

GPIS requires a dense cubic matrix of points as test set in order to qualitatively describe the implicit surface. When a VRS includes several objects, the matrix containing the training sample points (i.e. the point cloud) becomes large and the computational time increases drastically making the implementation of an online active perception algorithm for surface exploration infeasible. To overcome such problem a down-sampled subset is often heuristically selected and used [23]. An application of sparse kernels for mapping of large area is presented in [24] where authors propose a unified framework for building continuous occupancy grid maps. As already mentioned, merging haptic and visual sensory data into the same probabilistic model using GPs can lead to better shape representation [4] and planning [25]. An example of tactile sensing for object tracking with visual occlusion using particle filters is presented in [26]. Another efficient tactile perception algorithm for object manipulation and tracking is shown in [27].

Work in [28] and [29] showed that the morphology of an environment is very important for an unmanned ground vehicle (UGV) that autonomously try to traverse a particularly harsh environment. They showed also that occlusion and reflections, e.g. caused by a pot of water or broken glass, can lead to failure. This could be solved by asking a robotic arm to strategically explore the environment around the UGV. Nevertheless, the battery capacity, the computational power on board of the robot and limited time constraints, common during urban search and rescue missions, force the exploration task to reduce as much as possible the required elaboration time and the tactile interactions. On a different scenario, an interactive humanoid robot, which explore a table with objects for segmentation, can try to identify hidden elements by touching occluded regions and move its head only in case of positive tactile feedback.

To address those problems we propose a probabilistic method that identifies and analyses occluded regions in the working space area of a robot, where a VRS point cloud can be much larger than a single object (e.g. cup or bottle). We train a Gaussian Random Field (GRF) and a Gaussian Process Implicit Surface on the initial point cloud representing the VRS. We then infer the joint distribution of Gaussian Random Field model on regions of interest obtained from a 2.5D fast Delaunay triangulation. Delaunay triangulation on a discrete Euclidean d-dimensional point set corresponds to the dual graph of the Voronoi diagram [30] for the same

set. We use it to quickly identify and investigate large sparse areas in the visual point cloud that could potentially carry high uncertainty in the internal probabilistic model. We use a robot manipulator with tactile sensors for autonomously touching the isolated regions of the surface. We define a new training set for the GPIS using on-surface and off-surface tactile points obtained during each interaction by tactile sensors placed on the fingertips of the robotic hand. Finally we generate the new 3D shape inferring the GPIS on a subset of the VRS selected using the predicted mean of the GRF.

A surface exploration step in this paper denotes a single iteration of the algorithm that includes several physical interactions with the environment. During each exploration step the GPIS model is updated many times enlarging its training set with tactile information. We assume the environment to remain static during the whole analysis.

III. Surface Modeling

In this section we briefly discuss Gaussian Processes Regression (GPR) [8]. We describe how to exploit two dimensional GPR (Gaussian Random Field) and three dimensional GPR (Gaussian Process Implicit Surfaces [18]) for modeling terrain and object shapes.

A. Gaussian Random Fields

We denote $P_{VRS} = \{\mathbf{p_1}, \mathbf{p_2} \dots \mathbf{p_N}\}$ with $\mathbf{p_i} \in \mathbb{R}^3$ the set of measurements of 3D points lying on the visible reachable surface and $D_{RF} = \{\mathbf{x_i}, y_i\}_{i=1}^{N}$ a bi-dimensional training set where $\mathbf{x_i} \in \mathbf{X} \subset \mathbb{R}^2$ are the *xy*-coordinates of the points in P_{VRS} and y_i the *z*-coordinates (heights)[1]. We also define a set $\mathbf{X}_* \equiv \mathbf{X_{rf}}_* \subset \mathbb{R}^2$ of M test points. With a function $f : \mathbb{R}^2 \to \mathbb{R}$ we map a 2.5D surface where each vector of *xy*-coordinates generates a single height. Such a function can efficiently be modeled by a GPR which places a multivariate Gaussian distribution over the space of $f(\mathbf{x})$. The GP can be fully described by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x_i}, \mathbf{x_j})$. Assuming noisy observation $y = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ and $m(\mathbf{x}) = 0$ the joint Gaussian distribution on the test set $\mathbf{X}_*$ becomes

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^{\mathbf{T}} & \mathbf{k}_{**} \end{bmatrix}\right) \tag{1}$$

where $\mathbf{K}$ is the covariance matrix between the training points $[\mathbf{K}]_{i,j=1\dots N} = k(\mathbf{x_i}, \mathbf{x_j})$, $\mathbf{k}_*$ the covariance matrix between training and test points $[\mathbf{k}_*]_{i=1\dots N, j=1\dots M} = k(\mathbf{x_i}, \mathbf{x_{*j}})$ and $\mathbf{k}_{**}$ the covariance matrix between the test points $[\mathbf{k}_{**}]_{i,j=1\dots M} = k(\mathbf{x_{*i}}, \mathbf{x_{*j}})$.

The predictive function is obtained by conditioning on the training points

$$p(f_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\overline{f_*}, \mathbb{V}[f_*]\right) \tag{2}$$

$$\overline{f_*} = \mathbf{k}_*^{\mathbf{T}} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} \tag{3}$$

[1]Axis are described considering the frame represented in Fig.1

$$V[f_*] = \mathbf{k}_{**} - \mathbf{k}_*^{\mathbf{T}} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_* \tag{4}$$

For this study we choose to use the popular squared exponential kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = \sigma_e^2 \exp\left(-\frac{(\mathbf{x_i} - \mathbf{x_j})^T (\mathbf{x_i} - \mathbf{x_j})}{\sigma_w^2}\right) \tag{5}$$

Gaussian random field (GRF) is a common way to refer to Gaussian Process Regressors that generalize over bi-dimensional Euclidean vectors. Associating every coordinate to a single height is a big limitation when it comes to represent complex surfaces, e.g. mugs, inclined boxes. On the other hand, inferring a random field will directly produce 3D points by combining input-output into vectors of coordinates. This explicit behavior of the joint distribution permits to quickly obtain a DEM querying large portion of the VRS using only few bi dimensional testing points. The variance of the random field allows to directly highlight regions of low density data, e.g. occluded portion of the VRS, or high complexity portion of surface, e.g. different heights for the same coordinate.

B. Gaussian Processes Implicit Surfaces

Gaussian Process Implicit Surface (GPIS) models a function $f : \mathbb{R}^3 \to \mathbb{R}$ with supporting points defining an implicit surface. Whereas equations 1, 2, 3, 4 maintain the same form, $D_{IS} = \{\mathbf{x_i}, y_i\}_{i=1}^{N}$ becomes the new training set where $\mathbf{x_i} \in \mathbf{X} \subseteq P_{VRS}$ and $y_i \in \mathbb{R}$ defined as in [18]

$$y_i \begin{cases} = -1, & \text{if } \mathbf{x_i} \text{ is below surface} \\ = 0, & \text{if } \mathbf{x_i} \text{ is on the surface} \\ = 1, & \text{if } \mathbf{x_i} \text{ is above the surface} \end{cases} \tag{6}$$

We also redefine the set $\mathbf{X}_* \equiv \mathbf{X_{is}}_* \subset \mathbb{R}^3$ of M test points. The implicit nature of the GPIS does not allow to directly shape the VRS. It is needed to define a large set of test points, e.g. a dense cubic volume centered on a region of interest, and then find the isosurface of value 0 on the $\overline{f_*}$ associated with the inferred points in $\mathbf{X}_*$. This operation is very computationally expensive depending on the size of the VRS and $\mathbf{X}_*$. On the other hand GPIS allows to model complex surfaces and to use not only points belonging on the surface to shape the GPR but also empty region points (i.e. $f(x) \neq 0$).

As we show on Sec. V, this property is critical in defining the amount of interactions needed to describe the occluded VRS. For the GPIS we choose the same rbf covariance function as in Eq. 5.

Hyper-parameters were empirically chosen based on a set of experiments made on a $1m^3$ area. Having a covariance function that maps the uncertainty on input data similarly for the bi-dimensional case and the three-dimensional case is a fundamental assumption for our analysis as we show later in section V-B.1.

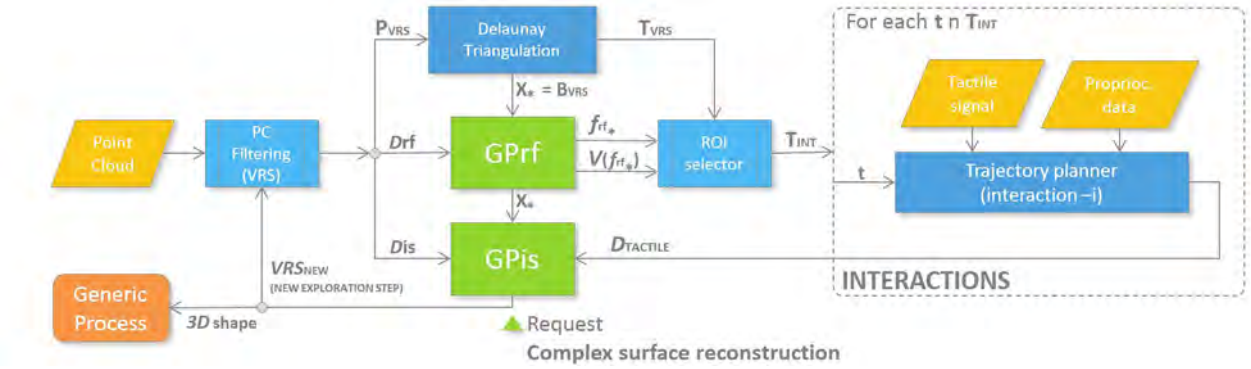


Fig. 3: Probabilistic framework processes flow

IV. Methodology

In our study we represent the environment assuming a limited number of interactions in a constrained time window considering a VRS containing multiple objects. This applies to all those robotics problems in which a robot needs to explore an environment in an online process that extends within limited amount of time (as those listed in Sec. II). In this section we describe the system and guide the reader through the framework process flow showed in Fig.3.

A. Strategy for modeling and inference

We initially model the VRS surface with a 2.5D function $f_{VRS}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ using a bi-dimensional Gaussian Process Regression defined in Sec. III-A. Generalization provided by equations 3 and 4 are generally used to obtain a DEM of the VRS and the level of confidence in the data for each point. The choice of $\mathbf{X}_*$ is therefore crucial for understanding properties of the environment. A naive approach could create a dense grid of bi-dimensional points on the whole surface (so that every single hole in the map is somehow inferred by the GPR). This creates a large set $\mathbf{X}_*$ that leads to a large covariance matrix even for the bi dimensional case. Instead of creating the grid we query only small empty regions analyzing the Voronoi diagram [30] of P_{VRS} defined in III-A. We run a fast 2.5D Delaunay triangulation[2] on the set P_{VRS} and fill $\mathbf{X}_*$ with the xy-coordinates of the barycenters ($\mathbf{B_{VRS}}$) of all the computed Delaunay triangles ($\mathbf{T_{VRS}}$). Elements in $\mathbf{X}_*$ represent coordinates of empty spaces and could easily be reduced in number, if needed, by putting a constraint on the area of the triangles to be analyzed (larger areas mean larger empty regions). We then use Eq. 4 to get the variance on the test points, i.e. the confidence seen as complexity of the shape or lack of information. Computing $V\left[f_{rf_*}\right]$ we can select the points in $\mathbf{B_{VRS}}$ carrying highest uncertainty (ROI selector box in Fig.3) and therefore also the vertices of the corresponding triangles that we denote as $\mathbf{T_{INT}} = [\mathbf{b_1}, \mathbf{tr_1}], [\mathbf{b_2}, \mathbf{tr_2}] \ldots [\mathbf{b_L}, \mathbf{tr_L}]$ with $L << N$. This step is crucial for the algorithm efficiency, allowing to obtain several regions of interest in a fast way without actually inferring a Gaussian Process Implicit Surface on the whole cube containing the VRS.

Each $\mathbf{b_i}$ is considered as a target position point for the trajectory planner for the arm . We define the approach vector for each target point (i.e. each interaction) by computing the normal vector to the plane defined by the triangle vertices constrained with direction going inside the surface (each $\mathbf{t} \in \mathbf{T_{INT}}$) .

We define an implicit surface by the support points of a function $\Psi_{VRS}(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ using a Gaussian Process implicit surface Regression defined in Sec. III-B. We train the model using all the 3D points in P_{VRS} labeled as 0 with the addition of a smaller[3] set of exterior points labeled according to Eq. 6. Artificial points above and below surface are created by increasing and decreasing the z-coordinates of copies of uniformly randomly selected points in P_{VRS} respectively.

B. Surface exploration

The tactile exploration task starts by sending trajectories to the robotic manipulator equipped with tactile sensors. From each sensor we obtain a temporal signal as a sequence of 3D points (sensor positions w.r.t. the world frame) along with their contact forces expressed as 3D vectors. We define a new training set $D_{tactile} = \{\mathbf{x_i}, y_i\}_{i=1}^{N}$ containing sampled 3D sensor positions labeled as 0 if there is contact (estimated from the module of the contact force vector) or 1 (i.e. above the surface) if there is no contact. In case of contact we further add below-the-surface samples as virtually generated 3D points placed a few millimeters from the contact positions along the direction of the contact force (see Fig.6). We train again the GPIS model alone adding $D_{tactile}$ to the initial training set. As we show in the experiment section on-surface points (contact points) and off-surface points are equally important when defining a surface shape. When the arm approaches the surface we start adding above-surface points to the GPIS model that will "push down" the uncertainty and redefine the implicit surface with a "clay-like" behavior. Inside an exploration step the framework updates the internal GPIS many times and queries only the GRF. This is done

[2]FADE25D C++ Library

[3]We used 1/5N points above and 1/5N points below the surface.

so that an external concurrent process can require the last world representation available at any time and interrupt the exploration step if needed (e.g. an external process queries a small portion of space using the GPIS and identifies that the analyzed occlusion contains an object so no further interactions are required). When all the trajectories have been used the exploration step is completed. The inferred mean of the Gaussian random field $\overline{f_{VRS_*}}$ along with its variance can help reducing the dimension of the test set of the GPIS in case a full 3D model of the VRS is needed. Alg. 1 shows one way to do it. It creates a test set as a grid of 2D points on the space covered by the VRS and then computes the mean function along with the variance from the GRF. It then generates 3D points for the GPIS test set using the means as xy-coordinates and the variances as confidence intervals where to span the z-coordinates. Such simple approach can help generating a test set with dimensionality considerably smaller than a dense cube of 3D points. The new VRS obtained inferring the GPIS on the new test set can be used to close the modeling loop and trigger a new iteration of the algorithm for better shape accuracy.

Algorithm 1 Generate a test set for a GPIS from an inferred GRF

procedure GENERATESUBSET
  $X_* \leftarrow 2Dgrid(size)$
  $f_{rf_*} \leftarrow gpmean(X_*)$ ▷ from the GRF
  $V_{rf_*} \leftarrow gpvar(X_*)$ ▷ return the diagonal
  for all $\mathbf{x}$ in X_* **do**
    $y \leftarrow f_{rf}(\mathbf{x}) - \left(m * V_{rf_*}(\mathbf{x}) + \tau_v\right)$ ▷ lower height, m and τ_v constants
    while $y < f_{rf}(x) + \left(m * V_{rf_*}(\mathbf{x}) + \tau_v\right)$ **do**
      $X_{is_*} \leftarrow addNewPointToSet(\mathbf{x}, y)$
      $y \leftarrow y + \Delta_y$ ▷ Δ_y incremental constant
    end while
  end for
  return X_{is_*}
end procedure

V. EXPERIMENTAL EVALUATION

In the following we describe the experimental setup showed in Fig. 1 and the experimental scenarios.

A. Experimental setup

The point cloud is obtained from a PrimeSense 3D camera placed 60 cm above a table oriented to form a 35° angle with the table plane. The table surface can be configured to contain holes, reflective surfaces or soft surfaces in order to recreate different scenarios. The tactile sensory system is composed of a Kinova Jaco2[4] robotic arm (6 dof) with a 3 fingered Kinova KG-3 gripper equipped with 3D OptoForce force sensors[5]. The tactile sensors can detect slipping and shear forces with high frequency. We use the sensor output to obtain 6D force-position signals which is used for generating the tactile training dataset. Proprioceptive data are less affected by noise with respect to vision sensor data. We generate contact and non-contact points after each physical interaction using the sensor orientation w.r.t. the world frame and the output contact force. We generate above-the-surface 3D points as square grids of 16 points placed along the downsampled fingertip trajectories (i.e. sensor position) oriented accordingly. The size of each grid is 8mm×8mm as the spherical sensor dimension and consistent with the VRS point cloud density. We use the contact force direction to orient the on-surface points grid to be orthogonal to the surface normal at the contact position. Below-the-surface points are virtually generated only in case of contact, translating a copy of the grid of on-surface points along the surface normal (see Fig. 6).

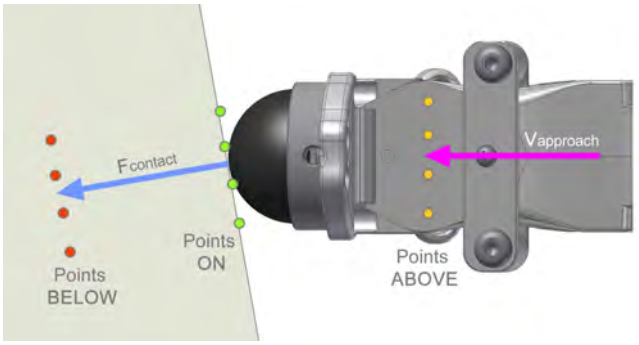


Fig. 6: Top view of a finger with positions of on and off surface points after contact.

The arm follows the approach vector on each target triangle until contact, until the arm reaches its work space limit (i.e. VRS border) or until it diverges too much from the target (e.g. terrain holes).

Hybrid position-force control [31] is used in proximity of the target point to allow small displacement along the plane orthogonal to the approach vector while imposing a minimum contact force along the approach.

B. Scenarios

1) Reflective surfaces: Reflective surfaces as metal plates or water pots generate ambiguity in the point cloud representation of the environment due to photometric effects as shown in the initial VRS in Fig. 4. In the first scenario we use the presented active perception algorithm to identify and model the difference between the two incomplete point cloud regions in an online, fully autonomous fashion. We repeated the experiment using 3 different reflective shapes and holes (reflective shapes RS 1,2 and 3 in Fig. 4) on a single exploration step. The first row of Fig. 4 shows the selected regions of interest on the point cloud with the corresponding Delaunay triangles laying on the areas of high uncertainty (estimated from the GRF imposing a minimum triangle area) and approach vectors. Each triangle selection triggers a physical interaction (visible in the second row of

[4]Kinova website: http://www.kinovarobotics.com/
[5]Optoforce website: http://optoforce.com/

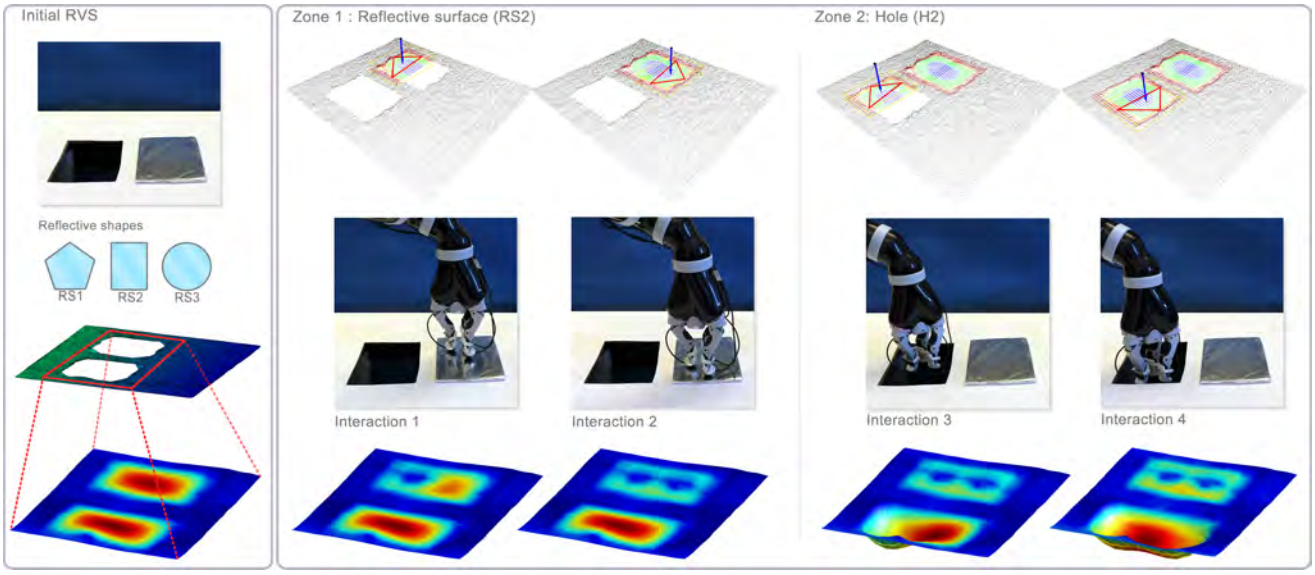


Fig. 4: Exp. setup 1: Different reflective objects are placed next to holes of similar shape which generates ambiguity on the VRS point cloud. The algorithm identifies regions of high uncertainty and starts poking the surface on different locations. After 4 interaction, the 3D model allows to clearly identify the two elements. Blue color in the first row indicates high variance in the GRF model queried in proximity of the select Delaunay triangles. Red color on the third row indicates high uncertainty in the GPIS model.

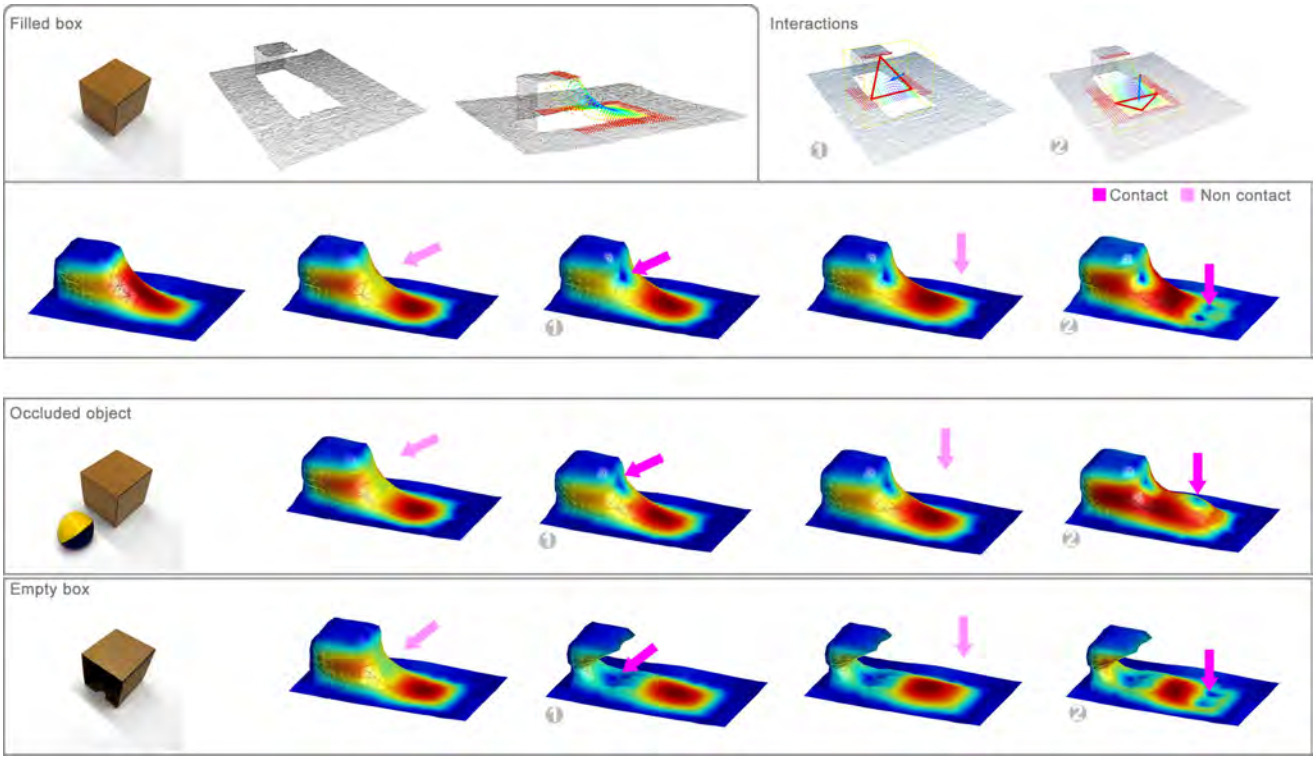


Fig. 5: Exp. setup 2: An object generates an occluded area on the VRS (first row). The algorithm analyses three different situations where the occluded area is a flat terrain (second row), hides a different object (third row) or has complex shapes (fourth row). After 4 interaction, the 3D model allows to describe the occluded area. Colormaps for points (first row) and shapes are chosen as described in Fig.4. In the last scenario the model identifies the empty area inside the box but creates an artifact (two missing faces) due to a lack in lateral interactions.

each column). After each action we train the GPIS with the new tactile training previously mentioned. The last row shows the isosurface of value 0 representing the implicit surface modeled by the GPIS after each interaction. We invite the reader to note how the last two interactions allow to shape the hole but do not reduce the representation uncertainty in that region. This is because we fed the model with above-the-surface points (i.e. non contact points since the arm could not reach the bottom of the hole) that only helped identify areas where the implicit surface could not be.

After the exploration step we used a simple threshold-based binary terrain classifier to automatically label the holes in the analyzed areas. Example of more advanced terrain classification can be seen in [32] and [33]. Results using 3 different reflective surfaces and hole shapes are shown on Table I on a 0.7m×0.6m×0.6m VRS.

Scen.	Shape in the first region	Shape in the second region	n° interact.	T_h Classifier [hole/flat/object]
1	RS1	H1	8	(?,?)→(flat,hole)
2	RS2	H2	4	(?,?)→(flat,hole)
3	RS3	H3	6	(?,?)→(flat,hole)
4	RS1	RS2	6	(?,?)→(flat,flat)
5	H1	H2	7	(?,?)→(hole,hole)

TABLE I: Different combinations of reflective surface (RS) and hole (H) shapes placed in two different regions. A simple threshold classifier T_h labels the presence of holes using two average heights centered on the query regions as shown in Fig.4.

The exploration steps, including elaboration time[6], planning and physical interactions lasted between 3-5 minutes for the above mentioned scenarios. The dimension of P_{VRS} varied between N = 5000 and N = 21000 points depending on the dimension of the occlusions and manipulability constraint. $\mathbf{B_{VRS}}$ contained between 12 and 21 baricenter points. Dimension of $\mathbf{T_{INT}}$ (L, resulting after reduction based on GRF variance) was between 4 and 8 (n° interact.).

2) Occluded areas: In the second experiment (Fig. 5) we demostrate how the algorithm can reconstruct occluded areas and how it can extract environmental properties which are not visible in a simple DEM. Similarly to the previous scenarios we analyse (now independently) elements placed inside the VRS that have the same point cloud representation as shown in Fig. 5. The first element is a full cubic box that generates a large occluded area on the initial VRS point cloud. The second element is an empty cubic box that hide its open face from the camera. The empty area inside the cube cannot be represented by a DEM (GRF) that would only consider the height of the upper side on the box. The third object is the same full cubic box that hides a third different object (a soft ball) placed a few centimeter behind it. By limiting the area of the Delaunay triangles we force the algorithm to only have 2 interactions on the exploration step for each scenario. The first row of Fig. 5 shows the the point cloud representation of the full box on the VRS together with its uncertainty distribution generated by the GRF and the process of action selection with the selected Delaunay triangles. Second row shows the evolution of the internal representation (computed offline) during each interactions. It is possible to notice how the implicit surface changes while the arm approach the target. Second row shows the evolution of the GPIS model for the second scenario. The occluded object becomes visible only during the second interactions after a physical contact. Last row shows that the GPIS can generalize information more complex than the ones embedded on a DEM. The empty cube shaped is revealed by the first interaction that do not add any on-surface point. The box is in fact carved by the off-surface points. The two triangles carrying higher uncertainty selected during the exploration step do not generate any lateral interaction with the box. This results in an artifact in the internal representation (last row of Fig.5) where the two lateral faces of the box disappear. Such situation can be avoided increasing the number of interactions for each exploration step. Similarly to the previous experiment we use a simple threshold classifier on the occluded area to identify the presence of objects as shown in Table II.

Scen.	Scenario description	n° interact.	T_h Classifier [object/flat/hole]
1	Full box	4	(obj,?)→(obj,flat)
2	Box with occluded object	4	(obj,?)→(obj,obj)
3	Empty box	4	(obj,?)→(obj,flat)

TABLE II: Detection of occluded objects. A simple threshold classifier T_h labels the presence of objects using two average heights centered on the box and on the occluded area behind the box respectively.

VI. Conclusions

We presented an efficient probabilistic framework for building a 3D model of a surface containing different occluded areas, objects and reflective surfaces [7]. The algorithm uses Delaunay triangulation and Gaussian Random Fields to quickly identify areas poorly described by the visual sensory system avoiding the computational cost of Gaussian Implicit Surfaces. The system generates subsequent target positions and orientations for a trajectory planner that brings a robotic arm equipped with tactile force sensor to touch the uncertain regions of the local map. On-surface and off-surface points generated during each physical interaction of the arm are used to update a Gaussian process implicit surface that keeps an internal complex representation of the environment. We did real experiments to show how very few interactions can unveil fundamental information hidden in the environment. We also showed how off-surface points alone (that are generated in case of non contact trajectories) can help to model simple terrain shapes. The algorithm can be used in an online process as opposed to other methods [4] and can be iterated to increase the quality of the 3D model. A limitation of the framework appears when the terrain complexity

[6]Using PCL, Eigen, ROS, Kinova SDK

[7]Video of an experiment available at: https://youtu.be/0-UlFRQT0JI

increases or when the covariance functions used for the GPIS and the GRF differ considerably. In such cases tactile interactions (that are planned using the GRF model) cannot bring enough information to the GPIS, resulting in wrong surface representations. A second weakness arises when the arm modifies too much the surface under analysis during the physical interactions and the internal representation of the environment diverges from the real world. In future work we plan to study these problems by segmenting objects in the environment and incorporating relative translations into the model. Variance values inside the triangles can help to generate sliding-on-surface acquisitions to obtain more tactile data from each interaction and embed additional surface properties. We also plan to test the algorithm on the robot shown in Fig. 2 and on a PR2.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding under the European Union's seventh framework program (FP7), under grant agreements FP7-ICT-609763 TRADR.

REFERENCES

[1] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. G. Martínez, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 686–693.

[2] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 630–635, June 2011.

[3] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," *CoRR*, vol. abs/1511.06065, 2015. [Online]. Available: http://arxiv.org/abs/1511.06065

[4] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," pp. 3180–3186, Nov 2013.

[5] H. B. Helbig and M. O. Ernst, "Optimal integration of shape information from vision and touch," *Experimental Brain Research*, vol. 179, no. 4, pp. 595–606, 2007.

[6] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[7] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI - Knstliche Intelligenz*, vol. 29, no. 2, pp. 193–201, 2015. [Online]. Available: http://dx.doi.org/10.1007/s13218-015-0352-5

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006. [Online]. Available: http://www.gaussianprocess.org/gpml/chapters/

[9] S. Vasudevan, "Data fusion with gaussian processes," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1528 – 1544, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889012001388

[10] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1054–1060.

[11] R. J. Peckham and G. Jordan, *Digital Terrain Modelling: Development and Applications in a Policy Support Environment*. Springer Science & Business Media, 2007.

[12] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair, "Gaussian process modeling of large scale terrain," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1047–1053.

[13] G. Turk and J. F. O'brien, "Variational implicit surfaces," 1999.

[14] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Trans. Graph.*, vol. 1, no. 3, pp. 235–256, July 1982. [Online]. Available: http://doi.acm.org/10.1145/357306.357310

[15] J. Solem and A. Heyden, "Reconstructing open surfaces from unorganized data points," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–653–II–660 Vol.2.

[16] F. Steinke, B. Schlkopf, and V. Blanz, "Support vector machines for 3d shape processing," *Computer Graphics Forum*, vol. 24, no. 3, pp. 285–294, 2005. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2005.00853.x

[17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463–470, July 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882293

[18] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," *Gaussian Proc. in Practice*, 2007.

[19] M. A. El-Beltagy and W. A. Wright, "Gaussian processes for model fusion," in *Artificial Neural NetworksICANN 2001*. Springer, 2001, pp. 376–383.

[20] M. Gerardo-Castro, T. Peynot, and F. Ramos, "Laser-radar data fusion with gaussian process implicit surfaces," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 289–302.

[21] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," pp. 2845–2850, May 2011.

[22] A. Melkumyan and F. Ramos, "A sparse covariance function for exact gaussian process inference in large datasets," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 1936–1942. [Online]. Available: http://dl.acm.org/citation.cfm?id=1661445.1661755

[23] N. D. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 609–616.

[24] K. Soohwan and K. Jonghyuk, "Gpmap: A unified framework for robotic mapping based on sparse gaussian processes," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 319–332.

[25] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Int. J. Rob. Res.*, vol. 32, no. 1, pp. 3–18, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1177/0278364912467485

[26] L. Zhang and J. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 3805–3812.

[27] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Touch based perception for object manipulation."

[28] K. Zimmermann, P. Zuzánek, M. Reinstein, T. Petrıcek, and V. Hlavác, "Adaptive traversability of partially occluded obstacles," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.

[29] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5177–5182.

[30] S. Fortune, "Handbook of discrete and computational geometry," J. E. Goodman and J. O'Rourke, Eds. Boca Raton, FL, USA: CRC Press, Inc., 1997, ch. Voronoi Diagrams and Delaunay Triangulations, pp. 377–388. [Online]. Available: http://dl.acm.org/citation.cfm?id=285869.285891

[31] W. D. Fisher and M. S. Mujtaba, "Hybrid position/force control: a correct formulation," *The International journal of robotics research*, vol. 11, no. 4, pp. 299–311, 1992.

[32] A. Krebs, C. Pradalier, and R. Siegwart, "Comparison of boosting based terrain classification using proprioceptive and exteroceptive data," in *Experimental Robotics*. Springer, 2009, pp. 93–102.

[33] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of field robotics*, vol. 23, no. 10, pp. 839–862, 2006.

Active Perception and Modeling of Deformable Surfaces using Gaussian Processes and Position-based Dynamics

Sergio Caccamo, Püren Güler, Hedvig Kjellström, Danica Kragic

***Abstract*— Exploring and modeling heterogeneous elastic surfaces requires multiple interactions with the environment and a complex selection of physical material parameters. The most common approaches model deformable properties from sets of offline observations using computationally expensive force-based simulators. In this work we present an online probabilistic framework for autonomous estimation of a deformability distribution map of heterogeneous elastic surfaces from few physical interactions. The method takes advantage of Gaussian Processes for constructing a model of the environment geometry surrounding a robot. A fast Position-based Dynamics simulator uses focused environmental observations in order to model the elastic behavior of portions of the environment. Gaussian Process Regression maps the local deformability on the whole environment in order to generate a deformability distribution map. We show experimental results using a PrimeSense camera, a Kinova Jaco2 robotic arm and an Optoforce sensor on different deformable surfaces.**

***Index Terms*— Active perception, Deformability modeling, Position-based dynamics, Gaussian process, Tactile exploration.**

I. Introduction

The knowledge of deformability properties of an object or part of an environment can improve robot navigation [1] or object manipulation [2]. A robot can, for example, avoid unstable terrains while driving, place non-rigid objects on stable positions after manipulation or apply proper forces during grasping. Visual sensors alone are not enough to extract the level of deformability. Active perception through integration with haptic exploration helps in estimating deformable properties by purposely interacting with and observing the environment. Most of the existing methods focus on estimating the deformability of single objects using computationally expensive force based simulators [1] and assume that the deformability is homogeneous. Some works consider heterogeneous deformability properties, i.e. deformability is different along the object, using a large number of interactions in a complex multi-camera setup [3].

We present an active perception framework for extraction of heterogeneous deformability properties of the environment, see Fig. 1. The system combines visual and haptic measurements with active exploration and builds deformability distribution maps. A fast Position-based dynamics (PBD) simulator is used to estimate the deformability of a portion of surface after a physical interaction.

The authors are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden. e-mail: {caccamo|puren|hedvig|dani}@kth.se

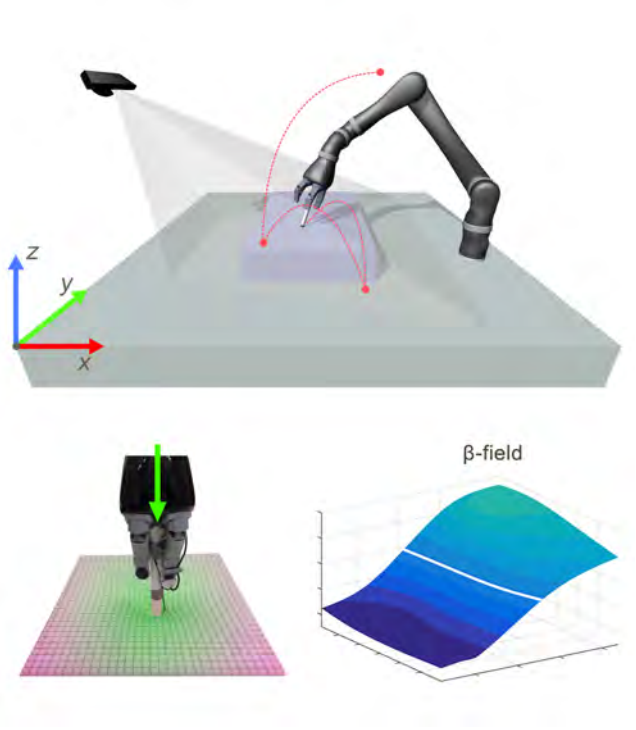


Fig. 1: Illustrative representation of the experimental setup. A Kinova Jaco2 arm equipped with Optoforce sensor interacts with a deformable surface observed by a PrimeSense camera. A Gaussian Process models the deformability distribution (β-field) of the surface from observation and maps it onto the geometric map.

We demonstrate the feasibility of our approach through a serie of experiments performed on scenarios representing terrains containing surfaces having different deformabilities.

A. System outline

The developed framework follows the process outlined in Fig. 2. Observations of the environment (initial and final Point Cloud - PC), extracted before and after a physical interactions are used to estimate the local deformability parameters (β) using a Position-based simulator. The probabilistic model (GPR β-field) gradually generalizes over the local deformability parameters to build a deformability map of the whole environment. Touch strategy and number of physical interactions are assessed using the joint distributions of the Gaussian Process models.

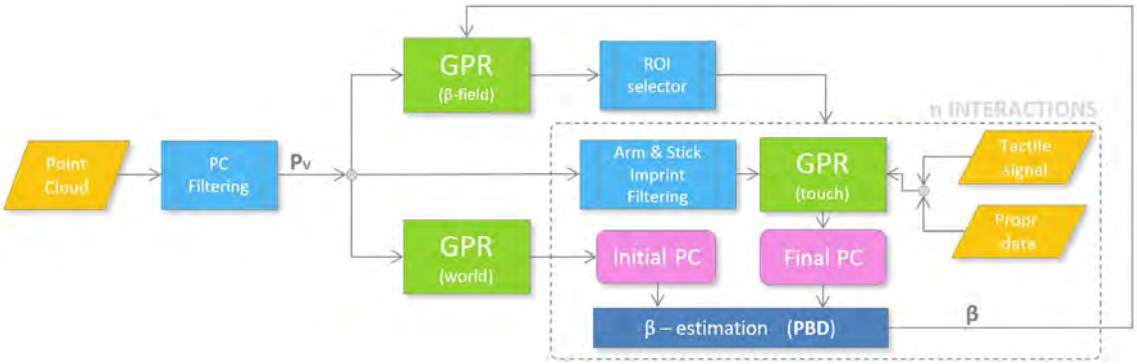


Fig. 2: The developed framework proces flow. After a pre-filtering stage, a Gaussian Process Regression (GPR world) is trained and it describes geometry of the environment. A second GPR (GPR β-field) is used to map deformability parameters (β values) onto the world model and determine whether and where to focus the next physical interaction. In each interaction, a new β value is locally estimated using a PBD simulator and the GPR β-field is updated. GPR-touch is used to obtain compact 3D representations of the environment when it is subject to deformation.

II. Related Work

Gaussian Process Regression (GPR) [4] have been widely used for modeling geometric surface properties [5] on a broad range of applications such as robotics [6], aeronautics or geophysics [7]. In robotics, merging visual and haptic sensor data into the same Gaussian Process probabilistic model leads to a better environmental shape representation [8] or improve planning [6]. Environmental observation can condition a GPR so that its posterior mean define the terrain property [9] of interest. Authors in [10], [8] show how to exploit the mean and variance of the joint distribution of a Gaussian Process for enhancing active perception algorithms in modeling geometric properties of objects.

Unlike the previous works, we use Gaussian Process Regression for mapping and modeling the deformability distribution of a surface in an active perception framework.

The problem of modeling the deformation of non-rigid objects have been widely studied in computer graphics [11] and computer vision communities [12]. The most commonly used approaches for modeling deformations are mesh-based models such as finite element method (FEM) and mass-spring model. FEM aims to approximate the true physic behavior of deformable objects by dividing them into smaller and simpler parts called finite elements. This numerical technique is computationally expensive and has high complexity. Mass-spring is computationally more efficient than FEM but difficult to tune in order to get the desirable behavior. In recent years, position-based dynamics (PBD) [13] have gained attention in the computer graphics community due to their speed and stability. PBD based methods converge to the problem solution by solving geometric constraints considering directly the object position and shape. They are computationally efficient, stable and are highly controllable. These are all important assets in the design of a robust and fast active perception framework. Meshless shape matching (MSM) [14] is a key algorithm among the field of PBD that simulates rigid and deformable objects [15]–[17]. In this study, we propose to estimate the parameter that define the elastic deformability of the object or part of the environment (β) from the observed real-world behavior using MSM.

Estimating parameters of a deformable model is a widely studied approach to realistically simulate the behaviors of objects [3], [18], [1], [19]. Frank et al. [1] learn the deformability model of an object by minimizing observed deformations and the FEM model prediction. Also, Boonvisut et al. [19] use a non-linear FEM-based method to estimate the mechanical parameters of soft tissues. However, these approaches assume homogeneous material properties. In [3], authors model heterogeneous soft tissues but they rely on a complex experimental setup consisting of several external cameras.

Unlike the previous approaches, we estimate the deformability of heterogeneous surfaces using MSM and Gaussian Process in a simpler, generic robotic experimental setup, i.e. a robotic arm and a depth camera sensor, see Fig. 1. It is showed in [20] that by matching real-world observation and MSM simulation, the deformability of objects can be estimated in a controlled 2D experimental setup. Here instead, we map the deformability of a surface with heterogeneous material properties by minimizing the error between the model prediction and observed deformations in 3D space.

III. Methodology

In this section, we describe Gaussian Processes for regression (GPR) [4] for 2.5 dimensional datasets[1] (Gaussian Random Field). We discuss how to exploit GPR to generate deformability distribution maps and geometric descriptions and show how to estimate the deformability parameters of an object through observation and simulation. This section ends with a description of the developed algorithms.

A. Gaussian Random Fields

A Gaussian Process Regression shaped over a bi-dimensional Euclidean set is commonly referred as Gaussian random field. We start defining the set $P_V = \{\mathbf{p_1}, \mathbf{p_2} \dots \mathbf{p_N}\}$,

[1]In a 2.5D dataset each xz coordinate has a single height y.

with $\mathbf{p_i} \in \mathbb{R}^3$, of measurements of 3D points generated by the visual sensor system. We define also $D_{RF} = \{\mathbf{x_i}, y_i\}_{i=1}^{N}$ a training set where $\mathbf{x_i} \in \mathbf{X} \subset \mathbb{R}^2$ are the *xy*-coordinates of the points in P_V and y_i the *z*-coordinates (heights)[2]. Similarly a set $\mathbf{X}_* \equiv \mathbf{X}_{\mathbf{rf}_*} \subset \mathbb{R}^2$ identifies a set of M test points. A terrain surface can be described with a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ where each vector of *xy*-coordinates generates a single height. This simplistic expression allows to efficiently describe 2.5D terrains but does not allow to model convex shapes which require multiple heights for a single *xy*-coordinate.

Such a function can efficiently be modeled by a GPR which places a multivariate Gaussian distribution over the space of $f(\mathbf{x})$. The GPR is shaped by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x_i}, \mathbf{x_j})$. The joint Gaussian distribution, assuming noisy observation $\mathbf{y} = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ and $m(\mathbf{x}) = 0$ on the test set $\mathbf{X}_*$ assume the following form

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^{\mathbf{T}} & \mathbf{k}_{**} \end{bmatrix}\right) \tag{1}$$

where $\mathbf{K}$ is the covariance matrix between the training points $[\mathbf{K}]_{i,j=1\ldots N} = k(\mathbf{x_i}, \mathbf{x_j})$, $\mathbf{k}_*$ the covariance matrix between training and test points $[\mathbf{k}_*]_{i=1\ldots N, j=1\ldots M} = k(\mathbf{x_i}, \mathbf{x}_{*\mathbf{j}})$ and $\mathbf{k}_{**}$ the covariance matrix between the only test points $[\mathbf{k}_{**}]_{i,j=1\ldots M} = k(\mathbf{x}_{*\mathbf{i}}, \mathbf{x}_{*\mathbf{j}})$.

The predictive function is obtained conditioning the model on the training set [4] :

$$p(f_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\overline{f_*}, \mathbb{V}[f_*]) \tag{2}$$

$$\overline{f_*} = \mathbf{k}_*^{\mathbf{T}} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y} \tag{3}$$

$$V[f_*] = \mathbf{k}_{**} - \mathbf{k}_*^{\mathbf{T}} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_* \tag{4}$$

We used the popular squared exponential kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = \sigma_e^2 \exp\left(-\frac{(\mathbf{x_i} - \mathbf{x_j})^T (\mathbf{x_i} - \mathbf{x_j})}{\sigma_w^2}\right) \tag{5}$$

which hyper-parameters σ_e, σ_w were empirically estimated based on a set of experiments made on a 1 m^3 area.

The mean of the joint distribution of a Gaussian random field allows to explicitly obtain the heightmap [21] of a terrain surface by simply using a grid of bi-dimensional testing points. The variance of the random field highlights regions of low density or noisy data, e.g. occluded portion of the map. In this paper, we use GPR for modeling both the geometric shape of the whole surface under analysis and its deformability properties that we denote β-field. For the latter, y_i of the training set D_{RF} contains the deformability parameter (β) of the surface estimated using MSM after a physical interaction on a selected target position.

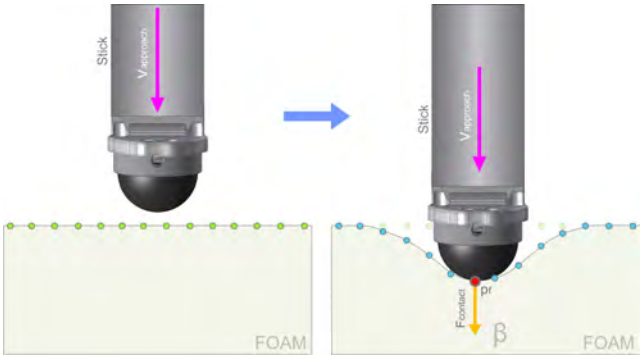


Fig. 3: Initial and final reconstructed point cloud used by the Position Based Dynamics algorithm. The Optoforce sensors ensure a constant normal force while collecting the second point cloud. The Gaussian Process Regressions allow to collect grid data points at uniform *xy*-coordinates while filtering noise.

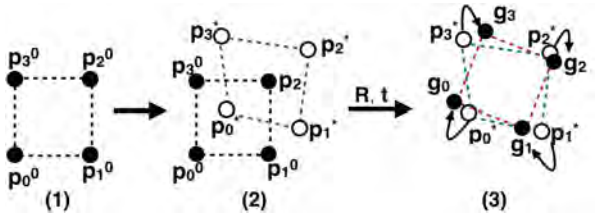


Fig. 4: For the sake of exposition, let's assume we want to maintain rigidity. (1) The initial shape of the object is represented with the point positions $\mathbf{p}_i^0$. (2) The points are displaced because of external forces and the intermediate deformed shape $\mathbf{p}_i^*$ occurs. The intermediate deformed shape does not embodies the knowledge of the object shape. (3) MSM determines the goal position $\mathbf{g}_i$ by calculating the optimal rotation and translation components that preserves the initial shape. Later, the intermediate deformed points $\mathbf{p}_i^*$ are pulled towards the goal positions $\mathbf{g}_i$.

B. Simulating deformation

The system uses MSM to simulate deformations. The simulation starts by storing the initial shape of the deformable object, $\mathbf{p}_i^0 \in \mathbb{R}^3$ where $i = 1, 2, 3, ..., K$ with K the number of points. The basic idea of MSM is shown in Fig. 4. In each time step, external forces such as gravity or collisions, move the points to unconstrained intermediate deformed positions $\mathbf{p}_i^* \in \mathbb{R}^3$. The unconstrained points are pulled to goal positions $\mathbf{g}_i \in \mathbb{R}^3$ which are determined by computing the optimal linear transformation between the initial shape $\mathbf{p}^0 \in \mathbb{R}^{3\times K}$ and intermediate deformed configuration $\mathbf{p}^* \in \mathbb{R}^{3\times K}$. We then extract the rotational $\mathbf{R} \in \mathbb{R}^{3x3}$ and translational components $\mathbf{t} \in \mathbb{R}^3$ of this linear transformation. The rotation and translation are the basis for the rigid transformation that moves the particles towards their goal position which respects the initial shape constraints.

To obtain rotational and translational components, a rotation matrix $\mathbf{R}$ and translation vectors $\mathbf{t}^0$ and $\mathbf{t}$ are determined by minimizing

[2]Axis are described considering the reference frame represented in Fig.1

$$\sum_i m_i(\mathbf{R}(\mathbf{p}_i^0 - \mathbf{t}^0) + \mathbf{t} - \mathbf{p}_i^*)^2 \ , \tag{6}$$

where m_i are the weights of the individual particles. The optimal translation vectors are the centre of mass of the initial shape and the deformed shape:

$$\mathbf{t}^0 = \frac{1}{m_c}\sum_i^K m_i\mathbf{p}_i^0 \ , \mathbf{t} = \frac{1}{m_c}\sum_i^K m_i\mathbf{p}_i^* \ , m_c = \sum_i^K m_i \ . \tag{7}$$

Finding the optimal rotation requires more complex steps than finding optimal translation vectors. In [13], authors relax the problem of finding the optimal rotation matrix $\mathbf{R}$ to finding the optimal linear transformation $\mathbf{A} \in \mathbb{R}^{3\times3}$ between the initial configuration $\mathbf{p^0}$ and the intermediate deformed configuration $\mathbf{p}^*$:

$$\mathbf{A} = \left(\sum_i m_i\mathbf{r}_i{\mathbf{s}_i}^\top\right)\left(\sum_i m_i\mathbf{s}_i{\mathbf{s}_i}^\top\right)^{-1} = \mathbf{A}_r\mathbf{A}_s \ , \tag{8}$$

where $\mathbf{r}_i = {\mathbf{p_i}}^* - \mathbf{t}$ and $\mathbf{s}_i = \mathbf{p}_i^0 - \mathbf{t}^0$ are the point locations relative to the center of mass. The matrix $\mathbf{A}_s$ is symmetric and contains only scaling information. Hence the rotational part can be obtained by decomposing $\mathbf{A}_r$ into the rotation matrix $\mathbf{R}$ and symmetric matrix $\mathbf{S}$ using polar decomposition $\mathbf{A}_r = \mathbf{RS}$ as in [13].

We determine the goal position in Fig. 4 as:

$$\mathbf{g}_i = \mathbf{R}\mathbf{s}_i + \mathbf{t} \ . \tag{9}$$

The steps described in Eq. (6-9) come from the well known Kabsch algorithm [22] and they only allow rigid transformation from the initial shape. To simulate deformation, [13] introduces linear deformation, e.g. shear and stretching by combining $\mathbf{R}$ and $\mathbf{A}$ as follows:

$$\mathbf{g}_i = ((1-\beta)\mathbf{R} + \beta\mathbf{A})\mathbf{s}_i + \mathbf{t} \ , \tag{10}$$

where β controls the degree of deformation, ranging from 0 to 1. If β approaches 1, the range of deformation increases, whereas if β is close to 0, the object behaves like a rigid body.

β is our parameter of interest for defining an object's deformability. Our goal is to estimate it by matching the simulated deformation and the observed deformation of the object.

Using linear transformation, only shear and stretch can be represented. To extend the range of deformation such as twist and bending modes, quadratic optimal transformation matrix $\bar{\mathbf{A}} \in \mathbb{R}^{3\times9}$ is calculated as follows and used instead of $\mathbf{A}$ in Eq. 10:

$$\bar{\mathbf{A}} = (\sum_i m_i\mathbf{r}_i\bar{\mathbf{s}}_i^\top)(\sum_i m_i\bar{\mathbf{s}}_i\bar{\mathbf{s}}_i^\top) = \bar{\mathbf{A}}_r\bar{\mathbf{A}}_s \tag{11}$$

where $\bar{\mathbf{s}}_i = [s_x, s_y, s_z,\ s_x^2, s_y^2, s_z^2,\ s_xs_y, s_ys_z, s_zs_x]^\top \in \mathbb{R}^9$.

For further expanding the range of deformation, the set of points are divided into overlapping clusters as seen in Fig. 5 and linear optimal translation A_j of each cluster j is calculated separately. The size of the cluster was empirically

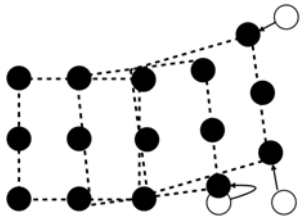


Fig. 5: Example point regions configuration with overlapping clusters with size 3x3.

chosen as described in Sec. IV-C. At each time step, the final position is determined by blending the goal positions of corresponding clusters:

$$\mathbf{g}_i = \frac{1}{M_i}\sum_{j\in\mathfrak{R}_i}\mathbf{g}_i^j \ , \tag{12}$$

where M_i is the number of clusters that particle i belongs to, $\mathfrak{R}_i$ is the set of clusters particle i belongs to, and $\mathbf{g}_i^j$ is the goal position which is associated with cluster $j \in \mathfrak{R}_i$.

C. Estimating deformability parameter

We model the shape of the virtual object as a surface fixed to the ground from edges. The initial shape of the object $\mathbf{p}^0$ is estimated before each physical interaction from the mean of the joint distribution of the GPR as shown in the left side of Fig. 3. To simulate the effects of a physical interaction, we select the point $\mathbf{p}_f$ closest to the manipulated region and fix its position in accordance with the disturbance as shown in the right side of Fig. 3. The simulator generates a goal configuration $\mathbf{g}^\beta$ for a specific β.

To estimate the deformability parameter β that best describes the locally deformed surface, we minimize an error function that measures the distance between the observed deformed shape $\bar{\mathbf{X}}$ and the simulated deformed shape $\mathbf{g}^\beta$. $\bar{\mathbf{X}}$ consists of the test set $\mathbf{X}_*$ and heightmap $\bar{f}_*$ modelled by GPR as described in Sec. III-A. The error function is calculated as follows:

$$E(\beta) = \frac{1}{K}\sum_{i=1}^K \min_{\bar{\mathbf{x}}_j\in\bar{\mathbf{X}}}(\|\mathbf{g}_i^\beta - \bar{\mathbf{x}}_j\|) \tag{13}$$

where $\bar{\mathbf{x}}_j$ and $\mathbf{g}_i^\beta$ are the jth and ith points from $\bar{\mathbf{X}}$ and $\mathbf{g}^\beta$ respectively. To find the minimum, the simulation runs for a number of β values uniformly sampled from the interval $[0, 1)$. The β that gives the lowest residual in Eq. (13) is selected as representing the deformability of the surface.

D. The algorithm process flow

The active exploration task starts with a full observation of the entire surface under analysis. The point cloud generated from this initial observation is cropped and filtered using a statistical outliers removal filter [23]. A Gaussian random field (GPR world, in Fig. 2) is trained on the 3D points of the filtered point cloud as described in Sec. III-A. Such GPR builds an internal geometric representation of the environment allowing to obtain compact representations

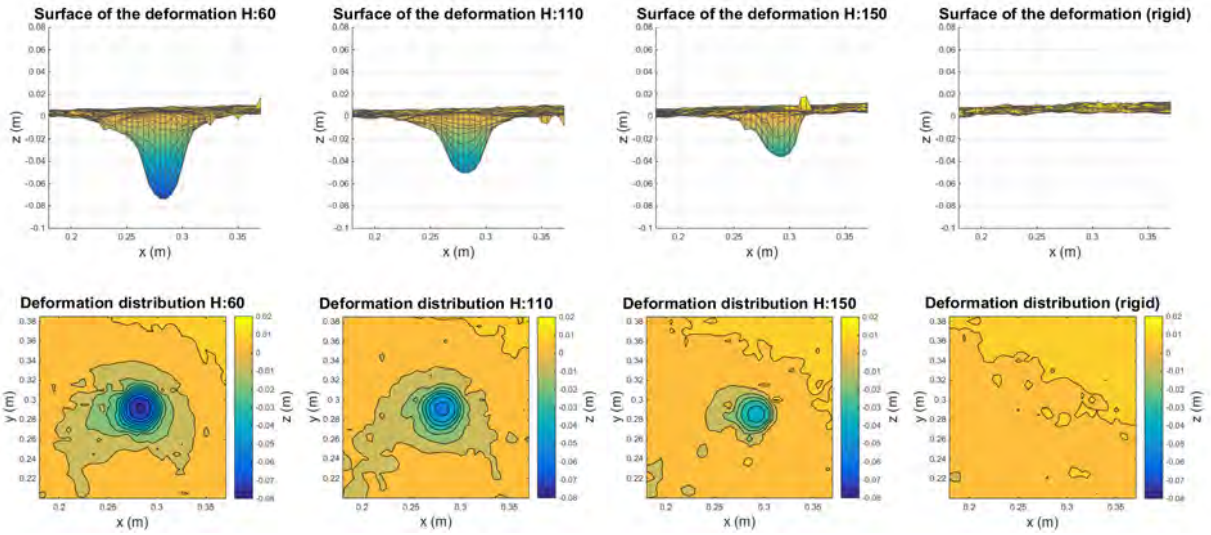


Fig. 6: Reconstruction of different deformations of different foams obtained applying the same force on the same contact point. H is the hardness of the surface defined as in Sec. IV-B.

of selected sub-regions (ROI). This is done by considering the mean of the joint distribution of the GPR world model inferred on a dense (0.5 cm) grid of 3D points centered on a ROI.

A second Gaussian Process (GPR β-field) is then initialized on the *xy*-coordinates of the whole geometric map. The block ROI selector of Fig. 2 analyzes the variance of the joint distribution of the GPR β-field model using a dense (0.5 cm) grid of two dimensional points (*xy*-coordinates) looking for regions of highest uncertainty - meaning that the β distribution is poorly modeled because of missing information or high noise. A randomly selected point[3] is a candidate target for the active exploration task.

In the successive step, the arm is moved toward the selected region. The approach vector has direction orthogonal to the surface under analysis on the target point location. We use hybrid position-force control [24] in proximity of the target point to impose a constant force on the direction of the approach vector while allowing displacements along the orthogonal directions.

When the Optoforce sensor detects a certain normal force the arm stops and the environment is observed again. From this second observation the system generates a new point cloud that contains the local environmental deformation. We use a convex hull filter to remove all the 3D points representing the robot hand and stick. The dimension and position of the convex hull is estimated from proprioceptive data using the robot model. The final point cloud contains occluded regions (incomplete point cloud) because of the hand and stick presence. In order to generate a compact representation of the deformation we train a third GPR (GPR-touch in Fig. 2) on a squared cropped ROI of the final point cloud that contains the deformation along with 10 tactile points. The tactile points are virtually generated considering the position and shape of the spherical surface of the Optoforce sensor along with the contact force direction as shown in Fig. 3. From the mean of the joint distribution of GPR-touch we create deformation shapes as shown in Fig. 6. Using the method described in Sec. III-C we use the two point clouds in order to get a local β value. The GPR β-field is finally trained on the locations subjected to deformation using the estimated β value as described in Sec. III-A. The exploration step is repeated until the ROI detector block does not find a new candidate point for the next physical interaction (meaning that the variance distribution of the β-field is low on the whole map). The threshold value for the variance was empirically estimated through several experiments. Its value determines the numbers of interactions needed and, as we will show in Sec. IV-C, the accuracy of the built β-field map.

[3]Selected among the regions carrying higher uncertainty.

IV. Experimental evaluation

A. Hardware setup

The hardware setup (see Fig. 1) used in the experiments consists of a PrimeSense RGB-D camera, a Kinova Jaco2[4] robotic arm equipped with a 3 fingered Kinova KG-3 gripper and a 3D OptoForce force sensor[5]. The camera is placed 80 cm above the table. The relative orientation between the camera and the table plane is 45°. We use a rigid 10 cm stick, mounted on the Kinova hand, for the interaction.

Sets of homogeneous and heterogeneous elastic foams of different shapes are placed on the table and explored by the arm. The OptoForce sensor, that can detect slipping and shear forces with high frequency, is placed on the tip of the stick. The haptic sensor output consists of a 9D force-position vector generated at 1 kHz. When the desired force is reached, contact force direction together with stick orientation and

[4]Kinova website: http://www.kinovarobotics.com/

[5]Optoforce website: http://optoforce.com/

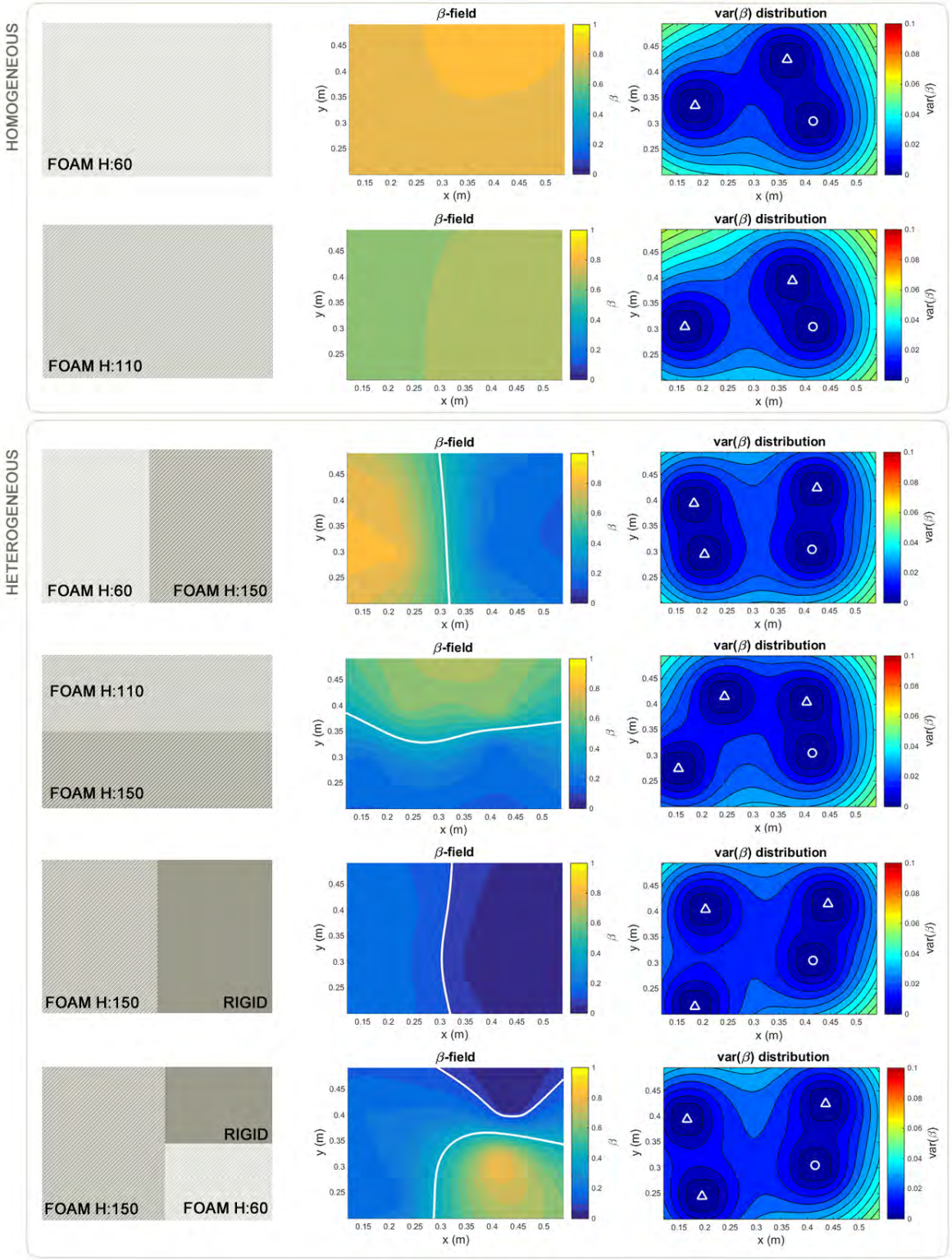


Fig. 7: Experiments setup. The first column shows an illustrative representation of the foams size, hardness and position for each experiment (rows). The second column shows the estimated deformability distribution (β-field). The third column shows the variance of the β-field along with the contact positions.

sensor position (proprioceptive data) are used to generate tactile 3D points.

All the software components (nodes) run under the robot operative system (ROS). Visual data are analyzed using the Point Cloud Library (PCL).

B. Experimental scenarios

To validate our approach, we tested the framework on six different scenarios described in Table I. All the analyzed surfaces covered an area of 60×40 cm. The filtered point clouds covering the analyzed regions counted ~12,000 points in average. The foams had different shapes and hardness but equal density. In the first two scenarios, homogeneous foams were physically explored by the arm. In the last four experiments, several foams having different sizes and hardness were attached together and used to assess the heterogeneous deformability properties.

The algorithm starts the active exploration task by interacting with a predefined initial xy-coordinate. The successive target points were randomly selected among those sub regions of the β-field having a variance higher than a given threshold.

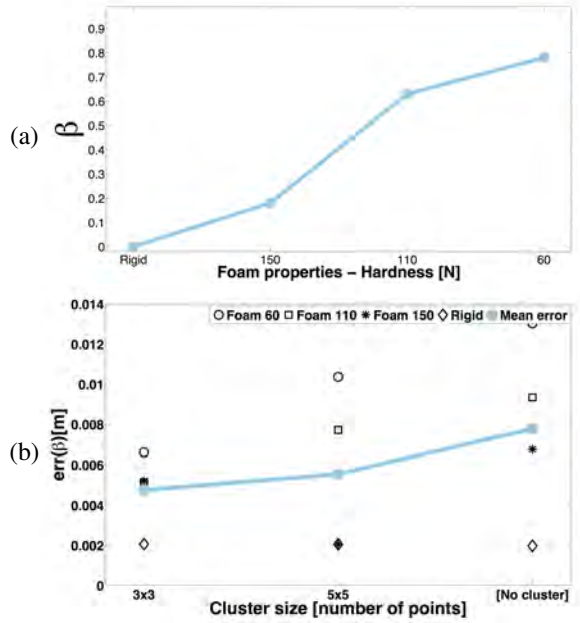


Fig. 8: (a) The optimal β as a function of deformability. The estimated deformability increases with decreasing firmness of the surface. (b) The points that represent the shape of the object are divided into overlapping clusters. The cluster size that gives the least error for optimal β estimation is selected.

C. Results

Fig. 8(a) shows the deformability values β, estimated following the approach described in Sec. III-C, as a function of decreasing hardness. This figure illustrates how the estimated deformability for each foam is in accordance with the real deformation characteristics. In Fig. 8(b), we also show the evolution of the error as a function of cluster size. The figure indicates that 3×3 cluster size models the observed deformation with the best accuracy. This happens because using small cluster size increases the range of deformation that can be modeled by MSM and therefore the accuracy of the estimation.

Scenario description (hardness)	n° interact.	T_h var(β)	β-regions
Homogeneous [60]	3	0.06	1
Homogeneous [110]	3	0.06	1
Heterogeneous [60/150]	4	0.06	2
Heterogeneous [110/150]	4	0.06	2
Heterogeneous [150/rigid]	4	0.06	2
Heterogeneous [60/110/rigid]	4	0.06	3

TABLE I: Different scenarios setup used during the experiments. The hardness of the foams (H), defined in terms of the force (N) required to compress the foam to 40%, is provided by the foam manufacturer. T_h represents the threshold value for the variance on the β-field.

Fig. 6 shows the mean of the joint distribution of the geometrical random field after contacts with different foams. It can be seen that the Gaussian random field creates a compact representation of the deformation shape that otherwise would be partially occluded by the stick and affected by noise.

Fig. 7 illustrates the interaction steps for the presented scenario and the corresponding evolution of the β-fields. The first column (ground truth) shows a representation of the setup with the position of the foams along with their hardness and relative dimension. The second column shows the corresponding estimated β-fields. A contour function identifies the best isopleth with the corresponding subdivisions of the β-fields into regions. It is possible to notice how the algorithm correctly identifies regions having different deformability and models the whole deformation map consistently with the ground truth. The last column indicates the variance distribution of the β-field along with the target points selected during the active exploration task. The first target point was pre-assigned and it is indicated with a circle whereas triangles indicate successive contacts. We invite the reader to note how the subdivision of the β-field into regions of the last experiment (last row of Fig. 7) slightly differs from its ground truth. This shows a sensibility of the proposed framework to the selected variance threshold. An increase in the variance threshold (which was empirically chosen during our experiments) helps limiting the number of interactions needed (less target points found) but at the same time decreases the accuracy of the β-field (regions having low variance are considered as explored). All the experiments lasted in average ~1.5 min including arm motion, planning and β-field calculation.

V. Conclusions

We have presented a novel active perception framework for modeling heterogeneous deformable surfaces[6]. The main

[6]A video of an experiment is available at: https://youtu.be/**hiddenlink**

contribution of our work is the ability to model the deformability distribution (β-field) of an environment from few physical interactions. The novelty of the approach is in the use of real-world observations in a PBD simulator for estimating the deformability parameters. PBD based methods are computationally efficient which is an important aspect for online active perception tasks. Our data-driven system relies on multisensory observations and selects regions to be interactively explored for assessing the deformability. The presented framework is particularly suitable for applications that require the robot to promptly investigate the environment minimizing the required environmental interactions.

We demonstrate the feasibility of our method through several real world experiments, using a simple setup consisting of a robotic arm, an RGB-D camera and a force sensor. We show how the obtained β-fields of the analyzed surfaces matched the ground truth.

There are several aspects of our method that deserve further attention. We have only modeled elastic, isotropic behaviors of heterogeneous surfaces. We want to increase the potentiality of our framework by capturing plastic and anisotropic behaviors. Another limitation is that the estimated deformability is expressed as a virtual (β value) of the deformation rather than a real physical measurement. Such values can change considerably if the simulator settings (e.g. cluster size) change. This can affect the accuracy of modeling surface deformability. Hence the variableness of PBD simulation should be investigated further as a future research. Finally, analysis of environmental visual appearance such as color and texture, can help the probabilistic framework to identify regions that are likely to have uniform material properties.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding under the European Union's seventh framework program (FP7), under grant agreements FP7-ICT-609763 TRADR.

REFERENCES

[1] B. Frank, C. Stachniss, R. Schmedding, M. Teschner, and W. Burgard, "Learning object deformation models for robot motion planning," *Robotics and Autonomous Systems*, vol. 62, no. 8, pp. 1153 – 1174, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014000797

[2] S. Rodriguez, J.-M. Lien, and N. M. Amato, "Planning motion in completely deformable environments," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2466–2471.

[3] B. Bickel, M. Bächer, M. A. Otaduy, W. Matusik, H. Pfister, and M. Gross, "Capture and modeling of non-linear heterogeneous soft tissue," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 89:1–89:9, July 2009. [Online]. Available: http://doi.acm.org/10.1145/1531326.1531395

[4] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006. [Online]. Available: http://www.gaussianprocess.org/gpml/chapters/

[5] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1054–1060.

[6] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," pp. 2845–2850, May 2011.

[7] C. K. I. Williams, *Learning in Graphical Models*. Dordrecht: Springer Netherlands, 1998, ch. Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond, pp. 599–621.

[8] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013*, Nov 2013, pp. 3180–3186.

[9] M. Gerardo-Castro, T. Peynot, and F. Ramos, "Laser-radar data fusion with gaussian process implicit surfaces," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 289–302.

[10] S. Caccamo, Y. Bekiroglu, C. H. Ek, and D. Kragic, "Active exploration using gaussian random fields and gaussian process implicit surfaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016*, Oct 2016.

[11] M. Salzmann and P. Fua, "Deformable surface 3d reconstruction from monocular images," *Synthesis Lectures on Computer Vision*, vol. 2, no. 1, pp. 1–113, 2010.

[12] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," in *Computer Graphics Forum*, vol. 25, no. 4. Wiley Online Library, 2006, pp. 809–836.

[13] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *J. Vis. Comun. Image Represent.*, vol. 18, no. 2, pp. 109–118, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2007.01.005

[14] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, "Meshless deformations based on shape matching," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 471–478, July 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073216

[15] Y. Tian, Y. Yang, X. Guo, and B. Prabhakaran, "Haptic-enabled interactive rendering of deformable objects based on shape matching," in *Haptic Audio Visual Environments and Games (HAVE), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 75–80.

[16] B. Zhu, L. Gu, J. Zhang, Z. Yan, L. Pan, and Q. Zhao, "Simulation of organ deformation using boundary element method and meshless shape matching," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 3253–3256.

[17] J. Liu and G. Su, "Multi-scale method for adaptive mesh editing based on rigidity estimation," in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*. IEEE, 2008, pp. 55–62.

[18] G. Bianchi, B. Solenthaler, G. Székely, and M. Harders, "Simultaneous topology and stiffness identification for mass-spring models based on fem reference deformations," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004*. Springer, 2004, pp. 293–301.

[19] P. Boonvisut and M. C. Cavusoglu, "Estimation of soft tissue mechanical parameters from robotic manipulation data," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 5, pp. 1602–1611, 2013.

[20] P. Guler, K. Pauwels, A. Pieropan, H. Kjellstrom, and D. Kragic, "Estimating the deformability of elastic materials using optical flow and position-based dynamics," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, Nov 2015, pp. 965–971.

[21] R. J. Peckham and G. Jordan, *Digital Terrain Modelling: Development and Applications in a Policy Support Environment*. Springer Science & Business Media, 2007.

[22] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, Sep 1976. [Online]. Available: http://dx.doi.org/10.1107/S0567739476001873

[23] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=1733343.1733366

[24] W. D. Fisher and M. S. Mujtaba, "Hybrid position/force control: a correct formulation," *The International journal of robotics research*, vol. 11, no. 4, pp. 299–311, 1992.

3D Registration of Aerial and Ground Robots for Disaster Response: An Evaluation of Features, Descriptors, and Transformation Estimation

Abel Gawel*, Renaud Dubé*, Hartmut Surmann†,
Juan Nieto*, Roland Siegwart* and Cesar Cadena*
*Autonomous Systems Lab, ETH Zurich, †Fraunhofer IAIS, University of Applied Sciences Gelsenkirchen

***Abstract*— Global registration of heterogeneous ground and aerial mapping data is a challenging task. This is especially difficult in disaster response scenarios when we have no prior information on the environment and cannot assume the regular order of man-made environments or meaningful semantic cues. In this work we extensively evaluate different approaches to globally register UGV generated *3D* point-cloud data from LiDAR sensors with UAV generated point-cloud maps from vision sensors. The approaches are realizations of different selections for: a) local features: key-points or segments; b) descriptors: FPFH, SHOT, or ESF; and c) transformation estimations: RANSAC or FGR. Additionally, we compare the results against standard approaches like applying ICP after a good prior transformation has been given. The evaluation criteria include the distance which a UGV needs to travel to successfully localize, the registration error, and the computational cost. In this context, we report our findings on effectively performing the task on two new Search and Rescue datasets. Our results have the potential to help the community take informed decisions when registering point-cloud maps from ground robots to those from aerial robots.**

I. Introduction

Multi-robot applications with heterogeneous robotic teams are an increasing trend due to numerous advantages. An Unmanned Ground Vehicle (UGV) can often carry high payloads and operate for extended periods of time, while an Unmanned Aerial Vehicle (UAV) offers swift deployment and the opportunity to rapidly survey large areas. This is especially beneficial in Search and Rescue (SaR) scenarios, see Fig. 1 as an example. Here, an initial overview can be made using UAVs before deploying UGVs for closer exploration in areas of interest. However, when it comes to efficiently combining the strengths of such robotic teams, we face numerous challenges. Additionally to the large difference in point of view, the sensor modalities used for mapping and localization are often drastically different for UAVs and UGVs. While UAVs typically use cameras as the prime sensor, UGVs often rely on LiDAR. This poses a major challenge in efficiently exploiting the UAV data on a UGV as registration between different sensor modalities is difficult to perform. Furthermore, using advanced functionalities such as traversability analysis and path planning for UGVs on UAV generated maps requires tight alignment between the data of different modalities, active localization and suitable map representations.

One critical step in using maps across several robots is the identification of the alignment between their maps. Several techniques are possible with increasing generality [1]. Firstly it is possible to impose a common origin of different robots' maps, e.g., by using common starting locations as done by Michael et al. [2]. Another option is to use global positioning sensors that allow for a good initial guess on the alignment of coordinate frames. In the case that several robots operate concurrently, it is also possible to find an alignment by a relative localization of the robots against each other [3]. However, the most challenging task is to register maps without any prior information regarding their mutual alignment. Furthermore, for the SaR application as in the "Long-Term Human-Robot Teaming for Robots Assisted Disaster Response" (TRADR) project, the scenarios are completely unpredictable which rules out the possibility of using supervised learning into the pipeline [4].

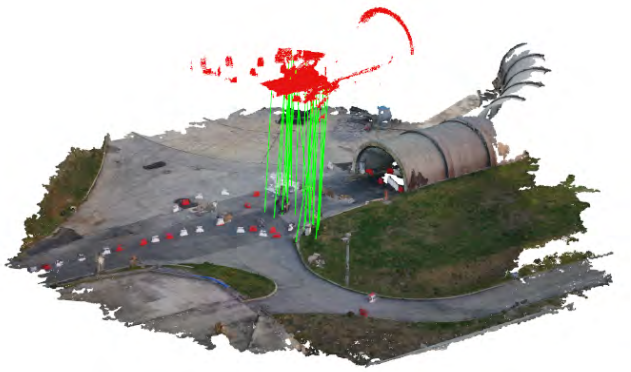


Fig. 1: Global localization of a *3D* UGV sub-map (red point-cloud) in a *3D* UAV reference map (coloured point-cloud). Green lines indicate the resulting matches associated to points in the two point-clouds. The data stems from the Montelibretti outdoor dataset and drawn from the complex experimental set-up.

While our previous work on online multi-robot SLAM for *3D* LiDARs [5][1] demonstrates a reliable registration among point-cloud maps taken from multiple ground robots, there is still the issue of dealing with differences in modality and in point of view between UAVs and UGVs.

The above mentioned challenges motivate us to evaluate several techniques to globally localize a UGV using its LiDAR sensor in a point-cloud map generated using the Multi-View Reconstruction Environment (MVE) [6] from images recorded by a UAV. The global registration (or localization) pipeline schematized in Fig. 2 consists of feature extraction, feature description and matching, and a 3D transformation estimation. We provide an evaluation and an analysis of the

[1]Within this paper, this system will be referred to as LaserSLAM.

implementation and performance of different choices for the modules in this registration pipeline. These choices are:

- Local feature extraction: key-points or segments.
- Feature descriptors: Fast Point Feature Histogram (FPFH), Unique Signatures of Histograms for Local Surface Description (SHOT) or Ensemble of Shape Functions (ESF).
- Transformation estimation: RANSAC based or Fast Global Registration (FGR).

The evaluation is conducted on two real world datasets of an indoor and an outdoor SaR scenario. This paper presents the following contributions:

- Extensive evaluation of global registration realizations for registering UGV and UAV point-clouds from LiDAR and camera data respectively.
- Two new datasets for multi-modal SLAM in SaR scenarios.

II. Related work

The field of *2D* metrical map-merging based on overlapping map segments is well studied in literature [7–9]. However, the task is increasingly difficult when moving to *3D* environments [1], especially when dealing with heterogeneous robotic teams, where *3D* data is generated from different sensors and with different noise characteristics [10]. Michael et al. [2] demonstrate a system for collaborative UAV-UGV mapping. The authors propose a system where a UGV equipped with a LiDAR sensor performs *2.5D* mapping, using the flat ground assumption and consecutively merging scans using Iterative Closest Point (ICP). In dedicated locations a UAV equipped with a 2D LiDAR is launched from the UGV and maps the environment using a pose-graph SLAM algorithm. Maps generated from the UAV are then fused online with the UGV map using ICP initialized at the UAV starting location.

Forster et al. [11] go a step further in fusing UAV-UGV map data from different sensors, i.e., RGB-D maps from the UGV and dense monocular reconstruction from the UAV. The registration between the maps is performed using a *2D* local height map fitting in x and y coordinates with an initial guess within a $3m$ search radius. The orientation is a priori recovered from the magnetic north direction as measured by the Inertial Measurement Unit (IMU)s. In a related setting Hinzmann et al. [12] evaluate different variants of ICP for registering dense *3D* LiDAR point-clouds and sparse *3D* vision point-clouds from Structure from Motion (SfM) recorded with different UAVs into a common point-cloud map using an initial GPS prior for the map alignment.

Instead of using the generated *3D* data for localizing between RGB and *3D* LiDAR point-cloud data, Wolcott and Eustice [13] propose to generate *2D* views from the LiDAR point-clouds based on the surface reflectivity. However, this work focuses only on localization and it is demonstrated only on maps recorded from similar points of view.

In our previous work [14] we presented a global registration scheme between sparse *3D* LiDAR maps from UGVs and vision keypoint maps from UAVs, exploiting the rough geometric structure of the environment. Here, registration is performed by clustering of geometric keypoint descriptors matches between map segments under the assumption of a known z-direction as determined by an IMU.

Zeng et al. [4] present geometric descriptor matching based on learning. However, this approach is infeasible in unknown SaR scenarios, as the descriptors do not generalize well to unknown environments.

Dubé et al. [15] demonstrate better global localization performance in *3D* LiDAR point-clouds by using segments as features instead of key-points. This approach has been demonstrated with multiple UGVs with the same robot-sensor set-up, but it is still to be studied how the approach performs under large changes in point of view.

Assuming good initialization of the global registration, Zhou et al. [16] perform a robust optimization. The work claims faster and more robust performance than ICP.

In summary, the community addresses the problem of heterogeneous localization. However, there is a research gap in globally localizing from one sensor modality to the other in full *3D* without strong assumptions on view-point, terrain or initial guess.

III. Aerial-Ground robot mapping system

In this section, we present our SLAM system. It extends our LaserSLAM system [5] with the component of global map alignment and localization in point-cloud maps from different sources, as well as online extension of these maps. Fig. 2 illustrates the architecture of the proposed system. While the major LaserSLAM system is running on the UGV, it also allows to load, globally align, and use point-cloud maps from other sources. As example, maps generated via MVE from UAVs or point-cloud maps resulting from bundle adjustment on data collected by another LiDAR equipped robot can all be leveraged.

A. Mapping algorithms

In the proposed system, we use a dual map representation for the different tasks of the robots, i.e., point-cloud maps and *OctoMaps* [17]. While the individual robots maintain point-clouds and surface meshes, these are integrated in a global *OctoMap* representation serving as the interface to other modules of the SaR system, e.g., traversability analysis as shown in [18]. Another advantage of the unified *OctoMap* representation is a persistent representation which also incorporates dynamic changes detection.

On the UAV's monocular image data we perform an SfM and Multi-View-Stereo-based scene reconstruction using the MVE [6]. The MVE produces a dense surface mesh of the scene by extensive matching and is therefore an offline method that is computed off-board the UAV. Although, efficient online mapping methods exist, we decide to use a method that produces high quality maps, that can be further used in the TRADR system, e.g., on the UGV for path planning or for situation awareness of first responders.

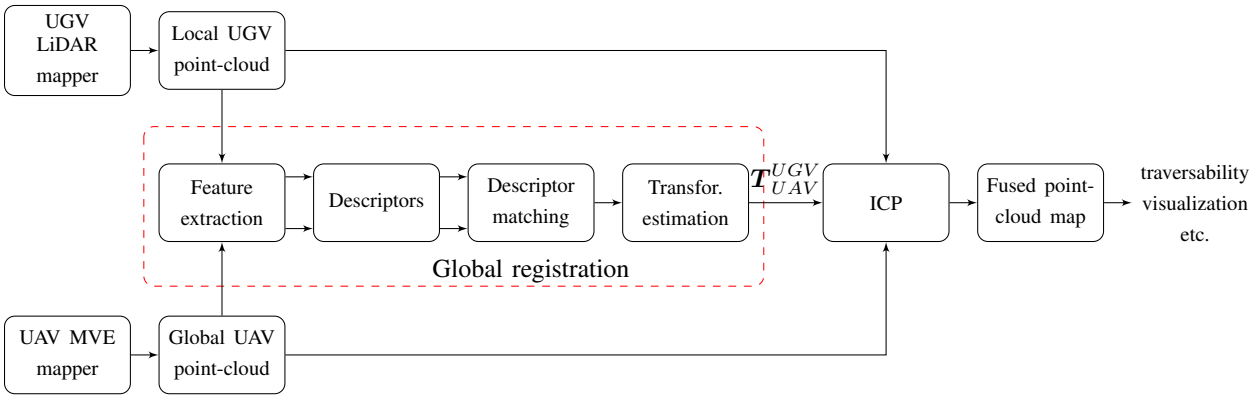


Fig. 2: Mapping System overview. The inputs to the system are the local UGV map and the global UAV reference map. If a global registration is triggered, the key-points are computed on both maps. Consecutively, the system performs descriptor extraction and matching. The initial global transformation is then refined by a step of ICP between the global and the local clouds, resulting in a fused map that is used for further functionalities of the system, such as path planning.

On the UGVs we use a variant of the LaserSLAM system which estimates in real-time the robot trajectory alongside with the *3D* point-cloud map of the environment. LaserSLAM is based on the *iSAM2* [19] pose-graph optimization approach and implements different types of sequential and place recognition constraints. In this work, odometry constraints are obtained by fusing wheel encoders and IMU data using an extended Kalman filter while scan-matching constraints are obtained based on ICP between successive scans.

After the creation of the UAV map, we facilitate a global registration scheme to localize the UGV in the UAV map as described in Sec. IV.

For UGV-only mapping, the LaserSLAM framework enables multiple robots to create consistent *3D* point-cloud maps. However, a different regime must be followed for generating and extending a consistent *3D* map by fusing in the UAV dense *3D* maps. Since the UAV maps are the result of an offline batch optimization process, the maps are already loop closed and represent a consistent initial basis for the global map. Furthermore, we treat the UAV maps as static, i.e., the map is taken *as is*. LaserSLAM is therefore extended to include a mode that allows the robot to use a given base map. This base map is then treated as the fully optimized map and extended with updates from the UGV LiDAR. It is important to note that the point-cloud map is only the internal representation for the robot to perform SLAM. All map updates as well as dynamic changes are maintained in the *OctoMap* representation that is derived from the point-cloud updates and serves as a unified representation for all processes using the mapping data.

B. Map usage

Thanks to the unified *OctoMap* interface, the merged map data can directly be used on other modules of the TRADR system. Notably, it can directly be used for traversability analysis and subsequent metrical path planning. The system therewith enables the UGVs to also use UAV generated maps for path planning. A UGV-loaded UAV map of one testing site (see Fig. 1) is depicted in Fig. 3 indicating traversable areas in green and non-traversable areas in red.

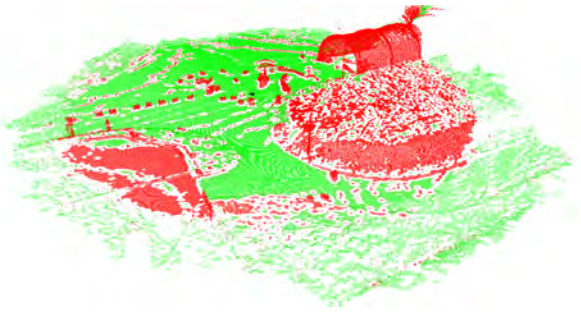


Fig. 3: Resulting UGV traversability estimation on the outdoor Montelibretti dataset. Traversable areas are marked in green, while non traversable areas are indicated with red. The parametrization is identical to the parametrization for LiDAR traversability analysis.

The *OctoMap* serves as interface for further modules of the system, such as novelty detection which is a separate contribution and out of the scope of this work.

IV. Global Registration

This section describes the pipeline that we use to globally register the UGV with respect to the UAV point-cloud map. The evaluation of different choices within this pipeline is the focus of our paper. The global registration consist of four modules: feature extraction, description, matching and, estimation of the $\mathbb{SE}(3)$ transformation.

A. Feature extraction

This module defines which components in the point-cloud map are going to be used for the registration. Key-point are samples from the full point-cloud that have some level of invariance to the point of view. Here, we use the Intrinsic Shape Signatures (ISS) detector [20]. The next option, is to add more information by clustering the point-cloud, resulting in segments. These segments are taken as the local features with the potential of being more descriptive than just 3D points [15]. Here, we follow a Euclidean based clustering

Realization	Feature	Descriptor	Trans. Estimation
FPFH	Key-point	FPFH	RANSAC-based
FPFH FGR	Key-point	FPFH	FGR
FPFH seg	Segments	FPFH	RANSAC-based
SHOT	Key-point	SHOT	RANSAC-based
SHOT FGR	Key-point	SHOT	FGR
SHOT seg	Segments	SHOT	RANSAC-based
ESF seg [15]	Segments	ESF	RANSAC-based

TABLE I: Global registration realization by different choices in the sub-modules.

as the segmentation algorithm. We do not explore global features as they are highly point of view dependent.

B. Descriptors

This module takes each feature and computes a descriptor with the aim of being descriptive enough such that it is reproducible on different maps of the same location. The descriptor is based on the key-point, and its neighborhood, or on the subset of points that belong to a segment. Here, we explore three descriptors:

- Fast Point Feature Histogram (FPFH) [21].
- Unique Signatures of Histograms for Local Surface Description (SHOT) [22].
- Ensemble of Shape Functions (ESF) [23].

C. Description Matching

The matching module is in charge of solving the data association problem between features from both maps by comparing their descriptors. In our implementation we use the nearest neighbor search in the space of the corresponding descriptor.

D. Transformation Estimation

Once a set of 3D point pairs is declared, this module computes the transformation such that the 3D points from one map are moved to the location of their correspondences in the reference map. In absence of outliers, the problem could be solved by minimizing a least square error function. Unfortunately, the presence of outliers is unavoidable and this module must deal with them. Here we explore two alternative methods. The first one is a RANSAC-based approach which is already available in PCL. The second one is based on the recent proposed FGR [16]. FGR uses the scaled Geman-McClure estimator as robust cost function into the optimization objective to neutralize the possible outlier matches.

E. Realizations

We explore different global registration alternatives by choosing different methods in each module. The realizations are as shown in Table I.

As global registration strategies, the evaluation focuses on 11 different configurations. Those that are shown in Table I plus their combinations when removing the ground plane prior to key-point detection, denoted by *gr* at the end. Ground removal is done by RANSAC based plane fitting.

F. Performance metrics

For the evaluation metrics, we use transformation errors $\Delta\boldsymbol{T}$ on the alignment between the UGV and UAV maps that are represented as

$$\Delta\boldsymbol{T} = \begin{bmatrix} \Delta\boldsymbol{R} & \Delta\boldsymbol{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{1}$$

with rotation matrix $\Delta\boldsymbol{R}$ and translation vector $\Delta\boldsymbol{t} = (\Delta\mathbf{x}, \Delta\mathbf{y}, \Delta\mathbf{z})^{\mathbf{T}}$. The translational error e_t is computed as follows:

$$e_t = \|\Delta\boldsymbol{t}\| = \sqrt{\Delta\mathbf{x}^2 + \Delta\mathbf{y}^2 + \Delta\mathbf{z}^2} \tag{2}$$

The rotational error e_r equates to:

$$e_r = \arccos \frac{trace(\Delta\boldsymbol{R} - \boldsymbol{I})}{2} \tag{3}$$

It is important to note that the two map types are not perfectly aligned in all locations due to the multi-modal nature of the data and we can therefore only evaluate errors down to a positional resolution of $0.2m$ and angular resolution of 2° respectively. Furthermore, we register the data using ICP in the basic experiment, to give an indication on the achievable alignment, as ICP always converged to a good solution in our experiments given a good initial guess. Motivated by the results of Hinzmann et al. [12], we consider registrations as successful when ICP is able to perform the final local alignment. Therefore, the thresholds for translational and rotational errors are set to $e_t = 3m$ and $e_r = 5^\circ$ above the resulting ICP solution of the basic experiment to count successful registrations.

V. Experiments

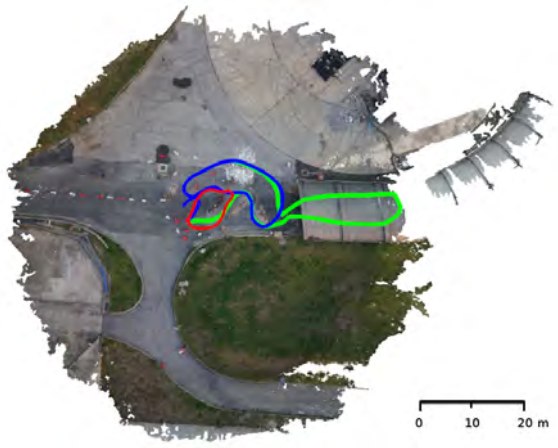
We evaluate our approach on two challenging SaR datasets recorded within the TRADR project which we make available with this publication[2]. The evaluation focuses on the global registration of the multi-modal point-cloud data.

A. Datasets

The first of these datasets was generated in an outdoor firemen training location in Montelibretti, Italy. The scenario simulates a car accident around a tunnel. It consists of six UGV runs in partly overlapping locations of the disaster area and one large UAV-generated map covering the whole site of approximately $80m \times 80m$ that was scaled using GPS information. The travelled trajectories of the robots are $2 \times 30m$, $2 \times 60m$, and $2 \times 110m$ of consecutive missions following the same paths twice. For the evaluation we use one UGV run of each size. Here, the UGV mapping data is fully covered in the UAV map, except for an indoor exploration of the tunnel which was not accessible to the UAV. The scenario and the trajectories are depicted in Fig. 4a.

The second dataset was recorded at a decommissioned power plant in Dortmund, Germany, consisting of several UGV runs and several large UAV-generated maps covering

[2]The datasets are available under http://robotics.ethz.ch/ asl-datasets/



(a) Montelibretti outdoor dataset.

(b) Dortmund indoor dataset.

Fig. 4: Top-down views of two SaR datasets considered in our experiments. The robot trajectories are indicated in green, red, and blue and overlaid on the colored UAV point-clouds.

different parts of the power plant, including the entire machine hall which was also visited by one UGV and has a size of $100m \times 20m$. The UGV run is fully covered in the UAV map and has a travelled distance of approximately $80m$, as illustrated in Fig. 4b. Since no GPS signals are available in the interior of the building, the UAV maps were scaled using the buildings' windows as reference to the outside maps and scaled accordingly.

B. Experimental setup

We evaluate several global registration strategies on increasingly challenging experimental set-ups. All set-ups consider the iteratively growing UGV map produced by laser-SLAM as local map. For the basic set-up, the global map is a cropped version of the UAV map, that approximately covers the space of the local map at any iteration. The more challenging intermediate set-up uses the cropped UAV map as seen at the last iteration of the UGV mapping as global map. Finally, the complex set-up considers the full UAV map for global localization.

As global registration strategies, the evaluation focuses on the 11 configurations presented in Sec. IV-E.

C. Registration performance

This section evaluates the global registration performance of the different algorithms considered, using the metrics presented in Sec. IV-F.

1) Parametrization: We choose the parametrization of the FPFH and SHOT descriptors to yield good performance across all data used, i.e., a histogram and search radius of $2.0m$ for FPFH and SHOT respectively. Our parameter choice is further motivated by extensive evaluation and shows plateauing performance in a large region around the chosen size, indicating robust performance. The matcher is based on performing fast nearest neighbor search in a FLANN tree [24], while the geometric verification is based on RANSAC and clustering. Furthermore, FGR is parametrized for the best possible performance we could find.

D. Results

Fig. 1 and Fig. 5 illustrate qualitative global registration between UGV sub-maps and global UAV maps on the tested datasets. In Fig. 6 and Table II the quantitative performance of the evaluated approaches is depicted as averaged over multiple runs with different initializations. While Fig. 6 shows translational and rotational error of the individual approaches over all datasets, Table II reports the minimal amount of cumulated UGV scans, i.e., the minimal travelled distance for reliable global registration. Here, we define reliable global registration, if from the associated UGV sub-map, the errors do not exceed the error thresholds e_t and e_r for 90% of the cases. Note, that we indicate combinations that failed to produce result within this margin as *N/A*.

In this experimental set-up the descriptor matching approach as described in Section IV performs best throughout all experiments. FPFH yields satisfying performance in the basic experiments. However, its performance drastically degrades in the more complex cases.

SHOT on the other hand shows reliable performance throughout all experiments, with the required overlap increasing with the complexity of our test-cases. Here, the ground removal does not provide a significant performance boost, especially since the ground plane extraction is unreliable on the large UAV map as we do not have incremental pose-updates from the robot on segments of the map. However, ground removal does not degrade the performance as it does for FPFH, expressing robust performance of SHOT over varying conditions.

While the RANSAC-based geometric verification can reject a large amount of mismatched descriptors and does not rely on the point of initialization, FGR is less robust to poor initialization as done for the intermediate and complex experiments. For the reduced search space in the basic and intermediate experiments, FGR is able to achieve reasonable registration performance and therefore shows high potential to be used for such reduced search problems, when carefully modelling its robust cost function.

While the segmentation approach shows very good performance for reproducible segmentations, e.g., single-modality localization [15], we found that the considered parametrizations of Euclidean segmentation in combination with the considered descriptors did not generalize well between the modalities and could not deliver interesting results in the experiments. For the sake of clarity of the plots, we therefore

Configuration	M_b	M_i	M_c	D_b	D_i	D_c
FPFH	1	45	60	2	N/A	N/A
FPFH gr	N/A	N/A	N/A	N/A	N/A	N/A
FPFH FGR	1	50	54	3	N/A	N/A
FPFH FGR gr	60	60	43	3	N/A	N/A
SHOT	1	10	36	1	1	3
SHOT gr	1	1	36	1	1	7
SHOT FGR	1	35	35	1	12	N/A
SHOT FGR gr	1	29	28	1	12	N/A
FPFH seg	N/A	N/A	N/A	N/A	N/A	N/A
SHOT seg	N/A	N/A	N/A	N/A	N/A	N/A
ESF seg	N/A	N/A	N/A	N/A	N/A	N/A

TABLE II: Minimal number of LiDAR scans for successful global registration experiments. Here, M_b, M_i, M_c, denote the basic, intermediate, and complex experiment on the Montelibretti dataset, while D_b, D_i, and D_c denote the different experimental set-ups on the Dortmund data. The UGV travels on average $0.8m$ between two scans.

Fig. 5: Resulting global localization between *3D* UGV sub-map (red point-cloud) and global *3D* UAV map (coloured point-cloud). Green lines indicate the resulting descriptor matches associated to points in the two point-clouds. The data stems from the Dortmund indoor dataset and drawn from the complex experimental set-up.

only show their performance in the basic experiments in Fig. 6. Since the remainder of the matching algorithm is identical to the well performing descriptor matching, we believe that given a reliable segmentation, the approach has the potential to yield very good performance for the global registration case. However, a purely geometric ground removal and segmentation on the full UAV point-cloud that is comparable to the segmentation on the UGV map is a hard problem, especially for cluttered SaR scenarios.

1) Timings: Additionally to the evaluation of residuals, computation times are an important factor in the choice of algorithms in robotics. Table III lists the computational times of the four main components of the global registration algorithms when executed on an Intel i7-4600U CPU @ 2.10GHz. The computation times are reported per UGV LiDAR scan which has an acquisition time of $3s$ on the considered platform.

Although our focus was not on maximizing computational efficiency, all approaches can be performed in this time window and therewith in real-time. We are confident that they can be further improved to also yield faster processing times. With key-point detection times increasing with the amount of points in the point-cloud, the segmentation approach is the fastest in the first step. Descriptor extraction, is fastest for SHOT descriptors, also scaling with the amount of points. However, the largest contribution to the computational time has the descriptor matching which is longest for SHOT and low for FPFH. For the segmentation, we report the timings for the high-dimensional ESF features and achieve low timings, due to the compact representation of segments. Finally, RANSAC-based geometric consistency is slower than the optimization-based FGR.

E. Discussion

The global registration of *3D* UAV and UGV point-cloud data is a difficult problem. Based on our evaluation, the most general solution that we devise is a key-point descriptor matching algorithm using SHOT descriptors. In our evaluation, FPFH descriptors performed well, with large overlap between the maps, but failed for the more complex experiments, and showed to be sensitive to ground removal.

The segmentation showed to not deliver satisfying results, as it requires repeatable ground removal and segmentation, which could not be achieved in the considered configurations and scenarios. Key-point detection on the other hand performed well.

SHOT descriptor matching is computationally more expensive than FPFH due to high descriptor dimensionality, but showed best performance throughout. The processing time can be speeded up by removing the ground in environments that allow for reliable ground removal.

Finally FGR can yield additional speed up of the transformation estimation. Yet, when using FGR, the cost function must be carefully considered as the technique is prone to converge to local minima. While the RANSAC-based transformation estimation takes generally longer than FGR the robustness to local minima is greatly increased. Also, the additional computational time for RANSAC was negligible in our experiments.

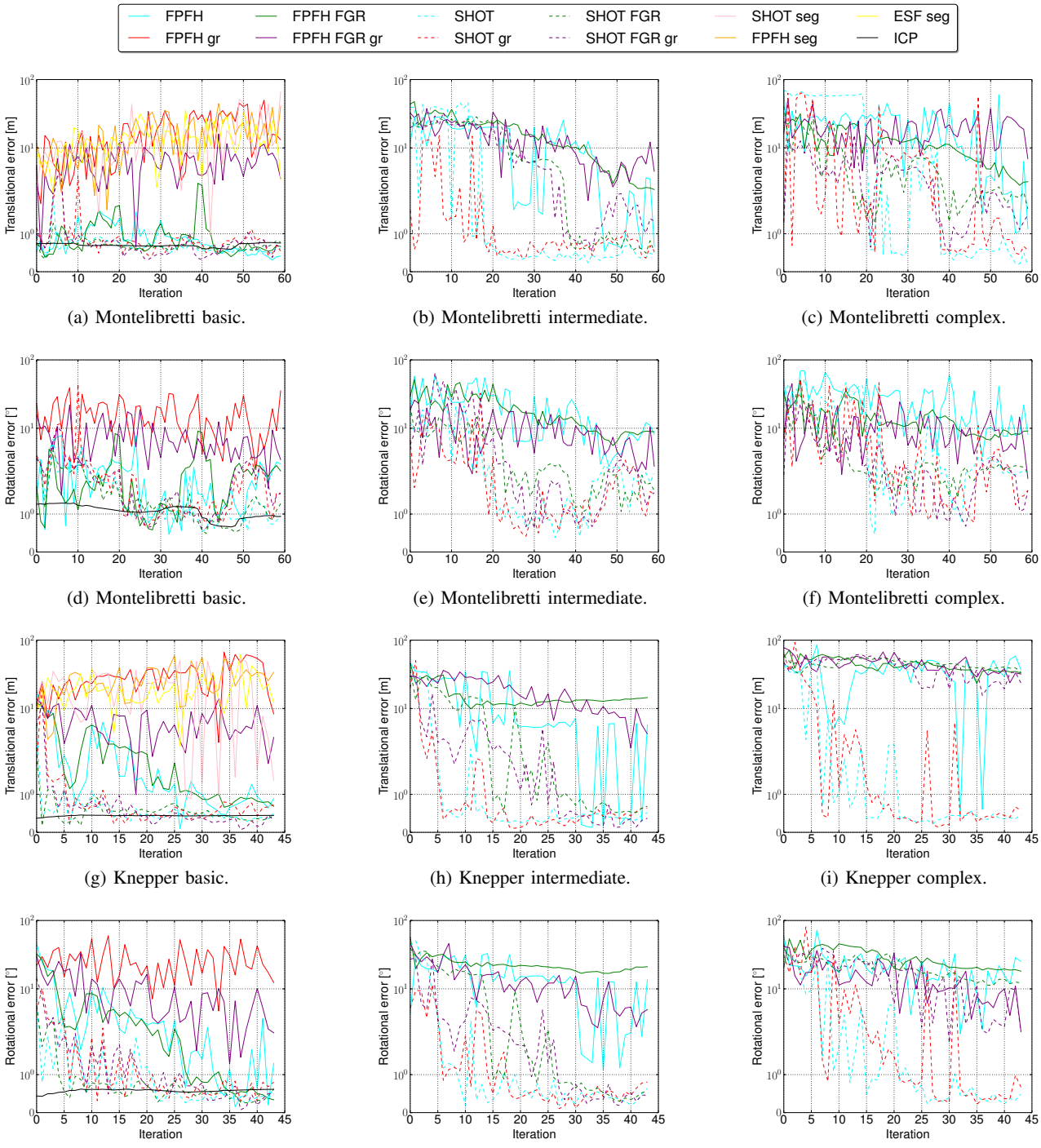
VI. Conclusion

This paper presented global registration algorithms for UGV and UAV point-clouds generated from heterogeneous sensors, i.e., LiDAR sensors for UGVs and cameras for UAVs, and drastically different view-points. The registration algorithm is based on geometrical descriptor matching. The approach was integrated with a full SaR robotic mapping system, bridging the gap between effective exploitation of UAV mapping data on UGVs. We evaluated several different *3D* descriptor-based registration techniques and identify the best performing approach for the problem of global point-cloud registration from heterogeneous sensors in SaR scenarios.

Future avenues of research could include point-cloud registration by using further informative cues than the geometrical information alone for data registration between the sensor modalities. This could benefit runtime and compactness of point-cloud description of the proposed algorithm.

VII. Acknowledgement

This work was supported by European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763.



(a) Montelibretti basic. (b) Montelibretti intermediate. (c) Montelibretti complex.

(d) Montelibretti basic. (e) Montelibretti intermediate. (f) Montelibretti complex.

(g) Knepper basic. (h) Knepper intermediate. (i) Knepper complex.

(j) Knepper basic. (k) Knepper intermediate. (l) Knepper complex.

Fig. 6: Translational and rotational error of plots for the Montelibretti (outdoor) and Dortmund (indoor) experiments on global registration. All errors are plotted over the number of iterations, i.e., the growing size of the UGV map. Here, we indicate experimental configurations as follows: *SHOT*, *FPFH* and *ESF* denote the used descriptors, an additional *FGR* denotes if we used the fast global optimization instead of the RANSAC-based outlier filtering, and we add *gr* for cases in which also ground removal was performed before descriptor extraction. The *ICP* solution is illustrated for the basic experiments. Plots a- c illustrate the translational errors for the Montelibretti experiment for the basic, indermediate, and complex set-up, while plots g- i show the translational errors for the respective experiments on the Dortmund data. Figures d- f illustrate the rotational errors for the Montelibretti experiment and Figures j- l the respective rotational errors on the Dortmund data.

Module	*FPFH*	*FPFH gr*	*FPFH FGR*	*FPFH FGR gr*	*SHOT*	*SHOT gr*	*SHOT FGR*	*SHOT FGR gr*	*Seg*
Key-point / Segmentation	35.97 ±13.17	22.92 ±13.30	39.01 ±16.04	24.57 ±14.90	33.1 ±12.16	18.97 ±11.07	35.76 ±13.30	23.06 ±14.87	9.32 ±5
Description	228.87 ±61.99	122.59 ±39.27	242.52 ±64.34	136.63 ±43.91	17.21 ±8.82	12.30 ±8.60	17.35 ±8.51	13.52 ±9.52	105.91 ±50.72
Matching	293.32 ±8.32	86.04 ±27.41	291.76 ±6.49	88.33 ±25.25	2897.94 ±44.01	1992.17 ±40.36	2937.60 ±65.13	2099.55 ±42.31	429.11 ±51.15
Geometric consistency / Optimization	15.91 ±11.54	22.72 ±17.28	2.38 ±1.09	2.05 ±1.05	7.55 ±6.11	8.24 ±6.79	2.23 ±0.99	1.82 ±0.94	0.80 ±1.60
Total	574.07	254.27	575.67	251.58	2926.01	2031.68	2992.94	2137.95	545.14

TABLE III: Mean computation times and standard deviations of the individual approaches in the complex Montelibretti experiment per LiDAR scan in ms as computed on a single core of an Intel i7-4600U CPU @ 2.10GHz.

REFERENCES

[1] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *JFR*, pp. 3–46, 2016.

[2] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida *et al.*, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *JFR*, pp. 832–841, 2012.

[3] B. Kim, M. Kaess, L. Fletcher, J. J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple Relative Pose Graphs for Robust Cooperative Mapping," in *ICRA*, 2010, pp. 3185–3192.

[4] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.

[5] R. Dubé, A. Gawel, J. Nieto, R. Siegwart, and C. Cadena, "Online multi-robot slam with 3d lidars: A full system," in *IROS*, 2017.

[6] S. Fuhrmann, F. Langguth, and M. Goesele, "Mve-a multi-view reconstruction environment." in *GCH*, 2014, pp. 11–18.

[7] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, pp. 1384–1397, 2006.

[8] J.-L. Blanco, J. González-Jiménez, and J.-A. Fernández-Madrigal, "A robust, multi-hypothesis approach to matching occupancy grid maps," *Robotica*, pp. 687–701, 2013.

[9] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Multiple robot simultaneous localization and mapping," in *IROS*, 2011, pp. 853–858.

[10] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Transactions on Robotics*, pp. 1309–1332, 2016.

[11] C. Forster, M. Pizzoli, and D. Scaramuzza, "Air-ground localization and map augmentation using monocular dense reconstruction," in *IROS*, 2013, pp. 3971–3978.

[12] T. Hinzmann, T. Stastny, G. Conte, P. Doherty, P. Rudol, M. Wzorek, E. Galceran, R. Siegwart, and I. Gilitschenski, *Collaborative 3D Reconstruction Using Heterogeneous UAVs: System and Experiments*. Cham: Springer International Publishing, 2016, pp. 43–56.

[13] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *IROS*, 2014, pp. 176–183.

[14] A. Gawel, T. Cieslewski, R. Dubé, M. Bosse, R. Siegwart, and J. Nieto, "Structure-based vision-laser matching," in *IROS*, 2016, pp. 182–188.

[15] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "*SegMatch*: Segment based place recognition in 3d point clouds," in *ICRA*, 2017.

[16] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *ECCV*, 2016, pp. 766–782.

[17] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *AuRo*, pp. 189–206, 2013.

[18] R. Dubé, A. Gawel, C. Cadena, R. Siegwart, L. Freda, and M. Gianni, "3d localization, mapping and path planning for search and rescue operations," in *SSRR*, 2016, pp. 272–273.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *IJRR*, pp. 216–235, 2012.

[20] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *ICCV*, 2009, pp. 689–696.

[21] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *ICRA*, 2009, pp. 3212–3217.

[22] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010, pp. 356–369.

[23] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *ROBIO*, 2011, pp. 2987–2992.

[24] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration." *VISAPP*, pp. 331–340, 2009.

X-View: Graph-Based Semantic Multi-View Localization

Abel Gawel*, Carlo Del Don*, Roland Siegwart, Juan Nieto and Cesar Cadena

***Abstract*—Global registration of multi-view robot data is a challenging task. Appearance-based global localization approaches often fail under drastic view-point changes, as representations have limited view-point invariance. This work is based on the idea that human-made environments contain rich semantics which can be used to disambiguate global localization. Here, we present *X-View*, a Multi-View Semantic Global Localization system. *X-View* leverages semantic graph descriptor matching for global localization, enabling localization under drastically different view-points. While the approach is general in terms of the semantic input data, we present and evaluate an implementation on visual data. We demonstrate the system in experiments on the publicly available *SYNTHIA* dataset, on a realistic urban dataset recorded with a simulator, and on real-world StreetView data. Our findings show that *X-View* is able to globally localize aerial-to-ground, and ground-to-ground robot data of drastically different view-points. Our approach achieves an accuracy of up to $85\,\%$ on global localizations in the multi-view case, while the benchmarked baseline appearance-based methods reach up to $75\,\%$.**

***Index Terms*—Localization, Semantic Scene Understanding, Mapping**

I. INTRODUCTION

GLOBAL localization between heterogeneous robots is a difficult problem for classic place-recognition approaches. Visual appearance-based approaches such as [1, 2] are currently among the most effective methods for re-localization. However, they tend to significantly degrade with appearance changes due to different time, weather, season, and also view-point [3, 4]. In addition, when using different sensor modalities, the key-point extraction becomes an issue as they are generated from different physical and geometrical properties, for instance intensity gradients in images vs. high-curvature regions in point clouds.

Relying on geometrical information, directly from the measurements or from a reconstruction algorithm, on the other hand shows stronger robustness on view-point changes, seasonal changes, and different sensor modalities. However, geometrical approaches typically do not scale well to very large environments, and it remains questionable if very strong view-point changes can be compensated while maintaining only a limited overlap between the localization query and database [5, 6].

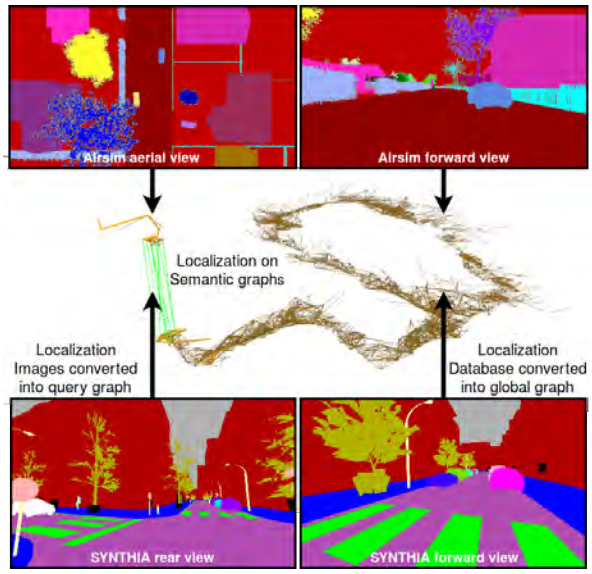


Figure 1: *X-View* globally localizes data of drastically different view-points using graph representations of semantic information. Here, samples of the experimental data is shown, i.e., semantically segmented images from the publicly available *SYNTHIA* and the *Airsim* datasets. The localization target graph is built from data of one view-point (*right images*), while the query graph is built from sequences of another view-point (*left images*). *X-View* efficiently localizes the query graph in the target graph.

Another avenue to address appearance and view-point changes are Convolutional Neural Network (CNN) architectures for place recognition [4, 7]. While these methods show strong performance under appearance changes, their performance is still to be investigated under extreme view-point variations.

Recently, topological approaches to global localization regained interest as a way to efficiently encode relations between multiple local visual features [8, 9]. On the other hand, the computer vision community has made great progress in semantic segmentation and classification, resulting in capable tools for extracting semantics from visual and depth data [10–12].

Based on the hypothesis that semantics can help to mitigate the effects of appearance changes, we present *X-View*, a novel approach for global localization based on building graphs of semantics. *X-View* introduces graph descriptors that efficiently represent unique topologies of semantic objects. These can

Manuscript received: September, 10, 2017; Revised December, 9, 2017; Accepted January, 16, 2018.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763 and by the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation.

* The authors contributed equally to this work.

Authors are with the Autonomous Systems Lab, ETH Zurich. gawela@ethz.ch, deldonc@student.ethz.ch, {rsiegwart, nietoj, cesarc}@ethz.ch

Digital Object Identifier (DOI): see top of this page.

be matched in much lower computational effort, therefore not suffering under the need for exhaustive sub-graph matching [13].

By using semantics as an abstraction between robot view-points, we achieve invariances to strong view-point changes, outperforming CNN-based techniques on RGB data. Furthermore, with semantics understanding of the scene, unwanted elements, such as moving objects can naturally be excluded from the localization. We evaluate our global localization algorithm on publicly available datasets of real and simulated urban outdoor environments, and report our findings on localizing under strong view-point changes. Specifically, this paper presents the following contributions:

- A novel graph representation for semantic topologies.
- Introduction of a graph descriptor based on random walks that can be efficiently matched with established matching methods.
- A full pipeline to process semantically segmented images into global localizations.
- Open source implementation of the *X-View* algorithm[1].
- Experimental evaluation on publicly available datasets.

The remainder of this paper is structured as follows: Sec. II reviews the related work on global localization, followed by the presentation of the *X-View* system in Sec. III. We present our experimental evaluation in Sec. IV and conclude our findings in Sec. V.

II. RELATED WORK

In this section we review the current state-of-the-art in multi-robot global localization in relation to our proposed system.

A common approach to global localization is visual feature matching. A large amount of approaches have been proposed in the last decade, giving reliable performance under perceptually similar conditions [1–3]. Several extensions have been proposed to overcome perceptually difficult situations, such as seasonal changes [14, 15], daytime changes [4, 16], or varying view-points using CNN landmarks [7, 17]. However, drastic view-point invariance, e.g., between views of aerial and ground robots continues to be a challenging problem for appearance-based techniques.

In our previous work, we demonstrated effective *3D* heterogeneous map merging approaches between different view-points from camera and LiDAR data, based on overlapping *3D* structural descriptors [5, 6]. However, 3D reconstructions are still strongly view-point dependent. While these techniques do not rely on specific semantic information of the scenes, the scaling to large environments has not yet been investigated, and computational time is outside real-time performance with large maps.

Other approaches to global localization are based on topological mapping [18, 19]. Here, maps are represented as graphs $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ of unique vertices $\boldsymbol{V}$ and edges $\boldsymbol{E}$ encoding relationships between vertices. While these works focus on graph merging by exhaustive vertex matching on small graphs, they do not consider graph extraction from sensory data or ambiguous vertices. Furthermore, the computationally expensive matching does not scale to larger graph comparisons.

With the recent advances in learning-based semantic extraction methods, using semantics for localization is a promising avenue [20–22]. In [21, 22] the authors focus on the *data association* problem for semantic localization using Expectation Maximization (EM) and the formulation of the pose estimation problem for semantic constraints as an error minimization. The semantic extraction is based on a standard object detector from visual key-points.

Stumm et al. [8] propose to use graph kernels for place recognition on visual key-point descriptors. Graph kernels are used to project image-wise covisibility graphs into a feature space. The authors show that graph descriptions can help localization performance as to efficiently cluster multiple descriptors meaningfully. However, the use of large densely connected graphs sets limitations to the choice of graph representation. Motivated, by these findings, we propose to use graph descriptors on sparse semantic graphs for global localization.

III. X-VIEW

In this section, we present our Graph-Based Multi-View Semantic Global Localization system, coined *X-View*. It leverages graph extraction from semantic input data and graph matching using graph descriptors. Fig. 2 illustrates the architecture of the proposed global localization algorithm, focusing on the graph representation and matching of query semantic input data to a global graph. The localization target map is represented as the global graph. *X-View* is designed to operate on any given odometry estimation system and semantic input cue. However, for the sake of clarity, we present our system as implemented for semantically segmented images, but it is not limited to it.

A. System input

We use semantically segmented images containing pixel-wise semantic classification as input to the localization algorithm. These segmentations can be achieved using a semantic segmentation method, such as [11, 12]. Also instance-wise segmentation, i.e., unique identifiers for separating overlapping objects of same class in the image space can be considered for improved segmentation, but is not strictly necessary for the approach to work. Furthermore, we assume the estimate of an external odometry system. Finally, we also consider a database semantic graph $\boldsymbol{G}_{db}$, as it could have been built and described on a previous run of our graph building algorithm as presented in the next sub-sections.

B. Graph extraction and assembly

In this step, we convert a sequence of semantic images $\boldsymbol{I}_q$ into a query graph $\boldsymbol{G}_q$. We extract blobs of connected regions, i.e., regions of the same class label l_j in each image. Since semantically segmented images often show noisy partitioning of the observed scene (holes, disconnected edges and invalid labels on edges), we smooth them by dilating and eroding the

[1]https://github.com/ethz-asl/x-view

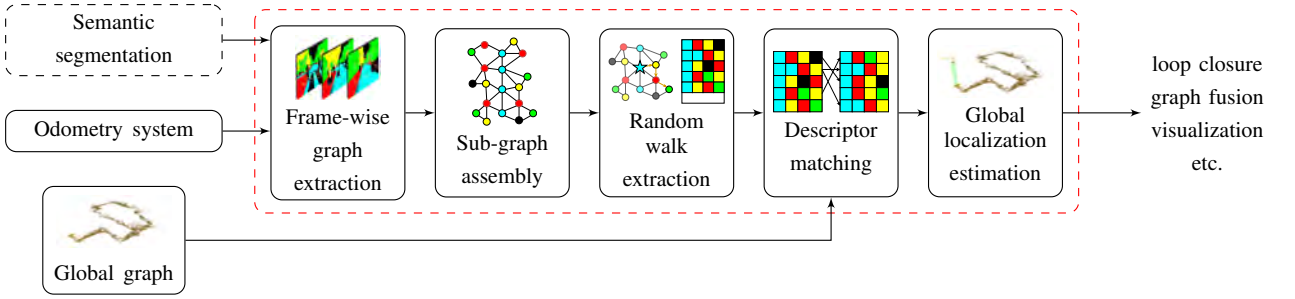


Figure 2: *X-View* global localization system overview. The inputs to the system are semantically segmented frames (e.g., from RGB images) and the global graph $\boldsymbol{G}_{db}$. First, a local graph is extracted from the new segmentation. Then, the sub-graph $\boldsymbol{G}_q$ is assembled and random walk descriptors are computed on each node of $\boldsymbol{G}_q$. The system matches the sub-graph random walk descriptors to $\boldsymbol{G}_{db}$, e.g., recorded from a different view-point. Finally, the matches are transferred to the localization back-end module to estimate the relative localization between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$. Consecutively, the relative localization can be used for various purposes such as loop closure, fusing $\boldsymbol{G}_q$ into $\boldsymbol{G}_{db}$ or for visualization.

boundaries of each blob. We furthermore reject blobs smaller than a minimum pixel count to be included in the graph, to mitigate the effect of minor segments. This process removes unwanted noise in the semantically segmented images. The magnitude of this operation is 4 pixels, and has a minor effect on the segmentation result. However, it ensures clean boundaries between semantic segments. Furthermore, the center location $\boldsymbol{p}_j$ of the blobs are extracted and stored alongside the blob labels as vertices $\mathbf{v}_j = \{\boldsymbol{l}_j, \boldsymbol{p}_j\}$. In the case that also instance-wise segmentation is available, it can be considered in the blob extraction step, otherwise the extraction operates only on a class basis.

The undirected edges $\boldsymbol{e}_j$ between vertices are formed when fulfilling a proximity requirement, which can be either in image- or *3D*-space. In the case of image-space, we assume images to be in a temporal sequence to grow graphs over several frames of input data. However, this is not required in the *3D* case.

Using a depth channel or the depth estimation from, e.g., optical flow, the neighborhood can be formed in *3D*-space, using the *3D* locations of the image blobs to compute a Euclidean distance. The process is illustrated for image data in Fig. 3 (top). Then, several image-wise graphs are merged into $\boldsymbol{G}_q$ by connecting vertices of consecutive images using their Euclidean distance, see Fig. 3. To prevent duplicate vertices of the same semantic instance, close instances in $\boldsymbol{G}_q$ are merged into a single vertex, at the location of the vertices' first observation. The strategy of merging vertices into their first observation location is further motivated by the structure of *continuous* semantic entities, such as streets. This strategy leads to evenly spaced creation of *continuous* entities' vertices in $\boldsymbol{G}_q$.

C. Descriptors

X-View is based on the idea that semantic graphs hold high descriptive power, and that localizing a sub-graph in a database graph can yield good localization results. However, since sub-graph matching is an NP-complete problem [13], a different regime is required to perform the graph registration under real-time constraints, i.e., in the order of seconds for typical robotic applications. In this work, we extract random walk descriptors for every node of the graph [23], and match them in a subsequent step. This has the advantage that the descriptors

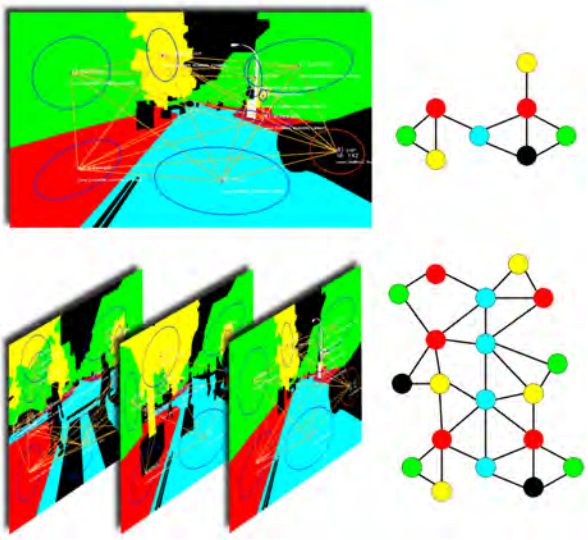


Figure 3: Extraction of semantic graphs from one image (top) and a sequence of images (bottom). Vertices are merged and connected from sequences of input data. Note that we omitted some vertices and edges in the sample graphs on the right side for visualization purposes and reduced the graph to a planar visualization, whereas the semantic graphs in our system are connected in *3D*-space. The ellipses around each vertex were added for visualization and represent a scaled fitted ellipse on a semantic instance of the segmentation image.

can be extracted and matched in constant or linear time, given a static or growing database-graph, respectively.

Each vertex descriptor is an $n \times m$ matrix consisting of n random walks of depth m. Each of the random walks originates at the base vertex $\mathbf{v}_j$ and stores the class labels of the visited vertices. Walk strategies, such as preventing from immediate returns to the vertex that was visited in the last step, and exclusion of duplicate random walks can be applied to facilitate expressiveness of the random walk descriptors. The process of random walk descriptor extraction is illustrated in Fig. 4.

D. Descriptor Matching

After both $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ are created, we find associations between vertices in the query graph and the ones in the database graph by computing a similarity score between the corresponding graph descriptors. The similarity measure is computed by matching each row of the semantic descriptor

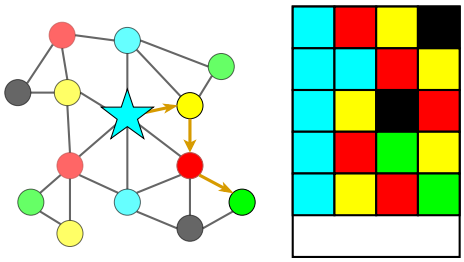


Figure 4: Schematic representation of the random walk extraction. (Left) From a seed vertex, cyan star, the random walker explores its neighborhood. This results in the descriptor of n random walks of depth m (here, $m = 4$). The highlighted path corresponds to the last line of the descriptor on the right. (Right) Each line of the descriptor starts with the seed vertex label and continues with the class labels of the visited vertices.

of the query vertex to the descriptor of the database vertex. The number of identical random walks on the two descriptors reflects the similarity score s, which is normalized between 0 and 1. In a second step, the k matches with highest similarity score are selected for estimating the location of the query graph inside the database map.

E. Localization Back-End

The matching between query graph and global graph, the robot-to-vertex observations, and the robot odometry measurements result in constraints $\boldsymbol{\theta}_i \subseteq \boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i)$ on the vertex positions $\boldsymbol{p}_i$ and robot poses $\boldsymbol{c}_i$ with $\boldsymbol{\theta}_i = \boldsymbol{e}_i^T \boldsymbol{\Omega}_i \boldsymbol{e}_i$, the measurement errors $\boldsymbol{e}_i$, and associated information matrix $\boldsymbol{\Omega}_i$. Specifically these three types of constraints are denoted as $\boldsymbol{\Theta}_{\mathbf{M}}(\boldsymbol{p}_i), \boldsymbol{\Theta}_{\mathbf{V}}(\boldsymbol{p}_i, \boldsymbol{c}_i)$, and $\boldsymbol{\Theta}_O(\boldsymbol{c}_i)$ respectively. The matching constraints $\boldsymbol{\Theta}_{\mathbf{M}}(\boldsymbol{p}_i)$ stem from the semantic descriptor matching of the previous step, while the robot odometry constraints $\boldsymbol{\Theta}_O(\boldsymbol{c}_i)$ are created using the robots estimated odometry between consecutive robot poses associated to the localization graph. The robot-to-vertex constraints encode the transformation between each robot-to-vertex observation. Using these constraints, we compute a Maximum a Posteriori (MAP) estimate of the robot pose $\boldsymbol{c}_i$ by minimizing a negative log-posterior $\mathbf{E} = \sum \boldsymbol{\theta}_i$, i.e.,

$$\boldsymbol{c}_i^* = \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \sum \boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i) \tag{1}$$

with $\boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i) = \{\boldsymbol{\Theta}_M(\boldsymbol{p}_i), \boldsymbol{\Theta}_V(\boldsymbol{p}_i, \boldsymbol{c}_i), \boldsymbol{\Theta}_V(\boldsymbol{p}_i)\}$ This optimization is carried out by a non-linear Gauss-Newton optimizer. Optionally, the algorithm also allows to reject matching constraints in a sample consensus manner, using RANSAC on all constraints between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$, excluding the specific constraints from the optimization objective. We initialize the robot position at the mean location of all matching vertices' locations from $\boldsymbol{G}_{db}$.

IV. EXPERIMENTS

We evaluate our approach on two different synthetic outdoor datasets with forward to rear view, and forward to aerial view, and one real world outdoor dataset with forward to rear view. In this section, we present the experimental set-up, the results, and a discussion.

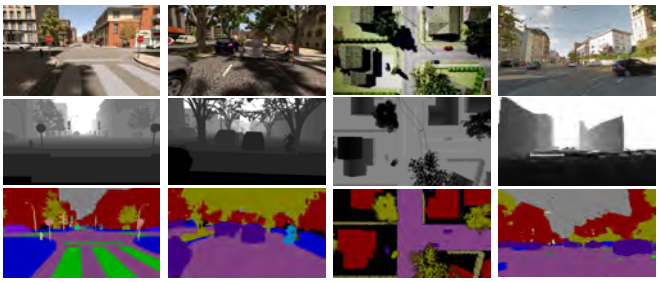


Figure 5: Sample images from the datasets used in the experiments: (top) RGB image, (middle) Depth image, (bottom) Semantic segmentation. (left) *SYNTHIA* with perfect semantic segmentation, (middle left) *SYNTHIA* with *AdapNet* semantic segmentation, (middle right) *Airsim* with perfect semantic segmentation, (right) *StreetView* with *SegNet* semantic segmentation.

A. Datasets

The first of the used datasets is the public *SYNTHIA* dataset [24]. It consists of several sequences of simulated sensor data from a car travelling in different dynamic environments and under varying conditions, e.g., weather and daytime. The sensor data provides RGB, depth and pixel-wise semantic classification for 8 cameras, with always 2 cameras facing forward, left, backwards and right respectively. The segmentation provides 13 different semantic classes which are labelled class-wise. Additionally, dynamic objects, such as pedestrians and cars are also labelled instance-wise. We use sequence 4, which features a town-like environment. The total travelled distance is $970\,m$.

In the absence of suitable public aerial-ground semantic localization datasets, we use the photo-realistic *Airsim* framework [25] to generate a simulated rural environment[2]. This environment is explored with a top-down viewing Unmanned Aerial Vehicle (UAV) and a car traversing the streets with forward-facing sensors. Both views provide RGB, depth and pixel-wise semantic classification data in 13 different classes with instance-wise labelling. Furthermore, both trajectories are overlapping with only an offset in $\mathbf{z}$-direction and have a length of $500\,m$ each. Please note that we used a pre-built environment, i.e., the objects in the environment have not specifically been placed for enhanced performance.

Finally, we evaluate the system on a dataset gathered from *Google StreetView* imagery. The RGB and depth data of a straight $750\,m$ stretch of Weinbergstrasse in Zurich are extracted via the *Google Maps API*[3]. Analogously to the *SYNTHIA* dataset, we use forward and backward facing camera views.

While the travelled distance between two image locations in the *Airsim* dataset is always $1\,m$, it varies between $0\,m$ to $1\,m$ in the *SYNTHIA* dataset, and is approximately $10\,m$ between two frames in the *StreetView* dataset. Sample images of all datasets are depicted in Fig. 5.

Our approach relies on semantic representations of scenes. While we do not propose contributions on semantic extraction from raw sensor data, recent advances on semantic segmentation show ever increasing accuracies on visual and depth

[2]http://robotics.ethz.ch/~asl-datasets/x-view/
[3]https://goo.gl/iBniJ9

data [10–12, 26]. We therefore evaluate the performance on *SYNTHIA* both using semantic segmentation with *AdapNet* [11], and the ground truth as provided by the dataset. On the *Airsim* data, we only use the segmentation from the dataset, and on the *StreetView* dataset, we use semantic segmentation with *SegNet* [12].

B. Experimental Setup

We evaluate the core components of *X-View* in different experimental settings. In all experiments, we evaluate *X-View* on overlapping trajectories and the provided depth and segmentation images of the data. First, we focus our evaluation of the different graph settings on the *SYNTHIA* dataset. We then perform a comparative analysis on *SYNTHIA*, *Airsim*, and *StreetView*.

In *SYNTHIA*, we use the left forward camera for building a database map and then use the left backward camera for localization. Furthermore, we use 8 semantic classes of *SYNTHIA*: *building*, *street*, *sidewalk*, *fence*, *vegetation*, *pole*, *car*, and *sign*, and reject the remaining four classes: *sky*, *pedestrian*, *cyclist*, *lanemarking*. The *AdapNet* semantic segmentation model is trained on other sequences of the *SYNTHIA* dataset with different seasons and weather conditions.

Analogously, we use the forward-view of the car in the *Airsim* dataset to build the database map and then localize the UAV based on a downward-looking camera. Here we use 6 classes *(street, building, car, fence, hedge, tree)* and reject the remaining from insertion into the graph *(powerline, pool, sign, wall, bench, rock)*, as these are usually only visible by one of the robots, or their scale is too small to be reliably detected from the aerial robot.

Finally, in the *StreetView* data, we use the forward view to build the database and localize using the rear facing view. Out of the 12 classes that we extract using the pre-trained *SegNet* model[4], we use five, i.e., *(road, sidewalk, vegetation, fence, car)*, and reject the remaining as these are either dynamic *(pedestrian, cyclist)*, unreliably segmented *(pole, road sign, road marking)*, or omni-present in the dataset *(building, sky)*.

We build the graphs from consecutive frames in all experiments, and use the *3D* information to connect and merge vertices and edges, as described in III-B. The difference between graph construction in image- and *3D*-space is evaluated in a separate experiment. No assumptions are made on the prior alignment between the data. The ground-truth alignment is solely used for performance evaluation.

C. Localization performance

We generate the PR of the localization based on two thresholds. The localization threshold t_L is applied on the distance between the estimated robot position $\boldsymbol{c}_i^*$ and the ground truth position $\boldsymbol{c}_{gt}$. It is set as *true*, if the distance between $\boldsymbol{c}_i^*$ and $\boldsymbol{c}_{gt}$ is smaller than t_L, i.e., $\|\boldsymbol{c}_i^* - \boldsymbol{c}_{gt}\| \leq t_L$, and to *false* for $\|\boldsymbol{c}_i^* - \boldsymbol{c}_{gt}\| > t_L$. The margin t_L on the locations is required, since $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ do not create vertices in the exact same spot. The same node can be off by up to twice the distance that we use for merging vertices in a graph. Here, we use $t_L = 20\,m$ for *SYNTHIA* and *StreetView*, and $t_L = 30\,m$ for *Airsim*. For the PR curves, we vary the consistency threshold t_c that is applied on the RANSAC-based rejection, i.e., the acceptable deviation from the consensus transformation between query and database graph vertices. The localization estimation yields a positive vote for an estimated consensus value s of $s \leq t_c$ and a negative vote otherwise.

Firstly, we evaluate the effect of different options on the description and matching using the random walk descriptors (i.e., random walk parameters, graph coarseness, number of query frames, dynamics classes, graph edge construction technique, and seasonal changes) as described in Sec. III-B - III-D. To illustrate the contrast to appearance-based methods, we also present results on two visual place recognition techniques based on BoW, as implemented by Gálvez-López and Tardos [2], and NetVLAD [4] on the datasets' RGB data. To generate the PR of the reference techniques, we vary a threshold on the inverse similarity score for BoW, and a threshold on the matching residuals of NetVLAD.

Furthermore, we show the performance of the full global localization algorithm on the operating point taken from the PR curves. Our performance metric is defined as the percentage of correct localizations over the Euclidean distance between $\boldsymbol{c}_i^*$ and $\boldsymbol{c}_{gt}$. As for BoW and NetVLAD, we take localization as the best matching image. The localization error is then computed as the Euclidean distance between associated positions of the matched image and the ground truth image. To improve performance of the appearance-based methods, we select the operating points with high performances, i.e., high precisions in the PR curves.

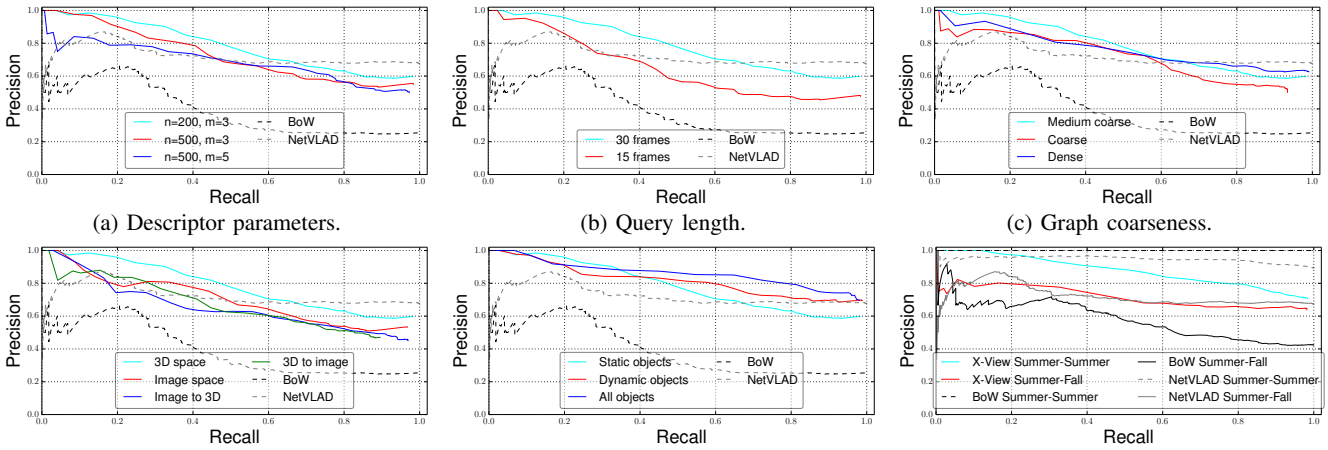
D. Results

While we illustrate the effects of different attributes of *X-View* in Fig. 6 as evaluated on *SYNTHIA*, we then also show a comparison on all datasets in Fig. 7.

Fig. 6a depicts the effect of varying the random walk descriptors on the graph. Here, a descriptor size with number of random walks $n = 200$ and walk depth m between $3 - 5$, depending on the size of $\boldsymbol{G}_q$ perform best. Both decreasing n or increasing m leads to a decrease in performance. These findings are expected, considering query graph sizes ranging between $20 - 40$ vertices. Under these conditions, the graph can be well explored with the above settings. Descriptors with larger walk depth m significantly diverge between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$, as the random walk reaches the size limits of $\boldsymbol{G}_q$ and continues exploring already visited vertices, while it is possible to continue exploring $\boldsymbol{G}_{db}$ to greater depth.

Secondly, Fig. 6b presents PR-curves for different sizes of $\boldsymbol{G}_q$, i.e., different numbers of frames used for the construction of $\boldsymbol{G}_q$. An increase in the query graph size leads to a considerable increase of the localization performance. Also this effect is expected as $\boldsymbol{G}_q$ contains more vertices, forming more unique descriptors. However, it is also desirable to keep the size of $\boldsymbol{G}_q$ limited, as a growing query graph size requires larger overlap between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$. Furthermore, the computational time for descriptor calculation and matching grows with increased query graph size.

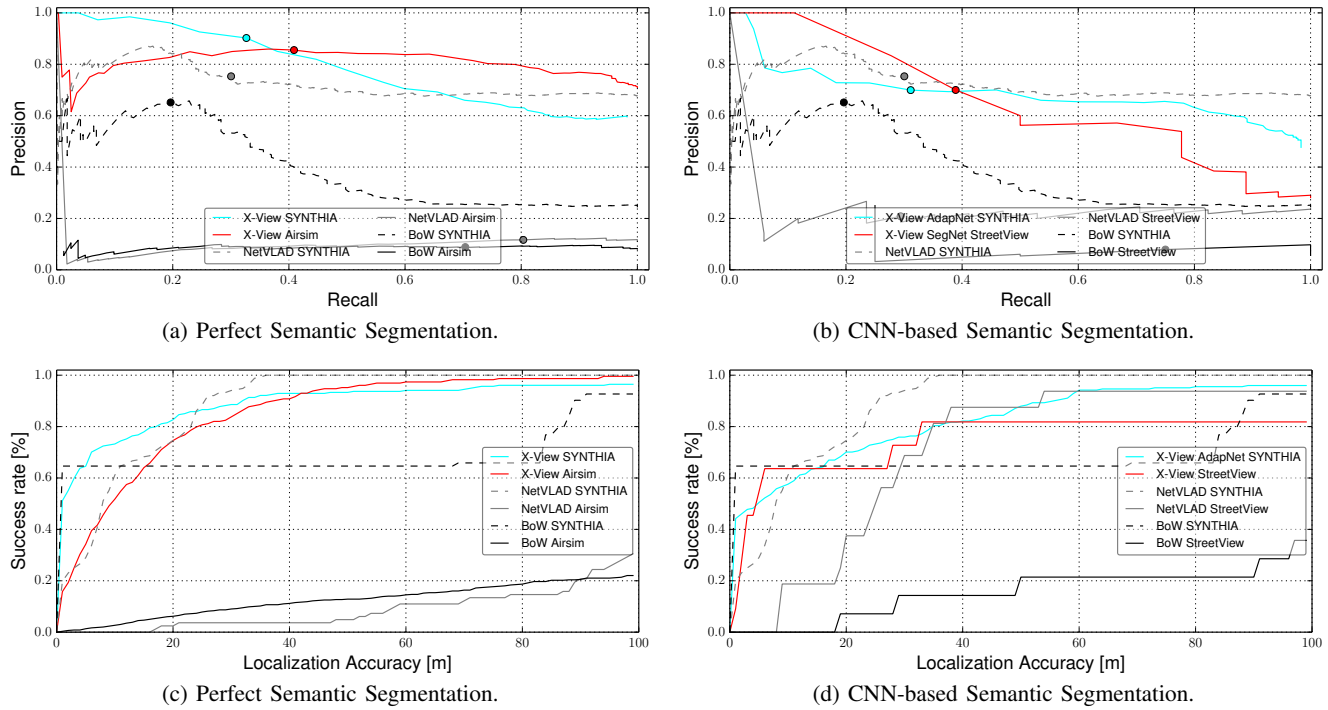
[4]goo.gl/EyReyn



(a) Descriptor parameters. (b) Query length. (c) Graph coarseness.

(d) Construction type. (e) Number of Semantic classes. (f) Seasonal changes on same view.

Figure 6: PR curves for localization of the rear view semantic images against a database graph built from the forward view on the *SYNTHIA* dataset (except (f)). For all plots we accept a localization if it falls within a distance of $20\,m$ from the ground-truth robot position. This threshold corresponds to the value up to which query graph vertices of the same semantic instance can be off from their corresponding location in the database graph, caused by the graph construction technique. (a) illustrates the effect of different descriptor settings on the localization performance. (b) shows the effect of increasing the amount of frames used for query graph construction, while (c) depicts the effect of using coarser graphs, i.e., a large distance in which we merge vertices of same class label. In (d) we compare the extraction methods in image-, and *3D*-space and in (e) the effect of including all semantic objects against including a subset of semantic classes. Lastly, in (f), we evaluate the localization performance on a configuration with the right frontal camera as query and the left frontal camera for the database, under the effect of seasonal changes. In contrast to the other plots where we use the ground truth, we use semantic segmentation with *AdapNet* on the data. The appearance-based techniques used are visual BoW [2] and NetVLAD [4].



(a) Perfect Semantic Segmentation. (b) CNN-based Semantic Segmentation.

(c) Perfect Semantic Segmentation. (d) CNN-based Semantic Segmentation.

Figure 7: Localization performance of *X-View* on the *SYNTHIA*, *Airsim*, and the *StreetView* data compared to the appearance-based methods [2, 4]. The operation points are chosen according to the respective PR curves in (a) and (b), indicated as dots. (c) illustrates the performance on perfectly semantically segmented data on *SYNTHIA*, and *Airsim*. (d) shows the system's performance on the *SYNTHIA*, and *StreetView* datasets using CNN-based semantic segmentation.

Thirdly, Fig. 6c shows the impact of increased graph coarseness, i.e., larger distances of merging vertices. Here, the coarseness cannot be arbitrarily scaled to low or high values, as it leads to either over- or under-segmented graphs. Our best performing results were obtained with a vertex merging distance of $10\,m$ for the *SYNTHIA* dataset, and $15\,m$ for *Airsim* and *StreetView* datasets, respectively.

Fourthly, Fig. 6d illustrates the effect of graph extraction in either image- or *3D*-space. The extraction in *3D*-space, taking advantage of the depth information as described in Sec. III-B shows superior performance. However, *X-View* still performs well when localizing a graph built in one space against a graph built in the other.

Fifthly, Fig. 6e explores the inclusion of different object classes. The configurations are: Only static object classes, static object classes plus dynamic object classes, and all object

Module	*SYNTHIA*	*Airsim*
Blob extraction	2.73 ± 0.65	1.76 ± 0.26
Construction of $\boldsymbol{G}_q$	337.39 ± 92.81	257.40 ± 28.30
Random Walks Generation	1.38 ± 0.82	1.07 ± 0.56
Matching $\boldsymbol{G}_q$ to $\boldsymbol{G}_{db}$	7.30 ± 4.51	4.33 ± 1.25
Localization Back-End	22.50 ± 9.71	5.15 ± 0.63
Total	371.3 ± 108.5	269.71 ± 31.0

Table I: Timing results in ms, reporting the means and standard deviations per frame on the best performing configurations on *SYNTHIA* and *Airsim*. The timings were computed on a single core of an Intel Xeon E3-1226 CPU @ 3.30GHz.

classes. Here, the results are not conclusive on the *SYNTHIA* dataset and more evaluations will be needed in the future.

Lastly, Fig. 6f shows *X-View*'s performance under seasonal change. We compare the performance of localizing the query graph built from the right forward facing camera of one season in the database graph built from the left forward facing camera of another season. Here, we consider the summer and fall sequences of *SYNTHIA*. The BoW-based techniques perform well in this scenario if the seasonal conditions are equal. However, its performance drastically drops for inter-season localization, while *X-View*, and NetVLAD suffer much less under the seasonal change.

The evaluation using PR-curves, and success rates over the localization error is depicted in Fig. 7. *X-View* has higher success rate in multi-view experiments than the appearance-based techniques on both synthetic datasets at our achievable accuracy of $20\,m$ for *SYNTHIA* and $30\,m$ on *Airsim* and using perfect semantic segmentation inputs as depicted in Fig. 7c. These accuracies are considered successful as node locations between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ can differ by twice the merging distance with our current graph merging strategy. On the considered operation point of the PR curve, *X-View* achieves a localization accuracy of $85\,\%$ within $30\,m$ on *Airsim*, and $85\,\%$ on *SYNTHIA* within $20\,m$.

Furthermore, *X-View* expresses comparable or better performance for multi-view localization than the appearance-based techniques using CNN-based semantic segmentation on the *SYNTHIA*, and *StreetView* datasets respectively. Here we consider successful localizations within $20\,m$ for both datasets. The achieved accuracies on the chosen operation points are $70\,\%$ on *SYNTHIA*, and $65\,\%$ on *StreetView*.

Finally, we also report timings of the individual components of our system in Table I. Here, the construction of $\boldsymbol{G}_q$ has by far the largest contribution, due to iteratively matching and merging frames into $\boldsymbol{G}_q$. As the graphs in *SYNTHIA* consider more classes and smaller merging distances, these generally contain more vertices and therefore longer computational times.

E. Discussion

Global registration of multi-view data is a difficult problem where traditional appearance based techniques fail. Semantic graph representations can provide significantly better localization performance under these difficult perceptual conditions. We furthermore give insights how different parameters, choices, and inputs' qualities affect the system's performance. Our results obtained with *X-View* show a better localization performance than appearance-based methods, such as BoW and NetVLAD.

During our experiments, we observed that some of the parameters are dependent on each other. Intuitively, the coarseness of the graph has an effect on the random walk descriptors as a coarser graph contains fewer vertices and therefore deeper random walks show decreasing performance as $\boldsymbol{G}_q$ can be explored with short random walks. On the other hand, an increasing amount of frames used for localization has the reverse effect on the descriptor depth as $\boldsymbol{G}_q$ potentially contains more vertices, and deeper random walks do not show a performance drop as they do for smaller query graphs.

Also the success rate curves indicate that *X-View* outperforms the appearance based methods particularly in the presence of strong view-point changes. While the appearance-based methods fail to produce interesting results for the *Airsim* dataset, they have a moderate to good amount of successful localizations on *SYNTHIA* and *StreetView*. On the other hand, *X-View* has generally higher localization performance and does not show a strong drop in performance among datasets. While computational efficiency has not been the main focus of our research, the achieved timings are close to the typical requirements for robotic applications.

Finally, we performed experiments both using ground truth semantic segmentation inputs, and CNN-based semantic segmentation. The performance with semantic segmentation using *AdapNet* [11] shows to be close to the achievable performance with ground truth segmentation on *SYNTHIA*. Using the *SegNet* [12] semantic segmentation on real image data from *StreetView* demonstrates the effectiveness of our algorithm's full pipeline on real data, resulting in better performance than the best reference algorithm. Despite the high performance, our system still receives a moderate amount of false localizations, which is due to similar sub-graphs at different locations, and we hope to mitigate this effect by including it into a full SLAM system in the future.

Furthermore, *3D* locations of the vertices are presently positioned at the blob centers of their first observation. We expect a more precise positioning technique to further disambiguate the associations between graphs.

V. CONCLUSIONS

In this paper we presented *X-View*, a multi-view global localization algorithm leveraging semantic graph descriptor matching. The approach was evaluated on one real-world and two simulated urban outdoor datasets with drastically different view-points. Our results show the potential of using graph representations of semantics for large-scale robotic global localization tasks. Alongside further advantages, such as compact representation and real-time-capability, the presented method is a step towards view-point invariant localization.

Our current research includes the investigation of more sophisticated graph construction methods, the integration of *X-View* with a full SLAM system to generate loop closures, and learning-based class selection for discriminative representations.

REFERENCES

[1] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[3] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[5] A. Gawel, T. Cieslewski, R. Dubé, M. Bosse, R. Siegwart, and J. Nieto, "Structure-based vision-laser matching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 182–188.

[6] A. Gawel, R. Dubé, H. Surmann, J. Nieto, R. Siegwart, and C. Cadena, "3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation," in *IEEE International Symposium on Safety, Security*, 2017.

[7] Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," 2017.

[8] E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter, and R. Siegwart, "Robust visual place recognition with graph kernels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4535–4544.

[9] Y. Su, F. Han, R. E. Harang, and X. Yan, "A fast kernel for attributed graphs," in *SIAM International Conference on Data Mining*, 2016, pp. 486–494.

[10] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[11] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4644–4651.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[13] S. A. Cook, "The complexity of theorem-proving procedures," in *ACM symposium on Theory of computing*. ACM, 1971, pp. 151–158.

[14] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.

[15] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4830–4836.

[16] M. Bürki, I. Gilitschenski, E. Stumm, R. Siegwart, and J. Nieto, "Appearance-based landmark selection for efficient long-term visual localization," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4137–4143.

[17] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," *Robotics: Science and Systems*, 2015.

[18] W. H. Huang and K. R. Beevers, "Topological map merging," *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.

[19] D. Marinakis and G. Dudek, "Pure topological mapping in mobile robotics," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1051–1064, 2010.

[20] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[21] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[22] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Semantic localization via the matrix permanent." in *Robotics: Science and Systems*, 2014.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," 2014, pp. 701–710.

[24] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[25] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.

[26] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

Aerial Picking and Delivery of Magnetic Objects with MAVs

Abel Gawel*, Mina Kamel*, Tonci Novkovic*,
Jakob Widauer, Dominik Schindler, Benjamin Pfyffer von Altishofen, Roland Siegwart, and Juan Nieto
*The authors contributed equally to this work.

***Abstract*— Autonomous delivery of goods using a Micro Air Vehicle (MAV) is a difficult problem, as it poses high demand on the MAV's control, perception and manipulation capabilities. This problem is especially challenging if the exact shape, location and configuration of the objects are unknown.**

In this paper, we report our findings during the development and evaluation of a fully integrated system that is energy efficient and enables MAVs to pick up and deliver objects with partly ferrous surface of varying shapes and weights. This is achieved by using a novel combination of an electro-permanent magnetic gripper with a passively compliant structure and integration with detection, control and servo positioning algorithms. The system's ability to grasp stationary and moving objects was tested, as well as its ability to cope with different shapes of the object and external disturbances. We show that such a system can be successfully deployed in scenarios where an object with partly ferrous parts needs to be gripped and placed in a predetermined location.

I. INTRODUCTION

Fast and customized delivery of goods is a major trend in transportation industry. MAVs are expected to be an important component in the future of autonomous delivery and are a means of transportation at the edge of consumer market entry [1]. Most solutions for handling goods with MAVs rely on mechanical gripping devices, as these can be realized lightweight and energy-efficient for high payloads [2], [3]. However, mechanical grippers usually require highly precise positioning of the gripper with respect to the object to yield a safe form closure or friction fit. High positioning accuracy cannot always be achieved with the MAV control alone, due to environmental disturbances, making either human intervention necessary, or requiring sophisticated additional actuators. Furthermore, the gripper design depends on the geometry of the objects to grip [4] making it necessary to use standard transportation containers or facilitate a variety of different mechanical grippers to enable MAVs to reliably grip differently shaped objects. Ferrous objects are interesting because they can be attracted by magnets. For gripping, these material properties can be exploited. In this case positioning accuracy can be considerably lower as a natural attraction force is generated between the magnetic gripper and ferrous material. However, using electro-magnets requires a constant power-supply to generate the magnetic field. On the other hand permanent magnets do not consume power, but they are problematic for releasing attracted objects. A new class of electro-permanent magnets overcomes both these limitations by providing a switchable permanent magnet [5].

Authors are with the Autonomous Systems Lab, ETH Zurich. {abel.gawel, mina.kamel, tonci.novkovic, dominik.schindler}@mavt.ethz.ch, {jwidauer, bpfyffer}@student.ethz.ch, {rsiegwart, nietoj}@ethz.ch

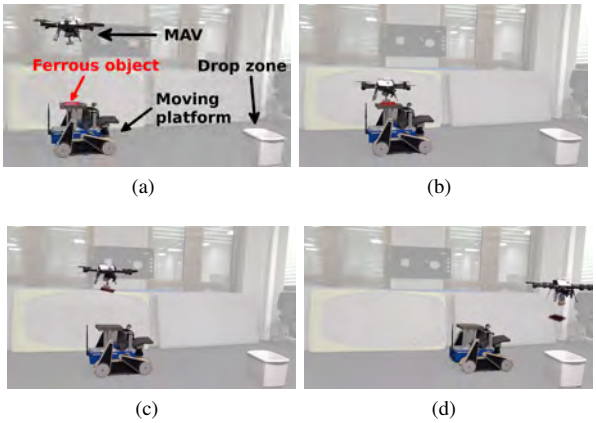


Fig. 1: Sequence of the autonomous aerial delivery approach: (a) The MAV detects object on moving platform and initiates servo positioning. (b) An Object is picked using a passively compliant electro-permanent magnetic gripper. (c) The MAV returns to operation height with an object attached and travels to the delivery zone. (d) The Object is dropped into the delivery container by deactivating the electro-permanent magnet after a short hover over the target location.

A second important challenge for aerial gripping is the correct positioning of the MAV towards an object of previously unknown shape and location, and deciding on a successful control for picking such objects. Here, servo-positioning techniques can enable a MAV to pick an object by providing relative localization to the object and a controller combined with an approaching strategy to yield robust object picking.

We present a novel system that is using electro-permanent magnets and is able to robustly and energy-efficiently pick and deliver stationary or moving objects with a partly ferrous surface of different shapes using a MAV. The current gripper design allows attachment to concave and convex objects with radius $> 90mm$.

This work is furthermore motivated by our participation in the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) in which MAVs are tasked to autonomously search a field for objects, pick them up and deliver them to a designated drop zone.

Our main contributions are:

- A low complexity and energy efficient electro-permanent gripper design that allows robust gripping with positional offset and different object shapes.
- A real-time visual servoing of the MAV position towards the object.
- An evaluation of the fully integrated system on different types of objects and in different conditions.

II. RELATED WORK

We focus our review of related work on recent advances in aerial gripping and servo positioning techniques for reliably detecting and approaching objects using a MAV.

A. Aerial Gripping

In [6] the authors propose an integrated object detection and gripping system for MAVs using IR diodes for detection and a mechanical gripper for gripping stationary objects. In contrast, our system aims to detect objects using a standard RGB camera and also grip moving objects with an partly ferrous surface.

Transportation of objects using MAVs was reported in [7], [8], [9]. However, the authors mainly focus on the control of MAVs transporting objects. In contrast to our work they do not implement a grip and release mechanism which is an important aspect for fully autonomous delivery.

An aerial manipulation task using a quadrotor with a two degrees of freedom (DOF) robotic arm was presented in [10]. The kinematic and dynamic models of the combined system were developed and an adaptive controller was designed in order to perform a pick and place task. Such system offers high manipulability, however, the shape of the objects to be picked is limited since the robotic arm is only able to pick thin objects in specific configurations, i.e., thin surfaces pointing upwards. Furthermore, this work assumes that the position of the object to be picked is known in advance.

A self-sealing suction technology for grasping was tested in [11]. A system capable of grasping multiple objects with various textures, curved and inclined surfaces, was demonstrated. Despite being able to achieve high holding forces, the gripping system requires a heavy compressor and an activation threshold force to pick up the objects. Also, all the tests were performed using a motion capture system with known object positions.

Another type of mechanical gripper was shown in [12]. The gripper uses servo motors to actuate the pins that penetrate the object and create a strong and secure connection. Similar design was also presented in [13]. The main limitation of such gripper is its restriction to pick only objects with penetrable surface. Furthermore, if the surface is not elastically deformable, the gripper might cause irreversible damage to the object.

In [14], a bio-inspired mechanical gripper was designed in order to allow quadcopters to carry objects with large flat or gently curved surfaces. In addition to being small and light, the gripper consists of groups of tiles coated with a controllable adhesive that allows for very easy attachment and detachment of the object. Nevertheless, the gripper is limited to smooth surfaces, requires tendon mechanism for attachment, and has a limited payload.

TABLE I: Properties of the magnetic material.

Material	Remanence	Intrinsic coercivity
Grade 5 Alnico	$1.25\,\mathrm{T}$	$48\,\mathrm{kAm^{-1}}$
Grade N45 Neodymium	$1.36\,\mathrm{T}$	$836\,\mathrm{kAm^{-1}}$

OpenGrab EPM[1] is a gripper developed using the principle of electro-permanent magnets [5]. It is a low-weight, energy efficient and high-payload solution developed for robotic applications and because of its advantages, we have decided to use the same principle for our own gripper. Since OpenGrab EMP is only able to pick flat surfaces, we have developed a more sophisticated design which allows our gripper to pick objects with curved surfaces, while maintaining an equal load distribution on all contacts between object and gripper.

B. Visual Servoing

Visual Servoing (VS) is a well established technique where information extracted from images is used to control the robot motion [15], [16], [6]. There are many approaches to deal with VS, however some of the most popular include:

1) Image Based Visual Servoing: In this approach, the control law is based entirely on the error in the image plane, no object pose estimation is performed. In [17] the authors employ this method to perform pole inspection with MAVs, while in [18] it is used to bring a MAV to a perching position, hanging from a pole.

2) Pose Based Visual Servoing: In this approach, the object pose is estimated from the image stream, then the robot is commanded to move towards the object to perform grasping or an inspection task for instance [19].

Our approach differs from the previous work in the sense that we apply servo positioning for gripping both static and moving objects.

III. ELECTRO-PERMANENT MAGNETIC GRIPPER

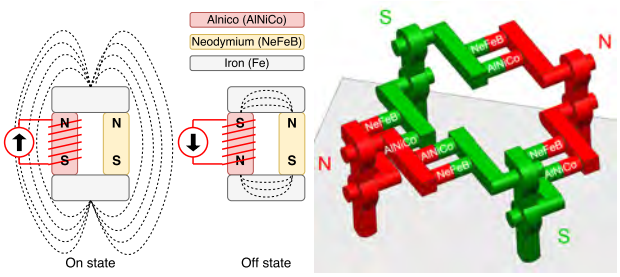
The proposed gripper features two main physical components, i.e., an electro-permanent magnet with electronics board and a passively compliant mechanical structure.

A. Electro-permanent magnet

The concept of an electro-permanent magnet is based on the physical properties of two different permanent magnets [5]. We consider Alnico and Neodymium magnets. The key properties are their remanence, which is the remaining magnetization after the removal of an external magnetic field and intrinsic coercivity, a measure for the necessary magnetic field to magnetize or demagnetize the material, see Table I.

In an electro-permanent magnet, both magnets are assembled in parallel while a coil is winded around the magnet with low intrinsic coercivity, here Alnico. Both magnets are connected to an iron carrier material, as illustrated in Fig. 2a.

[1] http://nicadrone.com/



(a) (b)

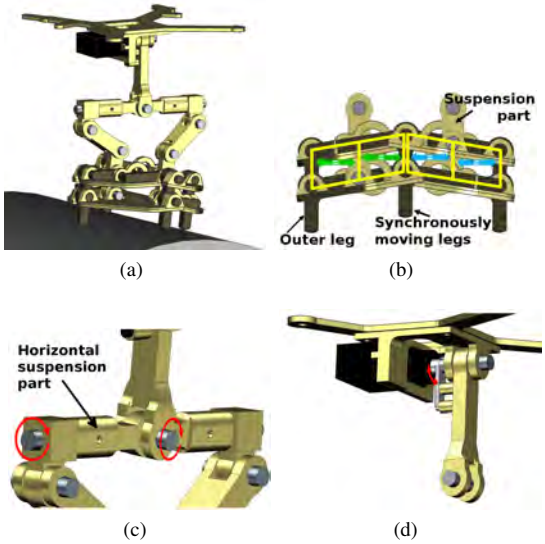
Fig. 2: (a) The figure illustrates the electro-permanent magnet principle. (b) Application of the electro-permanent magnetic principle in two circles as implemented for the design.

Sending a current pulse to the coil generates a magnetic field inside the coil which can switch the magnetic polarization of the Alnico, depending on the direction of the applied magnetic field. The Neodymium magnet stays magnetized in one direction throughout. If both magnets are magnetized in the same direction, the assembly acts as a permanent magnet to the outside, and if they are magnetized in opposite directions the magnetic field is circulating in the assembly and therefore does not act as a magnet to the outside.

B. Mechanical structure

The design of the mechanical structure aims to fulfill three main objectives, i.e., passive adaptivity to different surface geometries, integration of the electro-permanent magnets and functional connectivity to the MAV. We decided to implement 2 cycles of magnetic circuit on the gripper in order to realize a four-point contact to objects, ensuring secure hold. This is illustrated in Fig. 2b. For gripping objects of different shapes the functional parts are mounted on a carrier structure that allows for relative motion between the magnetic legs. The full design is illustrated in Fig. 3a. Equal force distribution between the legs is achieved by implementing a parallelogram-shaped support structure, as illustrated in Fig. 3b. A suspension with degrees of freedom in pitch and roll enables the gripper to account for attitude changes of the MAV, see Fig. 3c. To enable the gripper to extend below the MAV's feet, a retraction mechanism actuated by a servo motor is attached to the upper gripper suspension, see Fig. 3d.

The current design is calculated to a holding force of approximately $34\,\mathrm{N}$ which is tailored to the payload limit and the dynamics of the MAV considered. Furthermore, the attraction force can easily be scaled up with a moderate increase in energy consumption and weight of the electro-permanent magnetic components by increasing the diameter of the magnets. The manufactured gripper supplies a holding force of $30\,\mathrm{N}$ as further described in Section V-A but still within reasonable limits of the considered MAV.



(a) (b) (c) (d)

Fig. 3: (a) Full assembly of the gripper on a convex surface, carrying two circles of electro-permanent magnets as depicted in Fig. 2b (b) Parallelogram structure of end effectors for equal force distribution among all four contacts points between gripper and object. (c) Upper suspension to provide degrees of freedom in pitch and roll. (d) Retracting mechanism for pulling up the gripper.

IV. Visual Servo Positioning

The visual servo positioning module deals with the challenge of autonomously approaching and gripping a detected object. In a first step the MAV visually detects an object and localizes the relative pose of its Center of Gravity (CoG) to the object's CoG. Then a controller is activated to yield a desired x, y position. In the following step the MAV executes a strategy for approaching the object in $z-$direction, i.e., the direction of the gravity vector. Finally the MAV returns to its operation height and travels to a drop zone, where it releases the object. The drop zone is in a known location.

A. Relative localization

In our approach we use a simple frame-to-frame detector to estimate the object's CoG. Then, we estimate the relative transformation ${}_W\boldsymbol{T}_{m,o}$ between the MAV and the object's CoG in a global reference frame W. For the relative localization step, the object is approximated to have a flat surface for this calculation and the CoG to lie in the top plane of the object. The subscripts m and o denote the MAV location and the object location respectively.

The MAV uses its relative height estimate above the object h and attitude estimate ${}_W\boldsymbol{R}_m$. The location of the object is then estimated by first calculating the relative rotation ${}_W\boldsymbol{R}_{c,oi}$ between camera center c and object CoG in the normalized image plane oi via the rotation ${}_C\boldsymbol{R}_{c,oi}$ in camera frame C. This rotation is transformed into the world coordinate frame using the MAV's attitude estimate and the rotation between MAV base frame and camera center ${}_M\boldsymbol{R}_c$ in the MAV frame M, yielding the relative rotation ${}_W\boldsymbol{R}_{c,oi}$ in the global coordinate frame W.

$$ {}_W\boldsymbol{R}_{c,oi} = {}_W\boldsymbol{R}_{mM}\boldsymbol{R}_{cC}\boldsymbol{R}_{c,oi} \tag{1} $$

Using the translations between camera and object CoG in image coordinates, expressed in camera and world frame, ${}_C\boldsymbol{t}_{c,oi}$ and ${}_W\boldsymbol{t}_{c,oi}$ respectively, we calculate the translational component of the offset ${}_C\boldsymbol{t}_{oi}$.

$$ {}_C\boldsymbol{t}_{oi} = -\frac{h}{\begin{pmatrix}0 & 0 & 1\end{pmatrix}{}_W\boldsymbol{t}_{c,oi}}{}_C\boldsymbol{t}_{c,oi} \tag{2} $$

Finally we calculate the relative translation ${}_W\boldsymbol{t}_{m,o}$ between the MAV and CoG of the object in the global coordinate frame.

$$ {}_W\boldsymbol{t}_{m,o} = {}_W\boldsymbol{R}_m({}_M\boldsymbol{t}_{m,c} + {}_M\boldsymbol{R}_{cC}\boldsymbol{t}_{oi}) \tag{3} $$

Here ${}_M\boldsymbol{t}_{m,c}$ denotes the calibrated translation between MAV base frame and camera center.

The relative transform ${}_W\boldsymbol{T}_{m,o}$ between MAV and object is then

$$ {}_W\boldsymbol{T}_{m,o} = \begin{pmatrix} {}_W\boldsymbol{R}_{m,o} & {}_W\boldsymbol{t}_{m,o} \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{4} $$

One of the advantages provided by the design of our gripper is that we can simplify the calculation for planar objects as non-planar surface shapes will be passively handled by the mechanical structure of the gripper.

B. Approaching / Servoing

The x, y-offset and the z-offset are handled separately. The x, y-offsets are handled by a PID-controller. Based on the error between the object's CoG detected in the image frame and the gripper position, a $\Delta x, \Delta y$ command is generated and added to the current MAV pose. The control input is saturated and an anti-windup scheme is implemented. The newly generated MAV pose is then tracked by a trajectory tracking Model Predictive Controller (MPC) [20]. In the case of several objects being present in the MAV's current field of view, the strategy is to first target the object closer in the Euclidean x, y-distance.

As we assume objects to be on the ground, we define a set approach strategy to yield robust system performance, this is illustrated in Fig. 4. The approach-strategy is triggered when the MAV stays within a radius ε in x, y around a point at height h above the CoG of the object. If the MAV manages to stay within this radius for a set time $t_{wait} > t_{thresh}$, it then descends to a lower height h_{hover} above the object. The servo positioning checks whether the MAV is still within a sphere ε above the object's CoG and then initiates the final approach towards the object, which is a guided sequence of descending to the object before ascending to the operation height h. If the object is moving during the first descend, the MAV uses the previous velocity in x, y for its final descend making it possible to approach linearly moving objects. The MAV's trajectory controller accepts waypoints published by the servoing regime, as illustrated in Fig. 4.

If the MAV loses sight of the object in the approach sequence, it returns to height h and re-localizes the object in a wider field of view. Please note that a global search strategy is out of the scope of this paper. Presently, we also do not perform object tracking, but perform object detection over 3 frames. Therefore, if a different object is perceived closer to the MAV the algorithm will switch targets and approach the closest object.

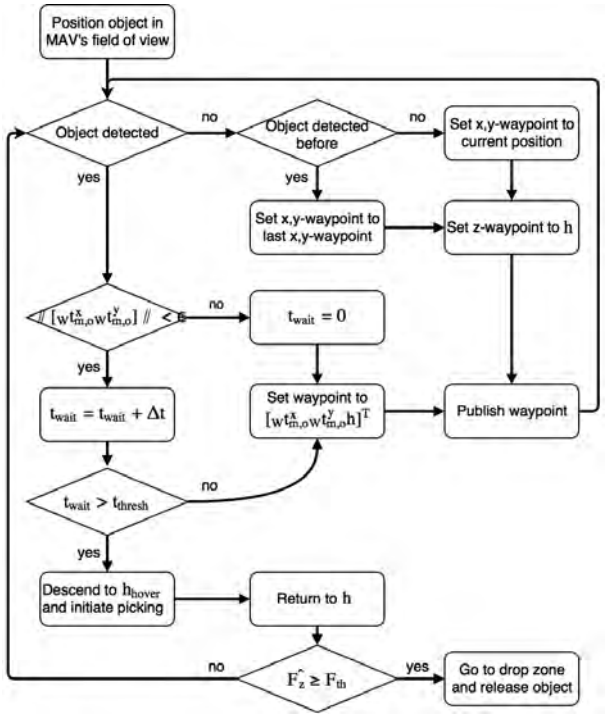


Fig. 4: State machine of the MAV for object picking and delivery. The MAV is positioned with an object in its field of view. Then the servoing strategy is performed, resulting in the delivery of the object to the drop zone, if all conditions are met.

C. Delivery

After successful gripping, the MAV flies to the drop zone of known location and releases the object.

An important aspect for robust aerial gripping and transportation is sensing successful gripping of the object. Given that a model-based external disturbances observer based on Extended Kalman Filter (EKF) is employed by the trajectory tracking controller [20] to compensate for external forces, we decided to employ it to detect successful grasping as well. A successful grasping is detected if the following equation holds:

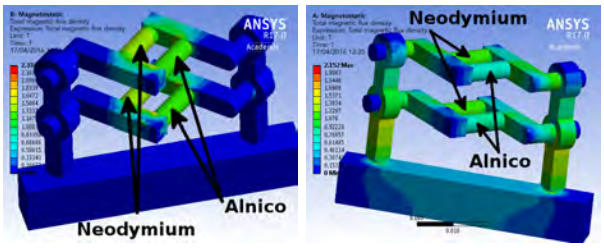
$$ \hat{F}_z \geq F_{th} \tag{5} $$

where $\hat{F}_z$ is the z component of the estimated external force expressed in world frame and F_{th} is a user defined threshold.

In case the object is lost during transport, the MAV returns to the location in which it detected the loss, re-detects the object and performs the servo-positioning from the start. The exact behavior is also shown in Fig. 4. The controller treats additional effects, e.g., drag forces by gripped objects or

TABLE II: Air gaps in magnetic flux simulation.

Loaction	Gap width
Between magnet and horizontal iron part	$50\,\mu$m
Between leg and horizontal iron part	$25\,\mu$m
Between leg and object surface	$100\,\mu$m



(a) (b)

Fig. 5: Simulation of the magnetic flux for the electro-permanent magnet, (a) in the off state forming a closed magnetic circuit inside the material and (b) in the on state inducing a magnetic field outside the gripper material. The magnetic flux density ranges from $0\,\mathrm{T}$ (blue) over $1\,\mathrm{T}$ (green) to $2\,\mathrm{T}$ (red).

additional weights as external disturbances and compensates the effects.

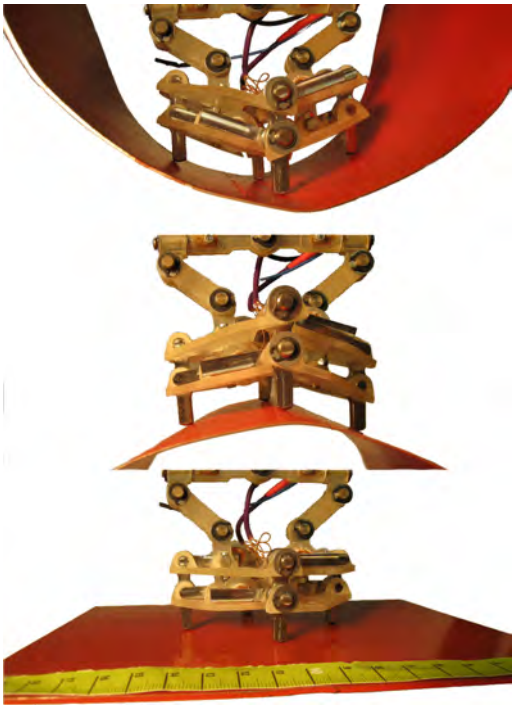
V. EVALUATION

The evaluation of our system is three-fold. We test the magnetic behavior of the gripper in simulation and real world experiments, perform a functional evaluation of the gripping with offsets and test the full system under varying conditions, i.e., external disturbances, differently shaped objects and moving objects.

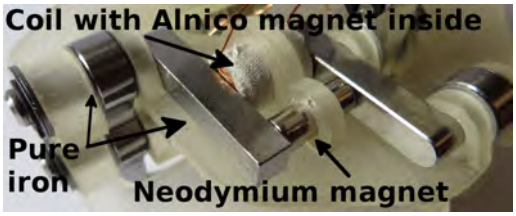
A. Magnetic gripper behavior

Simulation of the magnetic flux is shown in Fig. 5 for the gripper in the on and off state. With this configuration each of the 2 magnetic cycles of the gripper generates a force of $17\,\mathrm{N}$ per magnetic cycle while assuming air gaps between the functional part as depicted in Table II.

The physical gripper is illustrated in Fig. 6. Tests with the full assembly show that the gripper produces an attraction force of approximately $30\,\mathrm{N}$, with all legs connected, which is lower than the simulated value. We believe this is due to imperfect manufacturing of the gripper, resulting in slightly different air gaps in the assembly. Nevertheless, the force is still well within acceptable bounds. The attraction force is halved if only one of the magnetic cycles is closed. The functional parts of one of the magnetic cycles is illustrated in Fig. 6b. In order to switch between the gripper's on and off state, using the MAV's onboard $15\,\mathrm{V}$ batteries, a short $2.5\,\mathrm{ms}$ current pulse of $80\,\mathrm{A}$ is sent each time, resulting in consumption of $0.8\,\mathrm{mWh}$ per switch. The final assembly weighs $210\,\mathrm{g}$ including all functional components. However, the materials and the design of the support structure are not optimized yet, especially since we facilitate 3D printed



(a)



(b)

Fig. 6: (a) Full assembly of the gripper on differently shaped surfaces. (b) One of in total four Alnico / Neodymium assemblies in final gripper prototype.

plastic which requires considerably thicker parts to provide the required stiffness compared to light-weight composite materials. Furthermore, a circuit board for fast prototyping was used for the electronic parts, adding $120\,\mathrm{g}$ to the weight of the gripper assembly, although developing a Printed Circuit Board (PCB) would significantly decrease this weight.

B. Offset gripping

We evaluated the gripper assembly in an isolated experiment, using different square metal objects to evaluate its offset gripping behavior. The testing procedure defines a gripping procedure in an offset position in x, y to the objects's CoG followed by a vertical acceleration of $0.8g$. The test objects vary in weight and shape, i.e., we test the system on one heavy flat square metal plate and a lighter square metal plate with a bend of $30°$ in the middle. The tests were conducted by linearly increasing the offset of the grip position until the border of the object was reached. The results are illustrated in Table III.

Although the objects can be lifted statically regardless of

TABLE III: Results of offset gripping tests.

Object			Gripping max. values	
Dimensions	Weight	Bend	Offset	Pitch angle
$165 \times 165 \times 2\,\mathrm{mm}$	$520\,\mathrm{g}$	$30°$	$81\,\mathrm{mm}$	$55°$
$150 \times 150 \times 4\,\mathrm{mm}$	$870\,\mathrm{g}$	$0°$	$30\,\mathrm{mm}$	$27°$

TABLE IV: Results of integrated system (IS) tests.

Experiment type	Success rate	Experiments	Pick up tries
IS	$95.65\,\%$	23	25
IS + wind	$100\,\%$	5	8
IS + dynamic objects	$78.57\,\%$	14	27

the offset position, the vertical acceleration causes both tested objects to show a cutoff offset, i.e., the object is always lost in the lifting when this offset is exceeded. The major causes for losing contact with the object during the dynamic lifting is lateral slipping for the large, but lighter part and loss of contact with the innermost magnetic leg for the smaller, but heavier part. The failure mechanisms can be explained mechanically as follows. The large part starts slipping as the large offset gripping position causes the gripper suspension to have considerable offset rotation in pitch. This causes a lateral force on the magnetic legs that exceeds the friction induced by the magnetic attraction force at the contacts between the legs and the object, causing slippage. The small part fails, because the force is shifted onto the innermost leg due to the leverage effect, causing the combined force of gravity and acceleration to exceed the magnetic attraction force.

However, the MAV controller can reliably provide positional accuracy in x, y which is within the safe bounds of the evaluated gripper's behavior and can therefore provide safe means of gripping in the integrated system's context. Furthermore, we implemented a gripping detection in the integrated system, causing the MAV to re-attempt in case of unsuccessful gripping. Finally, we note that the tested accelerations in this experiment are higher than the ones in the integrated system.

C. Object detection

For the object detection in the integrated evaluation, we use a simple frame-to-frame recognition scheme using a down-facing camera that is rigidly attached to the base of the MAV. Objects are assumed to be arbitrarily shaped and of red, blue or black color. We aim to detect the CoG of the objects. Therefore, the images are undistorted, down-sampled to $\frac{1}{8}$ resolution and converted to HSV color space. The detector performs morphological opening to remove small foreground detections, and morphological closing to fill small holes in the foreground. Then it detects contours on the binary images, which are filtered by a threshold to reject very small (contour including $< 0.4\%$ of the image plane) and very large objects (contour including $> 90\%$ of the image plane). The remaining contours are detected as objects and their CoG calculated by using their 0th and 1st moments. The thresholding is performed on all three axes of the HSV color space to distinguish between red, blue and black objects.

D. Integrated System Evaluation

A MAV equipped with a down-facing Pointgrey Chameleon3 $3.2\,\mathrm{MP}$ camera with a fisheye lens running at $20\,\mathrm{Hz}$ and the presented gripper were evaluated in an indoor motion capture room, as illustrated in Fig. 1 and shown in the video supplement[2]. In a first experiment a bend ferrous object is placed in a random location across the room in one of two configurations, i.e., either with the bend facing upwards or downwards, providing a convex or concave object to pick. Then the MAV is brought manually to a hover in a random location with the object in its field of view. The MAV is then tasked to autonomously execute the object detection, servo positioning, gripping and transportation to a known drop-off location for releasing the object. This experiment is repeated 23 times using the bend metal object in different configurations. We also perform 5 trials while applying varying strengths of wind to the platform of up to $15\,\mathrm{m/s}$. Furthermore, another experiment is executed 14 times with the object placed on a moving platform that moves linearly in an arbitrary direction with a set velocity of $0.1\,\mathrm{m/s}$ which was not communicated to the MAV. We use the AscTec Neo MAV for these experiments and a motion capture system for tracking the MAV. However, the MAV is not limited to operate with a tracking system, and alternatively a state estimation on board the MAV can be used [21]. The results of these tests are presented in Table IV. Here we report the success rates along with the number of experiments and the number of pick up trials, i.e., the number of total pick up repetitions if the part is detected to not be picked and a re-picking is triggered. The procedure is illustrated in Fig. 1 for the dynamic experiments. The static experiments were performed in a similar setup with the object being placed on the ground.

E. Findings

Throughout all static experiments we recorded only one failure in the delivery action due to releasing the object next to the drop zone and therefore missing the drop zone container. Since the object landed too close to the container the MAV was not able to pick it up again because of the confined space and risk of crashing with the container. As expected, the picking quality decreased in the case of external disturbances since positioning accuracy achievable by the MAV decreases. Furthermore, in some cases, we noticed that reflections on the object can cause the detector to estimate a CoG that is off-centered, thus decreasing positioning accuracy, i.e., two repeated tries. In such cases, the MAV was not able to properly grip the object in the first approach, however, it was still able to detect this, recover, and retry the procedure. The controller was also able to handle objects of different weights without readjustment of its parameters.

[2]https://goo.gl/no0Bcz

In the dynamic case, when the object is placed on a moving platform we recorded decreased success rate for picking. We counted failure cases if the object was not accurately picked by the MAV causing it to slip and fall to the ground. Although the object could be recovered from the ground as static object, we cancelled the experiment in these cases and recorded failure. We noted that, since we perform frame-wise detection, we do not have an accurate velocity estimate of the moving platform which could be improved by implementing an object tracking over time.

VI. CONCLUSIONS & FUTURE WORK

In this paper, we have presented a full system for energy-efficient, autonomous picking and delivery of ferrous objects with MAVs. The integrated system is based on gripping technology with electro-permanent magnets.

We have evaluated the core innovations of our pipeline separately and the integrated system as a whole. Our results show that even under varying conditions the MAV is able to pick and deliver the objects in the static case and most of the times in the dynamic case as well. In contrast to state-of-the-art approaches which rely either on known object locations, known object shapes or high position accuracy of the MAV, our approach can handle all of these unknowns in an integrated manner while achieving very high delivery success rates. Furthermore, the proposed gripper design for MAVs combining passive compliance with electro-permanent magnets, to our best knowledge, has not been shown before.

For future work, we plan to further optimize our gripper design towards weight and compliance and integrate the camera in a next version of the gripper as its field of view is partly occluded in the current setup. We furthermore plan to implement object tracking and feed-forward control to increase the system performance for picking of moving objects. Another interesting avenue is a combination of our system with global search strategies and multiple MAVs. Finally, it would be beneficial to sense successful gripping with a dedicated sensor as our present approach relies on a minimum weight threshold (Eq. 5).

ACKNOWLEDGMENT

This work was supported by the European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763, European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS, and Mohamed Bin Zayed International Robotics Challenge 2017.

REFERENCES

[1] H. Zhang, S. Wei, W. Yu, E. Blasch, G. Chen, D. Shen, and K. Pham, "Scheduling methods for unmanned aerial vehicle based delivery systems," in *IEEE/AIAA 33rd Digital Avionics Systems Conference*, 2014.

[2] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *IEEE International Conference on Robotics and Automation*, 2011.

[3] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Robotics: Science and Systems*, 2011.

[4] S. B. Backus, L. U. Odhner, and A. M. Dollar, "Design of hands for aerial manipulation: Actuator number and routing for grasping and perching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[5] A. N. Knaian, *Electropermanent Magnetic Connectors and Actuators: Devices and Their Application in Programmable Matter*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2010.

[6] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[7] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[8] I. Maza, K. Kondak, M. Bernard, and A. Ollero, "Multi-uav cooperation and control for load transportation and deployment," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 417–449, 2010.

[9] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[10] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[11] C. C. Kessens, J. Thomas, J. P. Desai, and V. Kumar, "Versatile aerial grasping using self-sealing suction," in *IEEE International Conference on Robotics and Automation*, 2016.

[12] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[13] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. DAndrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, 2014.

[14] E. Wright Hawkes, H. Jiang, and M. R. Cutkosky, "Three-dimensional dynamic surface grasping with dry adhesion," *International Journal of Robotics Research*, vol. 35, no. 8, pp. 943–958, 2016.

[15] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-rye system," in *IEEE International Conference onRobotics and Automation*, 1995.

[16] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors," in *EEE International Conference on Robotics and Automation*, 2003.

[17] I. Sa, S. Hrabar, and P. Corke, "Inspection of pole-like structures using a vision-controlled vtol uav and shared autonomy," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[18] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Visual servoing of quadrotors for perching by hanging from cylindrical objects," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 57–64, 2016.

[19] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust visual tracking by coupling 2d motion and 3d pose estimation," in *IEEE International Conference on Image Processing*, 1999.

[20] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Robot Operating System, The Complete Reference, Volume 2*. Springer Press, 2017, (to appear).

[21] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013.