

FULL PAPER

**Designing, developing and deploying systems
to support human-robot teams in disaster response**

G.J.M. Kruijff^a, I. Kruijff-Korbayová^{a*}, S. Keshavdas^a, B. Larochelle^a, M. Janíček^a
 F. Colas^b, M. Liu^b, F. Pomerleau^b, R. Siegwart^b
 M.A. Neerincx^c, R. Looije^c, N.J.J.M Smets^c, T. Mioch^c, J. van Diggelen^c
 F. Pirri^d, M. Gianni^d, F. Ferri^d, M. Menna^d
 R. Worst^e, T. Linder^e, V. Tretyakov^e, H. Surmann^e
 T. Svoboda^f, M. Reinstein^f, K. Zimmermann^f, T. Petříček^f, V. Hlaváč^f

^a*DFKI, Saarbrücken, Germany;*

^b*ASL, ETH, Zurich, Switzerland;*

^c*TNO, Soesterberg, The Netherlands;*

^d*ALCOR, Sapienza University, Rome, Italy;*

^e*Fraunhofer IAIS, Sankt Augustin, Germany;*

^f*Department of Cybernetics, CTU, Prague, Czech Republic*

(submitted May 14, 2014, revised October 21, 2014)

This paper describes our experience in designing, developing and deploying systems for supporting human-robot teams during disaster response. It is based on R&D performed in the EU-funded project NIFTi. NIFTi aimed at building intelligent, collaborative robots that could work together with humans in exploring a disaster site, to make a situational assessment. To achieve this aim, NIFTi addressed key scientific design aspects in building up situation awareness in a human-robot team, developing systems using a user-centric methodology involving end users throughout the entire R&D cycle, and regularly deploying implemented systems under real-life circumstances for experimentation and testing. This has yielded substantial scientific advances in the state-of-the-art in robot mapping, robot autonomy for operating in harsh terrain, collaborative planning, and human-robot interaction. NIFTi deployed its system in actual disaster response activities in Northern Italy, in July 2012, aiding in structure damage assessment.

Keywords: robot-assisted disaster response; human-robot team; user-centric design; disaster response

1. Introduction

NIFTi was a large-scale four year integrated project funded by the EU Cognitive Systems unit [1].¹ The NIFTi consortium consisted of six academic partners (i.e., the institutions of the authors of this paper), sharing experience in human-robot interaction, human factors and cognitive user modeling, field robotics, spatial and visual modeling of outdoor environments, and flexible planning and execution; two end user organizations (the Italian National Firebrigade Corps and the Firebrigade of the City of Dortmund) and BLUEBOTICS², a company who developed

*Corresponding author. Email: ivana.kruijff@dfki.de

¹NIFTi was funded within the EU FP7 ICT programme, Jan 2010 – Dec 2013, grant No. 247870.

²<http://www.bluebotics.com/>

the novel rover platform Absolem for NIFTi (Fig. 20). NIFTi's aim was to investigate cognitive architectures which could meaningfully sense, act and cooperate with humans in real-life environments.

Regarding the issue of cooperation in cognitive architectures, when NIFTi started, it was entering a research landscape that primarily focused on autonomy, and high-level communication. Little or no attention was given to making the cognitive architecture adapt to the human in understanding the environment, planning and acting, communicating. In the words of the leading experts on human-robot teamwork: "Whereas early research on teamwork focused mainly on interaction within groups of autonomous agents or robots, there is a growing interest in better accounting for the human dimension. Unlike autonomous systems designed primarily to take humans out of the loop, the future lies in supporting people, agents, and robots working together in teams in close and continuous human-robot interaction." (Sierhuis & Bradshaw, p.c. 2009). NIFTi adopted the goal to bring the human factor into cognitive architectures while developing robots capable of collaborating with human team members under the complex outdoor circumstances of a disaster response.

To address this aim, NIFTi put strong emphasis on system integration, embedded within a user-centric approach to system development. The two firebrigade organizations included as partners in the NIFTi consortium enabled close involvement of end users, the ultimate stakeholders in this game, throughout the entire R&D cycle. They provided input to system specifications, participated in yearly exercises and evaluations, and provided feedback for further iteration cycles of the development process. Emphasis on system integration required that all developed functionality be integrated in the NIFTi system, and exposed to evaluation by the end users. To facilitate this across-the-board integration, NIFTi adopted a scenario-driven roadmap. The roadmap defined progressively more complex real-life scenarios: A tunnel accident (years 1 and 2), a chemical freight train accident (year 3), and an earthquake disaster response (year 4). These scenarios were all instantiated at end user training areas, subjecting the NIFTi system to realistic circumstances.

The roadmap simultaneously drove R&D, and brought the resulting systems closer to the possibility of real-life deployment. Needless to say, the lessons we learnt along the way were hardly ever foreseeable in advance. Experiments and tests each year brought new insights, to which we continuously adjusted our plans and the direction our R&D would take from that point on. The result was a convergence of R&D with real-life needs, real-life possibilities, and a real-life ability to actually aid in a disaster response. This ability was put to a test successfully during the summer of 2012, when NIFTi deployed its then-actual system to support a human-robot team in the structural damage assessment in Northern Italy during the aftermath of the July 2012 earthquake in the Emilia-Romagna region.

This article describes the experience NIFTi gained in designing, developing, and deploying human-robot team systems for robot-assisted disaster response. §2 provides more details on the NIFTi roadmap and scenarios. In §3 we describe the integrated system and its functional modules. §3.1 outlines the system architecture. §3.2 describes novel methods for building up robot-centric, three-dimensional models of dynamic environments – integrating various forms of perception (3D laser, visible range panoramic camera, thermal imaging). §3.3 addresses the functionality needed to bridge between system- and human-centric situation awareness. §3.4 then describes how we built on that to provide the robot capabilities for autonomously traversing complex terrain, including crossing gaps and climbing stairs. §3.5 provides an insight into handling human-robot collaboration. §3.6 provides more detail on how information about the situation is made available through various views in the graphical user interfaces for the human team members – portable, as well as a "static" setup in a remote command post. §3.7 addresses one crucial aspect of teamwork, namely handling cognitive load. §4 shows how our end user experiments reveal that this has a genuine, positive impact on ability for human team members to build up a situation awareness "through," together with, our robots. Finally, §5 recapitulates the experience of the actual deployment of our system during the earthquake response in Northern



Figure 1. Scenario impressions over the years.

Italy in July 2012. §6 provides the conclusions.

2. Scenario-driven R&D

NIFTi organised its R&D around a sequence of scenarios that gradually increased in complexity, including operational context complexity (from flat 2D, to semi-unstructured 3D) and collaborative context complexity, such as team size, its composition and geographical distribution (from 1 human/1 robot to a geographically distributed team consisting of multiple humans and robots). The robots used in NIFTi were an Unmanned Ground Vehicle (UGV, Fig. 20) and an Unmanned Aerial Vehicle (UAV microcopter, Fig. 21).

This scenario roadmap played a key role in providing an integrated conceptual picture for the project, to strongly drive integration of the various strands of R&D. Furthermore, by basing the scenarios directly in real-life situations in disaster response, we could ground R&D in real needs of Urban Search and Rescue (USAR) teams. The sections below describe the roadmap and the individual scenarios in more detail.

2.1 Roadmap

The NIFTi scenarios were designed in close cooperation between developers, and the USAR teams from the end user organizations involved in NIFTi as partners (Firebrigade Dortmund, Germany and National Firebrigade Corps, Italy). This was to ensure the scenarios would achieve a balance between practical relevance and feasibility, and necessary scientific progress. The result was a staged, iterative form of user-centric design cycle that addressed incrementally more complex situations.

On the one hand, the surroundings became incrementally harder for the robots to operate in. We went from flat, largely 2-dimensional terrain like the road surface in a tunnel, to semi-unstructured debris-strewn environments of an earthquake disaster. For robots, this necessitated

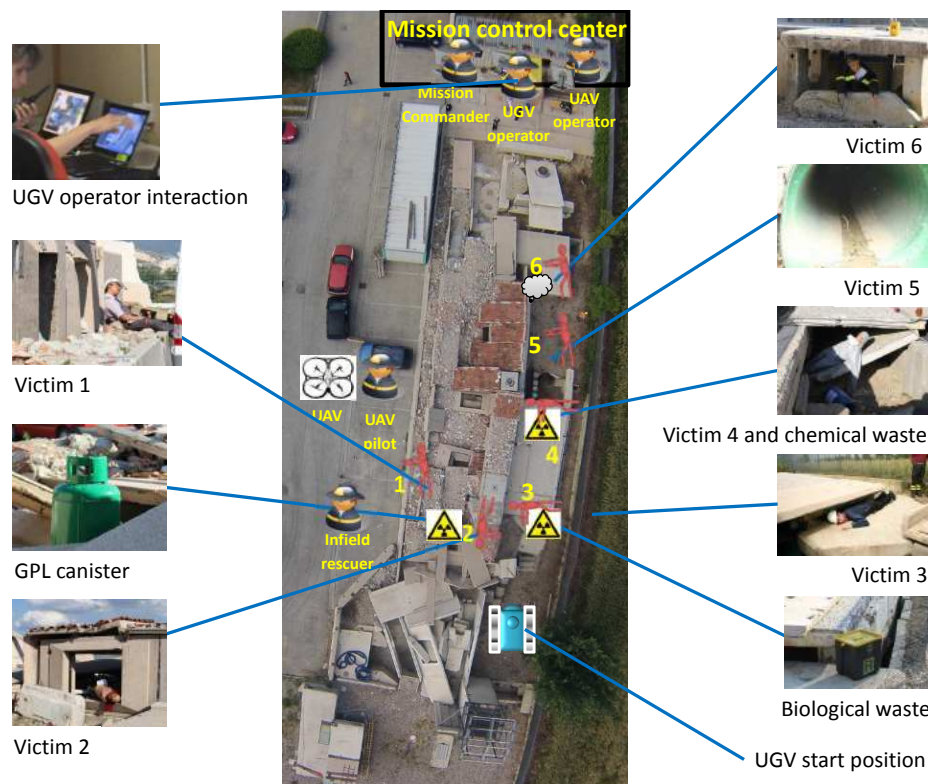


Figure 2. Year 4 scenario setup.

the development of increasingly more observational capabilities (from 2D to 3D), and progressively higher degrees of autonomy (3D path planning, adaptive morphology).

On the other hand, the organizational structure of the team become more realistic over the years. In Year 1, the organizational structure was non-existent, just a UGV and an operator. After that initial experience, we changed to a human-robot team setup. Eventually, the team included humans and robots, both UGV and UAV, working together in various locations – a genuinely geographically distributed team. By Year 4, robots were able to operate as team members in the sense of having the ability to build up situation awareness that was not immediately known by the operator and had to be shared.

2.2 Scenario Design

The joint scenarios that guided each year R&D were designed to evaluate all technical and operational requirements and to provide insight into the major determinants of human-robot team performance. To address important operational task demands, the environment provided realistic challenges, events and stressors like victim screams and (simulated) radioactive materials added at certain locations.

During the yearly scenario-based evaluation, the firefighter team-members (e.g., the commander, UGV operator, UAV operator or in-field rescuer) worked with the robots according to the role that was specified in the scenario. All the firefighters had an interface to show them (geographic) information [2]; the commander could contact both the UGV operator and the in-field rescuer, and vice-versa. Fig. 2 shows the final setting for the UGV-operator, where he can control the robot, get an overview of the situation and interact by walkie-talkie to the commander. Following an incremental R & D-approach, the complexity of the guiding scenario increased on three dimensions each year: scope of robot roles (i.e., level of autonomy and breadth of opera-

tions), team complexity (i.e., size and distribution), and terrain's complexity (i.e., accessibility and apparentness). Fig. 1 shows an impression of each year's setup.

The following is a brief chronological overview of how the NIFTi scenarios evolved and the convergence between R&D and real-life deployment came about.

The **first year** scenario comprised a truck accident in a tunnel, which was dangerous to enter for humans. A single remotely located operator teleoperated a UGV in the tunnel to create a 2D-map populated with car objects recognised by the robot. We focused on bringing together the various pieces of individual robot functionality (control, mapping, vision) with a basic, end user oriented graphical user interface for teleoperating the robot. Admittedly, the first end user pilot study at the training area of the Firebrigade of Dortmund, Germany was less successful than we would have wished. We faced highly familiar problems such as network issues (too low bandwidth, bad connectivity), robot hardware issues (short battery operating time, blowing a fuse every few minutes), and a wide range of human factor issues. Consequently, the local firebrigade chief commented, when he was asked whether such robots would ever play a role in disaster response: "No ... At least not in my lifetime." (D. Aschenbrenner, Dortmund Zeitung, January 14 2011). We concluded that we needed to move from the "single robot, single operator" setup to a full-scale human-robot team, and to get a better grip on technology.

In the **second year** we continued with the tunnel accident scenario, but now with a larger human team operating from a remote command post, and more difficult operational conditions, involving more smoke, flickering light and more debris. We introduced the new UGV platform developed in close collaboration with the end users involved in NIFTi (cf. Fig. 20). In addition, also a UAV microcopter was added to the rescue team (cf. Fig. 21). Furthermore, we moved to a human-robot team setup. The human team members took on various roles, such as a Mission Commander, UGV/UAV Operator and/or Mission Specialist.

During a joint exercise at the Firebrigade of Dortmund in summer 2011, end users and researchers teamed up, sitting in a real-life (large) command post to remotely operate the new UGV and to collaborate with an in-field UAV pilot, to explore a substantially more complex disaster site, a burning multi-story building. Robot control, vision, and mapping had significantly improved to move towards building up a robot-centric 3D understanding of the environment it was operating in. Human team members were provided access to this robot-centric situation awareness through an integrated user interface setup, facilitating multiple operational views (camera, map) and tactical views (team-level operations). All of this was simultaneously pushing the state-of-the art and showing how we were taking on board the lessons learnt from working with end users. When reporters once again asked the local firebrigade chief for his opinion on the practical feasibility of these robots, his response was: "The first deployment has moved to foreseeable future." (D. Aschenbrenner, Die Zeit, August 25 2011)

In the **third year** we moved to a scenario of an accident with a chemical freight train at a large terrain. Whereas the previous tunnel accident scenario was set in a relatively confined, enclosed 2D space cluttered with non-traversable objects, the train accident saw the introduction of large open space, a combination of static (bus, train, cars) and traversable 3D objects and structures (stairs, platforms, pallets), which made the terrain passable with difficulty. An in-field rescuer was added to the team, increasing team's size and geographical distribution. For R&D this put 3D robot-control and -mapping into the foreground, as well as the issue of building up and maintaining distributed situation awareness.

The real field-test of that system came during the summer of 2012: Not in the form of experiments at a training site, but during an actual deployment in Northern Italy during the aftermath of the July 2012 earthquakes. NIFTi deployed its then-actual system to support a human-robot team in exploring two large sites for structure damage assessment: A church, and a cathedral. During the missions in the cathedral, the robot was partly operating beyond line-of-sight. During this deployment, the enormous value of having worked with end users before became clear once more. Initial scepticism on the side of local rescue workers gradually changed into full-scale adoption as they came to realise what information our robots could really provide them with.

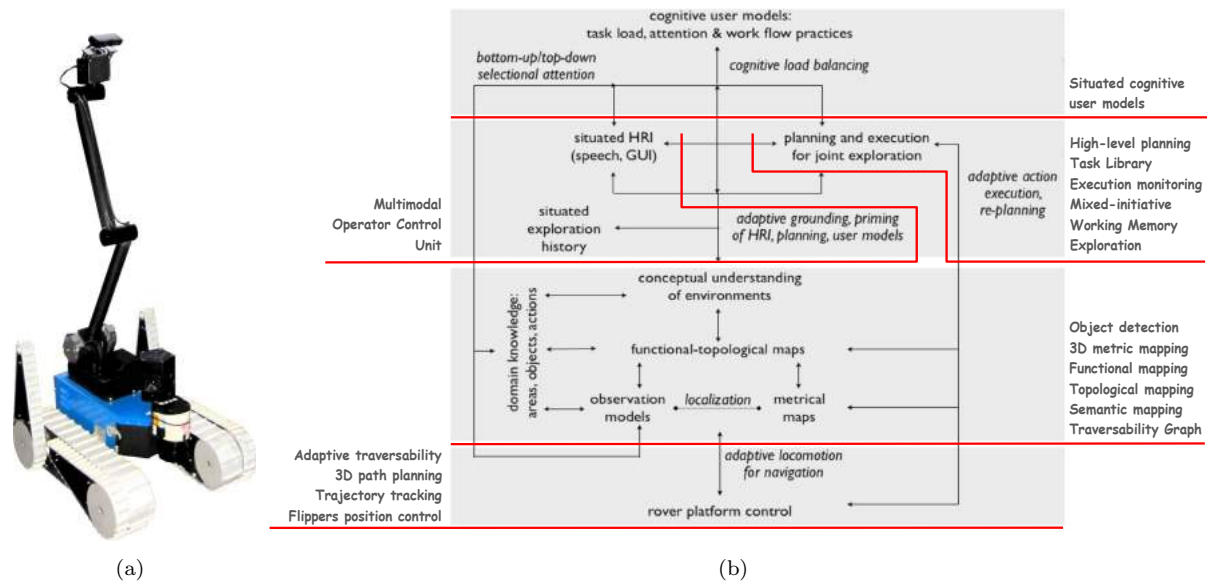


Figure 3. (a) NIFTi ground vehicle and (b) system architecture for this single robot considered as a team member.

Furthermore, the difficult circumstances under which we were operating there (long days, intense heat, stress) confirmed the need for human-robot teaming: A team can manage operating in these situations, where a single operator can hardly be expected to endure.

The **fourth year** built on that experience, moving on the roadmap to the earthquake scenario with multiple levels to explore. Specific dynamic areas could only be explored by an UAV, UGV or a human rescuer. Continuous team coordination and communication was crucial for an adequate disaster response. The experiments with this multi-human, multi-robot team were run at two sites of the Italian firebrigade: a USAR training area in Prato and an abandoned, partly destroyed hospital near Pisa.

3. System description

The NIFTi system constitutes a complex ecology of robots, network communication infrastructure, and a multitude of graphical user interfaces (mobile, smart-table, monitor). All of these components share the same network. We use the ROS framework [3] as main middleware for communicating information between these different components.

In summary, for the system & network infrastructure, we used ROS for running processes on the robot. Data was streamed over WiFi to one or more operator control units (OCU) and other visualization tools (RViz), and for logging purposes (`rosvbag`'s). Off-board computers were used for processing 3D laser range data (point clouds), and for the OCU and visualization. We used a 2.4GHz WiFi network, with an antenna nearby the entrance to the actual deployment area. The antenna was 50cm long, had 14dBi gain, and was extended with a Ubiquiti high power bullet enabling a transmission power of maximally 28dBm. Each robot (UGV and UAV alike) was also equipped with a bullet, and an omnidirectional rod antenna with a 9dBi gain. As we were mostly working in large open spaces, we did not experience substantial problems with network coverage.

3.1 Overall system outline

The NIFTi UGV has been developed in close collaboration between research partners and end users based on the system architecture shown in Fig. 3. The system uses the ROS framework [3] for lower-level control and the CAST middleware [4] for higher-level processing.

Besides the UGVs, light UAVs are used during USAR missions in NIFTi to get an overview. Because of their limited processing power, the UAVs are controlled by a significantly smaller software system, also based on ROS, such that they can be used mainly as flying cameras – either tele-operated or semi-autonomously.

The software system according to Fig. 3 is distributed between the on-board PC of the UGV and some stationary computers at the control center. Because of the message-passing features of ROS, this distribution can be very flexible and is defined by a set of launch files during run-time. However, the ROS stacks of the lowest layer, providing the drivers for the UGV and enabling the platform control, must definitely run on-board. In the layers above, there are several ROS stacks implemented to achieve particular capabilities needed for the USAR domain:

- object detection based on image processing, e.g. to detect victims automatically
- creation of metrical maps based on the LIDAR sensor mounted in front of the UGV
- 2D and 3D metric path planning
- multimodal (speech, GUI) human robot interaction based on the RViz package of ROS

These ROS stacks are connected to the CAST-based components by a bridge that transforms ROS messages to the CAST shared memory and vice versa. The higher layers, which are implemented as CAST components, include:

- conceptual and ontological understanding of the environment based on different kinds of maps (metrical, topological, functional)
- communicating the understanding from these concepts about the environment (including areas, objects and actions to be performed) to other high-level processes
- high-level planning and execution for joint exploration

These high-level components enable the UGV to communicate with a human through natural dialogue and to share its situation awareness within a mixed human-robot team. The following sections cover the essential contributions NIFTi achieved in its final year.

3.2 Robot-centric situation awareness

One of the ways to build robot-centric situation awareness is to design algorithms performing mapping of the robot sensory data into situation interpretation in the robot perspective (given for example by the interfaces of algorithms that the robot exploits). Such interpretation of the situation awareness vastly differs from the user perspective since its primary goal is the effectiveness of the used algorithms. In this field we contributed in two ways: first, we developed an algorithm for terrain perception—the *terrain-adaptive odometry* [5]; second, we implemented a robust *3D metric mapping* algorithm; both contributions improve the robot localization and are described below.

3.2.1 Terrain-adaptive Odometry

In our approach we exploited the concept of *sensing through body dynamics* [6] and combined state estimation techniques with machine learning. By using the same type of inertial sensors, the *Xsens MTi-G* unit, we even attempted to prove our concepts across platforms of vastly different morphologies. Initially, the proof of concept was tested and evaluated on a quadruped robot described in [7]. These results were then extended by a new methodology described in [8] and successfully applied to the NIFTi UGV platform [5], [9]. Furthermore, we developed algorithms necessary for processing the raw inertial data; see details in [10].

The *terrain adaptive odometry* algorithm was initially inspired by the research in robotic terrain classification (RTC) [11]. Contrary to the standard RTC approaches, we do not provide discrete terrain categories (usually labeled in the user perspective as sand, rubble, soil, etc.) but we classify the terrain directly by the values of coefficients correcting the robot odometry. These corrections make the odometry model naturally adaptable to the terrain due to inherent slip

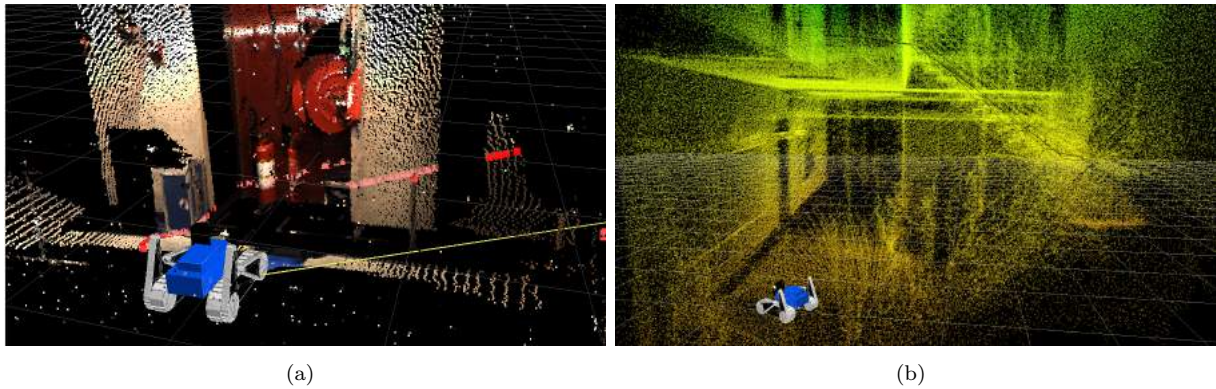


Figure 4. (a) Assembled point cloud with color information. (b) 3D Point cloud map of a staircase.

compensation. During experimental evaluation on rough outdoor terrain the overall improvement we achieved in root-mean-square error in position with respect to a state-of-the-art odometry model was approximately 68%.

3.2.2 3D metric mapping

Based on the terrain-adaptive odometry, we assemble individual laser scans into full 3D point clouds. Such a point cloud is a 3D representation of the contents of the field of view of the scanner at its current position. With proper calibration between the omnicaamera and the laser, it is possible to add color information to each individual point (see Fig. 4(a)).

In order to reconstruct a consistent representation, we use `libpointmatcher`¹, our open-source implementation of the Iterative Closest Point algorithm [12, 13]. With this implementation, we are able to process all point clouds online [14] and therefore to build a metric representation of the environment during exploration. An example of a result is shown in Fig. 4(b).

This processing also corrects the pose estimate of the robot which allows for a more precise localization information. It can also be used in a more robust localization framework fusing odometry, inertial measurements, visual odometry, and this laser odometry [15].

3.3 Bridging from robot-centric to human-centric situation awareness

Robots use their various sensors to build up a 3-dimensional structural representation of an environment. The primary purpose of that representation is to facilitate robot operations. These include localization, path planning, and autonomous adaptation and -driving.

We have introduced further steps of sensory interpretation, to facilitate more user-centric forms of situation awareness. Given the scenarios we focused on, these include object detection including victim detection, and functional mapping. The sections below describe these in more detail. Like the structural representations, these interpretations can be made available to the users in the multi-modal GUIs.

3.3.1 Object detection

Victim detection in disaster sites proved to be very difficult by using only visual information. We therefore build a layered sensory data representation, see Fig. 5. Thermal image (2D array of temperature) and depth images (2D array of distances) are reprojected (computed) as if they were captured from the same frame (possibly with some data misses due to occlusions). As the RGB-D sensor does not work reliably outdoors we construct a fused depth image, taking inputs from RGB-D and laser line scans, see Fig. 6 (a). The present detector uses depth as a scale estimate for aggregation of temperature and skin color likelihoods, sample results are shown in Fig. 6 (b) and (c), respectively. Distribution of human temperature is modeled as a mixture of

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

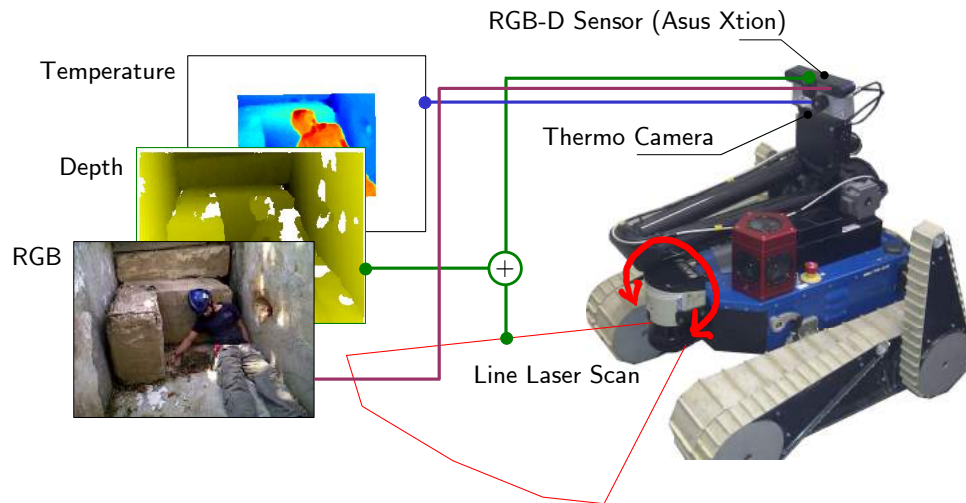


Figure 5. The scheme of multiple modalities (Color, depth, temperature) represented as a layered array computed from different sensors. The Lidar swivels from side to side, capturing full 3D.

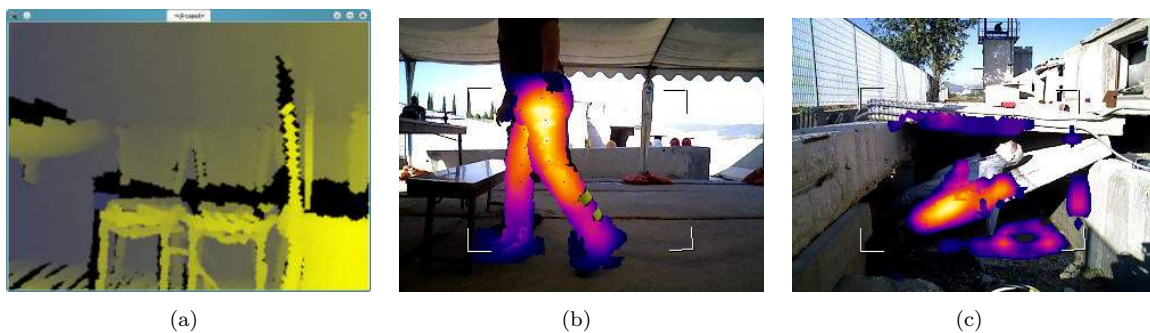


Figure 6. (a) Semi-dense depth computed from laser line scans, while robot undergoes motion (in this case rotation). (b),(c) Examples of human detection. In (c) the false detections are caused by concrete blocks being heated by the sun to the temperature around 36 degrees.

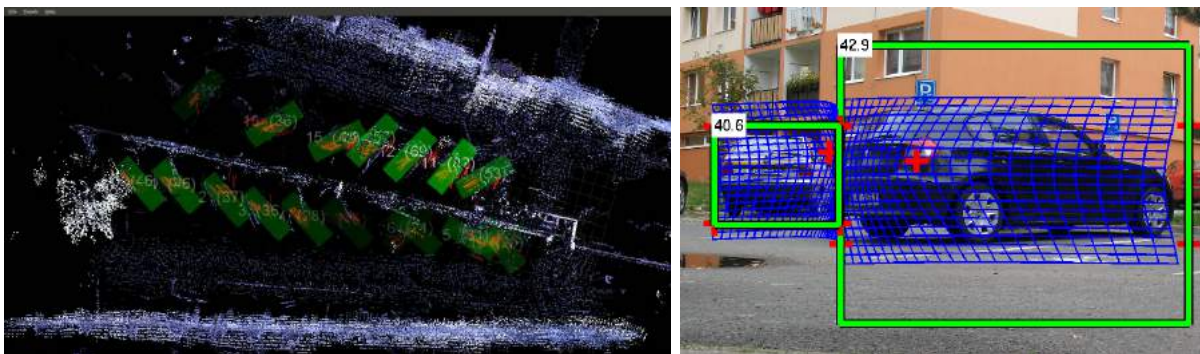


Figure 7. **Car detection and localization:** Left: colored point-cloud map with detected cars denoted by green rectangles. The number with the rectangle corresponds to the number of detections of one car accumulated over time. Right: Typical view point with car detections. Blue mesh corresponds to the estimated non-rigid alignment of the bounding box.

two components - skin and clothes. The parameters of the clothes' component are dynamically adapted according to the temperature of the background. In order to detect non-victim objects we developed a very efficient visual detector [16]. The detector discovers that the successively evaluated features in a sliding window detection process contribute not only to the confidence whether the object is present or not but also contain knowledge about object deformation. The standard sequential decision about the confidence is interleaved with feature warping. The same features are used for confidence and warp estimation. The interleaving process is an essential part of both learning and detection phase. A single detector can be thus used for deformed

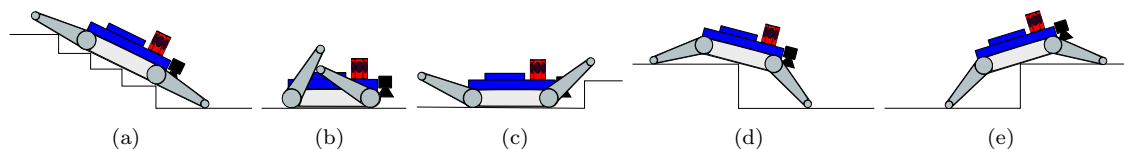


Figure 8. Five flipper modes corresponding to different morphological configurations of varying properties.

objects. The visual detections delineate relevant 3D point cloud data which are then used for distances and orientation estimation. Detections with 3D informations are sent to higher levels of the system. The detector has been applied mainly for car detection and localization, see Fig. 7.

3.3.2 Functional mapping

Robot-centric representations are, by definition, well-suited for robot tasks such as metric localization and navigation. However, in order to better communicate within the team, and especially with human operators, different representations are needed. The first user-centric representation we build is a topological decomposition of the environment into zones. We can perform this decomposition without any supervision [17] but we can also take hints from the users in order to seed the regions the user is interested [18].

These representations are also usually more compact and easier to reason with for high-level planning. Additionally to the topological representation, we also produce maps related to different functionalities the robot can perform; for example looking inside a car.

3.4 Autonomy and planning

Below we describe various levels of autonomous planning for different aspects of UGV locomotion. These include adaptive traversability, 3D metrical planning, and mixed initiative planning and dynamic control.

3.4.1 Adaptive Traversability

We define the *adaptive traversability* as means of motion control based on autonomous adaptation of robot morphology to traverse unknown complex terrain with obstacles. The aim is to reduce the degrees of freedom to be controlled (in our case the flippers) by the operator, therefore reducing the cognitive load. The main advantage of our approach is that no prior map or motion history is required, as only the latest incoming sensory data are processed—if some modalities provide data at lower frequency than others the latest measurement epoch is processed.

Input to the algorithm is laser data (expressed as Digital Elevation Map, DEM), motor torque signals, inertial data, and tracks odometry; the output is a robot control mode defined by a binary speed decision (stop or go) and a combination of discrete flipper configurations and their stiffness (hard or soft); see Fig. 8.

The control modes are selected automatically to maximize the *expected sum of discounted rewards defined by a reward function*; leading naturally to a reinforcement learning task. We defined the reward function as the weighted sum of (i) user denoted reward reflecting robot safety, (ii) high tilt angles penalty, i.e. negative reward, (iii) excessive flipper mode change penalty, (iv) robot forward speed reward (for making progress), and (v) motion roughness penalty.

To train the algorithm we created artificial obstacles using EUR pallets. To test the algorithm we were driving over natural complex obstacles in a forest environment and compared the performance to an expert operator using different criteria. For illustration a training obstacle and examples of testing obstacles are shown in Fig. 9. More details can be found in [9].

3.4.2 3D metric path planning

3D metric path planning for ground robot presents the difficulty to distinguish obstacles that need to be avoided from ground support. The standard approach is to have variants of elevation maps and do planning in 2D space [19]. This approach is clearly insufficient in case of multi-

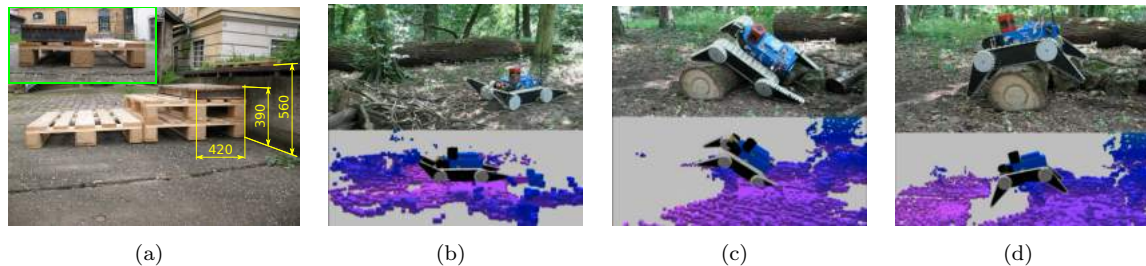


Figure 9. Examples of training (leftmost) and testing obstacles used for development of the adaptive traversability algorithm. The testing obstacles are shown with the corresponding DEM interpretation.

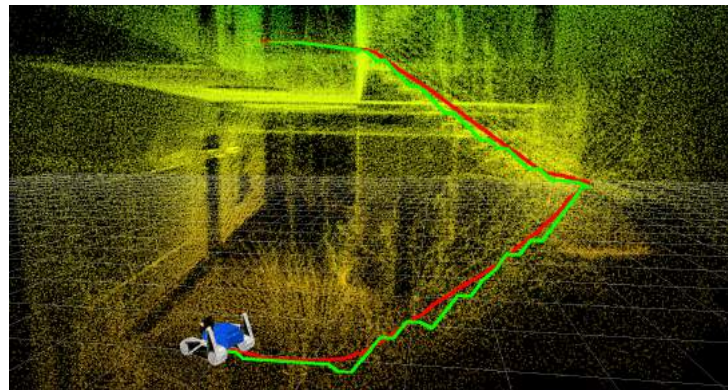


Figure 10. Autonomous descent of a complex staircase. Green line: the initial path planned by the robot. Red line: the actual robot path.

layered environments such as encountered in USAR applications.

Otherwise, planning is done on rich reconstructions of the environment such as polygon meshes. However this requires time-consuming processing and is not compatible with online planning in unknown environment [20, 21]. Instead of separating perception, path planning, and path execution, we propose to tightly integrate them in order to reduce latency.

Our algorithm takes a 3D point cloud as input, that will be processed on-demand with tensor voting [22] and uses the D*-lite search [23] in order to plan and replan efficiently during the path. We also select the best flipper position out of a repertoire allowing the robot to overcome most obstacles.

As an example, Fig. 10 shows a trace of execution of our ground robot in a rounded staircase. After a teleoperated exploration of the environment, the robot was asked to autonomously climb back down.

3.5 Planning, control and execution for human-robot collaboration

3.5.1 High-level representation of perception

To understand and reason about unstructured and dynamic environments, such as the USAR ones, high-level planning and execution have to primarily address a high level representation of the sensory inputs [24]. This requires to deal with the following problems: (i) extracting relevant features from raw data, gathered by the robot sensors; (ii) building a meaningful, higher level representation of this sensory information and (iii) mapping such a representation into a domain where both reasoning and decision making can take place. The above problems are faced by two complementary processes dealing with the topological and metric representation of the sensory data. The topological representation is defined by a graph of the environment (see Fig. 11). Nodes of this graph can be either objects, detected in the scene, by the visual detector system of the robot (see Section 3.3), or regions obtained by an unsupervised topological segmentation of the metric map (see the paragraph on *functional mapping* in Section 3.3, and

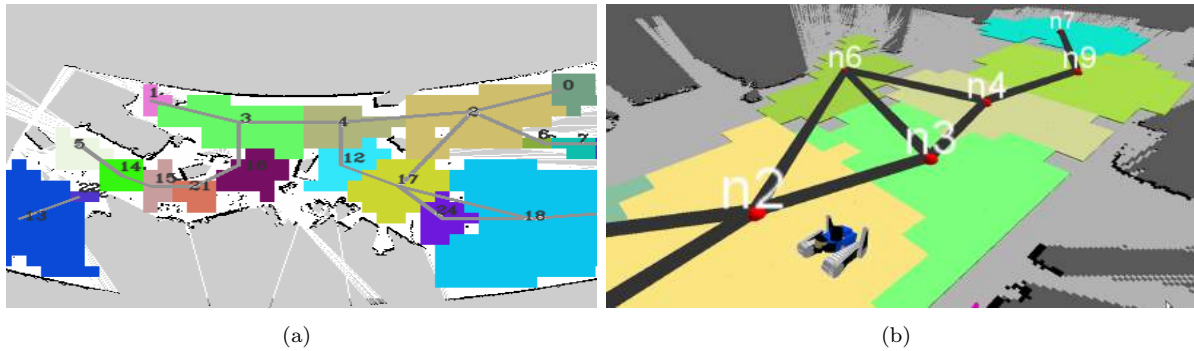


Figure 11. Graph-based representation of the topology of the firefighter training car accident scenario in Montelibretti (Italy)

also [17, 25]). Nodes are also annotated with properties related to the detected objects, wherein connections between nodes determine the traversability between the corresponding areas. This graph is directly translated into the domain knowledge supporting parametric planning [26].

Autonomous navigation tasks in a USAR environment demand a precise representation of the environment in terms of what is traversable and what is not, and in terms of what can be reached and what cannot, for this reason the topological map for 3D planning is complemented by a suitable processing of the point cloud. The point cloud data is segmented and each segment is labeled to provide a basic categorization of the environment, specifically defined for navigation purposes, that is, walls, ground, stairs, ramps, and obstacles that can be overcome (like fences, barriers, blocks of a specified height). Segmentation and labeling are made by the following steps: (1) point cloud filtering; (2) estimation of normals to the surface and curvature and, finally, (3) clustering and merging of the filtered point cloud. Clusters are labeled according to the geometrical constraints applied to the surface normals, to the mean curvature and to the points 3D coordinates [27, 28]. This process results in a classification of the point cloud into walls, stairs or ramps, and ground and surmountable obstacles as illustrated in Fig. 12(a). The semantic labeling of the clustered point cloud is then mapped to a suitable logic representation for inclusion in the knowledge base of the robot system.

3.5.2 Complementary strategies for planning

The robot system makes use of the above defined representation to reason about the environment [29], more specifically, on what can be reached and what cannot, which determines the system navigation strategies. Navigation strategies amount to choosing the best planning method to execute a task. Indeed, if the robot is currently moving on a flat terrain, such as the floor of a room, the robot system can select a 2D path planning algorithm to generate paths toward a temporary goal location, without resorting to a more complex algorithm, such as the 3D metric path planning (see Section 3.4). Another advantage of choosing among alternative planning strategies is that the system can recover from failures in the generation of valid plans, due to the high degree of complexity of the environment [30].

The robot system, in fact, chooses among three different path planning strategies, basing the choice on the terrain surface and topology and on the possible sources of planning failures. The three strategies are: (1) 2D path planning; (2) 3D path planning and (3) 3D graph-based planning [28]. The first strategy relies on the `move_base` navigation stack, provided by ROS¹. The second strategy is based on the 3D metric path planning, described in sub-section 3.4.2. The third strategy is performed on a graph-based representation of the environment (see Fig. 12(b)). This representation is obtained from the semantic labeling of the point cloud as follows. Points belonging to clusters, labeled as ground and stairs or ramps, are connected based on an iterative procedure taking into account both the model and the kinematic constraints of the robot, namely

¹<https://github.com/ros-planning/navigation>

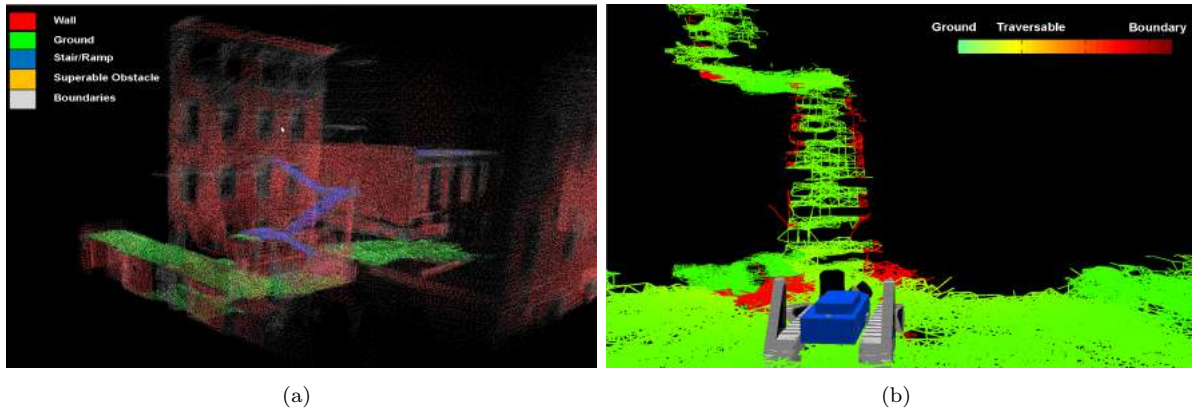


Figure 12. (a) Point cloud segmentation and labeling and (b) weighted graph representation of a fire escape stairs scenario.

its morphology as well as its ability to overcome obstacles. The result of this procedure is a graph connecting the different regions of the point cloud, denoting areas accessible by the robot. In parallel, both boundary and inflated obstacle regions are estimated by projecting the points labeled as walls onto the planes tangent to the surfaces approximating ground, stairs or ramps. Upon the estimation of the boundary regions, the edges of the connectivity graph are weighted by a factor taking into account the distance of the graph vertexes from these boundaries, the density of the neighborhood of the vertexes and the arc length of the edge. This traversability structure is used by the graph-based planning strategy to find minimum cost feasible paths toward target goals [27, 28].

3.5.3 Complementary strategies for motion control

The robot motion control system is augmented with an adaptive low-level control module taking care of switching among three different motion controllers: (1) a Reinforcement Learning (RL) controller (2) a trajectory tracking controller and (3) a flipper position controller [31]. The RL controller is based on the algorithm described in Section 3.4. The trajectory tracking controller, on the other hand, implements a control strategy based on input-output linearization via feedback [31]. The controller takes as input the current pose of the robot, obtained by fusing laser data with odometry and inertial data, the pose of a virtual reference frame, on the desired trajectory, a velocity profile, and generates the linear and angular control commands, in order to asymptotically stabilize the trajectory error to zero.

The flipper position controller locally adapts the position of the flippers to the surfaces on which the path lies, namely to the planes tangent to each point of the path. The controller computes the position commands of the flippers as follows. Four points are identified on the surface on which the current segment of the path lies. These points are representative of the contact points of the flippers with the surface. The controller estimates the normals of each of these points and generates the position commands of the flippers, on the basis of the orientation of the normals with respect to the global reference frame of the robot. Note that the estimation of the normals is not accurate. Moreover, the flippers are neither endowed with contact sensors nor with proximity sensors. Therefore, it is quite hard to correct the estimation as well as to determine the contact between the flippers and the surface. To face this limitation, the flipper position controller relies on a model of contact sensor. This model is based on a learned function, assessing the touch and the detach of the flippers from the surface. The parameters of this function have been learned as follows.

From the measurements of both the actual angles of the flippers and the electrical currents of the flipper motors the following features are extracted: (1) the average of the absolute values of the electrical currents within a fixed time window, (2) the average of the absolute values of the angular velocity of the flippers within the same time window, (3) the sign resulting from the product between the average of the electrical currents and the average of the angular velocity of

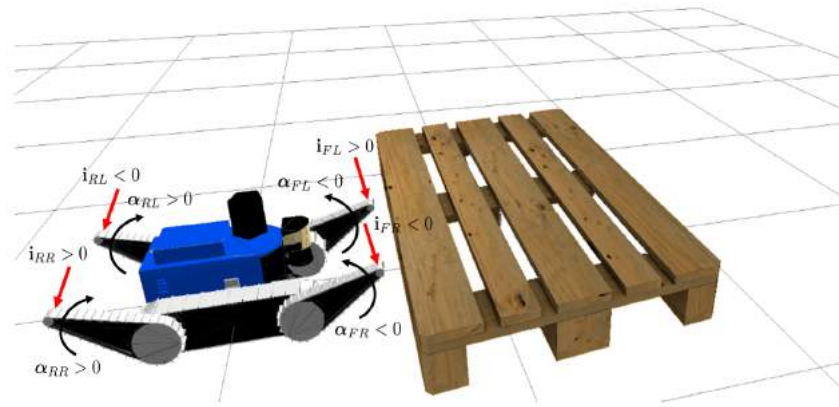


Figure 13. Relation between the values of the electrical currents and the directions of the forces applied to the flippers (red arrows). Relation between the values of the electrical currents and the values of the angles of the flippers (black arrows).

the flippers and, finally, (4) the average of the absolute values of the electrical current, filtered according to the Transposed-Direct-Form-II digital filter (see Fig. 13). The filter has been applied to reduce the oscillations of the signal current, during transient conditions of the servo motor, actuating the flippers. These features have been manually labeled to denote either the touch or the detach of the flippers from the surface. This data-set has been used to train a non-linear classifier, based on Support Vector Machine (SVM). A degree-d polynomial kernel has been chosen due to the non-linear separability of the data-set. The flipper position controller activate this contact sensor to correct the estimation of the position commands. The adaptive low-level control module can decide whether to activates the trajectory tracking and the flipper position controller on the basis of the values of the slip ratio of the tracks, to allow the robot to track a given 3D path and to ensure that the robot has a better traction on the harsh terrain [28].

3.5.4 Autonomous planning and mixed-initiative

The design of an autonomous planner taking into account mixed-initiative requires: (1) task sharing between the robot and the operator; (2) to provide the operator with a clear explanation of the robot behaviors, (3) to allow the operator to choose the level of autonomy of the robot during deployment, on the fly and, finally, (4) to generate goal strategies for information maximization. According to these requirements, we developed a control system for the UGV, with dynamic adjustment of the level of autonomy [26, 30]. The control system coordinates the interventions of the human operator and the low level robot activities, under a mixed-initiative planning perspective. More precisely, the control system is based on a declarative model of the activities of the robot, specified in the Temporal Flexible Situation Calculus (TFSC) [32–35]. The model explicitly represents the main components and activities of the robot, the cause-effect relationships as well as the temporal constraints between the activities. Further, the model integrates a representation of the activities of the human operator, enabling the control system to supervise his/her operations. A flexible planning engine (i) monitors the consistency of the robot and operator activities, with respect to the model, managing failures and (ii) incrementally generates plans, allowing the operator to locally assess the robot operations. Both the TFSC model and the flexible planning engine are implemented in ECLIPSE Prolog [36] which optimally combines the power of a constraint solver with logical inference, in order to generate plans. The model also ensures the continuous update of the system knowledge with incoming new information. A hybrid CAST subarchitecture has been designed to embed the TFSC model and the flexible planning engine, as well as the ROS nodes driving the communication tasks with the ROS layers (see Fig. 14). The choice of designing the hybrid CAST subarchitecture, effectively linking the ROS functionalities with the TFSC model is to overcome two main limitations of ROS: (1) the lack of a native mechanism for sub-typing and polymorphism to manage the data,

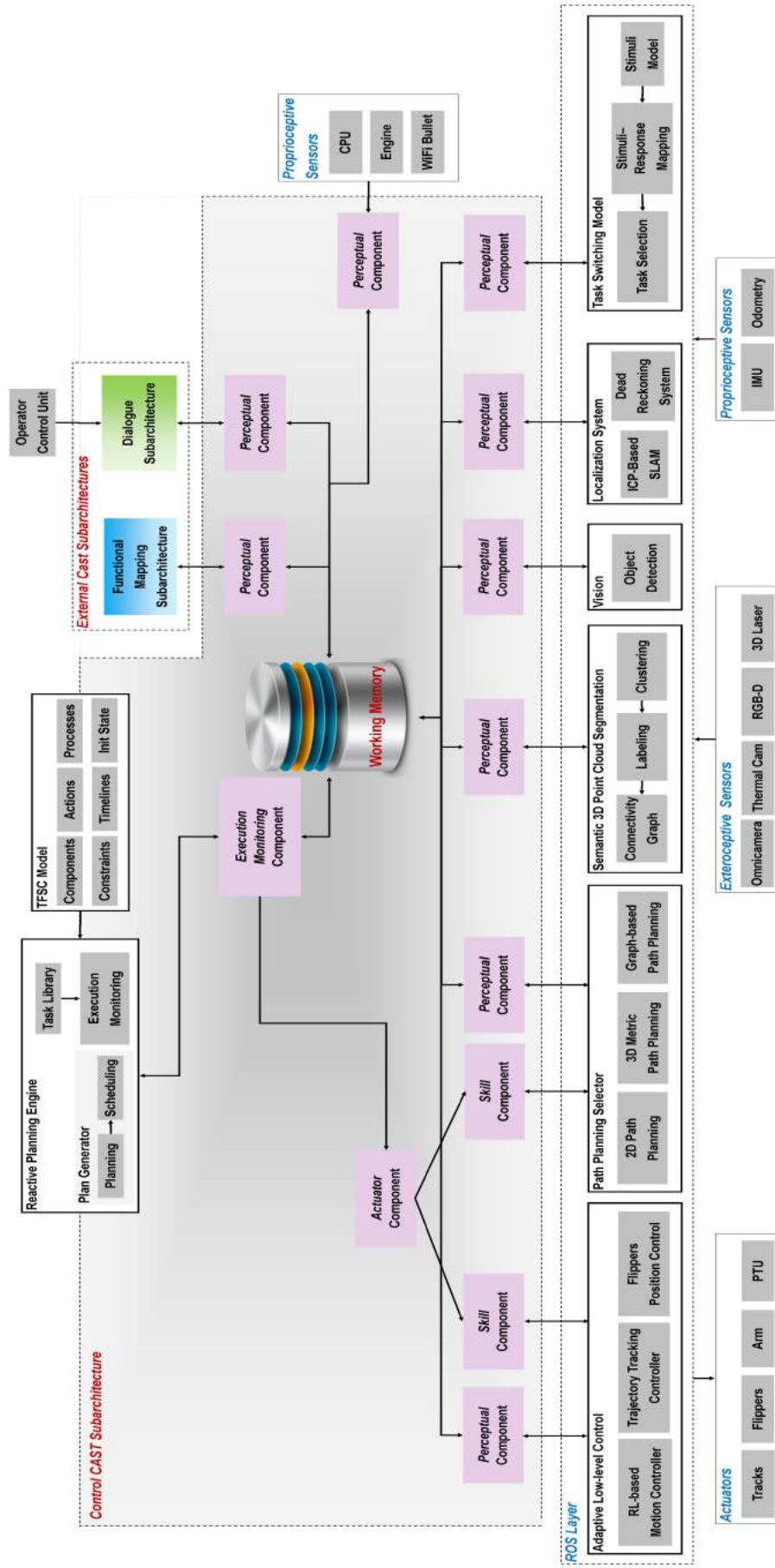


Figure 14. Subarchitecture CAST for semi-autonomous control.

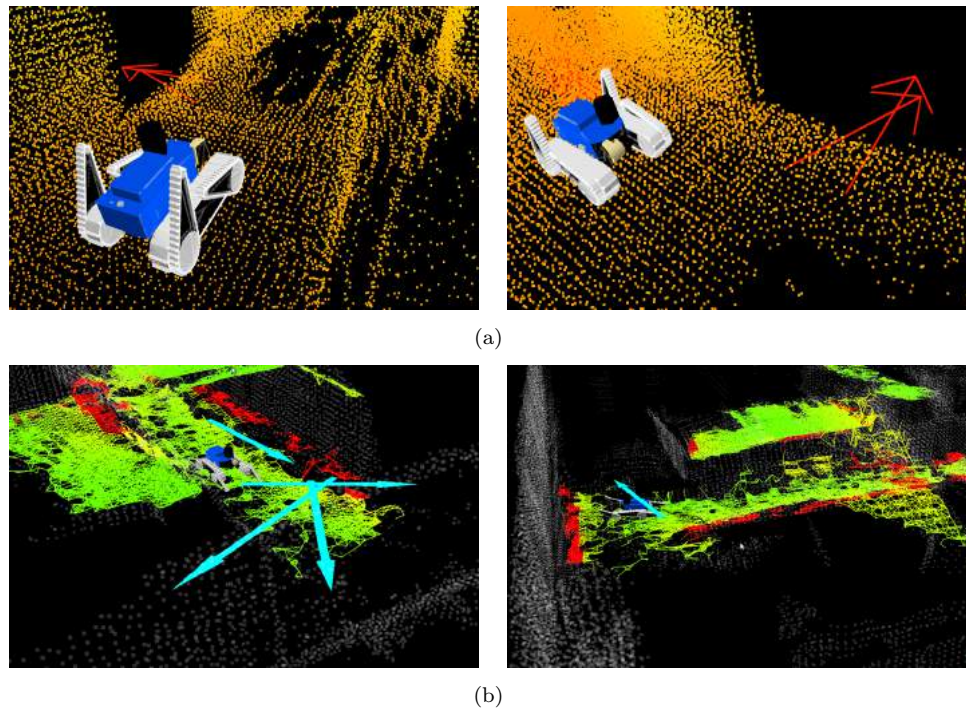


Figure 15. Next Best Pose generation: (a) the candidate poses (red arrows) suggest the robot to pass through a doorway of the corridor; (b) the candidate poses (cyan arrows) favor the acquisition of new information within the explored rescue area.

and (2) the lack of a structure to store the data, for planning and reasoning.

The core of the planning subarchitecture is implemented by the *Execution-Monitoring* (EM) component which plays a crucial role in orchestrating the other components of the subarchitecture. The EM both manages the communication with the human interfaces and maps the logical part of the control into the CAST subarchitecture. Furthermore, the EM manages the communication among all the components of the CAST subarchitectures, forming the robot system, and with the ROS modules implementing the sensory modalities of the UGV robot. The information gathered by the system is stored into the Working Memories (see Fig. 14), and mapped to the domain knowledge of the flexible planning engine. The EM sends task activation signals to the actuator components of the subarchitecture that finally execute the actions generated by the planning engine.

The overall control schema (see Fig. 14) implements several hybrid operative modalities ranging between autonomous and teleoperated modes, available during the execution of a task. The human operator can manually control some functional activities of the robot, scheduled by the flexible planning engine. For example, the operator can take control of the motion to explore an interesting location or escape from difficult environments, by suspending the robot autonomous navigation task. The operator can also modify the control sequence produced by the flexible planning engine, by skipping some tasks or inserting new operations. One of the most sought after operative modalities in navigation is to suggest exploration strategies for the rescue operators, based on information maximization. This modality requires to determine candidate poses, to be reached by the robot, from which most of the unknown space can potentially be observed by both the robot and the operator. Under this perspective, the problem of generating the next best observation positions of the robot can be formulated as a Next Best View (NBV) problem [37–39] where candidate poses are weighted by an utility function.

The utility function maximizes the information returned by a pose in space by estimating the volume of unknown space, falling into the viewing volume of the laser sensor or the robot. More concretely, this value is implemented as follows. For each of the rays cast from the origin of the laser reference frame toward all the possible directions a local utility is computed. The local



(a) Geo referenced information

(b) Mobile display

Figure 16. Content display.

utility takes into account the amount of free, occupied and unknown cells of the 3D occupancy map of the environment, hit by the ray.

Then the global utility associated with a pose is computed by integrating the local ones on the view cone domain. To speed-up the computation of the utility of a pose, rays are sampled within a solid angle, bounded by the limit angles of the laser sensor. Moreover, candidate poses are sampled from the connected component of the traversability graph (see Fig. 12(b)), on which the robot is moving, according to a Gaussian distribution, conditioned by the current position of the robot and the past position history. The next best view is the pose, among the sampled candidate poses, which maximizes the defined utility function. Fig. 15 shows how the robot infers a subset of next best views, on the basis of such exploration strategy.

The flexible planner, when in exploration modality, uses this strategy to generate goals and infer the adequate path to reach them, and evaluate with the EM the feasibility of the plan.

3.6 User-centric/adaptive interaction and situation awareness

Key to effective human robot teaming is proper communication of information between the different team members, to build up shared situation awareness. The human team members are ultimately the main stakeholders in this process. In keeping with the overall design methodology, we have also here adopted a strong (human) user-centric development. Below we describe the development of our interaction setup, and provide more details about two aspects that help bridging the gap between robot-centric environment models and human understanding.

We initially started by investigating and supporting multimodal one-to-one communication between a robot and an operator. After the experience of our initial pilot studies, we gradually shifted our focus to team-level communication. As the team was gradually extended, sharing and managing situation awareness became a crucial issue. To facilitate this, we designed and implemented ever more advanced multimodal user interfaces.

We initially focused on communication to support human-guided exploration by the robot. The human operator was remotely located, outside of visible range of the robot operating in the hot-zone. We designed and implemented an operator control unit (OCU) that facilitated multi-modal interaction (GUI, spoken dialogue) between a human and a robot. Based on the experience from the pilot study, we formulated a first approach to modelling the dynamics of the interdependent roles in a human-robot team. We focused primarily on determining how communication between roles in a team is affected when the team performs under varying stressful circumstances, so that a robot can adapt its multi-modal communication strategies given online human performance. This provided the setup for developing communication to support human-assisted exploration in the context of a human-robot *team*. This went beyond the originally envisioned setup of a single operator.

The physical setting for the interaction remained such that most of the human team is located

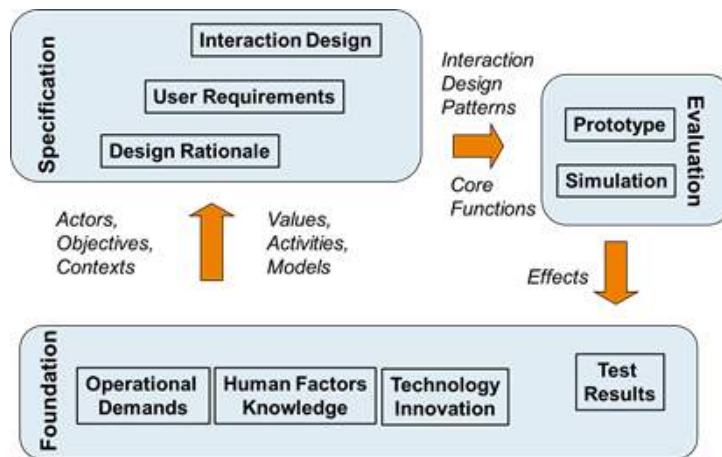


Figure 17. Situated Cognitive Engineering (sCE) design and evaluation process [42].

at a remote command post, outside of visible range of the robot operating in the hotzone. We developed a multi-view user interface to facilitate different views on information in the human-robot team, to help support different roles in the team. The interface provides multiple modes of communication, including touch and spoken dialogue. Views (possibly shown on multiple monitors) include the visualization of information from the various robots (UGV, UAV), and team situation awareness.

Beyond the remote command, we have extended the human-robot team setup to include an in-field human rescuer. This resulted in a further geographic distribution of both robot and human team members. We developed and investigated different versions of mobile interfaces to facilitate multi-modal communication between the in-field human rescuer, and the rest of the remotely located human team members. We followed the approach to support different team roles with different views and functionality for human-robot team interaction. In this context we also developed a novel method for content adaptation, i.e. presenting the “right material at the right time, in the right modality” [40, 41]. As information from an in-field rescuer typically comes “asynchronously” relative to when a robot is in the observed area, we have developed new means of storing and presenting such (geo-referenced) information at the operational and tactical levels of communication in the human-robot team (see Fig. 16).

3.7 Teaming

In order to support the team effort, the system builds up and maintains knowledge on the users, supporting them to stay in a continual workflow, by attuning the information processing and sharing to the task at hand. First, this involves making sense of what the users are doing, in terms of their current task, their cognitive task load (CTL) and emotional state. Second, the team and user context can be exploited by suggesting the appropriate level of autonomy for the task at hand, or notifying other team members to help their colleague.

The NIFTi system gives us a rich environment for collecting data on the users’ behavior, and for the design and implementation of such team support functionality. Based on the data gathered in our end user experimentation, we have tailored the CTL and emotional state models, and integrated them into the prototype system, where they were incorporated into a formal framework for dynamic task allocation and adaptive dialogues. We also collected a data-set for conducting 3D eye-tracking experiments to further develop computational visual attention models for top-down search tasks. Over the course of the project, we have also refined the methods for balancing the information transfers by means of policies for information exchange of team members with specific roles and capabilities. These working agreement policies, designed in collaboration with our end users and evaluated during the end user experiments, aim at establishing coherent and

	USAR robot	
Communication level	Interaction design pattern: for look and feel of smart question	Premise: smart questions are easy to use questions
Task level	Requirement: Provide automatic object recognition	Claim: Improved object detection
Group level	Requirement: Provide information on location and recognized objects teammates	Claim: Improved object detection

Table 1. Design patterns, premises, requirements and claims on different levels.

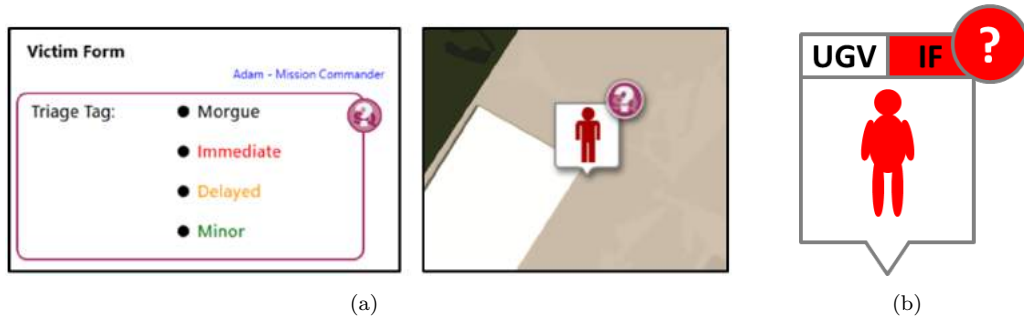


Figure 18. (a) Explicit unknown design and (b) re-design.

re-usable specifications of human-automation interaction at the communication level.

4. Experimentation

Each year the project went through a development cycle of a description of scenarios and use-cases, derivation of requirements and hypotheses, implementation of the system components, integration, and finally an evaluation. Fig. 17 shows the three components of this situated Cognitive Engineering (sCE) methodology: the foundation entails operational, human factors, and technological analyses to derive a sound and practical design rationale, the specification and maintenance of the requirements baseline, and the evaluation by means of simulation or a prototype, to validate and refine the requirements baseline [43], [42]. The requirements and hypotheses were described on three levels, the communication level, the task level and the group level [44]. An example of each level in the context of a robot for USAR is given in Table 1. The task level and group level can both be evaluated with small tasks (Task battery) or within a scenario.

4.1 Design patterns

The manner of information presentation influences the performance of the user enormously. To approach the evaluation of this presentation in a structured manner and to support development and reuse of good design solutions we use design patterns [45]. Table 1 provides an example of a design pattern that has been evaluated in the fourth year of the project and Fig. 18 provides the evaluated design and the suggested design improvement based on an evaluation. In the evaluation, 18 subjects evaluated the design using a questionnaire and they noted some problems in understanding who was responsible for answering the information request. Therefore information about who should react on the information request was added (the UGV (operator) in the case of the improved design figure).

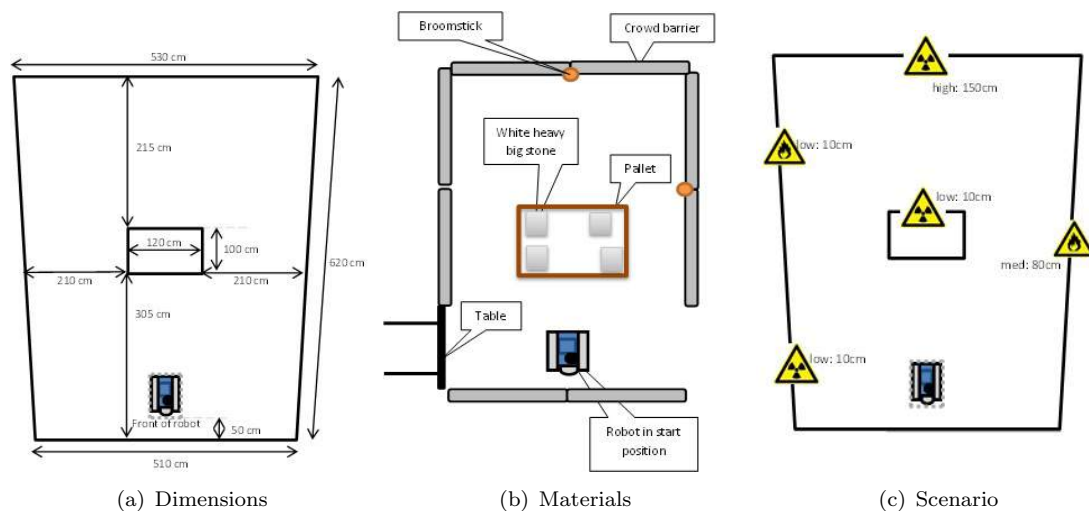


Figure 19. Set-up of the detect objects task

4.2 Task battery

To benchmark the progress in the yearly changing scenarios (see Section 2.2) we developed small tasks that evaluated some basic functionality (e.g. stopping the UGV before collision). This set of tasks was extended over the years to incorporate the growing capabilities of the systems. A first year task was for instance ‘detect objects’ in a maze like environment (see for example Fig. 19) to look at the performance of the combination UGV, UGV control interface, situation awareness display and operator. This task was kept over the years to see if changes in the combination had an (positive) effect on operator performance (task level). Because in the fourth year there was a complete team of two operators and a commander the same task was performed but with both a UAV and UGV (group level). This made it possible to not only compare the results over years, but also over different team compositions. The complete system with UAV and UGV and their operators and a commander provided in the fourth year of course involved opportunities which entailed an extra test where the complete team had to detect objects in an area.

4.2.0.1 Measures. Each task had its own set of measures. For the object detection task these were: performance (How many objects are detected) and situation awareness (participant has to draw how they drove through the area and draw the UGV’s start and end position). To measure the performance on a task like ‘explore area’, the performance was measured as amount of area covered, the efficiency was measured by taken path (ratio length of path and explored area) and the same measure for situation awareness was taken.

4.3 Scenario

Next to the task battery, the scenario as described in Section 2.2 was performed by multiple firemen each year. The setup of the fourth year (see Fig. 1) shows that a capability such as objects detection is also relevant here. The scenario is less structured than the task battery tasks but the same requirements can be tested. Another difference from the task battery is that multiple tasks are to be executed and the participants always work in a team and are free to use the operation mode of the UGV they prefer.

4.3.0.2 Results. One of the things we are interested in is if performance on the task battery is predictive for the scenario performance. In [46] it is shown that the “detect object” task does partially predict the scenario performance.

5. Deployment

In May 2012, two major earthquakes occurred in the Emilia-Romagna region, Northern Italy, followed by further aftershocks and earthquakes in June 2012. This sequence of earthquakes and shocks caused multiple casualties, and widespread damage to numerous historical buildings in the region. The Italian National Fire Corps (CNNF) deployed disaster response and recovery of people and buildings.

In June 2012, they requested the aid of NIFTi, to assess damage to historical buildings, and cultural artefacts located therein. To this end, NIFTi deployed a team of humans and robots (UGV, UAV) in the red-area of Mirandola, Emilia-Romagna, from Tuesday July 24 until Friday July 27, 2012. The team worked closely together with the members of the CNVVF involved in the red area. Below we briefly summarise our experience; see [47] for more detail.

In Mirandola we deployed a subset of the available NIFTi functionalities, described above. We focused on robust functionalities for robot control, video streaming from various omni-directional and monocular cameras (UGV,UAV), and laser-based 3D reconstruction of the environment (UGV), coupled to the NIFTi multi-modal OCU.

We deployed two NIFTi UGV platforms in Mirandola: One as the main system, and one in reserve should something go wrong. Fig. 20 shows the UGV platform used. In addition to the usual sensor suite, we mounted a 25cm-tall static mast on the battery compartment of the robot. On top of the mast was a pan-tilt unit with a Kinect camera. This provides a chase-style view of the robot, which is highly useful when navigating (tele-operating) the robot in tight or complex spaces – cf. also the recent experience with Quince reported in [48]. During the deployment we had the robot running all day long, under outside temperatures of 35 – 40 °C and operating temperatures inside the robot up to 95 °C. Batteries only needed a recharge in the evening.



Figure 20. NIFTi UGV with a rotating SICK-Laser (LMS100), a LadyBug3 omnicaam, active flippers and active/passive bogeys, IMU, GPS, and a static mast mounting a PTU with a Kinect sensor.

Two different types of UAVs were prepared for the mission (Fig. 21). One was a NIFTi UAV microcopter platform, the other a research platform which we could flexibly outfit with a variety of cameras, e.g. a high-definition camera or an ASUS Xtion Pro.

The human-robot team included operators for the UGV and UAV, UGV and UAV mission specialists, and a Mission Commander. Both the UGV Operator and the UAV Operator suffered from cognitive overload. UGV missions typically lasted about half an hour, and were characterized by interleaving driving, and observing. This interleaving made it possible for the UGV Operator to relax, momentarily; a luxury the UAV Operator did not have. The UAV did have some degree of autonomous flight control, but circumstances demanded that the UAV Opera-

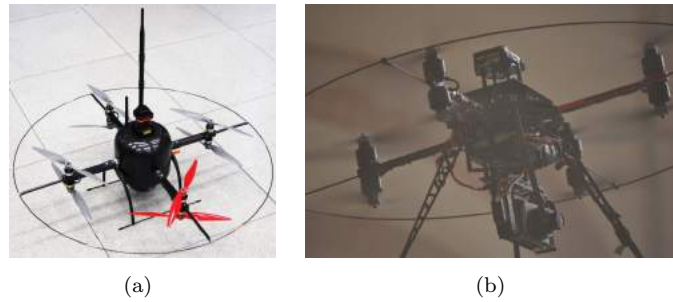


Figure 21. Standard NIFTi UAV octocopter with a standard configuration (a) and NIFTi UAV research octocopter (b) with a mounted on top camera, and a camera in a tilt unit under the main body. Another configuration flow includes a PC and a Kinect-style sensor mounted on top of the research UAV.



Figure 22. 3D reconstruction based on NIFTi UAV data

tor continuously attended to the UAV. This provides a first insight in possible roles of “robot autonomy.” In human-robot teams, humans and robots are (inherently) interdependent [49]. Robots can go where humans need to but cannot, whereas humans can aid robots in better understanding and operating in the environment. Both humans and robots are problem-holders – with the obvious “but” though that *the human users are the stake-holders*. Robot autonomy is ultimately to be in service of the human user, to reduce cognitive load (improved autonomous navigation, sensor data interpretation, collaboration) and to improve the possibility for the hu-

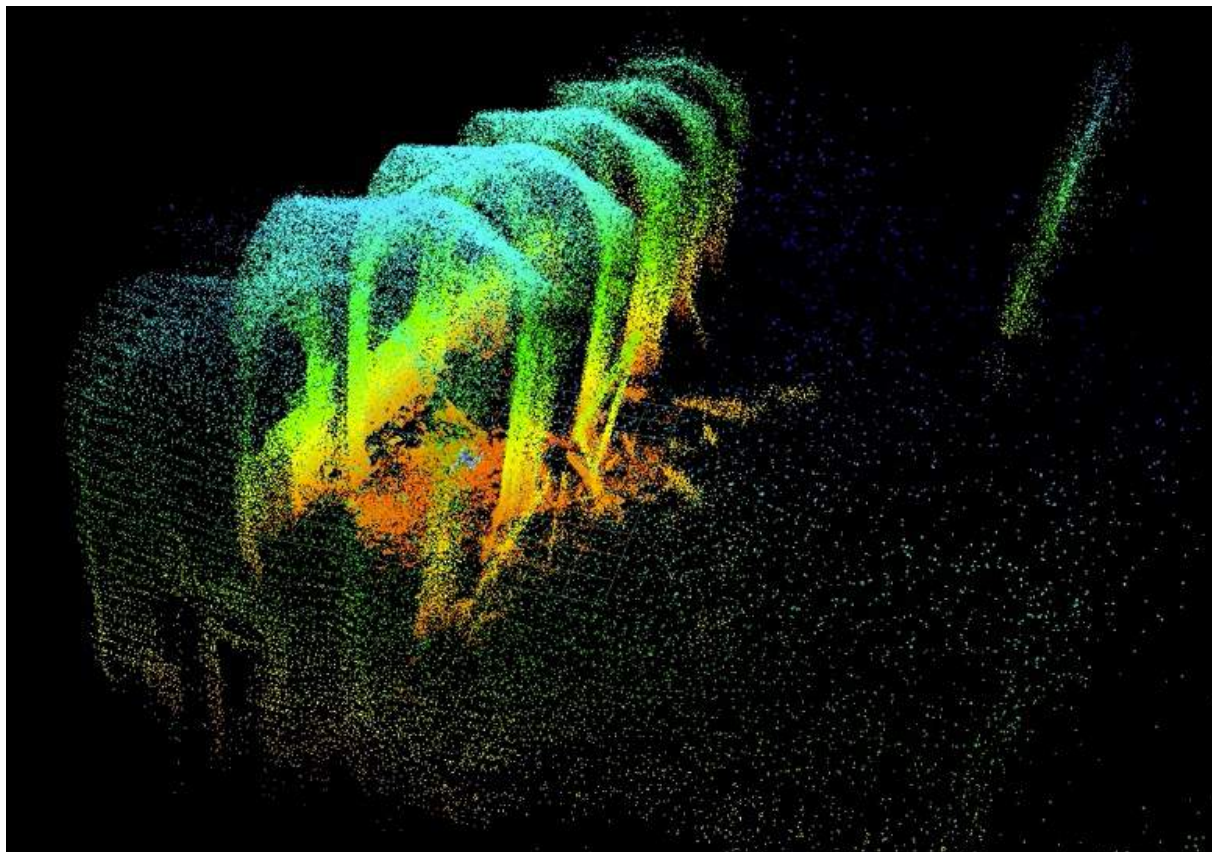


Figure 23. 3D map constructed by the NIFTi UGV.

man to collaborate with the robot as if “operating the world rather than the robot” [50]. We saw this over and over again during the deployment: Autonomy is to make life easier for the human to understand the environment.

The UAV serves as a good example here. The UAV mission specialist used augmented reality eyewear (*Vuzix WRAP 920AR+*) to watch the video stream from the camera mounted in a tilt-unit under the UAV. This quickly led to a pseudo-immersive experience, and the desire to look left-and-right and have the UAV and/or the tilt-unit follow suit. More (and better) flight control autonomy, enabling the UAV to simply hover and turn on the spot, would have facilitated this.

Further insights concern the flow of information between the UAV team and the UGV team, in terms of tactical (team-level) situation awareness (tacSA) and mission planning. During the entire deployment, the UAV team and the UGV team never operated in the same area simultaneously. Partly, the reasons were technical (network) and environmental (dust). Another reason regarded the *use*, the workflow which emerged in using information from the different teams in establishing further missions. Based on in-field line-of-sight observations of the area to be deployed in, and a first set of recon missions by the UAV team, we would establish a first sketch of the environment. Most importantly, we would identify important landmarks to navigate by, establishing explicit names for them (e.g. “column 4”), and determining targets for future missions. These targets typically included areas and objects to be observed, and how these observations were to be made. Targets were discussed together with members of the CNVVF.

Follow-up missions then helped detail out situation awareness and revise mission targets. Since awareness was coming from the different teams, we occasionally found mismatches in expectations which then required further missions; (as was to be expected, cf. [51]). For example, video from initial UAV recon missions in San Francisco church gave the impression that the top of the nave would be reachable from the western aisle, either from between the fourth and fifth columns, or the opening behind that. This would then make it possible for the UGV to drive

close to the altar, and provide close-up video. As it turned out at the end of the second UGV mission, what seemed accessible terrain from the viewpoint of the UAV, was not so in UGV-reality. The UGV did manage to take video of the altar, but an additional mission was then planned for the UAV to fly in over the main nave and record video from that viewpoint.

The UAV and the UGV thus supported each other, but indirectly so. It did result in the required situation awareness for the team, and the other stake-holders. At the same time, it also opened new questions as for how to optimally transfer data from one mission to the next, to make consolidated awareness available online. Before the deployment, we had developed a basic viewer for post-mission analysis. During a mission, a Mission Specialist could take snapshots in an OCU, annotate them with a description. Snapshots were stored with the text annotation and robot position information. For post-mission analysis, the viewer could then load snapshots and a 2D map, mark the snapshots on the map, and enable the user to browse snapshots. We did use some of this functionality, particularly to get high-definition snapshots of cultural artifacts, but what was missing was the possibility to correlate *geo-referenced video* from one mission, and show this during another mission in a *context-aware fashion*, i.e. show previously recorded video of the environment in which the robot in the current mission is located. This is a form of information fusion to provide continuous situation awareness across different missions within a single area. We made similar observations about map information. The UAV could be deployed to gather a 3D reconstruction of the environment. This map would not need to be so detailed as to enable the UGV to localize itself in it. All the map would need to make possible is a form of *forward mapping/scouting* for the UGV team to determine the optimal path amongst different alternatives. While operating in a harsh environment like the ones in Mirandola we would have greatly benefited from such functionality, as it could have saved time, or have indicated paths where none were obvious (like a traversal from the western aisle to the nave in San Francesco church). See Fig. 22 and 23: Coupling the UAV 3D information to the dense 3D metrical representation for the UGV could improve situation awareness for the Operator as well as the robot.

In summary, we observed several issues regarding the operations of a geographically distributed human-robot team, with team members operating both in-field and at a remote command post. As the UGV and UAV teams operated asynchronously, maintaining and transferring situation awareness between missions was an issue to the extent that system automatization could help (in the future) to make aspects of operational situation awareness from one team available to the next in an operational context-aware fashion. We thus need to address persistence of information.

6. Conclusions

We presented an overview of what we achieved in the NIFTi project. We have robust models of 3D dynamic environments, fusing information from a wide variety of sensors. We have highly robust and adaptive robot platforms (UGV, UAV) that use these models to operate in the complex environments typical for disaster response. And we have embedded all of that information, all these platforms, into the use context of a human-robot team. Humans have access to information at different operational levels, to form an assessment of the overall situation, and collaborate with robots as team members to guide further operations. The way information is presented takes into account that these contexts are stressful, with people working under varying cognitive load. What is displayed, how, and when, is adapted to fit the current load and usage. While further improvements are possible and needed at the level of the individual functionalities and system components, the important global achievement of NIFTi is that robots, and the information they provide, have been made useful to people. NIFTi achieved this through closely integrating research and development with a scenario-driven roadmap, end users, and real-life experiments. Reality, and real end user demand, drove the NIFTi R&D – and the NIFTi R&D showed that, despite the fact that robot-assisted disaster response is a complex and difficult task, we can

make substantial advances towards real-life deployments of these systems. The NIFTi Mirandola deployment is an example of that.

Through our practical experiences we have learnt that it is important to go beyond the single-robot single-operator paradigm and consider the operation of the response team as a whole with the robots as team members who collaborate with humans as well as among each other. Clearly, more research is needed in this area, to further improve distributed situation awareness as well as dynamic levels of autonomy, adapted to task, the situation and the team-members capabilities and needs.

Another aspect the importance of which we have identified through our practical experience is that such missions take time. In Mirandola we ran multiple missions over one or more days to explore a site. A robot does not just drive in and out, and the mission is done. In order to support this, a system needs to maintain and continue to update information over the course of missions. Information, and human-robot team experience as such, must become persistent. Based on our experience we see persistence as a key challenge for further progress in human-robot disaster response: we need persistent models for perception, of acting, of distributed joint situation awareness, of collaboration and human-robot teaming. We are addressing the challenges involved in taking such *long-term ecological perspective on robot-assisted disaster response* in our next project, TRADR [52].

Acknowledgments

The work reported in this article was supported by the EU FP7-ICT-Project NIFTi, Contract No. 247870.

Notes on contributors



Geert-Jan Kruijff is a Senior NLP Research Manager at Nuance Communications Germany, and a Research Fellow at the German Research Centre for Artificial Intelligence. From 2004 until 2013 he was active in the field of Human-Robot Interaction, focusing on situated dialogue processing. He led the HRI efforts at DFKI Saarbrücken, and was coordinator of the NIFTi project for the period 2010-2013. In 2013 he joined Nuance, where he leads the development of language resources for cloud-based natural language understanding, for a variety of consumer projects in the mobile and automotive domains.



Francis Colas is currently senior researcher at Inria Nancy Grand Est. After his Ph.D. in Computer Science from the INPG in 2006, he was postdoctoral fellow at Collège de France in Paris in the *Laboratoire de Physiologie de la Perception et de l'Action* (2007–2008). From 2009 til 2014 he was a post-doctoral fellow at the Autonomous Systems Lab of Prof. Siegwart (ETH Zurich). His research interests include Bayesian modeling of perception and action applied from cognitive sciences to robotics.



Mark Neerincx Mark Neerincx is full professor in Human-Centered Computing at the Delft University of Technology, and senior research scientist at TNO Perceptual and Cognitive Systems (Netherlands). He has extensive experience in fundamental and applied research on human-automation collaboration, among other things in the domains of health, security, defense, and space; as initiator and project leader, in large national and international research programs. Recent projects focus on the situated cognitive engineering of electronic partners (ePartners) that enhance the social, cognitive and affective processes in human-automation (robot) teams.



Fiora Pirri is currently full professor and head of the ALCOR Laboratory at the DIAG, Department of Computer, Control, and Management Engineering, "A. Ruberti" at Sapienza, University of Rome. She founded ALCOR Lab in 1998 and since then the Lab has widely to the theory and practice of Cognitive Robotics applied to rescue environments, winning several awards among which the AAAI-2002 Robot Exhibition, third award and finalist at the RESCUE RoboCup competitions of 2004, 2005 and 2006. Best paper award at ISVC 2010, finalist at SSR2013. Her research area is Cognitive Robotics, and in particular Perception, Attention and theories of Actions, on which she contributed with more than 130 peer reviewed papers which can be consulted on Google Scholar.



Dipl.-Inform. Rainer Worst: Senior researcher at Fraunhofer IAIS. Diploma in Computer Science (Univ. Bonn, 1981). IT-Expert at a consulting company in Wiesbaden. In 1984 joined the former GMD (National Research Center for Information Technology) in Sankt Augustin, which merged 1999 with Fraunhofer. Experience as a consultant, researcher and project leader in the areas Software Engineering, Quality Management, Autonomous Systems, and Virtual Environments.



Tomáš Svoboda: Ph.D. in artificial intelligence and biocybernetics from the Czech Technical University in Prague in 2000. Post-doc with the Computer Vision Group at the ETH Zurich (Swiss Federal Institute of Technology) for 3 years. Currently associate professor and deputy head of the department of Cybernetics at the Czech Technical University. Research interests include omnidirectional cameras, image based retrieval, and machine perception for robots in general.

References

- [1] NIFTi (Natural Human-robot Cooperation in Dynamic Environments). 2013. Available from: <http://www.nifti.eu>.
- [2] van Diggelen J, van Drimmelen K, Heuvelink A, Kerbusch PJ, Neerincx MA, van Trijp S, Ubink EM, van der Vecht B. Mutual empowerment in mobile soldier support. *Journal of Battlefield Technology*. 2012;15(1):11.
- [3] ROS home page. 2014 March. Available from: <http://www.ros.org/>.
- [4] Hawes N, Wyatt J. Engineering intelligent information-processing systems with CAST. *Advanced Engineering Informatics*. 2010;24:27 – 39. Available from: <http://www.cs.bham.ac.uk/~nah/bibtex/papers/haweswyatt10aei.pdf>.
- [5] Reinstein M, Kubelka V, Zimmermann K. Terrain adaptive odometry for mobile skid-steer robots. In: *Robotics and automation (icra), 2013 ieee international conference on*. 2013. p. 4706–4711.
- [6] Iida F, Pfeifer R. Sensing through body dynamics. *Robotics and Autonomous Systems*. 2006; 54(8):631 – 640. morphology, Control and Passive Dynamics.
- [7] Reinstein M, Hoffmann M. Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer. In: *Proc. ieee int robotics and automation (icra) conf*. 2011. p. 617–624.
- [8] Reinstein M, Hoffmann M. Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. *IEEE Transactions on Robotics*. 2013 April;29(2):563–571.

- [9] Zimmermann K, Zuzanek P, Reinstein M, Hlavac V. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In: Proc. IEEE Int Robotics and Automation (ICRA) Conf. 2014. Accepted for publication.
- [10] Kubelka V, Reinstein M. Complementary filtering approach to orientation estimation using inertial sensors only. In: Proc. IEEE Int Robotics and Automation (ICRA) Conf. 2012. p. 599–605.
- [11] Tick D, Rahman T, Busso C, Gans N. Indoor robotic terrain classification via angular velocity based hierarchical classifier selection. In: Proc. IEEE Int Robotics and Automation (ICRA) Conf. 2012. p. 3594–3600.
- [12] Chen Y, Medioni G. Object modeling by registration of multiple range images. In: IEEE International Conference on Robotics and Automation (ICRA). 1991. p. 2724–2729.
- [13] Pomerleau F, Colas F, Siegwart R, Magnenat S. Comparing ICP variants on real-world data sets. *Autonomous Robots*. 2013;34(3):133–148.
- [14] Pomerleau F, Magnenat S, Colas F, Liu M, Siegwart R. Tracking a depth camera: parameter exploration for fast ICP. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE. 2011. p. 3824–3829.
- [15] Kubelka V, Oswald L, Pomerleau F, Colas F, Svoboda T, Reinstein M. Robust data fusion of multi-modal sensory information for mobile robots. *Journal of Field Robotics*. in press;
- [16] Zimmermann K, Hurych D, Svoboda T. Non-rigid object detection with local interleaved sequential alignment (LISA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2014 April; 36(4):731–743.
- [17] Liu M, Colas F, Siegwart R. Regional topological segmentation based on mutual information graphs. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE. 2011. p. 3269–3274.
- [18] Liu M, Colas F, Pomerleau F, Siegwart R. A Markov semi-supervised clustering approach and its application in topological map extraction. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE. 2012. p. 4743–4748.
- [19] Tompkins P, Stentz A, Wettergreen D. Global path planning for Mars rover exploration. In: IEEE Aerospace Conference. 2004. p. 801–815.
- [20] Singh S, Kelly A. Robot planning in the space of feasible actions: Two examples. In: IEEE International Conference on Robotics and Automation (ICRA). Vol. 4. 1996. p. 3309–3316.
- [21] Wettergreen D, Moreland S, Skonieczny K, Jonak D, Kohanbash D, Teza J. Design and field experimentation of a prototype lunar prospector. *The International Journal of Robotics Research*. 2010; 29(12):1550–1564.
- [22] Medioni G, Tang CK, Lee MS. Tensor voting: Theory and applications. In: *Reconnaissance des formes et Intelligence Artificielle*, 2000. Proceedings of the Conference on. 2000.
- [23] Koenig S, Likhachev M. Fast replanning for navigation in unknown terrain. *Transactions on Robotics*. 2005;21(3):354–363.
- [24] Pirri F. The well-designed logical robot: Learning and experience from observations to the situation calculus. *Artificial Intelligence*. 2011;175(1):378–415.
- [25] Papadakis P, Gianni M, Pizzoli M, Pirri F. Constraint-free topological mapping and path planning by maxima detection of the kernel spatial clearance density. In: Proceedings of the international conference on pattern recognition applications and methods. 2012. p. 71–79.
- [26] Gianni M, Papadakis P, Pirri F, Liu M, Pomerleau F, Colas F, Zimmermann K, Svoboda T, Petricek T, Kruijff GJM, Khambhaita H, Zender H. A unified framework for planning and execution-monitoring of mobile robots. In: *Automated action planning for autonomous mobile robots*. Vol. WS-11-09. AAAI. 2011.
- [27] Ferri F, Gianni M, Menna M, Pirri F. Point cloud segmentation and 3D path planning for tracked vehicles in cluttered and dynamic environments. In: Proceedings of the 3rd IROS workshop on robots in clutter: Perception and interaction in clutter. 2014.
- [28] Menna M, Gianni M, Ferri F, Pirri F. Real-time autonomous 3D navigation for tracked vehicles in rescue environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2014.
- [29] Khambhaita H, Kruijff G, Mancas M, Gianni M, Papadakis P, Pirri F, Pizzoli M. Help me to help you: how to learn intentions, actions and plans. In: AAAI 2011 spring symposium “help me help you: Bridging the gaps in human-agent collaboration”. 2011. p. 30–35.
- [30] Gianni M, Papadakis P, Pirri F, Pizzoli M. Awareness in mixed initiative planning. In: AAAI fall symposium series. 2011.

- [31] Gianni M, Gonnelli G, Sinha A, Menna M, Pirri F. An augmented reality approach for trajectory planning and control of tracked vehicles in rescue environments. In: Proceedings of the IEEE international symposium on safety, security, and rescue robotics. 2013. p. 1–6.
- [32] Carbone A, Finzi A, Orlandini A, Pirri F. Model-based control architecture for attentive robots in rescue scenarios. *Autonomous Robots*. 2008;24(1):87–120.
- [33] Finzi A, Pirri F. Switching tasks and flexible reasoning in the situation calculus. *DIAG*. 2010. Tech Rep 7.
- [34] Finzi A, Pirri F. Representing flexible temporal behaviors in the situation calculus. In: Proceedings of the international joint conference on artificial intelligence. 2005. p. 436–441.
- [35] Carbone A, Finzi A, Orlandini A, Pirri F, Ugazio G. Augmenting situation awareness via model-based control in rescue robots. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2005. p. 3699–3705.
- [36] Apt KR, Wallace MG. *Constraint logic programming using eclipse*. Cambridge University Press. 2006.
- [37] Surmann H, Nüchter A, Hertzberg J. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*. 2003; 45(3):181–198.
- [38] Blodow N, Goron LC, Marton Z, Pangercic D, Rühr T, Tenorth M, Beetz M. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2011. p. 4263–4270.
- [39] Dornhege C, Kleiner A. A frontier-void-based approach for autonomous exploration in 3D. *Advanced Robotics*. 2013;27(6):459–468.
- [40] van Diggelen J, Grootjen M, Ubink EM, van Zomeren M, Smets NJ. Content-based design and implementation of ambient intelligence applications. In: *Ambient intelligence—software and applications*. Springer International Publishing. 2013. p. 1–8.
- [41] Streefkerk JW, Heuvelink A. Smart questions context-dependent mobile information exchange for military operations. In: Proceedings of the 6th international conference on knowledge and smart technology (kst). 2014.
- [42] Neerincx MA. Situated cognitive engineering for crew support in space. *Personal Ubiquitous Computing*. 2011 Jun;15(5):445–456.
- [43] Neerincx M, Lindenberg J. Situated cognitive engineering for complex task environments. In: Schraagen JMC, Militello L, Ormerod T, , Lipshitz R, editors. *Naturalistic decision making and macrocognition*. Aldershot, UK: Ashgate. 2008.
- [44] Mioch T, Ledegang W, Paulissen R, Neerincx M, Van Diggelen J. Interaction design patterns for coherent and re-usable shape specifications of human-robot collaboration. In: *Eics 2014*. Rome, Italy. ACM. 2014.
- [45] Dearden A, Finlay J. Pattern languages in hci: A critical review. *Human–computer interaction*. 2006; 21(1):49–102.
- [46] Mioch T, Smets N, Neerincx M. Predicting performance and situation awareness of robot operators in complex situations by unit task tests. In: *Achi 2012, the fifth international conference on advances in computer-human interactions*. 2012. p. 241–246.
- [47] Kruijff G, Tretyakov V, Linder T, Pirri F, Gianni M, Papadakis P, Pizzoli M, Sinha A, E P, Corrao S, Priori F, Febrini S, Angeletti S. Rescue robots at earthquake-hit mirandola, italy: a field report. In: Proceedings of the 10th iee international symposium on safety, security, and rescue robotics. iee international symposium on safety, security, and rescue robotics (ssrr-2012). College Station, TX, USA: IEEE Press. 2012 November 5-8.
- [48] Yoshida T, Nagatani K, Tadokoro S, Nishimura T, Koyanagi E. Improvements to the rescue robot quince -toward future indoor surveillance missions in the fukushima daiichi nuclear power plant. In: *Proceedings of field & service robotics (fsr) 2012*. Matsushima/Sendai, Japan. 2012.
- [49] Johnson M, Bradshaw J, Feltovich P, Hoffman R, Jonker C, van Riemsdijk B, Sierhuis M. Beyond cooperative robotics: The central role of interdependence in coactive design. *IEEE Intelligent Systems*. 2011 May/June;:81–88.
- [50] Olsen DR, Goodrich MA. Metrics for evaluating human-robot interactions. In: *Proceedings of permis 2003*. 2003.
- [51] Salmon P, Stanton N, Walker G, Jenkins D. *Distributed situation awareness: Theory, measurement, and application to teamwork*. Human Factors in Defence. Ashgate. 2009.
- [52] TRADR (Long-Term Human-Robot Teaming for Robot-Assisted Disaster Response). 2014. Avail-

able from: <http://www.tradr-project.eu>.