UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXIX Ciclo

Investigation on Visuo-Haptic Augmented Reality for Robotic Applications

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Stefano Caselli

Dottorando: *Giorgio Micconi*

Dicembre 2016

«In contrast with many engineers who make houses, cars, medicines, and clothing for human need and enjoyment, we make things that do not themselves directly satisfy human needs, but which others use in making things that enrich human living. In a word, the computer scientist is a toolsmith.»

F. P. Brooks, Jr. [1]

Contents

List of Figures

List of Tables

Introduction

The aim of this thesis is the investigation of the potential of *Visuo-Haptic Augmented Reality* (AR) technologies for robot applications. Augmented Reality has increased its relevance in several research fields in the last decade and it starts to have important applications also in the commercial field. Whereas for commercial products the term *Augmented Reality* is mainly used to indicate an application that augments the sense of sight, academic research has explored the augmentation of other senses too, like touch [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and, to a lesser extent, hearing [14, 15, 16, 17, 18]. Although the research activity on AR has been active for several years, the development of systems including the augmentation of more than one sense simultaneously needs more investigation.

Visuo-Haptic Augmented Reality (VHAR) is a technique that augments both the sight sense (Visuo) and the touch sense (Haptic, from Greek: haptos "palpable", haptikos "suitable for touch"). This approach allows a system to be more immersive than solutions augmenting only one sense. Visuo-Haptic augmentation can be used to generate highly immersive applications, enabling a user to receive artificial inputs through two different senses at the same time. Since the quantity of information and the human processing speed change from one sense to another (sight provides a large amount of information but it takes time to be processed; touch conveys less information than sight but it is more immediate), it can be useful to combine them, exploiting the advantages simultaneously.

As discussed by Azuma [2], AR is a way for a user to interact with information that is not directly available. From another point of view, we can say that AR is a way

to achieve the *Intelligence Amplification* proposed by Brooks [1] which can be translated, in this case, as making a task easier for a human user. Hence, AR can be used to provide information in a more intuitive way for the user, helping him/her to perform a real-world task.

This investigation aspires to highlight the benefits of VHAR interfaces for human-robot interaction, enabling a user to interact with a robot in a more intuitive way to perform a task. Often human-robot interaction, especially in industrial scenarios, requires the user to know the technical details of, at least, the robot platform. To achieve this level of knowledge, if the user is not an expert of robotics or if he/she is not a highly skilled operator, a specific training phase is needed. The use of a VHAR interface may reduce the need for training because it does not require such level of technical knowledge about the robotic system in use. VHAR interfaces have been used for human-machine interaction, as we will discuss in Chapter 1, but the literature on the use of these interfaces specifically for human-robot interaction is quite limited. Moreover, this work aims to extend the state of the art about VHAR interfaces for robot teleoperation, which includes even fewer contributions.

In this work we show that a Visuo-Haptic interface provides a valid and reliable way to enhance the user experience, enabling the operator to feel the execution of the task less artificial and more natural. We also show that VHAR interfaces can be applied in very different application contexts. Indeed, we adopt VHAR interfaces for human-robot interaction in scenarios in which the robots greatly differ but the benefits derived from the use are of the same nature. Enhancement of the user experience, higher level of immersion, and lower workload on the user are general results obtained across the different scenarios.

The aim of this thesis is to assess the potential of VHAR interfaces in coping with problems related to the interaction of a human operator with a robotic system. A generic human-robot interaction involves different robots and different environments. Moreover, the typology of the tasks greatly varies. Hence, to make this investigation more generic, the work in this thesis has addressed two quite different applications, which are developed in separate chapters 3, 4 and 5. The first use case is the development and evaluation of a VHAR application in an indoor scenario in which a

robotic manipulator has to learn how to perform pick-and-place tasks. The user provides several demonstrations of the task in a 3D physics-based AR scene, using a haptic interface. The interface generates the force feedback based on the detection of collisions in the physics simulation. The 3D AR scene is automatically generated using an object recognition and registration algorithm. The system learns how to perform the task generating a precedence graph based on the user's demonstrations. We show that this natural way of interaction, together with the use of physics simulation to simulate the behavior of the objects that must be manipulated by the robotic arm in the real task, creates a more intuitive human-robot interaction. Therefore, the user does not need to have prior knowledge of the technical details of the manipulator or the system itself.

The second use case of this thesis develops a VHAR application for a challenging scenario. Indeed a VHAR system is proposed in outdoor environment for the teleoperation of a radioactivity-detector equipped UAV (Unmanned Aerial Vehicle), using a novel sensor-based haptic teleoperation scheme. UAVs are becoming more and more autonomous thanks to the miniaturization of sensors and reduced power requirements but, at the moment, a fully autonomous aerial vehicle is not effective for many tasks and not allowed by several national laws. Hence, teleoperation of UAVs is still essential. Major agencies and organizations that are planning to develop fully autonomous UAVs in future, waiting for the technological progress and more permissive laws, state that the human factor, i.e. teleoperation, is still important and further research is needed. As mentioned by USAF (United States Air Force) [19] [20], UAV teleoperation research should move from the concept of *man-in-the-loop* to *man-on-the-loop*. This change would imply that a human operator will no longer control the UAV for the whole mission time but he/she will be able to override the autonomous behavior in any moment. Furthermore, USAF proposes that a single human operator could oversee more than one UAV simultaneously, intervening only in case of need. The operator would oversee the entire operation, being aware of the state of every component, for example the flight state of every UAV, sensor data acquisition and the potential obstacles or danger situations. This set of information is called *Situation Awareness* (SA). Maintaining a high level of SA, especially in complex scenarios,

causes a high level of mental workload for the human operator. Ruiz et al. faced this problem in a multiple-UAVs scenario [21], using different human-machine interfaces to increase the immersion level of the operator, and evaluating methods for the delivery of information to the human operator [22]. As their studies show, to enable a human operator to perform such a task, a good control and supervision interface is essential. The VHAR application for UAV teleoperation developed in this work addresses several of the issues in this field, proposing some solutions.
The subdivision of this thesis in two main parts, corresponding to two application scenarios, highlights the ability of VHAR in coping with similar problems even when the components of the systems are very different. In Table 1 and 2 a comparison of the two application scenarios is reported. As shown by the tables, the two applications rely on the same concepts and techniques, even though the implementations are different owing to the heterogeneous nature of the two scenarios. Hence, the VHAR interface model is suitable for the specific use cases. The common element of the two interfaces is the use of the same haptic device, the Novint Falcon. The algorithm and the software architecture developed in this thesis allow the same device to be used in the two scenarios in different ways.
To summarize, the contributions of this thesis are:

- The assessment of the use of VHAR interfaces for human-robot interaction;
- The development of a complete physics-based VHAR interface for intuitive industrial manipulator programming;
- A novel haptic teleoperation scheme for UAV teleoperation;
- An outdoor VHAR system for UAV teleoperation and mission control.

The results obtained show the importance of VHAR interfaces in two heterogeneous and innovative application fields.

The outline of the thesis is the follow. In Chapter 1 the state of the art about Augmented Reality, and in particular about Visuo-Haptic Augmented Reality, is presented. In Chapter 2 common techniques and problems of VHAR applications are

discussed. In Chapter 3 a first application of Visuo-Haptic AR in indoor scenario for industrial robot manipulator programming is described, while in Chapter 4 and Chapter 5 a similar approach is introduced for the guide of an environmental monitoring UAV, equipped with a radioactivity detector. In Chapter 6 common features of the two VHAR applications are discussed and conclusions are drawn.

Application	Rendering	Force feedback	Calibration	Environmental model
Indoor VHAR for robotic manipulation	3D + physics	Collision detection	Marker + laser	Object recognition and registration
Outdoor VHAR for robotic teleoperation	3D + 2D	Sensor based	Marker + GPS	GIS data

Table 1: Features of the VHAR applications developed in this thesis (1).

Application	User evaluation	I/O	Interaction	Workspace	Haptic device
Indoor VHAR for robotic manipulation	Multiple untrained users	3D	Programming By Demonstration	Indoor	Novint Falcon
Outdoor VHAR for robotic teleoperation	Expert operator	2D	Teleoperation	Outdoor	Novint Falcon

Table 2: Features of the VHAR applications developed in this thesis (2).

Chapter 1

State of the Art

This chapter presents the state of the art of the topics addressed in this thesis. The chapter is divided in four sections. In Section 1.1 presents an overview on AR state of the art. Section 1.2 discusses the state of the art of haptic feedback in general and the use of haptic interfaces for teleoperation. Section 1.3 illustrates the contributions related to indoor Visuo-Haptic AR while Section 1.4 describes the state of the art about outdoor Visual AR (without haptic augmentation). Indeed, the system presented in Chapter 5 is to our knowledge the first application of Visuo-Haptic AR in outdoor environment.

1.1 Augmented Reality

Augmented Reality has been studied for a long time but it is still under heavy research. In literature several survey contributions exist, showing the potential applications of AR as well as the techniques and the technologies used to develop these applications. In one of the surveys [2] Azuma presents a broad overview on the potential of AR, discussing several fields of application. As reported by the survey, AR systems find applications in six categories: medical, visualization and annotation, manufacturing and repair, robot path planning, entertainment, and military. The paper introduces AR highlighting the difference with Virtual Reality (VR): in VR the

user is completely surrounded by a synthetic environment and he/she cannot see the real environment around him/her. On the contrary, AR allows the user to see both the real and the virtual environments, with synthetic elements superimposed upon or composited with real ones. An example, proposed by [2], is shown in Figure 1.1. In the figure a real desk and a real telephone coexist with three virtual elements: two virtual chairs and a virtual lamp. It can be noticed that mixed occlusions are correctly shown: the desk occludes the two chairs and the lamp occludes the desk. Azuma also

Figure 1.1: Example of AR scene: a real desk and real telephone with virtual chairs and virtual lamp [2].

provides the requirements for a system to be identified as an AR system:

- combines real and virtual
- interactive in real time
- registered in 3D

While the combination of real and virtual elements can be considered as the target of an AR application, real-time interaction and 3D registration are the main problems that every AR system has to address. A wide-range description of these problems is proposed by the survey [2]. In [23] Azuma et al. present an updated survey of [2]. The survey reports new technologies and possible solutions for some of the previously cited problems.
Another survey on AR is [24], where Zhou et al. present a literature study on AR papers published in the ISMAR conference. From the study emerges that tracking and

calibration/registration, interaction, and display techniques are the most common research topics in this field. The authors suggest that the motivation is the importance of these topics in the development of an AR applications. The survey analyses a considerable quantity of papers, discussing the issues and the solutions described in those works.

One of the newest application field for AR, not reported by [2], is learning/education. Several education-oriented AR applications have been developed in the last years. In [25] the authors present an overview of the design and evaluation of some education-oriented AR applications. They conclude that AR can be considered as an effective tool for learning, mainly for real world annotation, contextual visualization and visuo-haptic visualization. As is shown in the next sections and chapters, Visuo-Haptic AR is one of the core elements of this thesis work.

AR can be used as an interface between a user and other systems, like robots [26, 27, 28]. Kasahara et al. [26] developed a tablet-based application to control a small robot or intelligent furnitures. The user can move the robot or change its orientation using the touchscreen of the tablet and the built-in IMU. The user can see the current state of the robot and the sent commands on the screen of the tablet, superimposed on the images coming from the built-in camera, generating an AR scene. In [27] Kruckel et al. present a mobile robot equipped with a spherical camera, teleoperated by a user wearing an Head Mounted Display (HMD). The user can look at the environment around the robot moving the head. The system generates the correct view using the images of the spherical camera and the built-in IMU of the HMD. Virtual cues are superimposed on the images, for example the artificial horizon line and an arrow indicating the moving direction. In [28] an AR application for mobile robot debugging is described. The application can overlay robot's sensory data on a real view of the robot and its environment. AR is also adopted to help users move robotic arms or manipulators. Hashimoto et al. [29] propose a touchscreen-based AR system for non-colocated teleoperation of a robotic arm. The system enables the user to teleoperate the arm using three different methods.

A combination of AR and Programming-by-Demonstration (PbD) technique can be exploited to help non-expert users to programming robotic task. In [30, 31, 32] the

authors present approaches for programming robot manipulators in AR to perform and simulate collision-free trajectories. In [33] an AR human-robot interaction system is presented for programming robot tasks in an industrial context.
In [34] an AR environment is proposed to enhance user skill transfer in simple assembly tasks without haptic interaction and physics-based simulation. In other works, physics-based augmented reality is investigated [35, 36, 37, 38]. In [35] a framework based on visual servoing is proposed for 3D object tracking in augmented reality with physics-based animation. In [36] an AR system is proposed featuring physics simulation and marker tracking. Lok et al. [37] developed a HMD system for incorporating dynamic real objects in a virtual environment. The system uses a volume query algorithm to determine plausible collision responses between virtual and real objects. In [39] a method is proposed for detection, registration and realistic augmentation of deformable shapes.
Nicolau et al. [40] present a purely visual AR environment for laparoscopic surgery. Nunez et al. [41] developed an AR system where the user can create a topological representation of the environment by setting and manipulating map nodes. In [42] the authors propose metrics for spatial collocation accuracy.
While the traditional purpose of an AR system is to augment user's experience in very specific use cases, Grubert et al. propose [43] a new class of AR systems, called Pervasive Augmented Reality (PAR), with the aim of generating a continuous and multi-purpose experience. This kind of AR applications supplements the concept of AR with that of Context-Awareness. Context-Awareness indicates techniques used to make a computer aware of the surrounding context, for example the places around the user position. The PAR applications, as explained by the authors, must be capable of adaptation to the new context, without the need of using different applications, each for every different use case. The paper [43] proposes a new taxonomy for PAR and analyses the state of the art about AR and Context-Awareness, discussing possible future scenarios.

1.2 Haptic feedback and haptic teleoperation

Haptic feedback for human interaction has been the target of several works. Adams et al. [3] present a study on fundamental stability and performance issues of haptic interaction and examine in depth the concept of virtual coupling network between the real device and a virtual world. The study presents and evaluates methods based on impedance and admittance virtual environment model. In [5] Hannaford et al. introduce a new energy-based method for haptic interface control and study the stability of the interaction under several circumstances, changing contact stiffness and time delay. Bimanual haptic interfaces are presented in [8]. Talvas et al. explore this field reporting the possible applications and point out the major issues. More work is needed, in authors' opinion, to achieve interesting applications and both hardware and software must be involved in the development of new specifically-designed techniques.

Medical surgery is one of the most involved field for haptic feedback studies. Rosen et al. present an experimental evaluation [6] of a specifically-developed haptic interface and control for surgery. The interface is evaluated against its capacity to operate with several materials and in usability tests. Based on the results, the authors claim that the interface allows the user to feel the contact and the interaction similarly to bare hands case. The work in [11] presents a survey on the use of haptic interfaces for robotic surgery and their impact on the performance of the task. In [10] and [9] surveys on haptic devices evaluation and haptic feedback for medical surgery are presented, respectively. Another survey [12] analyses the human perception of the haptic feedback (called "haptic illusion" by the authors) in order to better understand how to build haptic interface and control software. In [4] Prattichizzo et al. present a subtraction technique for haptic interaction. In their work they eliminate the kinesthetic component of the generic haptic feedback, leaving only the cutaneous component. They show that this alternative approach, for specific tasks (e.g. needle insertion in tissues), performs the same way of the complete haptic feedback and has advantages, e.g. improved robustness against time delay, simpler mechanical interface, and very localized cutaneous-feedback application.

Haptic feedback can also be used to augment standard visuo-audio multimedia contents, as explained in [7]. The authors discuss the possible evolution of the standard codecs for multimedia entertainment in a way to include information for haptic rendering. As pointed out, the main issues are the standardization of information type and format and the mechanical properties of the haptic device.
Haptic feedback is used for several applications of mobile robot teleoperation. The work in [44] uses a PHANToM OMNI haptic interface to teleoperate a mobile robot in indoor unknown environments where the wireless signal of the teleoperation link is not always reliable. The haptic feedback is used to inform the human operator of the level of the wireless signal in order to take the best decision for the success of the mission (e.g. move to a more wireless-covered area or switch to autonomous navigation). The authors report simulation results.
Haptic interfaces have been studied for collision avoidance in teleoperation tasks of aerial vehicles [45, 46, 47, 48, 49, 50, 51, 52, 53]. Lam et al. [45] investigated artificial force fields for the generation of haptic feedback in UAV teleoperation in simulated scenarios. In [50] an approach was presented for target identification and obstacle avoidance in indoor environments. A 3D map of the environment was built using computer vision and a bilateral control system was adopted for haptic teleoperation with force feedback. In [49] an intuitive teleoperation scheme was presented, including force feedback, to safely operate a UAV by an untrained user in a cluttered environment. An obstacle avoidance behavior was designed to autonomously modify the position of the UAV independently of the operator's commands. Masone et al. [53] proposed a method for semi-autonomous UAV path specification and correction where a human operator modifies the shape of the path of the UAV, while an autonomous algorithm ensures obstacle avoidance and generates force feedback. In [48] a novel haptic control scheme was introduced based on the position of the UAV, measured by an indoor visual tracking sensor and on the measurement of the force exerted by the user of the haptic device. In [54] the authors suggested that the perceptual awareness of a human operator when teleoperating a team of UAVs is best served by a haptic feedback based on UAVs velocity information rather than a cue based on obstacle avoidance.

Haptic teleoperation of unmanned aerial vehicles has been considered in previous works exclusively for collision avoidance or to make the flight process easier

1.3 Indoor Visuo-Haptic AR

An extension of Visual AR is the Visuo-Haptic AR where the user interface includes a haptic interface, which is a device capable of generating a force feedback for the user. In this case touch is augmented in addition to sight.
Harders et al. [55, 56, 57] propose a high precision system for visuo-haptic augmented reality where the user and the environment are co-located (the user and the haptic interface are inside the AR environment). The system includes a Phantom haptic device and it supports head mounted display with landmark-based registration. The system is successfully evaluated in entertainment applications (table tennis game with basic physics [55]) and in medical training by interaction with a fixed deformable model [56]. The approach is further extended in [57] for manipulation of deformable models using two-points interaction.
Cosco et al. [58] present an approach to eliminate visual obstruction generated by the co-located haptic device and to achieve a consistent composition of the user's hand and virtual objects using a video see-through head mounted display.
Phisical simulation and visual obstruction elimination is proposed in [59] for a Visuo-Haptic medical training simulation. The authors propose a mechanical prototype for palpation training with support of visual and haptic AR. In [60] Sutherland et al. have developed a training simulation for spinal needle procedures, simulating also the ultrasound probe signal used in real procedures. The work in [60], differently from [59] and beyond the specific medical application, does not implement a visual obstruction elimination. Kuroda et al. [61] propose a method to improve the registration of the 3D model of the target organ. This method considers lines corresponding to veins, nerves and bones to adjust the global registration of the patient body, and thus the target organ, w.r.t. the camera. Haptic behaviour and visual AR take advantage of this improved registration. Also, in the medical field the work in [62] describes an applications for tying secure knots and to reduce broken sutures, using a haptic interface

and visual feedback. Tests with users confirm that the use of haptic feedback enables better results in the fulfillment of the task.

Haptic augmented reality for robot programming by demonstration is investigated without physics-based animation in [63]. Hulin et al. focus on the development of training accelerators to increase the speed of skill acquisition. A marker-less wearable system for visuo-haptic AR with physics-based animation has been developed by Murakami et al. [64]. The system supports a HMD (Head Mounted Display) and it is evaluated on pick and place tasks. However, the work does not focus on programming by demonstration.

In [65] the VisHap system is described. VisHap uses visual tracking to seamlessly integrate force feedback with tactile feedback. A haptic device is controlled to meet the finger at the point of contact with a virtual object. However, the AR environment provides only a limited support for physics-based simulation. Vallino et al. [66] propose an AR system where the user can feel the surface and the weight of an object in a simple environment with an ad hoc support for physical simulation. In [67] a physically-based haptic rendering system is presented for macro and micro applications limited to two dimensional models.

In [68] an application of VHAR in the art field is proposed. In particular, a virtual painting application on virtual objects is shown to discuss improvements to which a HMD can lead compared to half-mirrors. Another application for virtual painting is presented by Bayart et al. [69]. Unlike the previous work, in this case the user virtually paints a real object, through the haptic device coupled with a remote haptic probe. The authors focus attention to maintain multi-modal coherence, between haptic and visual feedback.

An extension of a standard VHAR system is presented by Scheggi et al. [70] introducing a method for simultaneous rendering of object shape and weight using two different haptic devices and a HMD. The authors discuss the simultaneous interaction of the user, through the two haptic devices, with the virtual world.

The work in [71] focuses on AR telerobotics applications for the disposal of explosive ordnances with a bimanual haptic feedback and a HMD. The haptic feedback is used to avoid dangerous collision of the manipulators with the ordnances. A stereo

vision image flow is generated and transmitted to the HMD.
The first Visuo-Haptic system presented in this work (Chapter 3) combines several aspects that, individually, are discussed in the previous cited works. In particular, a Visuo-Haptic physics-simulation based AR system has been developed. The system is an affordable solution for desktop AR (since it is not colocated), which is oriented to object manipulation for programming by demonstration with coexisting real and virtual objects. Moreover, it supports marker-less object recognition and registration.

1.4 Outdoor Visual AR

Outdoor Visual AR has evolved in the last few years with the introduction of new technologies for the localization of real elements and their registration w.r.t. the corresponding virtual elements. Several contributions have been proposed for buildings and infrastructure alignment [72, 73, 74, 75]. Schall et al. [72] have developed a mobile AR system intended for professionals and water resources engineers. The handheld system allows an operator to see the 3D model of water pipeline infrastructure, superimposed on real images of the ground. As authors claim, this system should help the operator to localize the infrastructure under the ground, during inspection operations. In [73] Zollmann et al. extend the previous work considering several GIS (Geographic Information System) information, not only water pipeline. Moreover, they improve the manipulation feature and propose a new handheld prototype, with a better localization capability. The same research group proposes a similar mobile AR system to oversee building under construction [74]. In this work the authors face the problem of how to show the progress in building construction. The work presents several techniques for different view situations and methods to switch between two of them. An improvement of the previous work is [75] in which a UAV equipped with a video camera is used to generate an AR environment for construction site monitoring and documentation that supports registration and visualization of 3D building models and information annotation. A similar application is proposed by Sun et al. [76], but the aim of this work is the augmentation of an aerial view of already constructed building, using GIS software. The authors also reports the main problems

related to the generation of an AR scene from aerial views. The registration problem reported is also faced in this thesis work. In [77] another augmented telepresence system was developed for large-scale environments by exploiting an omni-directional camera mounted on an unmanned airship. Iwaneczko et al. [78] presented a heads-up display to be used in UAV ground control stations to improve the UAV manual control performance of the operator. The authors only provide simulation results. In [79] a mixed reality environment was developed where a user can interactively control a UAV and visualize range data, arranged as point clouds, in real-time. Augmented reality involving UAV systems has been investigated mainly by using videos from onboard cameras.

A study on how to generate an AR overview application and improve site understanding is presented in [80]. The authors propose two techniques that allow to observe the surrounding environment, multi-view AR and variable perspective view.

The closest work to the AR outdoor system developed in this thesis is proposed by Zollmann et al. [81] where an augmented reality system was developed using a fixed ground camera. However, the system in [81] is aimed to specify waypoints for the UAV from a touchscreen and to check for potential collisions with the surrounding environment.

Chapter 2

Methodologies in Augmented Reality

This chapter introduces the keypoints of every AR system, independently from the application, and the techniques used in this work to address the main issues that concern AR, as discussed in Chapter 1. The concepts behind these techniques are common to both applications developed later in this thesis.

2.1 AR setup

AR systems, and especially VHAR systems, have typical setup configurations, shown in Figure 2.1. These configurations are divided in two categories:

- *non-colocated setup* (Figure 2.1a);
- *colocated setup* (Figure 2.1b).

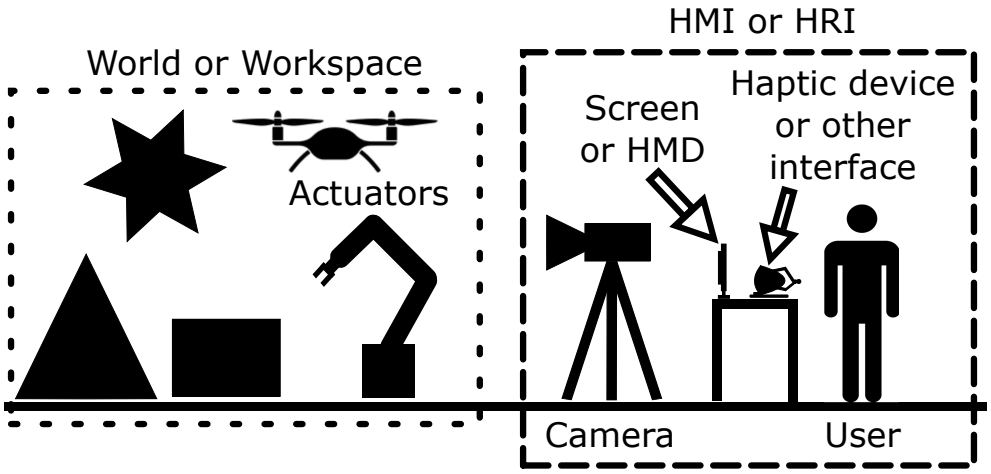
In a *non-colocated setup* the workspace (also called world, containing the place and elements with which the user interacts) is separated from the HMI (Human-Machine Interface, or HRI, Human-Robot Interface, depending from the application). Hence,

the user is not part of the workspace. The user with the HMI can be placed in proximity of the workspace, e.g. just a few meters away, or beyond the line of sight, e.g. kilometers away [82, 83]. In both scenarios the user has no direct physical contact with any element of the workspace. Although the user, when he/she is placed near the workspace, has direct line of sight to the workspace a camera is used to generate the visual augmentation of VHAR systems. The result of the visual augmentation is rendered on a screen in front of the user or on a HMD worn by the user. Depending on the context, the setup also needs an actuator, placed inside the workspace. In certain applications the actuator is teleoperated by the user.

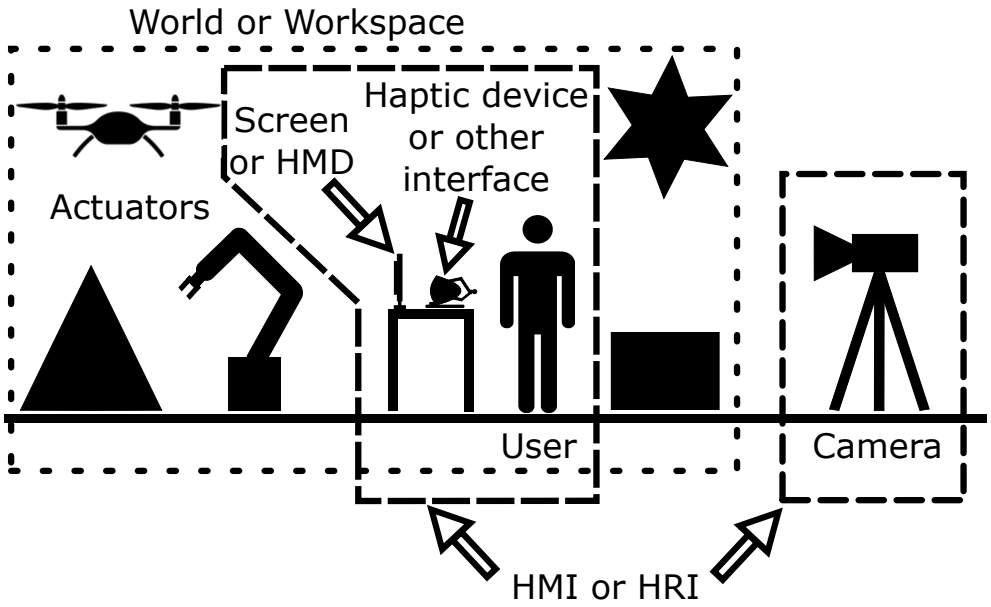
The *colocated setup* is generally composed by the same components of the non-colocated setup but the positioning is different. The user and the HMI are placed inside the workspace, i.e. they are part of it. The camera can be external (as in Figure 2.1b) or internal to the workspace, e.g. attached on the HMD worn by the user [55, 56]. Similarly to the non-colocated setup, the visual augmentation is shown on a screen or a HMD. Depending on the scenario, the user can directly interact with the elements of the workspace or use an HMI. Sometimes, in this kind of system an occlusion/obstruction elimination algorithm is implemented [58]. In this way the potential obstructions caused by the HMI and/or the user itself are addressed. As for the non-colocated case, colocated setups can include actuators inside the workspace and they can be teleoperated by the user.

Considering both configurations, if the actuator has no ability to produce a feedback for the HMI, the system will produce an artificial feedback using information based on the image flow captured by the camera, data from other sensors and/or the state of a simulation. Furthermore, methods to estimate the relative position of the camera w.r.t. the workspace and to create a *virtual camera*, simulating the real camera, are needed, as discussed in the next sections.

In this thesis, the configuration adopted for the two case studies is the non-colocated.



(a) Non-colocated VHAR setup.



(b) Colocated VHAR setup.

Figure 2.1: Typical VHAR setups.

2.2 Calibration and registration

The first and most important issue to be addressed in AR is *real-virtual objects registration*. For visual AR *registration* (also called *3D registration*) this means that virtual objects must be aligned with respect to the corresponding real objects. Hence, every AR system must execute a procedure that estimates the position and orientation of the real objects with respect to the camera or the inverse transformation (the position and orientation of the camera with respect of each object). Since the estimate is the result of an optimization procedure, e.g. a minimization, it is subject to an error. This is a key point of an AR system because if the alignment error is too high, perception of the augmentation of the real scene will be negatively affected and, moreover, the user could experience confusion and disorientation.
The transformation that defines the relative pose (position + orientation) of the objects and the camera is a homogeneous transformation. In this work, the estimation of the relative pose has been divided in two steps (Figure 2.2):

1. the relative pose ${}^{W}_{C}T$ of the *camera* with respect to a fixed reference system W, called *world* or *workspace*;

2. the relative pose ${}^{W}_{Ob_i}T$ of each real *object* with respect to the W reference system.

Hence, the AR system must be able to estimate the transformation of the camera and the set of transformations of the objects, separately.
The ${}^{W}_{C}T$ can be estimated using several methods, as discussed in [84] and [85]. These works presented methods that can be applied in different situations, like artificial and natural markers detection, stereo vision perception etc. Some of these methods, like the natural markers detection and the stereo vision approach, are techniques suited for applications that consider a moving camera because they do not require specific conditions, e.g. the positioning of artificial markers, to estimate the transformations. On the other hand, artificial markers detection has been studied for a long time and it is a robust technique to estimate the camera pose, especially in context with a steady camera. In this thesis, artificial marker detection and pose estimation has been chosen

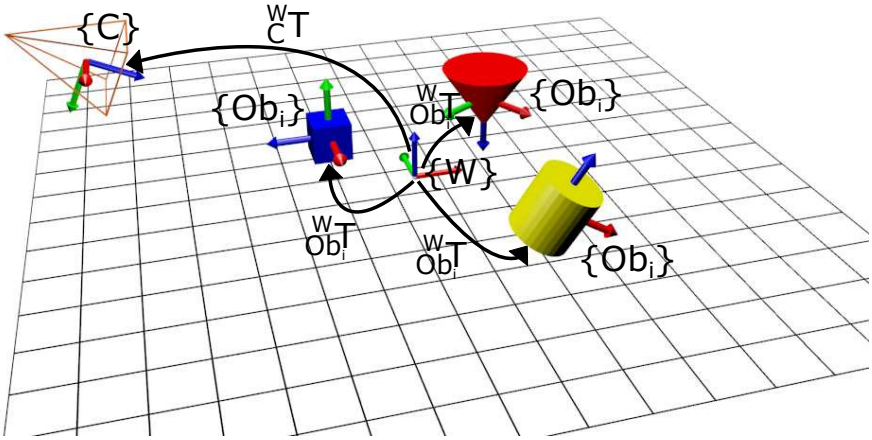
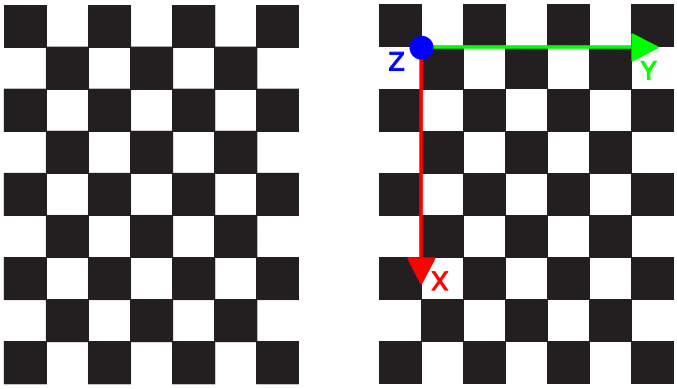


Figure 2.2: The homogeneous transformations that must be estimated by the AR application. ${}^{W}_{C}T$ *is the transformation of the camera reference system into world reference system.* ${}^{W}_{Ob_i}T$ *is the transformation of each object reference system into world reference system.*

as technique to perform the 3D registration. This choice is justified by the fact that, in the selected use cases, the camera will be in a steady pose and the use of a fixed marker does not limit the applications themselves. An example of artificial marker is shown in Figure 2.3a. The procedure to estimate the camera pose with respect to the artificial marker is the same of the camera extrinsic parameters calibration (see [86] for details on camera extrinsic parameters). The artificial marker reference system (shown in Figure 2.3b) is considered as coincident with the W reference system of Figure 2.2. To achieve the estimation of ${}^{W}_{C}T$, intrinsic camera parameters calibration is also needed. In this thesis work, the Camera Calibration Toolbox [87] with the Zhang's method [86] has been used to calibrate both intrinsic and extrinsic parameters for the indoor VHAR and intrinsic parameters only for the outdoor VHAR. The procedure for the intrinsic parameters is performed acquiring several images of a planar chessboard pattern (Figure 2.3a). Extrinsic parameters estimation requires only one image. For the extrinsic parameters of outdoor VHAR, the Aruco markers [88] [89] have been used (example in Figure 2.4).



(a) Standard chessboard pattern used for camera calibration.

(b) Standard chessboard pattern with axes aligned to the grid.

Figure 2.3: A chessboard pattern used as artificial marker.

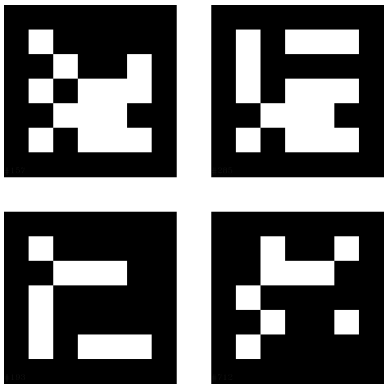


Figure 2.4: Aruco board, composed by 4 Aruco markers. The Aruco library finds each marker and estimates the camera pose using all the found markers. Each marker has a different ID expressed by the white and black pattern.

The AR application also needs to estimate the set of transformations ${}^{W}_{Ob_i}T$ aligning real and virtual objects. This set of transformations defines the relative pose of each object Ob_i with respect to the world reference system W. This estimation is strictly application-dependent. Typically it can be achieved using a 3D reconstruction, in in-

door use cases, or georeferencing the elements, for outdoor use cases. This estimation is subjected, as the estimation of ${}^{W}_{C}T$, to errors derived from the measurement process of the sensor involved.
After the estimations of all the previous transformations have been computed, the AR system can estimate the relative pose of each real element Ob_i with respect to the camera as follow:

$$ {}^{C}_{Ob_i}T = {}^{W}_{C}T^{-1} \cdot {}^{W}_{Ob_i}T = {}^{C}_{W}T \cdot {}^{W}_{Ob_i}T \tag{2.1} $$

With this information the AR system can render a virtual object registered with respect to the corresponding real object. The final registration error is affected by the errors generated by the estimation process of both ${}^{W}_{C}T$ and ${}^{W}_{Ob_i}T$. These error components can be caused by several reasons, including:

- errors in the 3D model of the real object;
- GPS or other position sensor measurement errors;
- errors from camera calibration;
- errors in the detection of artificial markers.

If the final registration error becomes high, the augmentation of the real scene is affected and a manual registration correction can be used to mitigate the problem. In [90] Sun et al. proposed a mobile AR system for registration of GIS (Geographic Information System). Their work is based on GPS-IMU pose estimation of the human operator and vision-based objects detection and pose estimation. They include the possibility for the operator to input a manual correction for the registration of the 3D models, using a gamepad and a keypad. A similar approach has been considered also for this thesis, as will be described in Chapter 5.

2.3 Virtual camera

AR scene generation needs a *virtual camera* (a camera in the 3D scene) that simulates the real camera that observes the workspace. Usually the image flow acquired

by the real camera is rendered as background of the AR scene and observed by the virtual camera, together with virtual elements. In order to generate a consistent AR scene, the parameters of the virtual camera must match the ones of the real camera.
Due to the different formation pipelines between computer vision and computer graphics, and the different reference frames, a procedure is required to ensure geometrical consistency between the real camera and the virtual camera. The virtual camera is defined using primitive functions of the OpenGL library.
In particular, the extrinsic parameters of the real camera are used for setting up the ModelView OpenGL matrix, while the intrinsic parameters of the real camera are used to compute in closed form the projection parameters of the virtual OpenGL camera as described below. A model of the real camera is considered with right-hand local (eye) coordinate system (with axis z pointing toward the viewing direction). The camera model is known after the calibration phase (see Section 2.2). The projection of a 3D point in eye-coordinates (X_e, Y_e, Z_e) onto a point (U_{real}, V_{real}) in the image plane is given by:

$$\begin{cases} U_{real} = f_x \frac{X_e}{Z_e} + u_0 \\ V_{real} = f_y \frac{Y_e}{Z_e} + v_0 \end{cases} \tag{2.2}$$

where f_x, f_y are the focal lengths of the camera in pixels and (u_0, v_0) are the coordinates of the principal point of the camera (intrinsic parameters estimated by the calibration procedure). In the OpenGL pipeline the 3D coordinates of a vertex may be converted into eye coordinates (X_e, Y_e, Z_e) by applying the ModelView matrix. 3D eye coordinates can be projected to windows coordinates (U_{vr}, V_{vr}) with the following transformation:

$$\begin{pmatrix} U_{vr} \\ V_{vr} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{w}{2}\frac{cot(\frac{\theta}{2})}{aspect} & 0 & -\frac{w}{2} - x_0 & 0 \\ 0 & \frac{h}{2}cot(\frac{\theta}{2}) & -\frac{h}{2} - y_0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{pmatrix} \tag{2.3}$$

where (w, h) are the size of the graphical window, $aspect$ is the aspect ratio $\frac{w}{h}$, (x_0, y_0) are the coordinates in pixels of the lower left corner of the viewport rectangle, and θ is the field of view angle of the perspective OpenGL frustum. The second transformation matrix in Equation 2.3 takes into account a change in orientation of the

two axes (y,z) between the 3D reference frame of the real camera with respect to the virtual camera. Indeed, the viewing direction of the virtual camera usually points in the opposite direction of the z axis. Moreover, V_{vr} must be scaled and translated into $V_{vr}^{'} = h - V_{vr}$ since the 2D reference frames of the real camera and the virtual camera on the image planes do not correspond. By combining Equation 2.2 and Equation 2.3 (the equality $U_{real} = U_{vr}$ and $V_{real} = V_{vr}^{'}$ must hold) it follows that:

$$\begin{cases} f_x \frac{X_e}{Z_e} + u_0 = \frac{cot(\frac{\theta}{2})}{aspect} \frac{w}{2} \frac{X_e}{Z_e} + (\frac{w}{2} + x_0) \\ \\ f_y \frac{Y_e}{Z_e} + v_0 = cot(\frac{\theta}{2}) \frac{h}{2} \frac{Y_e}{Z_e} + (\frac{h}{2} - y_0) \end{cases} \tag{2.4}$$

The parameters of the virtual camera, that are needed for setting up the AR scene, can then be derived as follows:

$$\begin{cases} x_0 = u_0 - \frac{w}{2} \\ y_0 = \frac{h}{2} - v_0 \\ \theta = 2cot^{-1}\left(\frac{2f_y}{h}\right) \\ aspect = \frac{w}{h}\frac{f_y}{f_x} \end{cases} \tag{2.5}$$

In order to assess the accuracy of the calibration algorithm a quantitative evaluation has been performed by measuring the mean error for a large number of 3D vertices ($N = 1000$) across several calibrated images of the indoor scenario of Chapter 3 where objects were manually registered. Projected points with real camera are computed using Equation 2.2. The virtual projection of a 3D vertex on the image plane is computed by invoking the *gluProject* OpenGL function which maps object coordinates to window coordinates. The mean error is computed in subpixel floating-point values as follows (x axis):

$$e = \frac{1}{N} \sum_{k=1}^{N} |U_{real}(k) - U_{vr}(k)| \tag{2.6}$$

Results prove that the calibration phase is very accurate, being the mean errors on both axes around one pixel ($e_x = 1.39, e_y = 1.34$).

Chapter 3

Visuo-Haptic AR for Industrial Robot Programming

This chapter presents a VHAR system in an indoor environment for intuitive programming of industrial manipulators (robot arms). Programming of an industrial manipulator requires specific technical skills about programming languages and robotics often unavailable to operators. Besides, every time a new task must be executed by the manipulator, a new sequence of instructions must be coded and a series of tests performed. This process can be tedious for the operator and, more important, the execution of a new task can constitute a danger for the objects involved and the manipulator itself, or it can not be immediately feasible. Since AR systems can simulate real objects, which may not be immediately available, an AR environment can be used by the human operator to check the task and to test the behaviour of the robotics system. Moreover, the use of an intuitive interface, like a VHAR interface, allows a non-expert operator to instruct the robotic system avoiding the need to explicitly code the sequence of instructions. Furthermore, the same VHAR system can be used as a training system for the operator.

The VHAR system developed in this chapter enables haptic interaction and physics-based simulation (including modelling of friction and gravity). The presence and pose of real objects in the workspace is detected performing a scan of the environment with

a planar laser scanner, mounted on the manipulator. An object recognition and pose estimation (*registration*) algorithm processes the 3D data extracted from the entire scan. The outputs of the algorithm are the identified objects and their poses. This information is used by the AR system to register the corresponding 3D models with the right pose. The VHAR interface has a desktop configuration and the user is not co-located with the real environment, hence the system is suitable also for interaction with remote physical systems (e.g. for training or remote operations). A three degree of freedom (3DOF) haptic device is adopted for interaction. The haptic device provides a force feedback computed from collision detection and the operator uses it to move a virtual proxy. This virtual proxy is used to push, select and manipulate virtual objects. A control algorithm ensures real-time coupling between the haptic device and the virtual proxy. The VHAR system processes the information regarding the execution of one or more demonstrations of the task, performed by the operator, and generate a *task precedence graph* of the actions composing the task. This graph of actions is used to perform a first simulation of the task using a virtual manipulator and, if the simulation performs well, to execute the task in the real environment, with the real objects and the real manipulator.
This VHAR interface, presented in [91, 92], extends previous works [93, 94, 95, 96] including automatic registration of 3D model with respect to real objects, AR visual clues, *Programming By Demonstration* technique for robotic task learning and robot path planning for real task execution.

3.1 System architecture

The system architecture is based on a non-colocated VHAR setup (Section 2.1). A representation of the setup is shown in Figure 3.1. In relation to the non-colocated template, Table 3.1 reports the implementation for this specific indoor VHAR application.
A video stream of the workspace is acquired continuously from the fixed monocular RGB camera (30 frames per second, 640×480 pixels). The images are continuously rendered as textures in the background of the user screen, generating the visual AR

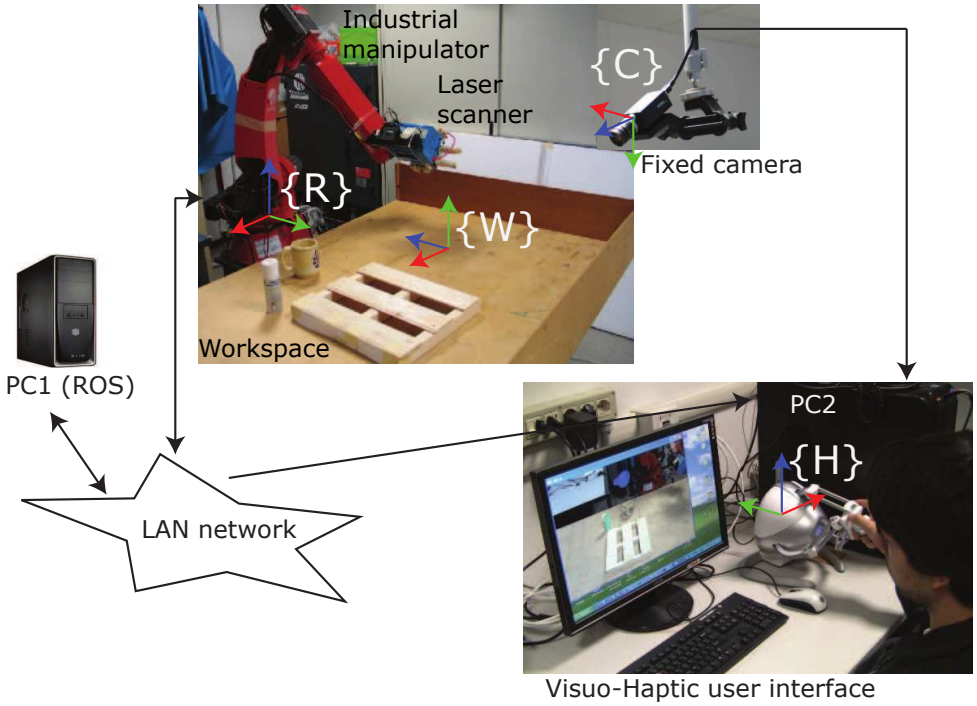


Figure 3.1: System architecture including the workspace, the manipulator, the fixed camera and the visuo-haptic user interface.

Non-colocated Template	Implementation
Workspace	Objects on a table and an industrial manipulator
HMI	Screen and haptic device (Novint Falcon)
Camera	Fixed monocular RGB camera
Actuators	Six degree of freedom industrial manipulator with laser scanner

Table 3.1: Implementation of the non-colocated VHAR setup template for the indoor VHAR application.

scene. The workspace includes a six degree of freedom robot arm (Comau SMART SiX) equipped with a gripper and a planar (2D) laser scanner (SICK LMS 400),

which is mounted in eye-in-hand configuration at the wrist of the manipulator. The laser scanner works at 50 nm (visible red light) and it emits 140 beams on a plane at 190 Hz. The 3D reconstruction of the workspace is generated by registering all the scans performed by the laser scanner while the manipulator is moving along a path, defined by a set of waypoints, that allows the scan of the entire workspace. The accuracy of range measurement is about 1.5 cm. A computer (PC1 in Figure 3.1, Core i7@3.40 GHz, 8 GB RAM), under the ROS (Robot Operating System) framework, is dedicated to laser data processing, including object recognition and pose estimation.

A second computer (PC2 in Figure 3.1, Core 2 Quad CPU@2.67 GHz, 4 GB RAM) runs the VHAR desktop user interface and manages concurrently the graphics rendering generated by the OpenGL library, physics-based processing, video stream acquisition and haptic rendering (1 kHz) of force feedback. The physics-based animation is developed upon the Bullet physics engine, that uses a hybrid impulse and constraint-based solver with a variable time step. The VHAR system, running on PC2, receives information regarding the real objects, processed by PC1, through a LAN network. The table where the objects are located is modeled in the AR environment as a hidden static plane surface. The user interface includes a 3DOF haptic device (Novint Falcon), which has a position resolution of 0.0635 mm and a maximum force feedback capability of about 10 N. The range of motion is about 10 cm^3. The user interaction feedback is artificially generated by the physics simulation and the industrial manipulator is not directly teleoperated by the user.

3.2 Building VHAR environments

The AR scene generated by this VHAR system supports the insertion of virtual objects either automatically, from robot perception (subject to noise), or through a standard manual registration procedure. The advantage of an automatic procedure is that there is no need for a manual setup of the AR scene. Indeed, automatic object recognition and registration enables a faster deployment of AR applications where the scene is not predetermined. Of course, important aspects that also need to be considered in the development of a VHAR system are the calibration of all the components of the

system and the development of a proper user interaction paradigm.
The steps needed to automatically build the VHAR environment are described in the following subsections.

3.2.1 Calibration

This subsection describes the procedure used for calibrating and registering the AR system, recalling the concepts discussed in Section 2.2 and 2.3.
There are four main reference frames (i.e. reference systems): the robot frame $\{R\}$ located on the base of the manipulator, the workspace or world frame $\{W\}$ located on the table, the camera frame $\{C\}$ and the reference frame of the haptic device $\{H\}$. The laser scanner is calibrated with respect to $\{R\}$. Hence, range data are available in frame $\{R\}$ as a point cloud (set of individual points). The camera extrinsic parameters, i.e. the fixed transformation matrix ${}^{W}_{C}T$, are computed by observing the chessboard pattern laying on the table (Figure 3.2).

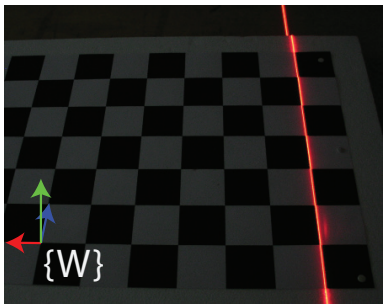


Figure 3.2: Chessboard pattern, located on the workspace table, used for calibrating the camera extrinsic parameters and laser line used for calibrating the workspace reference frame W with respect to robot reference frame R.

The chessboard pattern is aligned to the laser line so that the origin of $\{W\}$ (at a corner of the chessboard) is located at a known position in $\{R\}$ and so that the axes of the grid of the chessboard are parallel to the X and Y axes of $\{R\}$. The coordinates of the corner of the chessboard in $\{R\}$ can be easily identified by turning on the laser line at known coordinates $({}^{R}x, {}^{R}y)$. This arrangement allows calibration

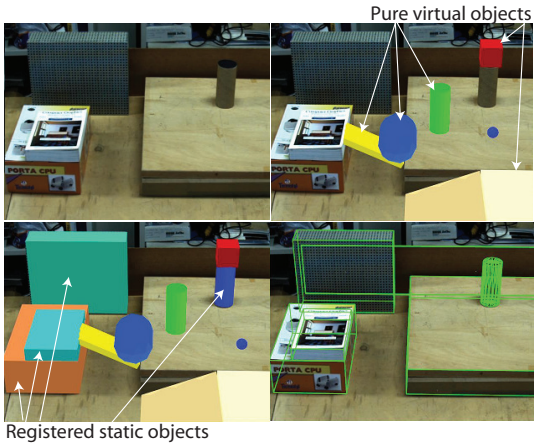


Figure 3.3: Results of camera calibration of the AR scene. Image from the camera (top left). AR scene with pure (dynamic) virtual bodies (top right). Static objects (bottom left). Wire-frame rendering (bottom right).

of the transformation matrix ${}^{R}_{W}T$ of the workspace frame with respect to the robot. The transformation ${}^{W}_{H}T$ of the haptic device with respect to $\{W\}$ is defined by the composition of a scale and an axis rotation.

Figure 3.3 shows an exemplar AR environment after calibration. The image collected from the camera is reported as well as the AR scene with pure (dynamic) virtual bodies superimposed to the environment. The Figure also shows an image with some of the underlying static real objects represented in the scene (which are not displayed in normal conditions) and the wireframe rendering which is helpful for appreciating the accuracy of calibration.

3.2.2 Automatic object recognition and registration

Augmented reality implies the ability to mix virtual and real elements in a single coherent environment. Adaptation to different scenarios requires suitable perception of the physical elements to be represented in the augmented environment. The proposed approach for automatic object recognition and registration is based on data acquired by a laser scanner. 3D range data (point cloud) acquired from the robot laser scanning

Algorithm 3.1 Algorithm for object recognition.

Input: O_i: centered object point cloud cluster;
M: point cloud dataset;

Output: the closest element $M_{target} \in M$;
T_{ICP}: alignment transformation;

1: $\text{FPFH}_{O_i} \leftarrow$ Compute_FPFH_descriptors(O_i);
2: **for** $M_j \in M$ **do**
3: $\text{FPFH}_{M_j} \leftarrow$ Compute_FPFH_descriptors(M_j);
4: $T \leftarrow$SAC_IA(O_i,FPFH_{O_i},M_j,FPFH_{M_j});
5: $f = fitness(TM_j, O_i)$;
6: **if** $f \leq f_{min}$ **then**
7: $T_{IA} \leftarrow T$;
8: $M_{target} \leftarrow M_j$;
9: $f_{min} \leftarrow f$;
10: **end if**
11: **end for**
12: $T_{ICP} \leftarrow \text{ICP}(T_{IA}, O_i, M_{target})$;

phase are processed to extract a point cloud subset (cluster) for each real object lying on the table of the workspace. Cluster extraction is performed by applying a flood fill algorithm after outliers removal and plane removal [97]. For each point cloud cluster O_i recognition is achieved by finding the most similar element M_{target} from a dataset M of complete point cloud models, i.e. the one that best aligns with the observed point cloud cluster. The point cloud dataset was generated by scanning each object alone in the workspace. The dataset also contains the triangulated mesh of each object.

Two algorithm has been developed to recognize the objects. The first, shown in Algorithm 3.1, is based on the computation and alignment of FPFH_{O_i} point cloud descriptors (Fast Point Feature Histogram [98]) of all the points of the observed point cloud O_i (centered on its centroid). Given a point p of a point cloud and its k nearest

neighbours $FPFH(p)$ is defined as

$$FPFH(p) = SPFH(p) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot SPFH(p_k) \tag{3.1}$$

where ω_k is a weight factor, and $SPFH$ is the Simplified Point Feature Histogram [98] which is a local descriptor that encodes information about estimated surface normals of the k-neighborhood of a point. Then, for each model in the dataset M_j the algorithm computes FPFH_{M_j} and it executes a sample consensus non-linear optimizer (SAC_IA, Line 4) between FPFH_{O_i} and FPFH_{M_j} on a set of triplets of corresponding points with similar features. The result of each optimization process is a transformation matrix T that best aligns the model in the dataset to the point cloud cluster. The closest element M_{target} in M to the observed point cloud O_i is the one that minimizes a fitness function, that is computed as a sum of squared distances of closest points from M_{target}, transformed by T, to the observed point cloud (Lines 5-10). The transformation matrix of the best matching model T_{IA} is further refined (to obtain the alignment transformation T_{ICP}) by applying the iterative closest point (ICP) algorithm (Line 12).

The second algorithm, presented in [99] uses the Particle Swarm Optimization (PSO) technique [100] to minimize the fitness function 3.2.

$$F(O_i, M'_j) = \frac{1}{N_{O_i}}\sum_{p\in O_i}\min_{q\in M'_j}\Big(dist(p,q)\Big) \tag{3.2}$$

The PSO is a bio-inspired optimization technique, based on the behaviour of swarms and flocks. The main advantages are the robustness against noise and occlusions and the ability to work with not differentiable problems. Despite this technique is an heuristic method, hence there are no warranties that the best solution found by the algorithm is optimal, PSO can be successfully used to perform object recognition and registration [101]. This technique can be applied to problems with arbitrary dimension. In our case the problem dimension is 6, since each point cloud that must be recognized and registered has 3 DOF of translation and 3 DOF of rotation. The PSO algorithm instantiates a parametrized number of entities called *particles*. Each

particle is a 6-dimension point that represents a transformation T (translation + rotation). This transformation is applied to a reference cloud $M_j \in M$ for fitness function evaluation.
The fitness function 3.2 compares the point cloud O_i to be recognized (composed of N_{O_i} points), with the reference cloud M_j extracted from a database, composed of N_{M_j} points. This reference is subjected to a transformation T encoded by a PSO particle, to obtain $M'_j = T(M_j)$. The fitness for a pair $<M_j,T>$ is the average of the minimum distance of each point p of O_i to the closest point q of the roto-translated reference M'_j. The $dist()$ function is a valid distance metric between points. In this case we selected the squared euclidean distance. Each point cloud is expressed in a local reference frame centered around its centroid. A model can do a full rotation around each axis while the range of translation is limited to 10 cm in each direction, which is enough to satisfy the constraints of the scenario we are considering. The evaluation of the fitness function is the solution for the pair $<M_j,T>$. The PSO updates each particle's position based on the temporary best general solution, temporary best local solution and the velocity of the particles. A termination criterion, based on time, iterations or fitness value must be set in order to stop the minimization process. When the optimization is finished, the pair $<M_{target},T_{PSO}>$ associated to the best general solution is returned together with the fitness value. This result indicates which model M_j is recognized as O_i, how to register the model with the target, and how well they match. The PSO implementation used in this work [102] has been developed using the Nvidia CUDA framework. CUDA allows parallelization of the PSO algorithm on a GPU in two different ways (Figure 3.4). Indeed, the update of the position of all particles is executed in parallel, as well as the application of the associated transform to M_j. The total number of GPU threads created by CUDA is equal to $K * M$ (Figure 3.4). K is the number of particles, M is the number of blocks of points of the reference cloud M_j transformed and evaluated simultaneously. The distance between each transformed point and each point of the target cloud is calculated in parallel for all the particles. A serial reduce function evaluatse the best solution (the particle that produces the minimum average distance).

When an observed object is recognized the corresponding mesh model in the

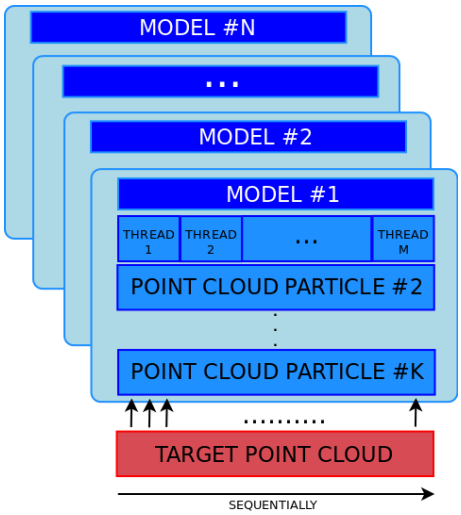


Figure 3.4: Parallelization of fitness function evaluation using the CUDA framework.

Figure 3.5: Objects from the dataset used in the experiments.

dataset is inserted (registered) in the augmented reality environment using the alignment transformation matrix (T_{ICP} or T_{PSO}, for FPFH or PSO algorithms respectively) translated to the centroid of the object (with graphics rendering for visible objects or without graphics rendering for invisible objects). Insertion of new objects by the user in the AR environment after the laser scanning phase requires a re-scanning of the environment. Figure 3.5 shows some of the objects from the dataset used in the

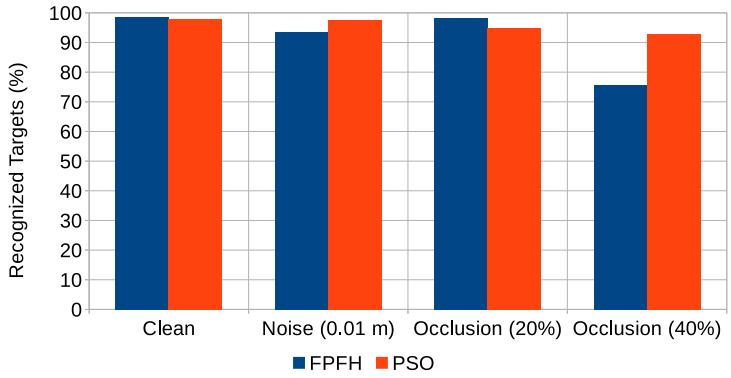


Figure 3.6: Results of the recognition comparison tests (FPFH vs PSO) in 4 scenarios: clean, 0.01m of noise, 20% of occlusion, 40% of occlusion.

experiments.

The time required for recognizing an object depends on the size of the point clouds and on the size of the dataset. For point clouds containing 6K points the recognition time is about 20 seconds for the FPFH algorithm and 45 seconds for the PSO algorithm (PC1 in Figure 3.1) with about ten objects in the dataset. The recognition performance of the two algorithms was evaluated executing 50 independent tests for each of the 10 models in the dataset. In each test a model is used as the target cloud and compared to all the others under normal conditions, simulating noise and occlusion. In Figure 3.6 the results of the tests are shown. As illustrated in the figure, the FPFH algorithm achieves high recognition rate in normal condition (clean) and with a small amount of occlusion (20%). The FPFH recognition rate drops in presence of noise and with higher occlusion. Instead, the PSO algorithm is able to better maintain a constant recognition rate across all the test scenarios, in particular with noise and high occlusion.

We also compared the registration error of both algorithm. The graphs in Figure 3.7 are obtained running 100 tests for each level of noise. Increasing the noise FPFH obtains less sparse results than PSO but the FPFH errors tend to be greater than the PSO errors. Moreover, given the iterative nature of the Particle Swarm Optimization

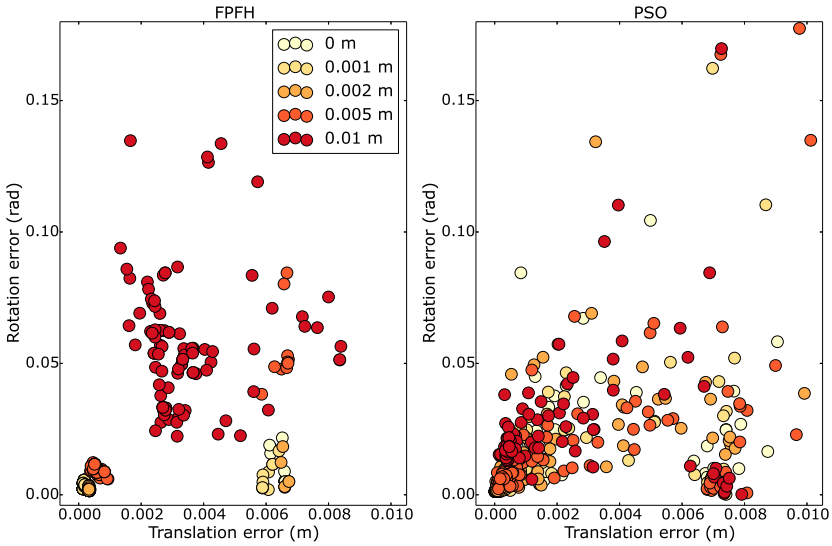


Figure 3.7: Results of the registration comparison tests (FPFH vs PSO). 6 noise levels are applied, from 0 m of noise (light circles) to 0.01 m of noise (dark red). We can notice that the PSO is more scattered than FPFH but in average it achieves better results

technique, the PSO can obtain even better results if higher termination criterion are set (longer optimization time, lower fitness value, ecc.).

Triangulated mesh models occasionally contain irregularities due to noisy observations. Such irregularities could affect the stability of a physically-simulated dynamic body in the AR environment. For example, a registered body that should stand still on top of the table could fall down due to excessive noise at its bottom part. Thus, to avoid stability issues in the physics-based simulation the triangulated mesh models in the dataset may require to be manually repaired using a CAD software. Of course, ideal CAD models of real objects, if available, can be used in the dataset. Figure 3.8 shows examples of point cloud models and the corresponding mesh models saved in the dataset. Figure 3.9 shows a successful alignment of three observed objects with their corresponding point cloud models in the dataset. Accuracy of the object registration procedure is illustrated in Figure 3.10, which reports the distribu-

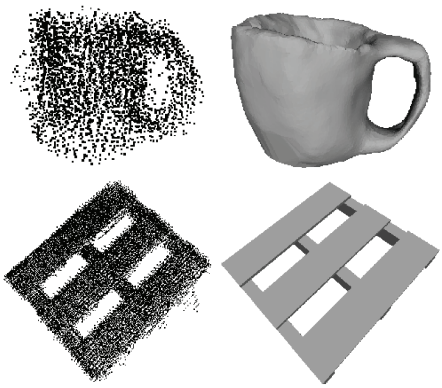


Figure 3.8: Examples of point cloud (left column) and mesh models (right column) in the dataset.

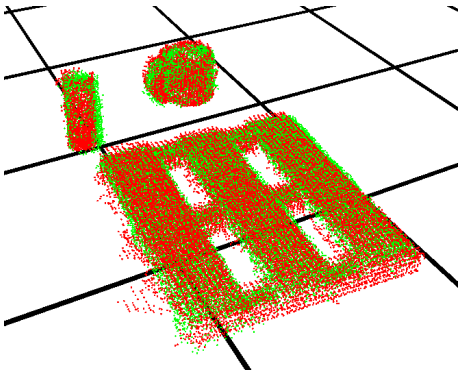


Figure 3.9: Example of point cloud alignment. Observed point cloud clusters (green) and aligned point cloud models from the dataset (red).

tion of alignment error in manually annotated images. The average error is 4.7 pixels, which agrees with the accuracy of the laser scanner since images have a 640×480 resolution and the visible workspace is about $1.8m \times 1.8m$.

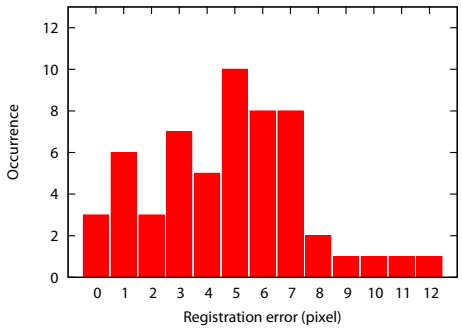


Figure 3.10: Alignment error distribution.

3.2.3 Occlusion handling

Occlusion handling in the AR environment between static (hidden) virtual bodies, corresponding to real objects, and dynamic virtual bodies is achieved taking advantage from the graphics rendering pipeline. The method adopted in this work is suitable for a monocular camera system. Since the position of static virtual bodies is known either from manual registration or automatic object recognition, static physical bodies corresponding to real objects are rendered only in the OpenGL depth buffer and filtered out from the color buffer. This is a fast approach that allows depth values to be rendered directly into the Z-buffer of the graphics hardware. Figure 3.11 shows an example scene where occlusion is handled properly. In the example a virtual red

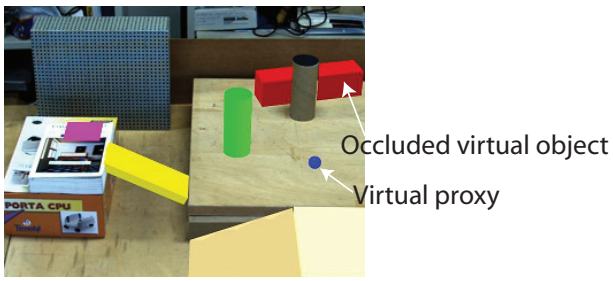


Figure 3.11: Example of a virtual object (red box) being occluded by a real object (brown cylinder). The image also shows the virtual proxy.

box is occluded by a real object (brown cylinder). The occlusion handling algorithm

is also continuously applied to the virtual proxy. Indeed, the virtual proxy correctly disappears once it goes behind one of the objects populating the environment.

3.2.4 Haptic interaction paradigm

As previously explained, the system supports different types of virtual bodies that coexist in AR. Indeed, the 3D model of each recognized real object, that is registered in the AR environment, can either be rendered as a dynamic virtual body or hidden as a static virtual body. Static virtual bodies are hidden as they correspond to static real objects. Additional static bodies can be manually added to the environment. Dynamic bodies without a real counterpart in the workspace can also be instantiated in the AR environment when performing a task.

Haptic interaction within the AR environment is achieved using a virtual proxy paradigm (Figure 3.11). A small sphere with uniform mass density serves as a proxy for the haptic device. The virtual proxy can be moved in the AR environment and it can physically interact with both static and dynamic bodies. Objects can be pushed by the spherical proxy, selected or deselected (i.e. attached to or detached from the virtual proxy), translated and rotated.

Selection is performed by pressing a button on the haptic device when the virtual proxy is in close proximity to a dynamic object (deselection is performed by pressing the same button). A selected dynamic object can be translated or rotated. As mentioned in Section 3.1, our setup includes a 3 DOF haptic device (rather than a full 6 DOF device). Hence translation and rotation are managed in two separate modes of operation (triggered by pressing a button on the device). Translation of the virtual proxy in free space and translation of a selected dynamic virtual object are performed by the human operator by moving the haptic device. Linear spring-dampers produce a translational central force F that enables the virtual proxy or the selected object to follow the motion of the haptic tool and to convey force feedback to the user (in the opposite direction). Force F is given by the following expression:

$$F = k_T d - b_T v \tag{3.3}$$

where d and v are the relative position and translational velocity of the virtual proxy

(or the selected virtual object) with respect to the tool of the haptic device, while k_T and b_T are the translational spring and damping constants. Force feedback on the haptic device is generated as force $-F$ when collision between the virtual proxy (or the selected virtual object) and another object in the environment is detected. Rotation of a selected virtual object is performed using a semi-physics approach. The cubic workspace of the haptic device is divided into a uniform grid of $3 \times 3 \times 3$ cells and the measured position $({}^H x, {}^H y, {}^H z)$ of the haptic device is used to set the angular velocity $(\omega_x, \omega_y, \omega_z)$ of the object about the local reference frame. For example, to rotate the object about its vertical axis the ω_z component is set as:

$$\omega_z = \begin{cases} -\omega & \text{if } {}^H x < -x_c, \\ 0 & \text{if } -x_c \leq {}^H x \leq x_c, \\ +\omega & \text{if } {}^H x > x_c \end{cases} \tag{3.4}$$

where ω and x_c are constant values. The angular velocity about the other axes is defined analogously. A more advanced physics-based method that treats separately translation and rotation modes was presented in [103]. Such method, however, is not applicable to the Novint Falcon device due to the limited range of motion and performance of the device.

3.3 Robot programming by demonstration in augmented reality

The haptic user interface described in Section 3.2.4 allows the user to perform basic manipulation actions, such as a sequence of pick and place operations of objects, and it was conceived for programming manipulation tasks by demonstration. *Programming by Demonstration* (PbD) [104, 105] is a technique that makes programming a robot easier and faster for the user by letting the system learn how to perform a task from human observations. The system is able to recognize elementary actions as they occur by detecting the change of state of the objects and the change of their relative pose. The elementary actions include moving an object A "next to", "on top of" or "inside" another object B. In particular, the system determines that an action has

completed when a manipulated object, after being detached from the virtual proxy, comes to rest. The type of action is determined by computing the relative position of the manipulated object (which has come to rest) with respect to the closest objects in the environment and their contact state. The relative position of two objects is determined by computing the relative position of the centers of the corresponding bounding boxes.

The PbD approach developed in this work uses task precedence graphs for learning manipulation tasks from multiple user demonstrations in the VHAR environment. A *task precedence graph* (TPG) [104] is a directed, acyclic graph $G = \{V, E\}$ where V is the finite set of elementary actions $V = \{t_1, t_2, \ldots, t_n\}$ and $E \subset V \times V$ is a set of arcs. Each arc (t_a, t_b) of E defines a precedence relations stating that action t_a must be completed before the execution of action t_b can start. Each elementary action has a number of pre- and post-conditions describing the state of the environment at the beginning and at the end of each action. A single user demonstration, that defines a way to solve a task, is a sequence of fixed length of elementary actions $D_i = \{t_{i_1}, t_{i_2}, \ldots, t_{i_n}\}$ with $t_{i_j} \in V$. A TPG can be learned incrementally from a sequence of m demonstrations of the same task. After the first demonstration $D_1 = \{t_{1_1}, t_{1_2}, \ldots, t_{1_n}\}$ the set of constraints $E_1 = E^{D_1} = \{(t_{1_i}, t_{1_j}) | i < j\}$ is initialized with all the precedence relations in D_1. For each additional demonstration D_k, showing alternative ways to perform the same task (by permuting the elementary actions), the TPG is updated to refine the set of valid precedence relations ($E_{k+1} = E_k \cap E^{D_{k+1}}$). Namely, the updated TPG is computed by the intersection of the past knowledge about the task with the precedence relations introduced by the latest demonstration. A sequence of elementary actions of length n is compliant with a precedence graph $G = \{V, E\}$ if every pair (t_{i_k}, t_{i_l}) with $k < l$ satisfies the task constraints defined by E. Hence, the cardinality of the set of precedence relations decreases as multiple demonstrations are provided, since new demonstrations have the effect of removing previous constraints by showing alternative possible solutions for the task.

3.4 Experimental evaluation

This section presents the experimental results of the user interaction with the interface and the ability of the system to execute a manipulation task in the real environment. Relevant results of the previous works are briefly reported, for further details see [93, 94, 95, 96].

3.4.1 Object manipulation

A user study was conducted to assess the performance of the visuo-haptic system for object manipulation. The task, shown in Figure 3.12, consists of two consecutive pick and place operations. Two objects (a thin yellow box and a green bunny) must

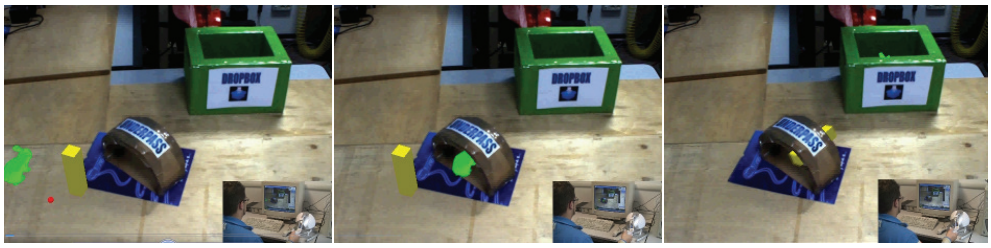


Figure 3.12: User study consisting of pick and place operations to assess object manipulation.

be picked up, lifted, and placed inside a drop box after passing under an arc-shaped object (which requires the yellow box to be rotated before passing under the arc). A total of 11 participants were recruited among students of the University of Parma and signed a consent form. After a training session each subject performed two trials of the manipulation task, in random order, with and without force feedback. The task completion time was measured in each trial. All the subjects were able to complete the tasks. Results confirm that the use of force feedback reduces the task completion time as a One-way ANOVA shows that there is a significant effect of force feedback ($p < 0.02$). The average completion time was 32.6($\pm$8.7) seconds with force feedback and 46.4 ($\pm$14.9) seconds without force feedback. Moreover, a subjective evaluation of the visuo-haptic interface scored 8.4 on a Likert scale of $1-10$ (where 1 is not satisfactory and 10 is highly satisfactory). In this task force feedback is mainly

helpful while inserting the objects in the box as it provides a perception cue that assists the subject. This experiment served also to get users acquainted with the system before asking them to perform the robot PbD experiments described hereafter.

3.4.2 Evaluation of robot programming by demonstration tasks

This subsection investigates the usability of the VHAR environment to programming robotics task through PbD technique. The three PbD experiments reported next were performed by exploiting automatic object registration as described in Subsection 3.2.2. All subjects that participated in the experiments were able to complete the tasks after a short training session.
A first PbD experiment, shown in Figure 3.13 and 3.14, is conceived as a prototype that resembles a pallet arrangement. There are four relevant actions: picking the box

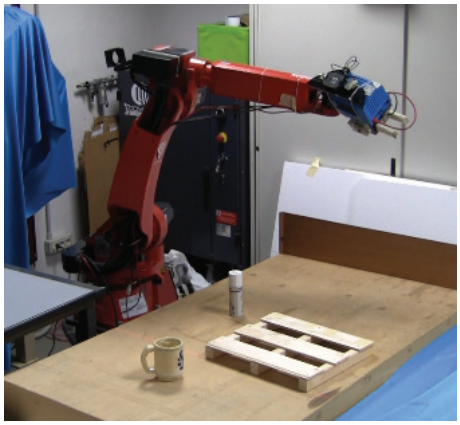


Figure 3.13: Experiment 1: pallet arrangement. Robot laser scan phase.

or the container and placing them on the pallet (actions A_1 and A_2); picking the cylinder object or the cup and putting them inside the container (A_3 and A_4). The pallet 3D model is registered as a hidden static object, the box and the container are not real and they are instantiated by pressing a button on the haptic interface when needed. The user provides two demonstrations of the task. In the first demonstration D_1 the user performs the sequence of actions $A_1A_2A_3A_4$, while in the second demonstration D_2

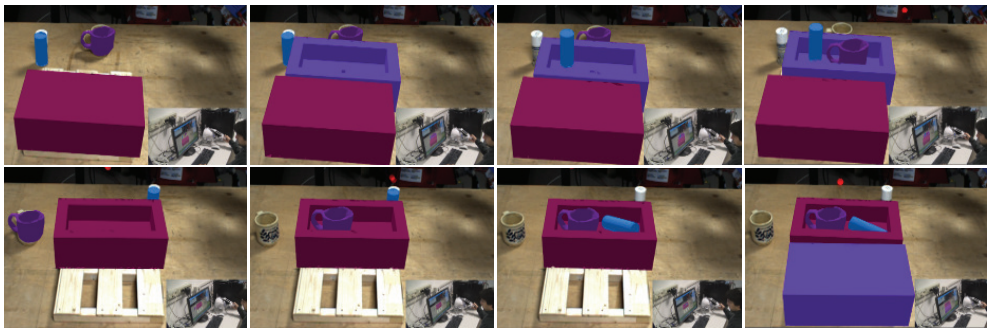


Figure 3.14: Experiment 1: pallet arrangement. Top row: first demonstration D1; bottom row: second demonstration D2. Each demonstration is a sequence of four elementary actions. Object colors are not relevant to the task.

the sequence is $A_2A_4A_3A_1$. The second demonstration is performed with a different initial configuration of the objects, thus requiring the environment to be re-scanned using the range sensor after the first demonstration. The system learns that precedence relations (A_2,A_3) and (A_2,A_4) hold, i.e. the two objects must be put in the container after placing the container on the pallet.

A second PbD experiment, shown in Figure 3.15, shows a kitchen task of laying a table. There are four relevant actions: picking the dinner dish and placing it on the

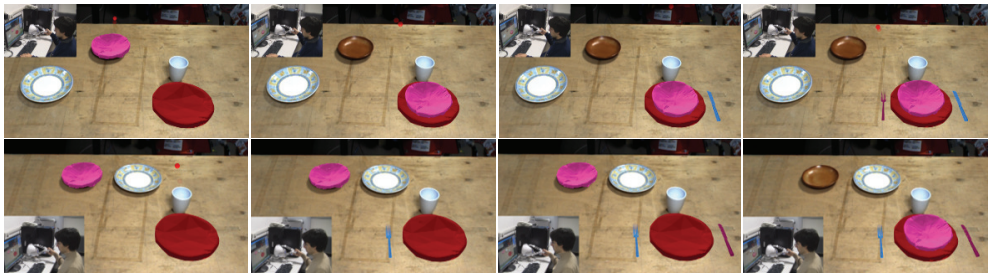


Figure 3.15: Experiment 2: laying the table. Top row: first demonstration D1; bottom row: second demonstration D2. Each demonstration is a sequence of four elementary actions. Object colors are not relevant to the task (a video of the experiment can be downloaded at `www.ce.unipr.it/%7Emicconi/HAVE2014.mp4`*).*

table below the glass (A_1); picking the soup dish and placing it on top of the dinner

dish (A_2); picking the knife and the fork and placing them on the table right of, and left of the dinner dish (A_3 and A_4). The glass 3D model is registered as a hidden static object, the fork and the knife are instantiated as virtual objects in the AR scene. The user provides two demonstrations of the task. In the first demonstration D_1 the user performs the sequence of actions $A_1A_2A_3A_4$, while in the second demonstration D_2 the sequence is $A_1A_4A_3A_2$. The second demonstration is performed with a different initial configuration of the scene. The system learns that precedence relations (A_1,A_2), (A_1,A_3) and (A_1,A_4) hold, i.e. the soup dish, the fork and the knife can be placed in any order but only after the dinner dish has been placed.

A final complete PbD experiment is presented where the task is to build a small bridge. After user demonstration in the AR environment and task learning, the system scans again the environment using the laser scanner and the manipulator successfully executes the learned task as explained next. The user provides two demonstrations of the task (Figure 3.16). There are three relevant actions: picking a wooden block and

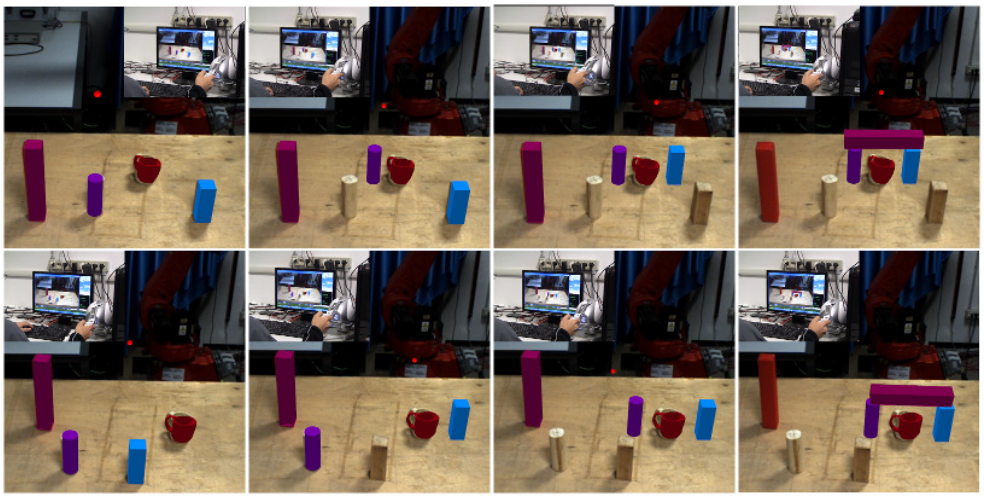


Figure 3.16: Experiment 3: building a bridge. Top row: first demonstration D1; bottom row: second demonstration D2. Each demonstration is a sequence of three elementary actions. Object colors are not relevant to the task.

placing it on the table right of a cup (A_1); picking a wooden cylinder and placing it

on the table left of the cup (A_2); picking a long plastic block and placing it on top of both the small block and the cylinder (A_3). All objects are recognized and registered as dynamic bodies from laser scanning. Pick and place action A_3 is more challenging for the users than the previous ones due to the reduced size of the supporting objects of the bridge. In the first demonstration D_1 the user performs the sequence of actions $A_2A_1A_3$, while in the second demonstration D_2 the sequence is $A_1A_2A_3$. The second demonstration is performed with a different initial configuration of the scene. The system learns that precedence relations (A_1,A_3), (A_2,A_3) hold, i.e. in order to build the bridge the two wooden objects (bridge piers) should be placed in any order on the table before the long block (bridge deck).
Once the task has been learnt, a simulation to plan the the trajectories for the manipulator is run. Figure 3.17 shows the planned robot manipulation task starting from a different initial configuration of the objects. The robot manipulation planner has been developed upon the OpenRAVE engine as described in [97]. Motion planning ensures collision avoidance and it takes less than 4 seconds for the whole task, which is far less than the time required for laser scanning and object recognition. Figure 3.18 shows the successful execution of the planned manipulation task by the real robot.

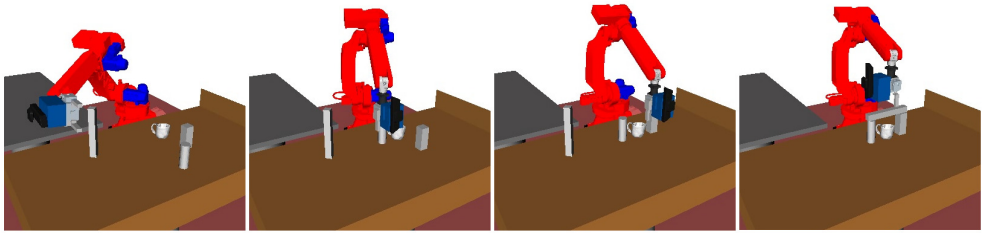


Figure 3.17: Experiment 3: planned robot motion of learned task.

3.5 Discussion

Results reported in Subsection 3.4.1 have confirmed that force feedback improves task completion time and performance for experiments involving interaction and manipulation of objects in AR scene. Moreover, they suggest that force feedback is

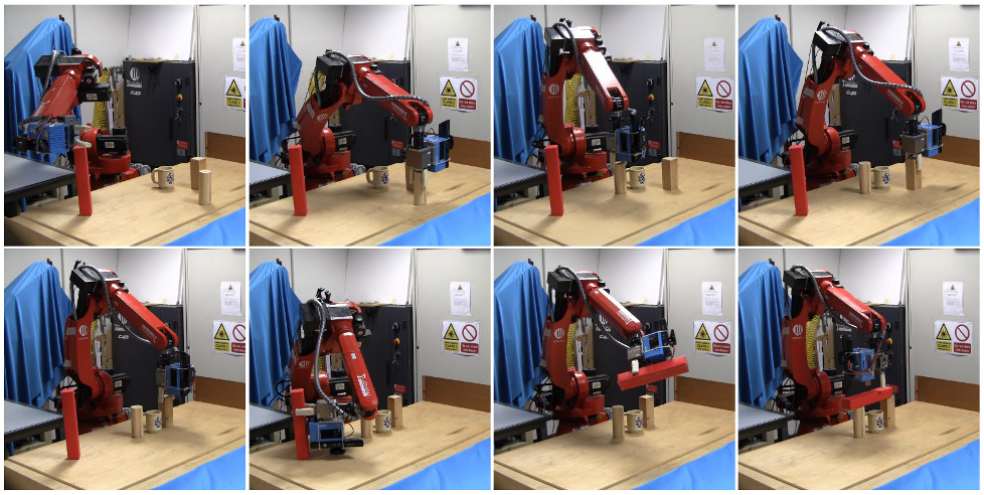


Figure 3.18: Experiment 3: robot execution of learned task after user demonstration and laser scanning. Initial configuration is shown in the top-left image, final configuration is displayed in the bottom-right image. A video of the experiment can be downloaded at `www.ce.unipr.it/%7Emicconi/MMSJ2014.wmv`.

considered an important feature that improves the feeling of immersion in the AR environment. The performance of the two automatic object recognition and registration algorithms, FPFH and PSO, has been estimated quantitatively (Subsection 3.2.2). Moreover, the comparison between the two algorithms indicates that the FPFH has good results in normal conditions and with small occlusion, executing the recognition and registration process in 20 s. The FPFH fails with noise and larger occlusion. The PSO obtains better results in those conditions where the FPFH fails, both in recognition and registration. The main disadvantage of PSO is the higher execution time to obtain a comparable result to FPFH, 45 s. The results of both algorithms, in normal conditions or with small occlusion, confirm that the accuracy is adequate enough to allow the user to perform manipulation tasks. Of course, object registration may suffer due to strong occlusions in case of cluttered scenes with many objects. Furthermore, this is one of the first VHAR system including physics simulation and PbD technique for a real robot scenario. Although the presented experiments are rather

simple, compared to the complexity of robot tasks in industrial applications, results reported in Subsection 3.4.2 indicate that desktop-based VHAR is a viable solution for providing user input to a robot working cell.

Chapter 4

Haptic AR for Aerial Environmental Monitoring

In this chapter an outdoor Haptic AR interface for a radioactivity detector-equipped *Unmanned Aerial Vehicle* (UAV) teleoperation is presented. This HAR interface is only a part of the complete outdoor VHAR system. The presentation of the entire VHAR interface is deferred to Chapter 5, where more emphasis is given to the visual part. The development of such an interface is motivated by the fact that there is an increase in demand for outdoor environmental monitoring and surveillance applications related, for example, to nuclear inspection [106]. Also, there is a rapid expansion of decommissioning activities. Hence, there are many opportunities in the development of UAV-based inspection systems and, in particular, in the design of advanced user interfaces, for skilled personnel, enabling rapid sensor data gathering.
The proposed haptic-based teleoperation approach relies on the use of an UAV equipped with a custom CdZnTe-based X-ray detector (radioactivity detector) providing radiation information and a wireless communication infrastructure. Potential applications are the localization and identification of radioactive materials in industrial plants (such as steel mills), construction, recycling factories and landfills. The discovery of illicit radiological materials clearly generates a reduction of the environmental impact of dangerous substances and an increased safety for operators, as well as an

increased security for citizens. The use of an UAV equipped with a detector has the advantage of not exposing a human operator to the radioactive materials.
UAV teleoperation with a human in the loop allows an expert operator to focus on candidate areas of the environment, thereby remote guidance overcomes the problem of limited duration of each flight afforded by onboard batteries. However, teleoperation of aerial vehicles at a distance in direct sight is complex due to the lack of *Situation Awareness* (SA) of the operator. Hence, it is crucial to augment the information available to the pilot by exploiting the UAV onboard sensors. In the proposed AR approach, force feedback is used as an additional sensory channel to reduce operator's perceptual load in localizing the hazardous materials. In particular, an AR user interface has been developed integrating two haptic control modes, both based on impedance control. Both control modes provide an attractive force feedback around the location of the most intense detected radiation source, augmenting the information returned to the user by the visual feedback. The purpose of the attractive force feedback is to keep the UAV close to the radiation source once a target is found. The approach for haptic teleoperation supported by the AR system, presented in [107, 108, 109], is novel compared to previous studies where haptic teleoperation of UAVs has mainly been investigated for collision avoidance (Section 4.1.1).
In this chapter and Chapter 5 we investigate the potential of VHAR in the context of a highly relevant fielded robotic application. Results show potential benefits of a multimodal AR feedback to the user that can be expected to hold in other applications as well. In particular, in this chapter we focus on the haptic augmentation.

4.1 Architectures for aerial environmental monitoring

4.1.1 Related works on UAVs for environmental monitoring

Before illustrating the proposed system, we briefly report prior work on UAVs deployed for environmental monitoring task. UAVs have already been used for environmental monitoring and radioactivity detection in a few situations. As discussed in [110] small multi-rotor unmanned aerial systems have the advantage of flying at reduced speed, with greater maneuverability, and they can obtain high spatial resolu-

tion maps of radiological contamination sources. Several works investigated the use of UAVs for radiation detection [111, 112, 106, 113, 114, 115, 116, 110, 117, 118]. However, none of these works have explored the use of haptic teleoperation for radiation detection. Indeed, standard teleoperation interfaces (RC) were adopted or preprogrammed flight missions were used for autonomous UAVs. Boudergui et al. [111] reported preliminary experiments of nuclear and radiological risk characterization using a teleoperated quad-rotor equipped with a CdZnTe sensor and a gamma camera. However, the system was conceived for security applications in indoor environments. In [112, 113] two systems were presented for radiation detection using fixed wing UAVs flying at high altitude and high speed. Neumann et al. [114] investigated a remotely controlled UAV, in field tests, for measuring hazardous gaseous substance sources. In [115] a remote sensing system for radiation detection and aerial imaging was developed using an autonomous helicopter. Experiments were reported in a simulated environment. In [116] multiple UAVs contour mapping of nuclear radiation was simulated with formation flight control. Martin et al. [117] developed an unmanned aerial system, carrying a gamma radiation mapping unit, for the radiological characterization of uranium mines. Sanada et al. [118] presented an unmanned helicopter for aerial radiation monitoring experimented around the Fukushima Dai-ichi nuclear power plant. The radioactive cesium deposition on the ground was successfully measured. Radiation mapping has, of course, been carried out by autonomous mobile robots, for example in [106], where a gradient-based Bayesian method was proposed which used a prior knowledge of the radiation distribution on the ground.

4.1.2 HAR teleoperation architecture

This subsection describes the system architecture of the Haptic AR teleoperation approach developed in this chapter, shown in Figure 4.1. The operator of the haptic device, which is connected to a ground station, interacts with the UAV by sending motion commands on the xy plane while receiving a 2D force feedback, i.e. the UAV flies at a constant height. A second human operator, using a standard remote controller, is responsible for take off, landing, setting the altitude set point and he/she can take full control of the UAV, at any time, as required by the Italian regulation for

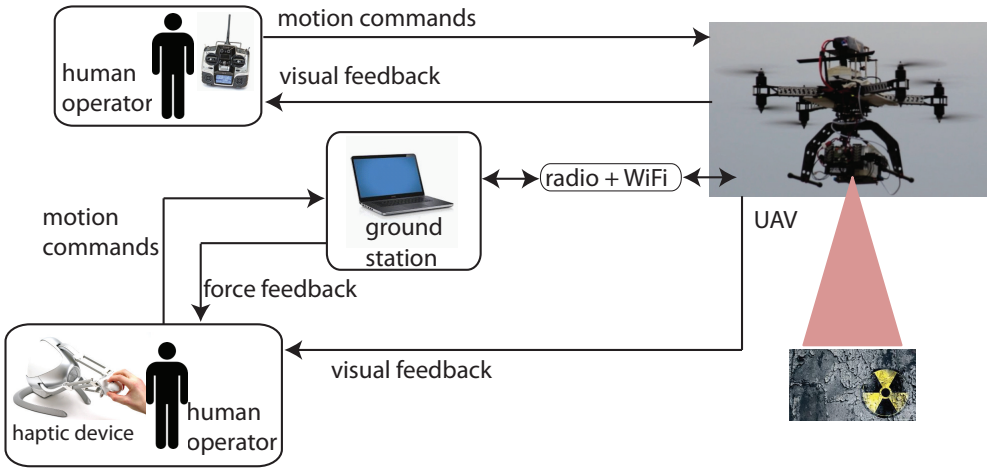


Figure 4.1: Overall system architecture of the HAR teleoperation approach.

UAVs. Both operators receive a direct visual feedback of the UAV.

The main functions of the ground station are the following: processing motion commands provided by the operator of the haptic device and sending them to the UAV, receiving sensor data from the UAV, computing force feedback and sending force signals to the haptic device. Information received by the ground station includes UAV telemetry data (position, speed, height, battery charge, etc..), sent through the UAV radio link, and sensor data from the onboard X-ray detector (number of photon counts for each energy band in a fixed time frame), sent through a wireless link.

The system architecture is based on a non-colocated VHAR setup (Section 2.1) but the visual component is ignored in this chapter (see Chapter 5 for the description of the complete system). A representation of the setup is shown in Figure 4.2. In relation to the non-colocated template, Table 4.1 reports the implementation for this specific outdoor HAR application. The world (or workspace) reference system W is the main frame. The W frame represents the position and orientation of the UAV before the takeoff. The haptic reference system H is placed behind the W frame, in a way that allows the operator to see the UAV and the exploration area. The movement commands sent by the haptic interface to the UAV are expressed assuming that the relative position and orientation of the W and H frames are as shown in Figure 4.3.

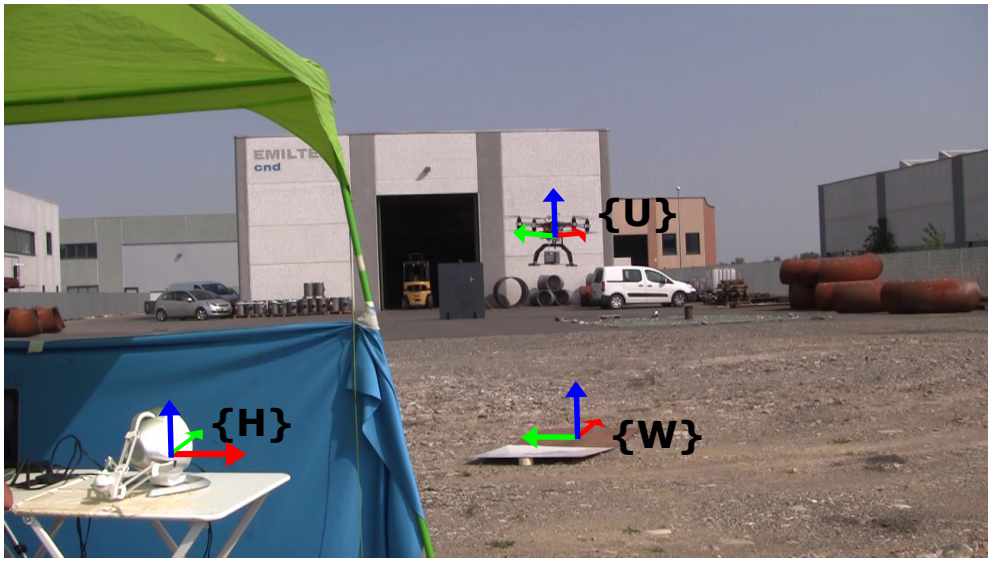


Figure 4.2: Outdoor HAR interface setup including the workspace, the UAV and the haptic user interface.

Non-colocated Template	Implementation
Workspace	Outdoor area with possible radioactive materials
HMI	Screen and haptic device (Novint Falcon)
Camera	-
Actuators	UAV equipped with X-ray detector

Table 4.1: Implementation of the non-colocated VHAR setup template for the outdoor HAR application.

Before the takeoff, the UAV frame U is coincident to the W frame. During the flight the U frame is moving, indicating the pose of the UAV. The relative pose of U with respect to W is estimated using the position and orientation information calculated by the UAV autopilot. The estimation is based on an Extend Kalman Filter (EKF) that fuses the data acquired by GPS, IMU and barometer. The relative transformation ${}^{W}_{U}T$ is expressed in the cartesian reference system. To translate the EKF position (also

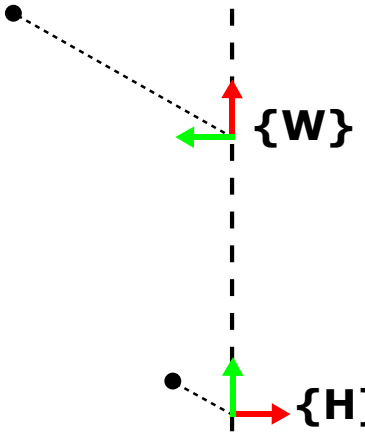


Figure 4.3: Relative pose of W and H frames for the haptic teleoperation. The black circle represents the haptic toolpoint position (bottom) and the direction/position for the UAV (top).

called GPS position or GPS coordinates), expressed in WGS84 coordinates (i.e. latitude and longitude), to cartesian coordinates we apply a formula that calculates the metric distance between two WGS84 points and we consider the *W* frame position as starting point. All the other WGS84 positions of the UAV during the flight are referred to that point.

4.2 UAV platform

4.2.1 Aerial vehicle

The UAV equipped with the CdZnTe X-ray detector is shown in Figures 4.4 and 4.5. The aerial vehicle is the Tiger Shark multicopter (octocopter in coaxial configuration) produced by VirtualRobotix Italia (`www.virtualrobotix.it`), with payload up to 4 Kg and maximum flight time of about $10-15$ minutes in the complete operational configuration including the onboard electronics and radiation detector. The UAV sends telemetry data in real-time to the ground station and sensor data measured by the X-Ray detector in near real-time. Detailed technical features of the UAV are as follows: the UAV is built on a light frame with a mixed Carbon and Alu struc-

ture; the size of the frame is within 550 mm without propellers. The UAV is equipped with MEMS accelerometer, gyro, magnetometer and GPS sensors. The VRBrain autopilot system is used (based on the Arducopter firmware, adapted by VirtualRobotix Italia), which comprises a 168 MHz ARM CortexM4F microcontroller with DSP and floating-point hardware acceleration (8 RC Input standard PPM, 8 RC Output at 490 Hz). The autopilot system supports multiple flying modes like loiter, return to home and guided mode. An integrated high speed data flash memory is used for data logging.

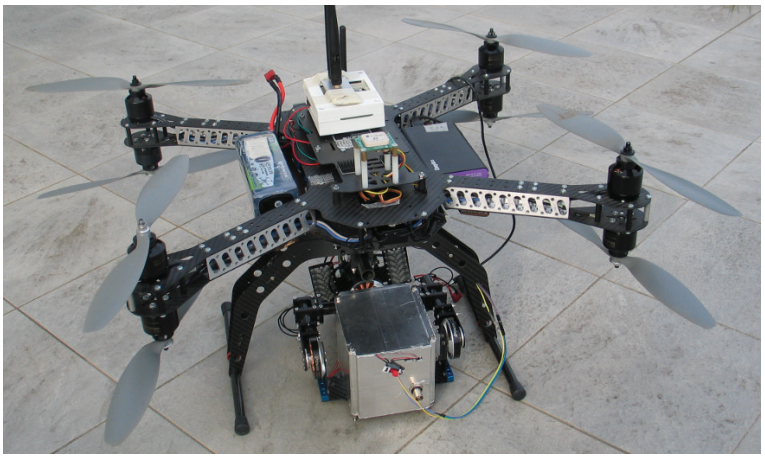


Figure 4.4: UAV equipped with CdZnTe X-ray detector (mounted at the bottom).

The CdZnTe X-ray detector detector is enclosed in a box and mounted on a two-axes brushless gimbal unit with an anti-vibration system. The gimbal unit is automatically controlled by a micro-controller with stabilization software based on IMU sensor and it can also be manually operated, if necessary, by the second operator using the remote controller input. The size of the gimbal unit is about $200 \times 150 \times 160$ mm. An embedded PC based on the Intel Galileo platform reads sensor data from the X-Ray detector and sends the data stream to the ground station through a 5 GHz WiFi connection. Intel Galileo is an x86 platform (i586) with a single core @ 400 MHz and 256 MB RAM. Sensor data from the CdZnTe detector is written on the serial port of the Intel Galileo. The dedicated wireless data connection link avoids

Figure 4.5: UAV equipped with CdZnTe X-ray detector in a flying test in a real environment.

bandwidth contention on the UAV RF channel and does not affect the UAV autopilot system, which runs on a real-time operating system. Two external antennas are connected to the embedded platform allowing a WiFi communication range up to 170 m. The embedded system is powered by an external Li-Ion 10 Ah, 5 V battery pack that ensures long-lasting use (about 9 hours).

4.2.2 CdZnTe X-ray detector

The purpose of the X-ray detector is the detection of nuclear sources on the ground in a wide energy range to reveal most dangerous contaminants that may be dispersed in the environment. The detector, designed and developed for this work, is lightweight (about 0.3 Kg) and it has low power consumption. The measurable energy range is from 10 KeV to 1.3 MeV (up to Co 60). Radioactivity data measured by the detector is represented as a histogram of 2800 energy bands. Each energy band contains the number of counts detected in a time frame $\triangle t = 2s$. The count rate C for each energy

band i varies according to the inverse square law

$$C \propto \frac{1}{l^2} \tag{4.1}$$

where l is the distance to the nuclear source. Radioactivity data are transmitted to the ground station through the WiFi link. The time required to send a single spectrum to the ground station with all 2800 energy bands (including sensor reading and transmission) is 2.2 s (0.45 Hz). The sensor features a high energy resolution (> 10 counts/sec for a Co 57 source of 1 mSv/year at 2 meters from the detector) in a large field of view. Since the highest the energy of the photon to be revealed the largest the thickness of the detector that must be used, a 6 millimeter thick CdZnTe (Cadmium Zinc Telluride) detector (single carrier) has been exploited. The choice of a single carrier device was due to the fact that hole transport properties are not good enough to ensure optimal charge drift over several millimeters. Such detector can be obtained by using a particular contact geometry known as "coplanar grid". The area of the detector is 20×20 mm.

As one of the most interesting applications of the proposed system is the detection of nuclear sources that can damage operators and workers in industrial or material stocking plants, the detector was designed to measure nuclear sources whose average effective dose is 1 mSv/year at 1 m distance. Indeed, 1 mSv/year is the dose limit set by the law for workers. Table 4.2 reports the typical number of counts per second measured by a $20 \times 20 \times 6$ mm detector at 2 m distance from some nuclear sources. Reported values show that by using the proposed detector on the UAV at about 2 meters from the ground it is possible to measure a number of counts per second that is enough for localizing nuclear sources that are relevant for workers safety.

An automatic procedure is performed at the beginning of each flight for background radiation removal assuming that the UAV take off location is far away from all the nuclear sources of the environment. Indeed, once the UAV is airborne and hovering at the desired height a set of radiation measurements is taken for some time frames $\triangle t_j$, $j = 1 \ldots K$, then the intensity of the background radiation is set as $I_b = \max_{\triangle t_j} \sum_i C_b[i]$, i.e. the largest measured value of the sum of the counts over all energy bands $i = 1 \ldots 2800$. Then, when the UAV is flying during the teleoperated

Table 4.2: Measured radioactivity from different nuclear sources by a $20 \times 20 \times 6$ *millimeter detector at* 2 *meter distance.*

Nuclear source	dose (mSv/year)	source activity (Bq)	counts/sec
Americium 241	1	$1.6E8$	1270
Cobalt 57	1	$5.4E7$	411
Cesium 137	1	$8.2E6$	159

mission the current radiation intensity I in a time frame, which is used to update the position on the ground with the highest radiation that in turn is needed to generate force feedback (as explained in Section 4.3), is computed as the difference between the current measured value $\sum_i C_m[i]$ and the background radiation acquired at the beginning of the mission, i.e.

$$I = \begin{cases} \sum_i C_m[i] - I_b & \text{if } (\sum_i C_m[i] - I_b) > \Gamma, \\ 0 & \text{otherwise ,} \end{cases} \tag{4.2}$$

where Γ is a small threshold.

4.3 Haptic teleoperation scheme

In aerial monitoring for detection of environmental hazards safety is a major issue which is often dictated by strict national regulations. Hence, we propose a teleoperation system including two operators: a main operator in charge of guiding the vehicle by means of the haptic device during the hazard search and localization phase, and a secondary operator overseeing the flight mission with a standard remote controller. Also, in the proposed system to optimize sensor performance the UAV maintains its current altitude and it flies at constant speed. The second human operator is responsible for take off, landing, and setting the altitude set point of the UAV. Moreover, the second operator can override at any time haptic commands thereby taking full control of the UAV. Both operators have direct visual feedback of the UAV and are also supported by a GUI returning the UAV position overlaid on a map and the measured

radiation value. Technically, the two roles could be fulfilled by a single operator by exploiting the three degrees of freedom of the haptic device but at the cost of additional operator's overload and increased safety risk. The haptic device adopted is the same of Chapter 3, the Novint Falcon.

Assuming a planar environment, the detector will measure a maximum intensity when it is on the vertical of the radiating target. The main operator controls the UAV position in the x, y plane. The GPS coordinates of the UAV will be assumed as the coordinates of the radiating part at the ground.

Two impedance control modes (Figure 4.6) have been developed where the operator sends motion commands (via the haptic device tool point) and receives force feedback limited to the xy plane. In particular, the ground station processes the motion

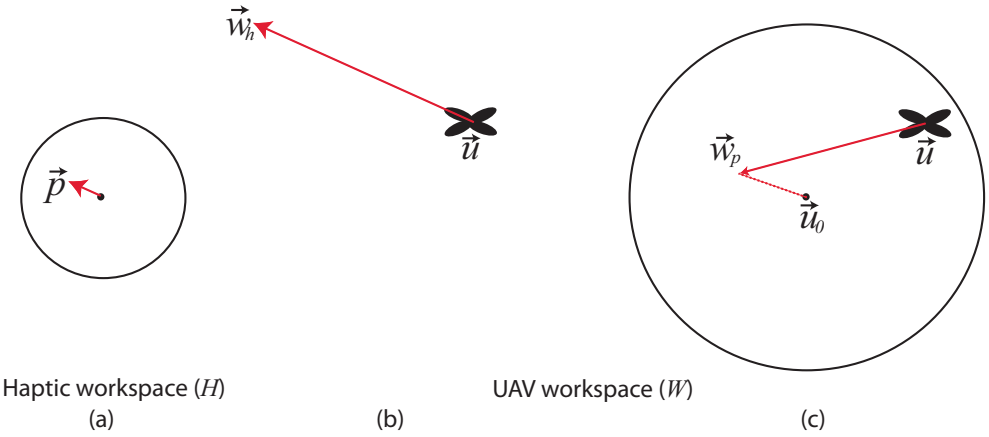


Figure 4.6: Diagram illustrating the two haptic control techniques. Haptic workspace (a) on the x, y plane. Position-heading technique (b), the current displacement $\vec{p}$ of the haptic device is used to compute a heading direction with respect to the current UAV position. Position-position technique (c), $\vec{p}$ is used to compute a waypoint with respect to the center of operation $\vec{u}_0$.

commands provided by the operator of the haptic device and sends them to the UAV, receives sensor data from the UAV, computes the force feedback and actuates the force signals to the haptic device.

Since inspection tasks require constant flight speed, in both control modes there is

no direct mapping from the haptic device to the speed of the UAV, which is set to 0.5 m/s. Moreover, the two control modes have been designed to map the limited workspace of the haptic device to an arbitrarily large UAV workspace.

In the first control mode ($M_{heading}$) position information read by the haptic interface (namely, the x, y coordinates of the tool point) is used to compute the horizontal heading direction of the UAV. In particular, the current displacement $\vec{p} = (p_x, p_y, p_z)$ of the tool point of the haptic device with respect to its center is converted to a waypoint $\vec{w_h} = (x, y, z)$ for the UAV in world coordinates as follows

$$\vec{w_h} = \frac{\eta}{\|\vec{p}\|} \underbrace{\begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}}_{{}^{W}_{H}\mathbf{T}} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{4.3}$$

$$\eta = \begin{cases} 0 & \text{if } \|\vec{p}\| < D \\ 1 & \text{otherwise} \end{cases} \tag{4.4}$$

where $\vec{u} = (u_x, u_y, u_z)$ is the current position of the UAV, α is a constant and ${}^{W}_{H}\mathbf{T}$ is the rotation matrix from the haptic coordinate system H to the world reference frame W. The world reference frame corresponds to the UAV pose at take off. A flat earth model is assumed for the conversion between cartesian and GPS coordinates. The UAV flies in guided mode following the current waypoint with yaw heading pointing towards the waypoint itself. Each waypoint $\vec{w_h}$ is not meant to be reached by the UAV as it is continuously updated and placed at a long distance from the current UAV position ($\alpha \gg 0$), thus resulting in a position-heading approach. In Equation 4.3 the altitude of the generated waypoint (z coordinate) does not depend on the displacement of the tool point of the haptic device p_z. Indeed, as explained above the UAV altitude set point u_z is provided by the second operator via the standard remote controller. If the tool point of the haptic device is within a fixed range D from the center of the haptic reference frame the waypoint is not updated ($\eta = 0$ in Equation 4.4) and the UAV hovers.

The second control mode is a hybrid approach that allows the operator to switch between the position-heading technique ($M_{heading}$), discussed above, and a position-position technique ($M_{position}$). Mode switching is achieved by pressing one of the buttons of the haptic device. The position-position technique, inspired by the approach proposed in [49], enables a direct mapping between the haptic device to the UAV position around a fixed center of operation. In particular, in this second control mode a waypoint $\vec{w}_p = (x, y, z)$ for the UAV in world coordinates is computed, for both control modes, as follows

$$\vec{w}_p = \begin{cases} \frac{\eta}{\|\vec{p}\|} {}^{W}_{H}\mathbf{T} \begin{bmatrix} \delta & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} u_{0x} \\ u_{0y} \\ u_{0z} \end{bmatrix} & \text{if } M_{position}, \\ \\ \vec{w}_h & \text{if } M_{heading}. \end{cases} \tag{4.5}$$

where $\vec{u}_0 = (u_{0x}, u_{0y}, u_{0z})$ is the center of operation, defined as the position of the UAV when $M_{position}$ is activated, and $\delta \ll \alpha$ is a constant. The idea is that the position-heading control technique can be used for long transition motions, e.g. when the UAV has to be moved to explore different regions of the environment, while the position-position technique is helpful when the UAV explores an area in close proximity of a nuclear source.

For both control modes a force feedback $\vec{f}$ is provided to the operator, acting as a basin of attraction, to let the UAV fly close to the region where the radiation is maximal. Indeed, as the remotely operated UAV travels away from the location of the most intense perceived radiation the haptic device exerts a force in the horizontal direction towards the center of the haptic workspace. The force feedback is computed as

$$\vec{f} = \frac{\zeta}{\|\vec{r} - \vec{u}\|} \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}}_{{}^{H}_{W}\mathbf{T}} \begin{bmatrix} \beta & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_x - u_x \\ r_y - u_y \\ r_z - u_z \end{bmatrix} \tag{4.6}$$

$$\beta = \frac{f_{max}}{L^2} d^2(\vec{r}, \vec{u})_{xy} \tag{4.7}$$

$$\zeta = \begin{cases} 0 & \text{if } d(\vec{r},\vec{u})_{xy} > L \\ 1 & \text{otherwise} \end{cases} \tag{4.8}$$

where $\vec{r} = (r_x, r_y, r_z)$ is the estimated position of maximum radiation discovered so far by the UAV, which is updated over time. Radiation intensity I at each position of the UAV in time frame $\triangle t$ from Equation 4.2. If the latest measured radiation intensity is greater than the maximum radiation found so far, the latest radiation intensity is set as the current maximum radiation. The force feedback has a quadratic profile (Equation 4.7) where $d(\vec{r},\vec{u})_{xy}$ is the distance between the projection of the UAV on the ground and $\vec{r}$; f_{max} is the maximum exerted force, and L is the radius of the basin of attraction. No force feedback is provided outside the basin of attraction ($\zeta = 0$ in Equation 4.8).

The provided force does not overpower the user who can also guide the UAV outside the basin of attraction if he/she desires. Moreover, the user can reset force feedback to zero by pressing a button on the tool point of the haptic device. This feature can be useful when there are multiple concentrated radiation sources to be explored.

4.3.1 Software architecture

The software architecture, illustrated in Fig. 4.7, consists of three main modules: the first software module consists of a set of processing nodes based on the ROS (Robot Operating System) middleware, the second module is the QGroundControl (QGC) application and the third module is the software running on the UAV. Both the ROS module and QGroundControl run on the ground station computer. The software running on the UAV includes the Arducopter firmware of the VRBrain board and the software on the embedded system for acquisition and transmission of radioactivity data from the CdZnTe detector. The telemetry channel between the VRBrain board and the ground station uses the MAVLink protocol. The embedded system runs a Debian Wheezy operating system. QGroundControl works as a proxy for the ROS module by receiving telemetry data from the UAV and forwarding them to the MAVLink communication node. Moreover, QGroundControl transmits the GPS waypoint to

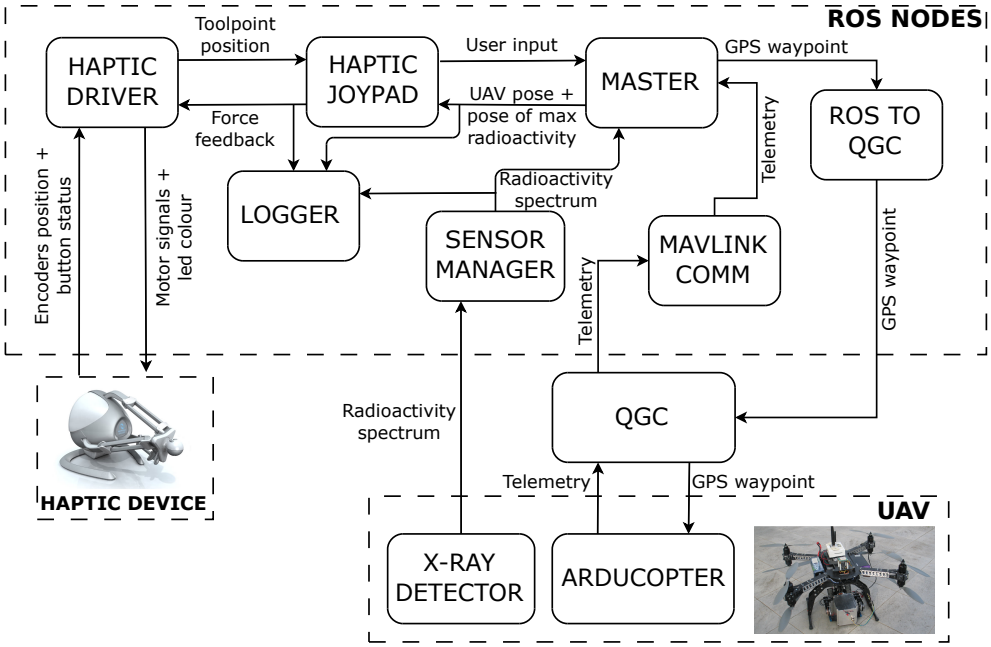


Figure 4.7: Software architecture. Three main modules can be identify: the ROS nodes, the QGC ground station and the UAV system.

the UAV. The MAVLink communication node is in charge of converting MAVLink telemetry to ROS messages. The Master node subscribes to ROS topics to receive telemetry data, user inputs commands and radioactivity data. The Master node generates the waypoint for the UAV and updates the position of the maximum measured radiation. The waypoint is converted from Cartesian world coordinates into GPS coordinates by using the current GPS position of the UAV and its orientation at take-off. The generated waypoint message received by the "ROS to QGC" node, is then forwarded to QGroundControl and, subsequently, to the UAV. The Haptic driver node publishes the actual position of the tool point of the haptic device and applies force feedback to the haptic device. The Haptic joypad node computes the desired UAV direction in Cartesian coordinates, which is published on a topic subscribed by the Master node, and it also computes force feedback for the Haptic driver, based on the position of the current maximum radioactivity. The radioactivity spectrum from

the detector is sent to the Sensor manager node on the ground station using custom MAVLink messages over a dedicated Wi-Fi UDP connection. The Master node continuously updates the value and the position of maximum radiation found so far during the mission. If the latest measured radiation intensity is greater than the maximum radiation found so far, the latest radiation intensity is set as the current maximum radiation. The position of the maximum radiation source and the current position of the UAV are published as ROS topics and are used by the Haptic joypad node to determine whether or not a force feedback must be provided to the user. Finally, a Logger node was developed to record data of the mission including the UAV telemetry, the radioactivity spectrum and the applied force feedback.

4.4 Experiments

4.4.1 Preliminary evaluation

Initial evaluation of the CdZnTe detector was carried out in an indoor laboratory setup by measuring the number of counts at each energy band for a light Cobalt-57 source that was located very close to the detector (UAV turned on but not flying). The measured radiation spectrum is shown in Figure 4.8 (top). Test flights of the teleoperation system were then performed with the CdZnTe detector mounted on the UAV in an open field without radiation sources as shown in Fig. 4.9. The radiation spectrum of the background radiation signal measured in the open field is reported in Figure 4.8 (bottom). It can be notice that the sensor exhibits a peak at low radiation which is due electrical noise. Hence, in the experiments reported in Subsection 4.4.3 the energy bands below 50 KeV were ignored.

4.4.2 Experiments with simulated radiation sources

This subsection illustrates experiments that have been performed to assess the haptic teleoperation interface in detection of nuclear sources in partially simulated environments. A small UAV, which does not carry the X-ray detector, was teleoperated in a real environment with simulated radiation sources on the ground. Flights were

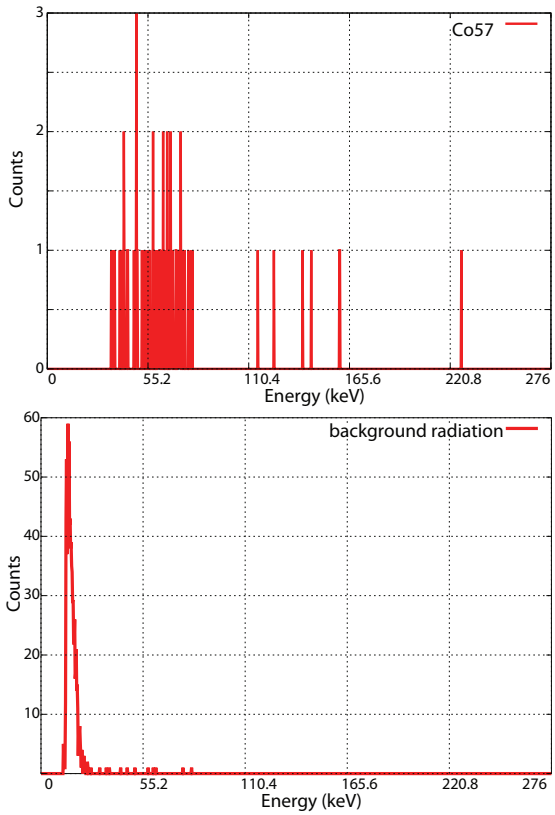


Figure 4.8: Number of counts measured in laboratory for a light Cobalt-57 source (top histogram), and in the test flight shown in Figure 4.9 (bottom histogram) in time frame $\triangle t = 2s$; the low-energy peak is due to noise. The right tail of the histogram is truncated after 276 *KeV.*

performed in an open grass field by an expert operator. Positions of the simulated radiation sources were unknown to the operator and were only used to compute the simulated sensor readings. The small UAV flew using the same haptic teleoperation and software architecture described in Section 4.3.1. Figure 4.10 shows images of the ground station and the QGroundControl program.

In particular, experiments have been performed with up to two simulated radiation sources on the ground: Am241 (Americium) and Cs137 (Cesium). Basin of force feedback attraction was $L = 9$ m with $f_{max} = 9$ N. Flight time was about 6 minutes.

Figure 4.9: Teleoperation experiment in an open grass field. The sensor equipped UAV measures background radiation.

Figure 4.10: Ground station with haptic device (left) and QGroundControl program (right).

The UAV maintained a steady height from the ground at about 3 meters. Figure 4.11 shows a picture of the experiments. Figure 4.12 shows two example states of the exploration and the radiation counts given by the simulated sensor. The simulated measurement of the total detected radioactivity (sum of the contributions of the two nuclear sources) is quite similar to the radiation of the closest nuclear source to the UAV.

The complete flight path of one of the experiments is reported in Figure 4.13, highlighting the regions where a radiation was measured by the simulated sensor and the estimated location of maximum radiation. Figure 4.14 reports the plot of force

Figure 4.11: A teleoperation experiment using a small UAV with simulated radiation sources and haptic feedback.

feedback returned to the user in the experiment (Equation 4.7) as well as the distance $d(\vec{r},\vec{u})_{xy}$ between the projection of the UAV on the ground and the location of maximum radiation. Figure 4.14 also reports the radiation intensity I. It can be noticed that force feedback intensity increases and decreases as $d(\vec{r},\vec{u})_{xy}$ increases and decreases. Force feedback drops to zero when the operator presses the reset button and when the UAV flies outside the basin of attraction.

The usefulness of force feedback was evaluated in 10 experiments that were repeated with force feedback disabled. When force feedback was disabled the expert operator could only feel the presence of a radioactive source by looking to a plot of

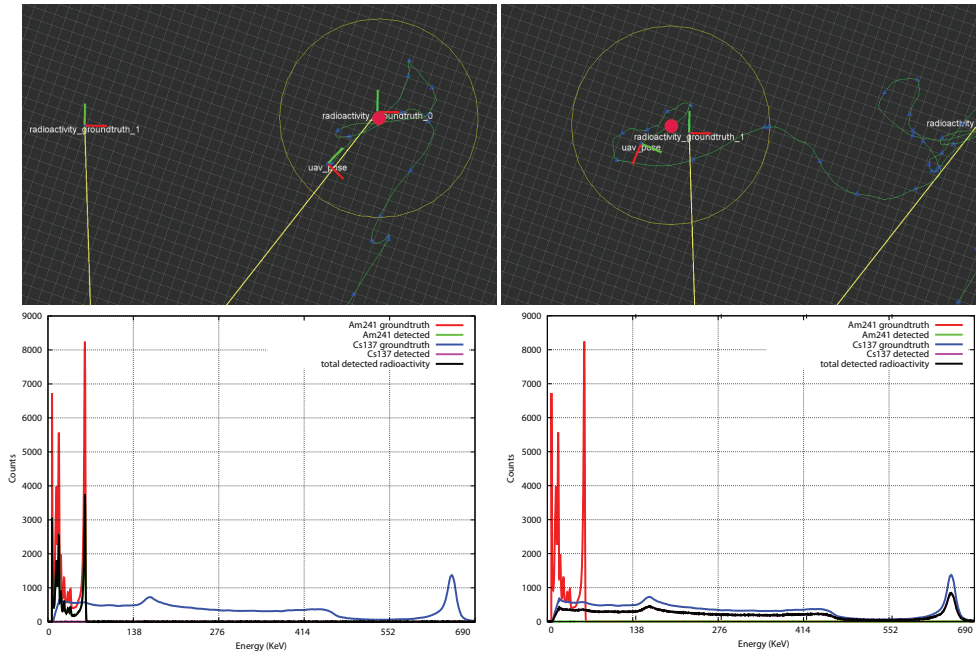


Figure 4.12: Two states of the experiment with the small UAV. UAV flying close to the Am241 source (left column) and UAV flying close to the Cs137 source (right column). Top row shows the current UAV location, the actual location of the two radiation sources (unknown to the operator), the traveled path of the UAV (green line) and the current estimated location of maximum radiation (red dot). Bottom row shows the ground-truth spectrum of each simulated source, the number of detected counts for each source and the total detected radioactivity (sum of the measured radiation due to both sources). It can be noticed that in both states the total detected radioactivity (black line) is quite similar to the radiation of the closest radioactivity source to the UAV. In the second state, after exploring the Am241 source, the user reset force feedback and the estimated location of maximum radiation by pressing a button on the haptic device.

the detected spectrum displayed on the screen of the ground station (Figure 4.15). The operator always had direct sight of the UAV during the experiments; however, when force feedback was disabled the operator had to switch his attention to the screen of the computer, as shown in Figure 4.16 (right). Each exploration experiment

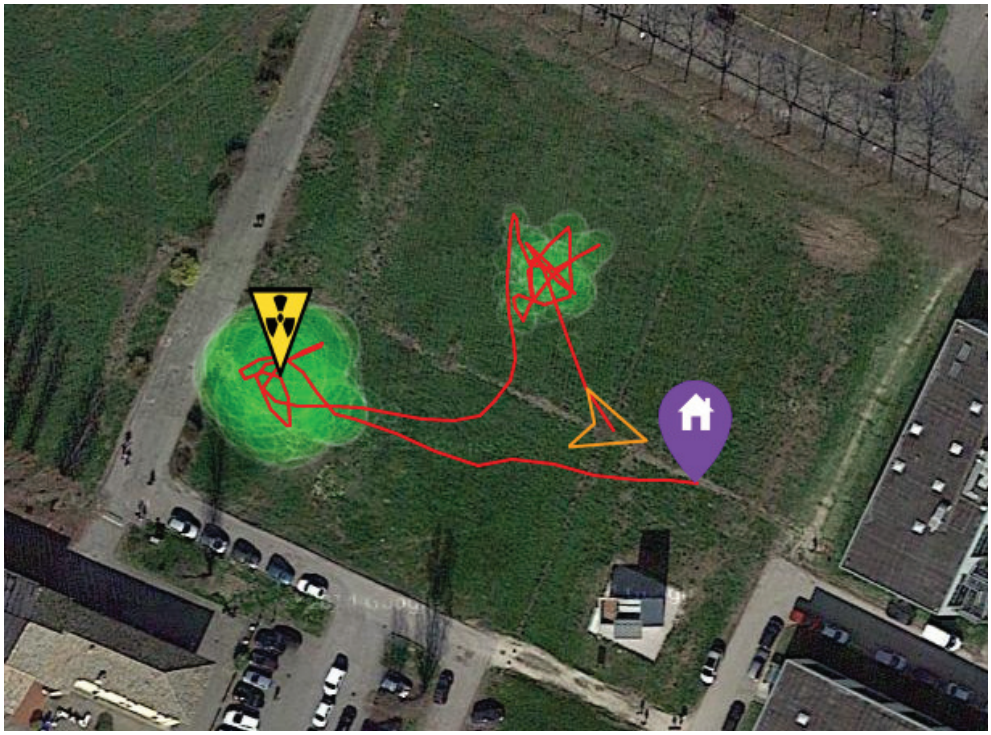


Figure 4.13: Flight path of one of the experiments with the small UAV including the regions where a radiation was measured by the simulated sensor (green circles with different size and shading) and the estimated location of maximum radiation (apex of yellow triangle).

of a simulated radioactive source ended when the operator believed that the current position of maximum radiation was close enough to the radiation source.
Table 4.3 summarizes the results including the average time taken to detect a radioactive source, the detection accuracy (measured as the difference between the position of maximum radiation at the end of the experiment and the actual one using GPS coordinates) and the fraction of time $\frac{t_s}{T}$ spent looking at the screen of the computer (where t_s is the time spent looking at the screen and T is the total time of exploration). Both the average exploration time and the average detection error were slightly lower (i.e., better) for haptic teleoperation, although the difference was not found to be statistically significant. A noticeable result is that in case of absence of force feedback

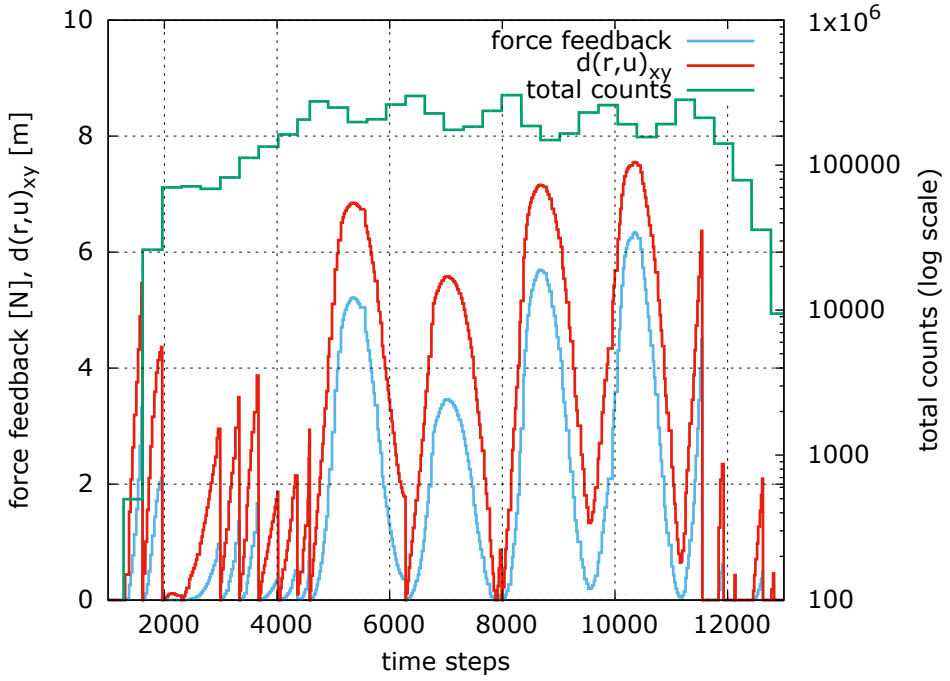


Figure 4.14: Plot of force feedback intensity, distance $d(\vec{r},\vec{u})_{xy}$ and radiation intensity I for one of the experiments with the small UAV and simulated radiation sources.

Table 4.3: Comparison of haptic and pure visual feedback in detection of radioactive sources.

	detection time (s) avg. (std.dev.)	detection error (m) avg. (std.dev.)	$\frac{t_s}{T}$
haptic	129.7($\pm$16.4)	3.0($\pm$3.3)	0%
visual	144.7($\pm$25.4)	5.9($\pm$7.2)	22.8%($\pm$1.7%)

the operator had to spend about 22.8% of the time looking at the screen, whereas with the haptic feedback turned on he/she never had to check the auxiliary visualization devices and rather remained steadily focused on the flying UAV. This result shows that the use of force feedback is helpful in reducing operator mental workload. Mental workload indicates the amount of mental effort an operator provides to accomplish a task. A high level of mental workload is normally associated to a high

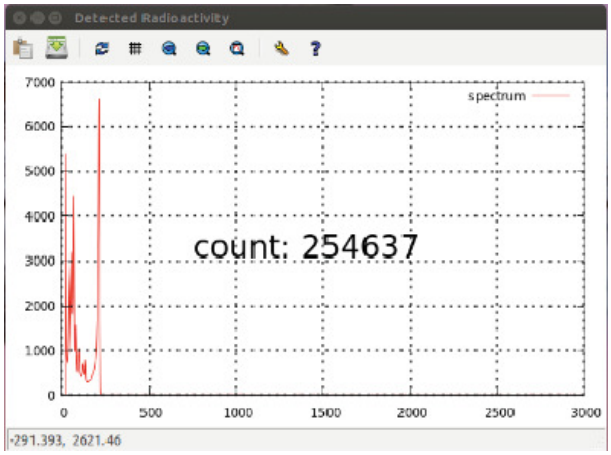


Figure 4.15: Radioactivity spectrum displayed on the screen of the ground station in teleoperation experiments with pure visual feedback.

Figure 4.16: An image taken during haptic teleoperation with force feedback enabled (left) and an image taken when force feedback was disabled with pure visual feedback of radioactivity data displayed on the screen of the ground station (right).

level of effort, concentration and/or possibly frustration. Hence, in conclusion the use of force feedback causes a lower mental workload level that helps to minimize the risk of dangerous conditions like collisions or crashes.

4.4.3 Experiments in real environments

The haptic teleoperation system has been finally validated in an outdoor environment with a real nuclear source under the supervision of the regional environmental

protection agency. Experiments have been performed with an intense nuclear source (Iridium 192) located outdoor in a service area of an industrial plant located in Gossolengo, Piacenza, Italy. The nuclear source was put into a vertical lead pipe placed on the ground to prevent lateral radiation emission (Figure 4.5). The operator was kept at a safety distance of about 30 m from the nuclear source. The UAV maintained a height from the ground ranging from 1.5 to 3 meters.

A first experiment has been performed in position-heading control mode with prior knowledge of the location of the nuclear source, that was placed in direct sight of the operator. Figure 4.17 shows two images of the experiment, the first at the beginning of the experiment once the UAV is airborne, the second at the end, as well as the corresponding total accumulated spectra. Figure 4.18 shows the to-

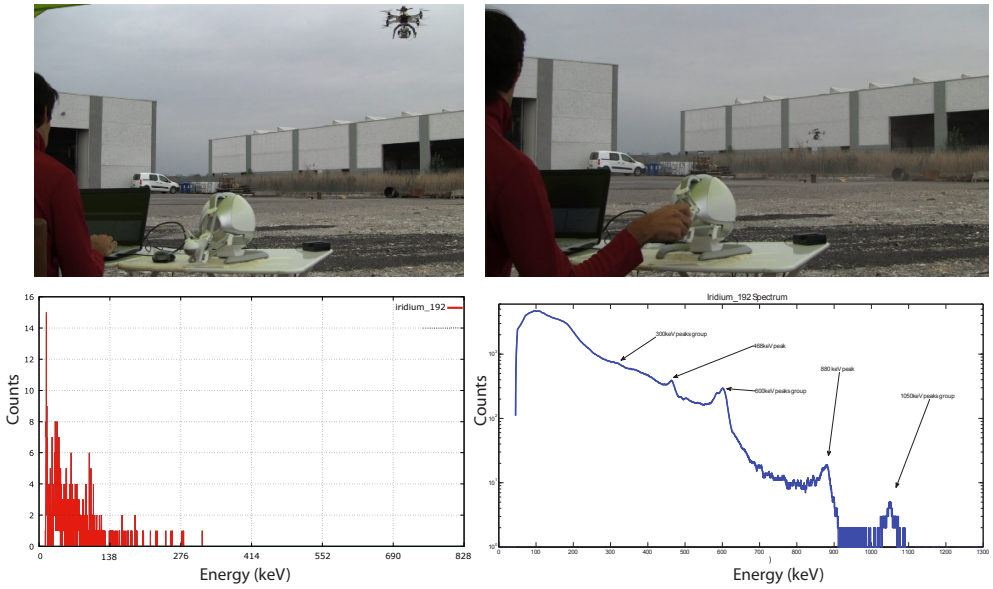


Figure 4.17: Detection of Iridium 192 in real environment (operator has direct sight of the nuclear source): image at the beginning of the experiment (top left) and at the end of the exploration task (top right). Total radiation spectrum (obtained by summing up all measurements) at the beginning of the experiment (bottom left) and at the end of the exploration task (bottom right).

tal flight path of the UAV and the radioactivity data. Flight time of the experiment

Figure 4.18: Detection of Iridium 192 in real environment (operator has direct sight of the nuclear source): flight path, regions where a radiation was measured by the CdZnTe detector (green circles with different size and shading) and the estimated location of maximum radiation (apex of yellow triangle).

was about 7 minutes. In this experiment the error between the actual position of the nuclear source and the location of the maximum measured radiation intensity was about 2.5 m (estimated using GPS coordinates). A video of the mission is available at `www.ce.unipr.it/%7Emicconi/experimentIridium192.wmv`.

A second set of experiments with the same real source has been carried out. In these experiments the operator view is partially covered by a cloth and he/she is not able to see the ground, i.e. he/she can not see where the radioactivity source has been placed (Figure 4.19 top). The operator can only see the UAV for safety reasons. The position of the source has been changed for each experiment. These tests aim to

Figure 4.19: Detection of Iridium 192 in real environment using the position-position teleoperation mode. Experimental setup (top), where the UAV is highlighted by the red circle. Image taken from the onboard camera (bottom) when the operator affirmed that the current position of maximum measured radiation was close enough to the real nuclear radiation source (located at the center of the target within the lead container). The cloth in front of the operator prevented him from seeing the location of the nuclear source on the ground. In this set of experiments the gimbal unit was not used to get a clearer view of the ground from the onboard camera. The low speed of the UAV ensured that the X-ray detector remained parallel to the ground during the mission.

compare the two control modes, position-heading mode and position-position mode. A video camera is installed on the UAV pointing to the ground to record each flight.

The recorded videos are used offline to measure the error between the estimated position of the source, while the operator is holding the UAV above that specific position, with respect to its actual location. To be able to detect significant differences in spatial accuracy between the two modes, i.e. with more resolution with respect to the GPS, a target with concentric circles (0.5 m radius step size) has been placed on the ground below the source (Figure 4.19 bottom).
On average using the position-position teleoperation mode the operator was able to detect the location of the nuclear source with a position error less than 1.5 m, while using the position-heading teleoperation mode the position error was about 2.5 m (as the previous test). The distance between the take-off position of the UAV and the location of the source was, on average, 20 m. The average flight time was 5 minutes.

One last test has been performed in an real application context. The mission has been conducted in a garbage dump, as shown in Figure 4.20. The goal of the test was the detection of radioactive materials not known a priori using the developed haptic teleoperation interface. No radioactivity source has been discovered and the result has been confirmed by an external examination.

4.5 Discussion

In this chapter we discuss a Haptic AR interface for the teleoperation of a X-ray detector equipped UAV. This interface is the haptic component of the complete outdoor VHAR interface presented in Chapter 5. The HAR interface allow a human operator to perform a radioactivity scan of an area with an high level of safety, without the risk to be exposed to hazards. A novel haptic rendering algorithm, based on two control modes, has been developed and evaluated in simulated and real scenarios.
The haptic teleoperation approach generates an attractive force feedback around the location of the most intense detected radiation enabling the discovery of radiation sources. Although several works have been proposed on the haptic teleoperation of ground and aerial vehicle, the majority use the haptic feedback to avoid collisions. This is one of the first attempts of use the haptic feedback to attract the vehicle to-

Figure 4.20: Test in a real application context: haptic teleoperation of X-ray detector equipped UAV in a garbage dump.

wards a point.

The assessment of the tests indicates that the temporal and spatial performance improvement of the haptic teleoperation is not statistically significant with respect to the visual feedback only teleoperation. However, the haptic teleoperation obtains more constant results with respect to visual feedback only teleoperation. Moreover, the haptic teleoperation has the advantage of reducing the mental workload of the operator, allowing him/her to focus entirely on the flight of the UAV, without the need to look at the radioactivity data on a screen. This feature also increases the safety of the flight. Hence, the HAR interface is able to increase the SA level of the operator, communicating the radioactivity data to him/her with less mental workload. The comparison of the two control modes highlights that the position-position mode enable the operator to better identify the radiation source position, decreasing the spatial error from 2.5 m to 1.5 m.

The HAR interface has been also tested in a real application scenario, a garbage

dump, without discover any radioactive materials and the result has been confirmed by an external certified analysis.

Chapter 5

Visuo-Haptic AR for Aerial Environmental Monitoring

This chapter completes the description of the outdoor VHAR interface whose haptic part has been introduced in Chapter 4. In particular, emphasis is given to the Visual part of the interface, since the Haptic component has been discussed in the previous chapter. The VHAR interface aims to improve the *Situation Awareness* (SA) of the human operator during the UAV flight. In Chapter 4 we showed that the haptic augmentation helps the operator to stay focused on the flight and the exploration of an area while looking for radioactive materials. In a generic UAV mission, the human operator must focus on the task but he/she is also required to verify sensor data acquisition, be aware of the state of the UAV, and check for potential obstacles or dangerous conditions related to the flight. However, maintaining a high level of SA is challenging since the task imposes a high mental workload for the human operator. In this chapter we show how expanding the interface with an additional augmentation channel, in this case a visual feedback, can provide additional assistance to the operator and improve SA. The overall VHAR user interface shows the potential of Visuo-Haptic augmentation in outdoor applications.

In the proposed system, visual feedback from a fixed camera on the ground, that observes the environment where the UAV is flying, is displayed on a computer screen

and is augmented with both 2D and 3D graphical overlays. Three types of graphical overlays are rendered in real-time. First, a 3D histogram of the measured radiation intensity is displayed on top of the video stream to let the operator see the most recent measured intensity values as well as the location of the maximum radiation found during the mission. Second, a 2D virtual cursor is displayed around the UAV that is computed from a vision-based tracking algorithm. Visual-tracking not only facilitates the operator in detecting the UAV on the image when the UAV is flying at large distances, but also improves the estimation of the UAV 3D pose and, therefore, the localization of the radioactive materials, compared to the accuracy that would result by using the UAV onboard sensors alone. Third, other elements such as buildings in close proximity to the UAV are retrieved from a geographic information system (GIS), registered using a semi-automatic approach, and highlighted on the screen. Simple bounding boxes of the building are also used to generate geo-fences for the UAV, i.e. a virtual perimeter for collision avoidance. In addition to the visual feedback, the haptic feedback is generated as an attractive force around the location of the most intense detected radiation source as described in Chapter 4.

The proposed VHAR interface has been evaluated in a real environment with the same radioactivity source of Chapter 4. In the real experiments the UAV was clearly visible to the operator for safety reasons, although the developed VHAR interface allows the UAV to be teleoperated by looking to the computer screen without a direct sight to the UAV. Usability experiments, performed in a simulated environment, are also reported. The purpose of these tests was to assess the ability of the interface to reduce the mental workload and to increase the SA level.

5.1 System architecture

The system setup is an extension of the non-colocated setup of Chapter 4. As shown in Figure 5.1 and in Table 5.1, the VHAR interface includes the same components of the HAR interface plus a fixed monocular RGB camera on the ground. The camera is needed to add the Visual component of the VHAR interface. The reference system C of the camera is also added. Furthermore, the image flow acquired by the camera is

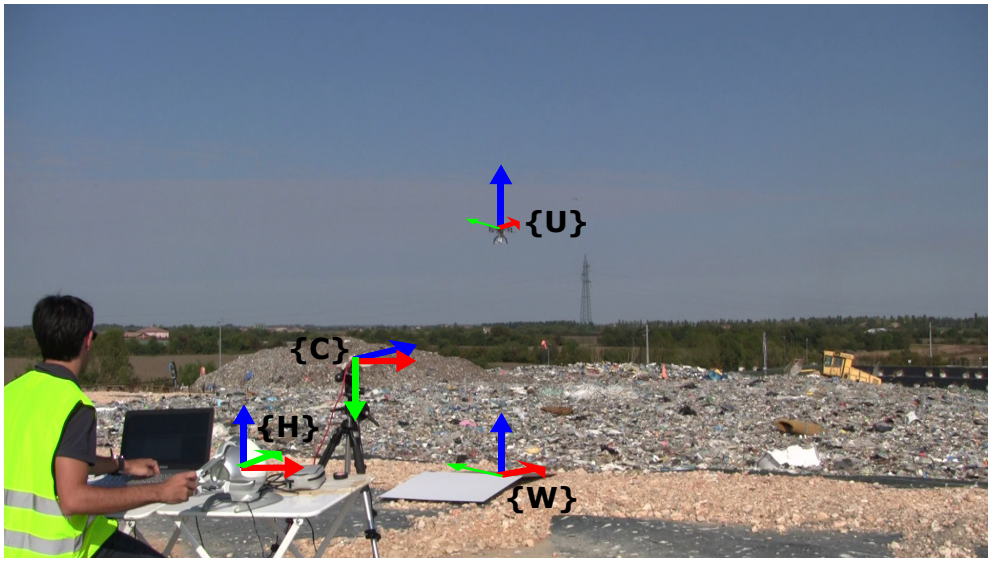


Figure 5.1: Outdoor VHAR interface setup including the workspace, the UAV, the camera, and the haptic user interface.

Non-colocated Template	Implementation
Workspace	Outdoor area with possible radioactive materials
HMI	Screen and haptic device (Novint Falcon)
Camera	Fixed monocular RGB camera
Actuators	UAV equipped with X-ray detector

Table 5.1: Implementation of the non-colocated VHAR setup template for the outdoor VHAR application.

processed using a custom computer vision algorithm, described in Section 5.2. The algorithm, implemented as a ROS node, is responsible for the detection of the UAV in the images and publication of the position information to the other ROS nodes.
In contrast to the HAR interface of Chapter 4, using the VHAR interface the operator must look at a screen (as shown in Figure 5.1). The advantage of the VHAR interface over the HAR interface is the ability to show the environment surrounding the UAV

and augmenting it, showing other important information related to the flight and the mission, while the operator is focused on the teleoperation of the vehicle supported by force feedback.

5.1.1 Calibration

In this subsection we illustrate, in relation to Section 2.2, how the VHAR interface is calibrated and how its components are mutually registered one to another. The intrinsic parameters of the camera are calibrated as reported in Section 2.2. The extrinsic parameters calibration is performed, for each mission, using an Aruco marker (Figure 2.4) placed with a known relative pose ${}^{M}_{W}T$ with respect to the world (or workspace) reference system W (Figure 5.2). The estimation of the extrinsic parameters is coincident to the estimation of the transformation ${}^{C}_{M}T$. The marker is not

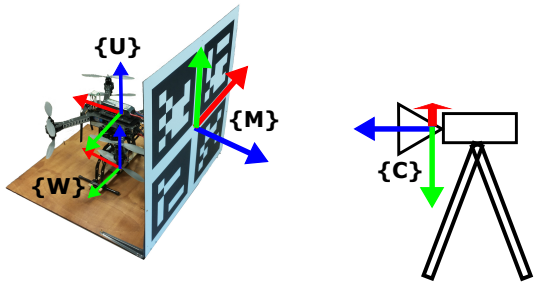


Figure 5.2: UAV takeoff platform with marker (80×80 *cm) and reference frames.*

visible in Figure 5.1 because it is lowered before the UAV takeoff. If the takeoff position does not change from one mission to another, there is no need to repeat the extrinsic parameters calibration procedure.

The current 3D position ${}^{C}p^{t}_{ekf}$ of the UAV in the camera reference frame C is given by ${}^{C}p^{t}_{ekf} = {}^{C}_{M}T\,{}^{M}_{W}T \cdot {}^{W}p^{t}_{ekf}$, where ${}^{W}p^{t}_{ekf}$ is the current 3D position of the UAV in the world reference frame W (located at the base of the takeoff platform) provided by the autopilot EKF. The EKF fuses data from the GPS sensor, IMU and barometric altimeter and, therefore, its output is strongly affected by the position error of the GPS receiver.

The spatial knowledge established by the two matrices ${}^{C}_{M}T$ and ${}^{M}_{W}T$ allows computation of the UAV 3D position with respect to the camera, and therefore with respect to the images, and generation of the Visual AR environment.

5.2 Vision-based UAV detection

In order to help the operator to visually track the UAV while it is flying and to better register the histogram bars representing the measured radioactivity intensity, a vision-based UAV detection algorithm has been developed. The proposed approach, illustrated in Algorithm 5.1, exploits motion segmentation and SVM-classification using SURF local descriptors. The algorithm was developed on the OpenCV library.

Main input data are the intrinsic camera parameters, the current image I from the camera, the UAV position estimated by the EKF in camera coordinates ${}^{C}p^{t}_{ekf}$, the projection of the UAV position on I at previous frame $\overline{p}^{t-1}_{ekf}$, a search window SW (defined by a center pixel and a size) and the last UAV predicted state $\overline{p}^{t-1}$ on I. The output is the current estimated position of the UAV in camera coordinates ${}^{C}p^{t}_{v}$. The estimated position of the UAV using visual detection is more accurate than the one provided by the autopilot EKF and, therefore, it also improves georeferencing of the radioactivity sources.

After projecting ${}^{C}p^{t}_{ekf}$ on the image plane (Line 1) the foreground images $F(I)$, representing moving objects, is computed (Line 2) using the MOG2 background subtraction algorithm [119]. A dilation is then applied to $F(I)$ (Line 3). An efficient search of the UAV is then performed in a sub-window SW of the dilated image $\overline{F}(I)$, centered at $\overline{p}^{t}_{ekf}$ (Lines 4-16). In particular, blobs are extracted from the sub-window and then, the Axis Aligned Bounding Box (AABB) Box_i of each blob is computed.

Afterwards, each box is evaluated by a binary Bag-of-Words SVM classifier, trained to detect the UAV on the image, using SURF features (Lines 9-16). The box with the highest positive score $BestBox$, if any, is selected as the candidate AABB that contains the UAV (this case is named $SURF$). The 3D position of the UAV in camera coordinates ${}^{C}p^{t}_{v}$ is estimated by back projection of the center pixel of $BestBox$ using the z component of ${}^{C}p^{t}_{ekf}$ (Line 18). The current state $\overline{p}^{t}$ of the UAV on I is also

updated using the *BestBox* center (Line 19). Finally, the size of *SW* is updated to the size of *BestBox* enlarged by a constant value (Line 20).

If a valid box is not found by the SVM classifier, the Box_i having the largest intersection with the search window and whose distance to the predicted position $\widetilde{p}^t$ of the UAV does not exceed the threshold value Th (Lines 22-30) is selected (this case is named *Closest Box*). Then, the box center is used to compute the 3D position of the UAV in camera coordinates (Line 32) as above. Prediction (Line 22) is computed using an alpha-beta filter. The size of *SW* is also incremented by a small value (Line 34). If a valid box is still not found, prediction $\widetilde{p}^t$ is used to compute the 3D position of the UAV in camera coordinates (Line 36) and $\overline{p}^t$ is updated using the difference vector between the last two measured positions provided by the EKF (Line 37) (this case is named *Prediction*). Since the last case is valid for short-term prediction only, after a fixed time frame the 3D position of the UAV is directly set as ${}^{c}p^{t}_{ekf}$.

Algorithm 5.1 Vision-based UAV detection.

Input: Intrinsic camera parameters
I: Image at t
${}^{C}p^{t}_{ekf}$: 3D UAV position at t from EKF in camera frame
$\overline{p}^{t-1}_{ekf}$: UAV position from EKF at $t-1$ on I
SW: search window (center, size)
Th: displacement threshold between consecutive frames
$\overline{p}^{t-1}$: last UAV predicted state on I
Output: ${}^{C}p^{t}_{v}$: estimated UAV position

$\overline{p}^{t}_{ekf} \leftarrow {}^{C}p^{t}_{ekf}$ projection on I
$F(I) \leftarrow$ Compute foreground image
$\overline{F}(I) \leftarrow F(I)$ dilation
$SW \leftarrow updateCenter(SW, \overline{p}^{t}_{ekf})$
$C \leftarrow$ Extract blobs in $SW(\overline{F}(I))$
for $C_i \in C$ **do**
  $Box_i \leftarrow$ Extract AABB of C_i
end for
$Score_{max} \leftarrow 0$
for all Box_i **do**
  $Score_i \leftarrow$ Classify Box_i with SVM
  if $Score_i \geq Score_{max}$ **then**
    $BestBox \leftarrow Box_i$
    $Score_{max} \leftarrow Score_i$
  end if
end for
if $Score_{max} > 0$ **then**
  ${}^{C}p^{t}_{v} \leftarrow Backproject(center(BestBox), {}^{C}p^{t}_{ekf}.z)$
  $\overline{p}^{t} \leftarrow center(BestBox)$
  $SW \leftarrow updateSize(SW, size(BestBox))$
else
  $\widetilde{p}^{t} \leftarrow predict(\overline{p}^{t-1})$ predicted position on I
  $Area_{max} \leftarrow 0$
  for all Box_i **do**
    $Area_i \leftarrow Box_i \cap SW$
    if $Area_i \geq Area_{max} \wedge \|center(Box_i) - \widetilde{p}\|^{t} \leq Th$ **then**
      $BestBox \leftarrow Box_i$
      $Area_{max} \leftarrow Area_i$
    end if
  end for
  if $Area_{max} > 0$ **then**
    ${}^{C}p^{t}_{v} \leftarrow Backproject(center(BestBox), {}^{C}p^{t}_{ekf}.z)$
    $\overline{p}^{t} \leftarrow center(BestBox)$
    $SW \leftarrow incrementSize(SW)$
  else
    ${}^{C}p^{t}_{v} \leftarrow Backproject(\widetilde{p}^{t}, {}^{C}p^{t}_{ekf}.z)$
    $\overline{p}^{t} \leftarrow \overline{p}^{t-1} + \overline{p}^{t}_{ekf} - \overline{p}^{t-1}_{ekf}$
    $SW \leftarrow incrementSize(SW)$
  end if
end if

5.3 Visual feedback displayed in augmented reality

An augmented reality environment is generated and displayed on a computer screen to provide a real-time visual feedback to the operator during the mission. 2D and 3D graphical overlays, called *widgets*, are rendered using the OpenGL library on top of the video stream acquired by the video camera. Geometric registration of the virtual camera is performed thanks to the calibration parameters obtained as described in Section 5.2. An example of the augmented reality scene is shown in Figure 5.3.

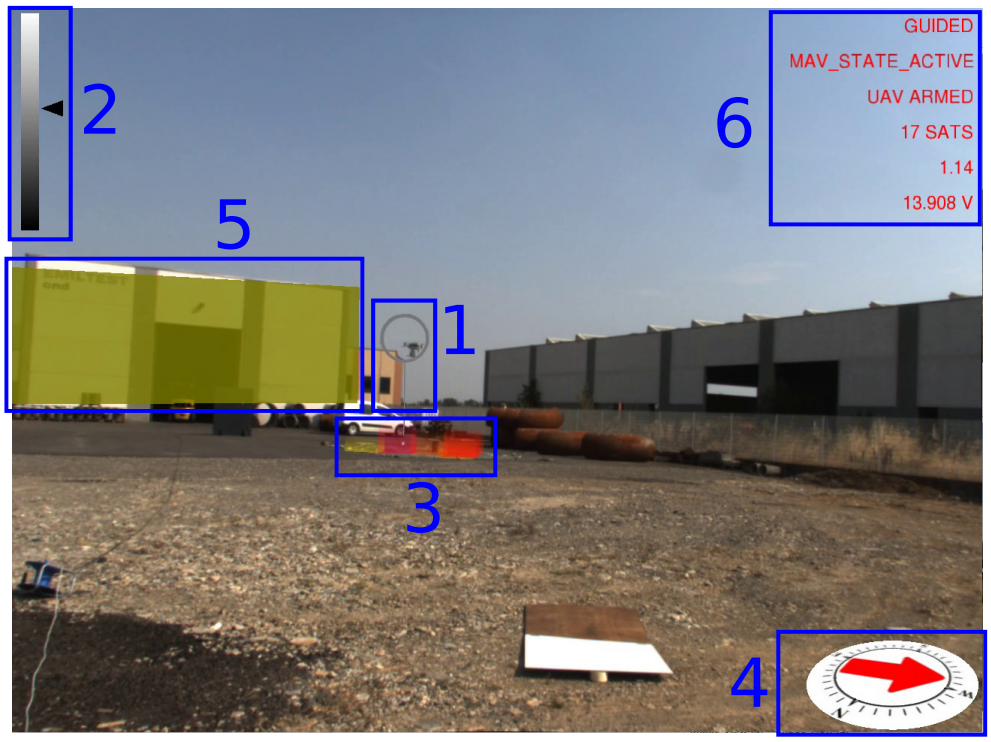


Figure 5.3: Augmented reality scene example with graphical overlays highlighted by blue boxes (for better visibility).

Widget 1 is a virtual cursor displayed as a ring around the UAV, that is computed thanks to the vision-based UAV detection algorithm. The virtual cursor tracks the UAV and, therefore, it helps the operator to quickly identify its current position. A

vertical line is also displayed to show the current projection of the UAV on the ground. The ring is displayed on a gray scale according to the distance of the UAV to the camera. This information is also displayed by a 2D cursor moving on a vertical bar (widget 2, top left) that acts as a distance-color map. The maximum distance (white color) in the experiments was set to 50 m.

Widget 3 is a colored 3D histogram that represents the distribution of the geo-referenced radiation intensity Γ on the ground. The histogram shows the bin of the maximum detected radiation intensity during the mission and the closest $k = 25$ bins to the current UAV position. The higher the bin, the higher the radiation intensity inside a cell of the grid. The highest bin has a fixed height and all other bins are scaled in real-time. The grid has a 1 m $\times$ 1 m cell size. The color of each bin also changes according to the radiation intensity value, ranging from green (low radiation intensity) to red (high radiation intensity). The current UAV heading and the magnetic north are displayed on a virtual compass (widget 4).
Geographical features in close proximity to the UAV workspace are retrieved from a query to the Open Street Map (OSM) database (`http://www.openstreetmap.org`), registered in the augmented reality scene, and highlighted on the screen (widget 5) when the UAV flies close to them. Each building is displayed by computing an oriented bounding box from a 2D polyline provided by the geographical data. Buildings are also used to generate geo-fences for the UAV, i.e. a virtual perimeter for collision avoidance. Automatic registration only provides a coarse alignment of the geographical elements due to calibration errors and imprecisions in the OSM data. Hence, registration is manually refined during the calibration phase. More advanced approaches for automatic registration of buildings could be adopted [120, 121, 90], but these methods are beyond the scope of this work. Buildings and woods are drawn using yellow and green boxes respectively, streets and rivers are displayed using red and blue cylinders respectively. Since OSM does not provide information about the height of buildings and woods, the height of the bounding boxes was fixed to a constant value.
The last graphical overlay (widget 6) provides textual information to the operator about mission critical parameters including current flight mode, flight status, hori-

zontal dilution of precision (HDOP) and number satellites currently visible.

5.4 Software architecture

The software architecture, illustrated in Figure 5.4, extends the architecture of Chapter 4 (see Figure 4.7). The system is based on the ROS framework. The *camera* node is a ROS driver that gets images from the fixed camera and publishes them including intrinsic calibration parameters. The *cv_uav_detection* node computes and publishes the estimated UAV pose using the computer vision algorithm. The *ar_scene* node displays the graphical widgets of the augmented reality scene. In particular, it receives the estimated UAV pose that is used to display the 3D virtual cursor around the UAV. The *ar_scene* node also reads the measured radiation intensity and it displays the 3D histogram on the ground. Finally, it displays information about the geographical features around the UAV. The *geo_service* node is responsible for publishing the geographical features to be displayed that are retrieved offline from Open Street Map. The OSM API is used to download geographical data close to a specified GPS position (the takeoff position) within a given radius R. More specifically, the *geo_service* node reads the current UAV position and publishes only the geographical features that are within a certain distance to the UAV. As discussed in Section 5.2, both *cv_uav_detection* and *ar_scene* nodes require the relative pose of the camera w.r.t. the UAV takeoff pose (*Home*).

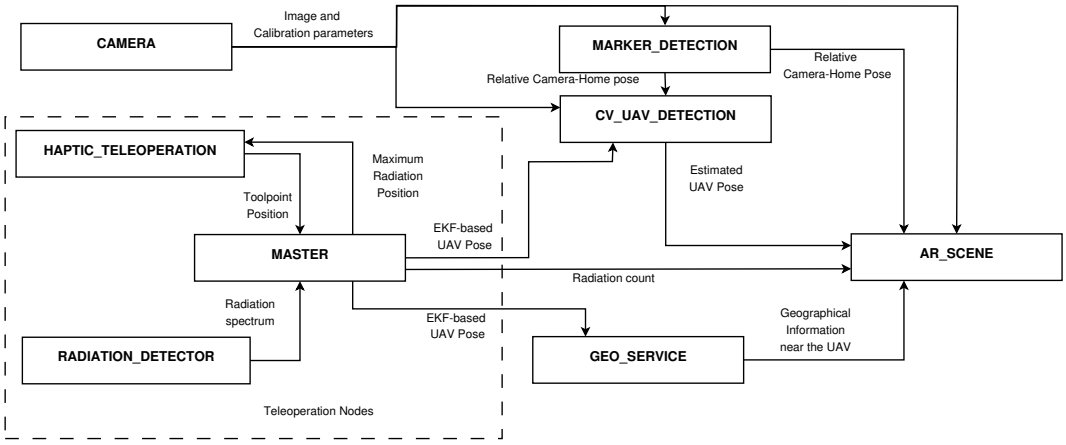


Figure 5.4: Software architecture of the VHAR system: at bottom left, in the dashed-line box, a simplified version of Figure 4.7 is shown.

5.5 Experiments

Experiments were performed in the same industrial area of Chapter 4 with the same radioactivity source (Iridium 192). Images of the augmented reality environment are reported in Figure 5.5. As in previous experiments, the human operator piloted the UAV in the exploration task using the haptic device. The fixed ground camera used in the experiments (AVT Mako G125C) has a resolution of 1292×964 and a maximum frame rate of 30 fps. The camera was placed at approximately 1.5 m height from the ground. The vision-based UAV detection algorithm takes, on average, 16.72 ms of execution time (59.8 Hz on an Intel Core i7-4720HQ). The training set (Figure 5.6) contains 4000 images (2000 negative samples and 2000 positive samples) acquired in a different environment (rural area).

The SVM classifier, implemented by the OpenCV library, has been trained using Bag-Of-Words with SURF features and a codebook of size 1000. The *train_auto* method was used, that implements a 10-fold cross-validation. The test set contains 3944 images taken from the industrial setting (Figure 5.5). The UAV detection algorithm was evaluated in a distance range between 5 to 45 meters.

Figure 5.7a shows the frequency of activation of the three UAV detection strate-

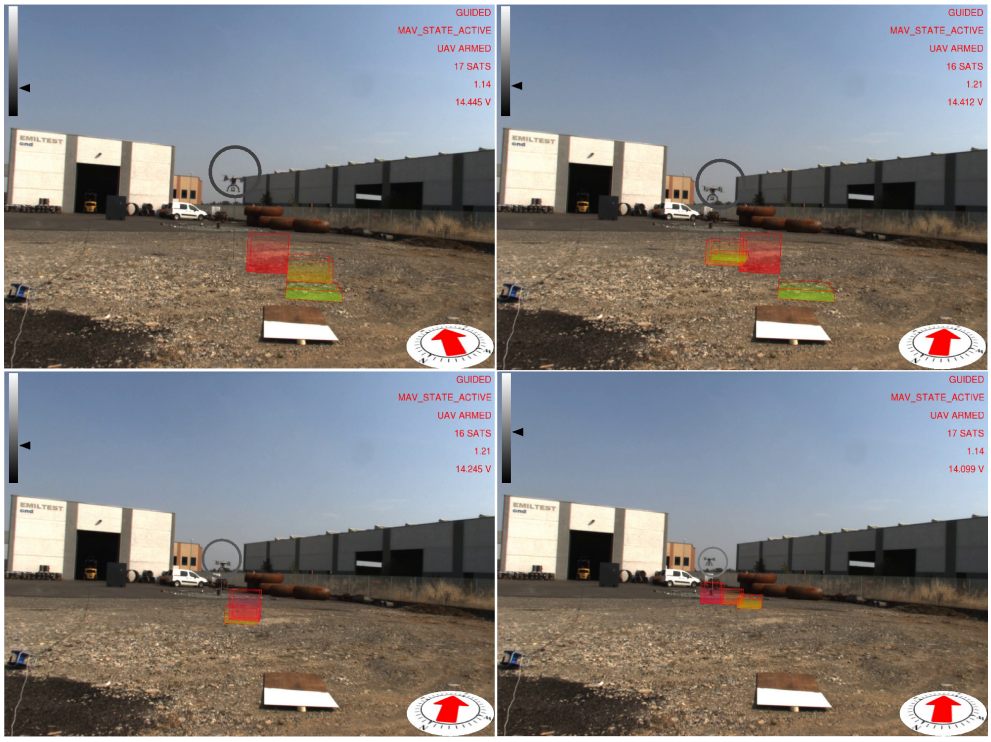


Figure 5.5: Example images of the augmented reality environment in an industrial setting (from top left to bottom right).

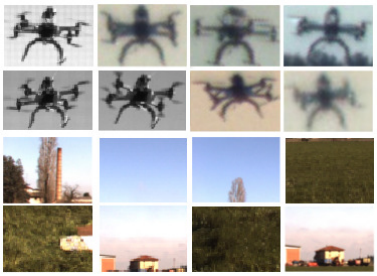


Figure 5.6: Examples of the images used in the training set for vision-based UAV detection.

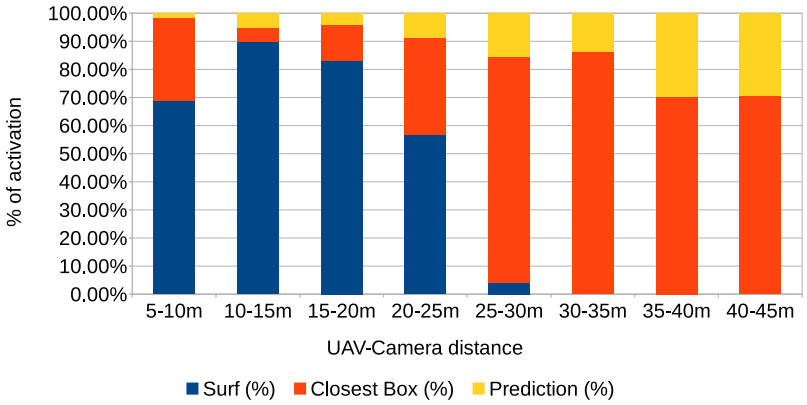
gies (*SURF*, *Closest Box* and *Prediction*) at different horizontal distances of the UAV to the camera. It can be noted that *SURF*-based estimation is mainly active when the

UAV distance to the camera is under 25 m. Beyond that distance the size of the UAV on the image reduces so that SURF features are no longer usable, thus, *Closest Box* is mainly active. Activation frequency of the *Prediction*-based estimation method also increases with the distance.

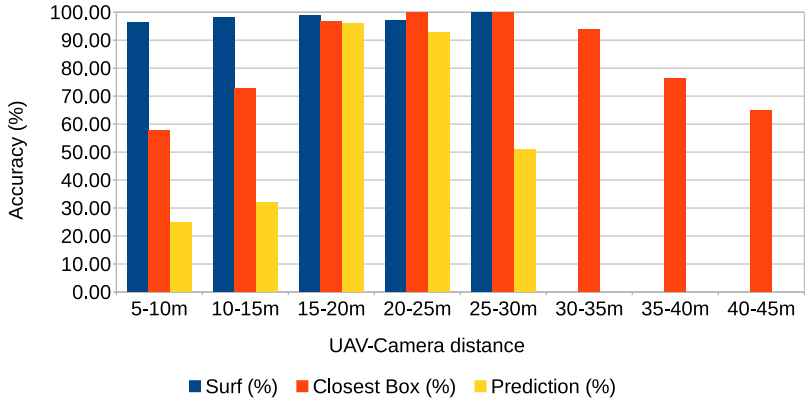
Accuracy was computed by comparing the estimated UAV location on the image with a manually annotated ground truth. A distance threshold of 15 pixels, estimated from a ROC curve, was used. The overall detection accuracy of the algorithm on the test set is about 90%. Figure 5.7b reports detailed detection results. The *SURF* method, when active, achieves an accuracy above 95%. The *Closest Box* method achieves an accuracy rate of about 60% on the short range (when it is rarely active) due to the fact that when the UAV is close to the camera the search window includes spurious blobs from the ground (e.g. moving dust or grass). *Closest Box* also achieves a 60% accuracy rate at long distances as background subtraction can fail to detect the UAV due to its small projection on the image plane. Although the accuracy rate of the *Prediction* method, which is mainly active at long distances, is very low, it still provides an estimated position of the UAV on the image plane that is useful to the operator to identify the UAV, as confirmed by the user study reported next. A comparison between the vision-based UAV detection algorithm and the result provided by the autopilot EKF is shown in Figure 5.8 and in the video available at `http://www.ce.unipr.it/~micconi/aruav.mp4`.

3D registration of buildings required a manual correction. Indeed, the coarse registration of the geographical elements based on the OSM data has an average position error of 5 m and an average orientation error of 5°. An example of building registration is shown in Figure 5.9. It can be expected that open GIS services will gain more accuracy in the future, reducing the alignment errors. Furthermore, using automatic approaches, as suggested in [120, 121, 90], the need for a manual registration caused by other errors, like camera pose estimation error, will drop.

The usability of the VHAR interface was evaluated in a simulated environment based on the same software architecture described in Section 5.4. The evaluation aims to determine whether the complete VHAR interface causes a lower mental workload and a higher SA level with respect to the HAR interface (Section 4.4.2). As for the



(a) Frequency of activation.



(b) Accuracy.

Figure 5.7: Vision-based UAV detection performance.

previous tests, the task the users were asked to perform was a radiation source detection. In this case the tests were completely simulated. The augmented reality environment (Figure 5.10) displayed a video stream of the real workspace, and it included a simulated 3D model of the UAV (animated using the ArduCopter SITL simulator) as well as a simulated radioactivity source. A total of 10 non-expert participants were recruited for the evaluation. Each participant, after a training session, performed two tests in random order. In one case the user performed the radiation source detection

Figure 5.8: Comparison of UAV position estimation methods: vision-based (left), EKF (on-board sensors) (right).

Figure 5.9: Building visualization in augmented reality: result from coarse registration (left) and after manual correction (right).

task using the VHAR interface. In the other case graphical augmentation on the video stream was disabled and the task was performed using the previous HAR interface, with additional information displayed on a side screen. In particular, the side screen displayed the radiation spectrum and a 2D plot of the UAV trajectory with a mark indicating the current location of the maximum radiation found during the mission.

Evaluation of the mental workload was carried out from the NASA-TLX (Task Load indeX) questionnaire [122] while the evaluation of the SA was based on the Situation Present Assessment Method (SPAM) [123].
Four categories of NASA-TLX have been considered, for the mental workload estimation, in form of questions:

Figure 5.10: Simulated environment developed for the usability tests. The UAV is a 3D model simulated using the ArduCopter SITL simulator.

Mental demand How mentally demanding was the task?

Performance How successful were you in accomplishing what you were asked to do?

Effort How hard did you have to work to accomplish your level of performance?

Frustration How insecure, discouraged, irritated, stressed, and annoyed were you?;

Two question related to physical effort and temporal demand have been considered not useful to the evaluation and discarded. Each question required an answer in form of rating within a 100-points range with 5-point steps. High values mean high mental demand, unsuccessful performance, great effort and high frustration or stress. The users answered the questionnaire after the conclusion of the mission. The same evaluation has been performed for both scenarios. Figure 5.11 shows the results in a

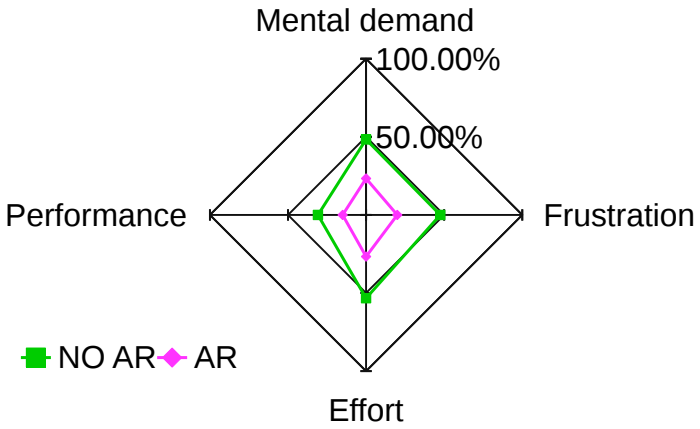


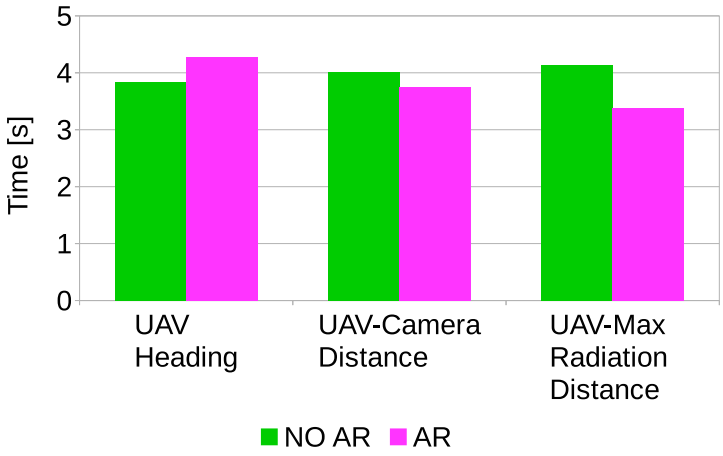
Figure 5.11: NASA-TLX questionnaire results. The AR line indicates average results of the VHAR interface; the NO AR line indicates average results without visual-feedback.

star-like diagram. The VHAR interface received better scores (lower rates) for each parameter, i.e. it caused a lower mental workload level. One-way ANOVA analysis showed that the results were statistically significant for all parameters ($p \leq 0.05$).

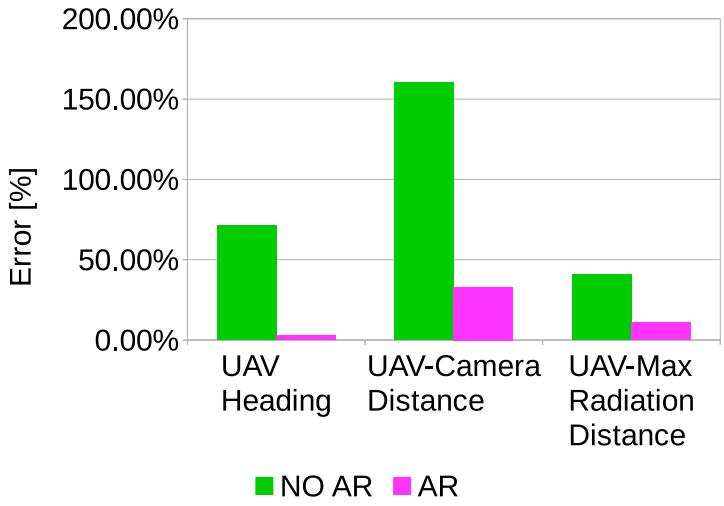
Participant were asked questions during the execution of the task according to the SPAM method to estimate the SA level. Questions were related to the current state of the task. In particular, three questions were asked:

- What is the current heading of the UAV with respect to the magnetic north?
- How far is the UAV from the current location of maximum radiation?
- How far is the UAV from the camera?

The nature of the SPAM questionnaire aimed to highlight the SA level of the user, focusing on the spatial relation between the surrounding environment and the UAV. Participants explicitly stated that the visual feedback of the VHAR interface was very useful to detect the radioactivity source and to determine distances. Indeed, statistically significant results (Figure 5.12) were found for all the parameters estimated by the users ($p \leq 0.05$).



(a) Response time.



(b) Parameter estimation error.

Figure 5.12: SPAM questionnaire results. The AR line indicates average results of the VHAR interface; the NO AR line indicates average results without visual-feedback.

However, the results of the questionnaire indicated no significant differences in response time. Hence, it can be concluded that the VHAR interface is perceived as

intuitive and that it improves the situation awareness of the task rather than its execution time.

5.6 Discussion

In this chapter we have presented the complete VHAR interface, partially introduced in Chapter 4. In particular, the Visual component of the interface has been described. The Visual component, alongside the Haptic component, is able to increase the SA level of the human operator during the UAV teleoperation.
In the previous chapter we showed how the HAR interface enables the operator to perform the exploration task without the need to look at a screen and, consequently, without loosing focus on the UAV. The complete VHAR interface enables the operator to focus on the mission and the UAV, while he/she is keeping under control other important information related to the mission, like flight parameters.
A vision-based algorithm for detection of the UAV in an image flow has been developed and evaluated. The algorithm helps the operator to track the UAV in the image and allows better registration, in the Visual AR scene, of the 3D models representing the radioactivity intensity on the ground. The AR scene also contains other 2D and 3D widgets that represent important information about the mission and the environment.
The VHAR interface has been evaluated to assess its usability and the associated mental workload. Tests with non-expert users show that the interface is intuitive and helps the operator to reach a higher level of SA compared with the HAR interface. Furthermore, the mission is accomplished with less mental overload and frustration than the HAR interface.
In summary, results in Chapter 4 and this chapter confirm that proper inclusion of Visual and Haptic feedbacks into a coherent AR environment can provide tangible benefits even in critical outdoor guidance tasks. Furthermore, this is, at our knowledge, the first complete outdoor Visuo-Haptic AR interface. Several research investigations have addressed outdoor Visual AR [72, 73, 74, 75, 77, 80, 81], but only a few involving UAV, even less using a ground camera and none of these works has considered

haptic teleoperation. The main limitation of the VHAR interface described in this dissertation is that the use of a single fixed camera reduces the available field of view of the environment. Moreover, additional work is needed to achieve a fully automatic registration of geographical data that are affected by large estimation errors.

Chapter 6

Conclusions

This thesis has presented an investigation on the potential of Visuo-Haptic Augmented Reality technologies, especially for robotic applications. Alongside the use of Visual AR for commercial software, in particular for the mobile application field, several research studies have examined the application of such technologies in different scenarios and the augmentation of other senses. Beside the mere ability to overlay information on a view of a real environment or scenario, AR has been proven useful and intuitive to augment the entire user experience when interacting with several kinds of systems. For example, VHAR, a particular category of AR that augments sight and touch, has been used widely in studies about medical training. The analysis of use of the VHAR technologies for Human-Machine and/or Human-Robot interaction has been the purpose of this thesis.

To perform a complete assessment of VHAR interfaces, this investigation has focused on two different scenarios, highlighting the common advantages of such interface, in both use cases. The investigation has led to the development of two VHAR interfaces that implement unique features with respect to the state of art. The developed interfaces address key issues of AR, like calibration and mutual registration of the components of the interfaces and the environment. The study has shown that in both scenarios the use of VHAR enables a human operator, interacting with a robotic application, to perform tasks in a more natural and intuitive way, decreasing the frus-

tration incurred and the mental workload.
In the first scenario, the VHAR interface has been used for human-robot interaction in an indoor environment for intuitive programming of an industrial robotic arm. The normal programming process requires a skilled operator. Moreover, each time the robotic arm must execute a new task, the operator has to manually change the instruction sequence, a process that can be tedious. Hence, the use of an intuitive VHAR interface allows a non-expert operator to instruct the robotic system avoiding the need to explicitly code the sequence of instructions. To achieve a more realistic user interaction with the virtual elements inside the automatically generated AR scene, a physics simulation has been included in the interface. Usability evaluation with non-expert operators has shown that the haptic feedback, generated when the object controlled by the user collide with others, increase the realism of the interaction and the intuitiveness of the interface. Finally, a Programming-By-Demonstration algorithm has been implemented to allow the system to acquire knowledge about the the task, based on user demonstrations. Using this knowledge the system is able to control a real manipulator to perform the task with the real objects. Successful tests have been performed to prove the ability of the system to learn the task and to reproduce the task in the real workspace.
A VHAR interface has also been developed and tested in an outdoor scenario. In this application case, the VHAR interface has been used for the teleoperation of a UAV equipped with an X-ray detector. The interface allows the operator to avoid safety hazards during the exploration of possibly contaminated areas. Indeed, the UAV, carrying the radioactivity detector, is teleoperated by the operator from a safety distance. Furthermore, during the mission the operator must be focused on the exploration task but also be aware of the UAV flight and the environment surrounding the UAV. While the Haptic AR component of the interface helps the user to explore the area to find any radioactive materials, the Visual AR component of the interface increases the level of Situation Awareness and decreases frustration and metal workload, as proven by the usability evaluation. Indeed, the operator looks at a screen and, thanks to augmented image flow, stays focused on the UAV while he/she remains aware of the state of the flight, of the data acquired by the sensors, and of the environment.

Even though the two application studies concern quite different use cases, Visuo-Haptic Augmented Reality has proven a promising technology for Human-Machine and Human-Robot interaction. In both cases, the operator interacts more naturally and intuitively with actuators in the field (the robot arm or the UAV) and with the workspace (the objects or the radioactivity source). Moreover, the frustration and the level of skills required are lower than using traditional technologies and interfaces. Once some of the technical problems highlighted by this work (like object recognition and registration, mutual calibration of each component of the system) have been fully engineered, the Visuo-Haptic Augmented Reality will become a reliable choice for many kinds of interaction with computers and robots.

Future research on VHAR interfaces will focus on more advanced solutions for AR interaction with head mounted displays, to enable viewpoint rotation, and 6 DOF haptic rendering for improved immersiveness.

Bibliography

[1] F. P. Brooks, Jr. *The Computer Scientist As Toolsmith II*. Communications of the ACM, 39(3):61–68, March 1996. URL: `http://doi.acm.org/10.1145/227234.227243`.

[2] R. Azuma. *A Survey of Augmented Reality*. In Presence: Teleoperators and Virtual Environments 6, pages 355–385, August 1997.

[3] R. J. Adams and B. Hannaford. *Stable haptic interaction with virtual environments*. IEEE Transactions on Robotics and Automation, 15(3):465–474, June 1999.

[4] D. Prattichizzo, C. Pacchierotti, and G. Rosati. *Cutaneous Force Feedback as a Sensory Subtraction Technique in Haptics*. IEEE Transactions on Haptics, 5(4):289–300, October 2012.

[5] B. Hannaford and J.-H. Ryu. *Time-domain passivity control of haptic interfaces*. IEEE Transactions on Robotics and Automation, 18(1):1–10, February 2002.

[6] J. Rosen, B. Hannaford, M. P. MacFarlane, and M. N. Sinanan. *Force controlled and teleoperated endoscopic grasper for minimally invasive surgery-experimental performance evaluation*. IEEE Transactions on Biomedical Engineering, 46(10):1212–1221, October 1999.

[7] F. Danieau, A. Lecuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie. *Enhancing Audiovisual Experience with Haptic Feedback: A Survey on HAV*. IEEE Transactions on Haptics, 6(2):193–205, April 2013.

[8] A. Talvas, M. Marchal, and A. Lécuyer. *A Survey on Bimanual Haptic Interaction*. IEEE Transactions on Haptics, 7(3):285–300, July 2014.

[9] T. R. Coles, D. Meglan, and N. W. John. *The Role of Haptics in Medical Training Simulators: A Survey of the State of the Art*. IEEE Transactions on Haptics, 4(1):51–66, January 2011.

[10] M. Khan, S. Sulaiman, A. M. Said, and M. Tahir. *Usability studies in haptic systems*. In International Conference on Information and Communication Technologies (ICICT), pages 1–5, July 2011.

[11] N. Enayati, E. De Momi, and G. Ferrigno. *Haptics in Robot-Assisted Surgery: Challenges and Benefits*. IEEE Reviews in Biomedical Engineering, PP(99):1–1, 2016.

[12] S. J. Lederman and L. A. Jones. *Tactile and Haptic Illusions*. IEEE Transactions on Haptics, 4(4):273–294, July 2011.

[13] K. Salisbury, F. Conti, and F. Barbagli. *Haptic rendering: introductory concepts*. IEEE Computer Graphics and Applications, 24(2):24–32, March 2004.

[14] R. W. Lindeman, H. Noma, and P. G. de Barros. *Hear-Through and Mic-Through Augmented Reality: Using Bone Conduction to Display Spatialized Audio*. In IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pages 173–176, November 2007.

[15] R. W. Lindeman, H. Noma, and P. G. de Barros. *An Empirical Study of Hear-Through Augmented Reality: Using Bone Conduction to Deliver Spatialized Audio*. In IEEE Virtual Reality Conference, pages 35–42, March 2008.

[16] F. Ribeiro, D. Florencio, P. A. Chou, and Z. Zhang. *Auditory augmented reality: Object sonification for the visually impaired*. In IEEE International Workshop on Multimedia Signal Processing (MMSP), pages 319–324, September 2012.

[17] K. Kondo, N. Anazawa, and Y. Kobayashi. *Characteristics comparison of two audio output devices for augmented audio reality*. In Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pages 1–6, October 2013.

[18] S. Wei, G. Ren, and E. O'Neill. *Haptic and audio displays for augmented reality tourism applications*. In IEEE Haptics Symposium (HAPTICS), pages 485–488, February 2014.

[19] USAF. *United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047*, May 2009.

[20] USAF. *Small Unmanned Aircraft System (SUAS) Flight Plan: 2016-2036*, April 2016.

[21] J. J. Ruiz, A. Viguria, J. R. Martinez-de Dios, and A. Ollero. *Immersive displays for building spatial knowledge in multi-UAV operations*. In International Conference on Unmanned Aircraft Systems (ICUAS), pages 1043–1048, June 2015.

[22] J. J. Ruiz, M. A. Escalera, A. Viguria, and A. Ollero. *A simulation framework to validate the use of head-mounted displays and tablets for information exchange with the UAV safety pilot*. In Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), pages 336–341, November 2015.

[23] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. *Recent advances in augmented reality*. IEEE Computer Graphics and Applications, 21(6):34–47, November 2001.

[24] F. Zhou, H. B. L. Duh, and M. Billinghurst. *Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR*. In IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), pages 193–202, September 2008.

[25] M. E. C. Santos, A. Chen, T. Taketomi, G. Yamamoto, J. Miyazaki, and H. Kato. *Augmented Reality Learning Experiences: Survey of Prototype Design and Evaluation*. IEEE Transactions on Learning Technologies, 7(1):38–56, January 2014.

[26] S. Kasahara, R. Niiyama, V. Heun, and H. Ishii. *exTouch: Spatially-aware Embodied Manipulation of Actuated Objects Mediated by Augmented Reality*. In International Conference on Tangible, Embedded and Embodied Interaction, TEI '13, pages 223–228, February 2013. ACM. URL: `http://doi.acm.org/10.1145/2460625.2460661`.

[27] K. Kruckel, F. Nolden, A. Ferrein, and I. Scholl. *Intuitive visual teleoperation for UGVs using free-look augmented reality displays*. In IEEE International Conference on Robotics and Automation (ICRA), pages 4412–4417, May 2015.

[28] T. H. J. Collett and B. A. MacDonald. *Augmented Reality Visualisation for Player*. In IEEE International Conference on Robotics and Automation, (ICRA), May 2006.

[29] S. Hashimoto, A. Ishida, M. Inami, and T. Igarashi. *TouchMe: An augmented reality interface for remote robot control*. Journal of Robotics and Mechatronics, 25(3):529–537, April 2013.

[30] H. C. Fang, S. K. Ong, and A. Y. C. Nee. *Interactive robot trajectory planning and simulation using Augmented Reality*. Robotics and Computer-Integrated Manufacturing, 28(2):227–237, 2012.

[31] J. W. S. Chong, S. K. Ong, A. Y. C. Nee, and K. Youcef-Youmi. *Robot programming using augmented reality: An interactive method for plan-*

ning collision-free paths. Robotics and Computer-Integrated Manufacturing, 25(3):689–701, June 2009.

[32] H. C. Fang, S. K. Ong, and A. Y.-C. Nee. *Robot Programming Using Augmented Reality*. In International Conference on CyberWorlds, pages 13–20, 2009.

[33] A. Pichler and M. Ankerl. *User centered framework for intuitive robot programming*. In IEEE International Workshop on Robotic and Sensors Environments (ROSE), pages 1–6, 2010.

[34] N. Pathomaree and S. Charoenseang. *Augmented Reality for Skill Transfer in Assembly Task*. In IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), August 2005.

[35] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. *Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework*. IEEE Transactions on Visualization and Computer Graphics, 12(4):15–628, July 2006.

[36] C. Chae and K. Ko. *Introduction of Physics Simulation in Augmented Reality*. In International Symposium on Ubiquitous Virtual Reality, (ISUVR), July 2008.

[37] B. Lok, S. Naik, M. Whitton, and F. P. Brooks. *Incorporating Dynamic Real Objects Into Immersive Virtual Environments*. ACM Transactions on Graphics, 22(3):701–701, 2003.

[38] D. S. M. Liu, C. H. Yung, and C. H. Chung. *A Physics-Based Augmented Reality Jenga Stacking Game*. In Workshop on Digital Media and Digital Content Management (DMDCM), pages 1–8, 2011.

[39] J. Pilet, V. Lepetit, and P. Fua. *Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation*. International Journal of Computer Vision, 76(2):109–122, 2007.

[40] S. A. Nicolau, L. Goffin, and L. Soler. *A low cost and accurate guidance system for laparoscopic surgery: validation on an abdominal phantom*. In ACM Symposium on Virtual Reality Software and Technology, 2005.

[41] R. Nunez, J. R. Bandera, J. M. Perez-Lorenzo, and F. Sandoval. *A Human-Robot Interaction System for Navigation Supervision Based on Augmented Reality*. In IEEE Mediterranean Electrotechnical Conference (MELECON), May 2006.

[42] D. Wang, Y. Zhang, W. Zhou, H. Zhao, and Z. Chen. *Collocation Accuracy of Visuo-Haptic System: Metrics and Calibration*. IEEE Transactions on Haptics, 4(4):321–326, 2011.

[43] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. *Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality*. IEEE Transactions on Visualization and Computer Graphics, PP(99):1–1, 2016.

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

[45] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. van Paassen. *Artificial Force Field for Haptic Feedback in UAV Teleoperation*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 39(6):1316–1330, November 2009.

[46] S. Stramigioli, R. Mahony, and P. Corke. *A novel approach to haptic teleoperation of aerial robot vehicles*. In IEEE International Conference on Robotics and Automation (ICRA), pages 5302–5308, May 2010.

[47] A. Y. Mersha, S. Stramigioli, and R. Carloni. *On Bilateral Teleoperation of Aerial Robots*. IEEE Transactions on Robotics, 30(1):258–274, February 2014.

[48] X. Hou, R. Mahony, and F. Schill. *Representation of vehicle dynamics in haptic teleoperation of aerial robots*. In IEEE International Conference on Robotics and Automation (ICRA), pages 1485–1491, May 2013.

[49] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel. *Bilateral haptic teleoperation of VTOL UAVs*. In IEEE International Conference on Robotics and Automation (ICRA), pages 2393–2399, May 2013.

[50] R. Carloni, V. Lippiello, M. D'Auria, M. Fumagalli, A. Y. Mersha, S. Stramigioli, and B. Siciliano. *Robot Vision: Obstacle-Avoidance Techniques for Unmanned Aerial Vehicles*. IEEE Robotics Automation Magazine, 20(4):22–31, December 2013.

[51] A. Y. Mersha, X. Hou, R. Mahony, S. Stramigioli, P. Corke, and R. Carloni. *Intercontinental haptic teleoperation of a flying vehicle: A step towards real-time applications*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4951–4957, November 2013.

[52] P. Stegagno, M. Basile, H. H. Bulthoff, and A. Franchi. *A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback*. In IEEE International Conference on Robotics and Automation (ICRA), pages 3862–3869, May 2014.

[53] C. Masone, P. R. Giordano, H. H. Bulthoff, and A. Franchi. *Semi-autonomous trajectory generation for mobile robots with integral haptic shared control*. In IEEE International Conference on Robotics and Automation (ICRA), pages 6468–6475, May 2014.

[54] H. I. Son, J. Kim, L. Chuang, A. Franchi, P. R. Giordano, D. Lee, and H. H. Bulthoff. *An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments*. In IEEE World Haptics Conference (WHC), pages 149–154, June 2011.

[55] G. Bianchi, B. Knoerlein, G. Szekely, and M. Harders. *High Precision Augmented Reality Haptics*. In Proceedings of EuroHaptics, 2006.

[56] M. Harders, G. Bianchi, B. Knoerlein, and G. Szekely. *Calibration, Registration, and Synchronization for High Precision Augmented Reality Haptics*. IEEE Transactions on Visualization and Computer Graphics, 15(1):138–149, 2009.

[57] S. Jeon and M. Harders. *Extending haptic augmented reality: Modulating stiffness during two-point squeezing*. In IEEE Haptics Symposium, pages 141–146, 2012.

[58] F. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. A. Otaduy. *Visuo-Haptic Mixed Reality with Unobstructed Tool-Hand Integration*. IEEE Transactions on Visualization and Computer Graphics, 19(1):159–172, 2013.

[59] T. R. Coles, N. John, D. A. Gould, and D. G. Caldwell. *Integrating Haptics with Augmented Reality in a Femoral Palpation and Needle Insertion Training Simulation*. IEEE Transactions on Haptics, 4(3):199–209, 2011.

[60] C. Sutherland, K. Hashtrudi-Zaad, R. Sellens, P. Abolmaesumi, and P. Mousavi. *An Augmented Reality Haptic Training Simulator for Spinal Needle Procedures*. IEEE Transactions on Biomedical Engineering, 60(11):3009–3018, 2013.

[61] Y. Kuroda and O. Oshiro. *Visual and haptic augmentation technologies for surgical navigation*. In International Symposium on Micro-NanoMechatronics and Human Science (MHS), pages 71–76, 2010.

[62] T. Akinbiyi, C. E. Reiley, S. Saha, D. Burschka, C. J. Hasser, D. D. Yuh, and A. M. Okamura. *Dynamic Augmented Reality for Sensory Substitution in Robot-Assisted Surgical Systems*. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (EMBS), November 2006.

[63] T. Hulin, V. Schmirgel, E. Yechiam, U.E. Zimmermann, C. Preusche, and G. Pöhler. *Evaluating exemplary training accelerators for Programming-by-*

Demonstration. In IEEE International Symposium on Robot and Human Interactive Communication (ROMAN), pages 440–445, 2010.

[64] K. Murakami, R. Kiyama, T. Narumi, T. Tanikawa, and M. Hirose. *Poster: A wearable augmented reality system with haptic feedback and its performance in virtual assembly tasks*. In IEEE Symposium on 3D User Interfaces (3DUI), pages 161–162, 2013.

[65] G. Ye, J. J. Corso, G. D. Hager, and A. M. Okamura. *VisHap: Augmented Reality Combining Haptics and Vision*. In IEEE Intl Conference on Systems, Man and Cybernetics, volume 4, pages 3425–3431 vol.4, October 2003.

[66] J. Vallino and C. Brown. *Haptics in augmented reality*. In IEEE International Conference on Multimedia Computing and Systems, July 1999.

[67] J. Kim, F. Janabi-Sharifi, and J. Kim. *A Physically-Based Haptic Rendering for Telemanipulation with Visual Information: Macro and Micro Applications*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), September 2008.

[68] C. Sandor, S. Uchiyama, and H. Yamamoto. *Visuo-Haptic Systems: Half-Mirrors Considered Harmful*. In EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC), pages 292–297, 2007.

[69] B. Bayart, J. Y. Didier, and A. Kheddar. *Force Feedback Virtual Painting on Real Objects: A Paradigm of Augmented Reality Haptics*. In EuroHaptics, pages 776–785, 2008.

[70] S. Scheggi, G. Salvietti, and D. Prattichizzo. *Shape and weight rendering for haptic Augmented Reality*. In IEEE International Symposium on Robot and Human Interactive Communication (ROMAN), pages 44–49, 2010.

[71] A. Kron, G. Schmidt, B. Petzold, M.I. Zah, P. Hinterseer, and E. Steinbach. *Disposal of Explosive Ordnances by Use of a Bimanual Haptic Telepres-*

ence System. In IEEE International Conference on Robotics and Automation, (ICRA), May 2004.

[72] G. Schall, D. Schmalstieg, and S. Junghanns. *VIDENTE-3D visualization of underground infrastructure using handheld augmented reality*. GeoHydroinformatics: Integrating GIS and Water Engineering, 2010.

[73] S. Zollmann, G. Schall, S. Junghanns, and G. Reitmayr. *Comprehensible and Interactive Visualizations of GIS Data in Augmented Reality*. Advances in Visual Computing, pages 675–685, July 2012. URL: `http://dx.doi.org/10.1007/978-3-642-33179-4_64`.

[74] S. Zollmann, D. Kalkofen, C. Hoppe, S. Kluckner, H. Bischof, and G. Reitmayr. *Interactive 4D overview and detail visualization in augmented reality*. In IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pages 167–176, November 2012.

[75] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. *Augmented Reality for Construction Site Monitoring and Documentation*. Proceedings of the IEEE, 102(2):137–154, February 2014.

[76] M. Sun, N. Dong, C. Jiang, X. Ren, and L. Liu. *Real-Time MUAV Video Augmentation with Geo-information for Remote Monitoring*. In International Conference on Geo-Information Technologies for Natural Disaster Management (GiT4NDM), pages 114–118, October 2013.

[77] F. Okura, M. Kanbara, and N. Yokoya. *Augmented telepresence using autopilot airship and omni-directional camera*. In IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pages 259–260, October 2010.

[78] P. Iwaneczko, K. Jędrasiak, and A. Nawrat. *Augmented Reality in UAVs Applications*. In Aleksander Nawrat and Karol Jędrasiak, editors, Innovative Simulation Systems, pages 77–86, 2016.

[79] Z. Ai, M. A. Livingston, and I. S. Moskowitz. *Real-time unmanned aerial vehicle 3D environment exploration in a mixed reality environment*. In International Conference on Unmanned Aircraft Systems (ICUAS), pages 664–670, June 2016.

[80] E. Veas, R. Grasset, E. Kruijff, and D. Schmalstieg. *Extended Overview Techniques for Outdoor Augmented Reality*. IEEE Transactions on Visualization and Computer Graphics, 18(4):565–572, April 2012.

[81] S. Zollmann, C. Hoppe, T. Langlotz, and G. Reitmayr. *FlyAR: Augmented Reality Supported Micro Aerial Vehicle Navigation*. IEEE Transactions on Visualization and Computer Graphics, 20(4):560–568, April 2014.

[82] A. Hosseini and M. Lienkamp. *Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality*. In 2016 IEEE Intelligent Vehicles Symposium (IV), pages 1366–1373, June 2016.

[83] P. M. d'Orey, A. Hosseini, J. Azevedo, F. Diermeyer, M. Ferreira, and M. Lienkamp. *Hail-a-Drone: Enabling teleoperated taxi fleets*. In 2016 IEEE Intelligent Vehicles Symposium (IV), pages 774–781, June 2016.

[84] A. Thangarajah, J. Wu, B. Madon, and A. K. Chowdhury. *Vision-based registration for augmented reality-a short survey*. In IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pages 463–468, October 2015.

[85] E. Marchand, H. Uchiyama, and F. Spindler. *Pose estimation for augmented reality: a hands-on survey*. IEEE Transactions on Visualization and Computer Graphics, PP(99):1–1, 2016.

[86] Z. Zhang. *A Flexible New Technique for Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

[87] J.-Y. Bouguet. *Camera Calibration Toolbox*.

[88] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. *Automatic generation and detection of highly reliable fiducial markers under occlusion*. Pattern Recognition, 47(6):2280–2292, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0031320314000235.

[89] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer. *Generation of fiducial marker dictionaries using mixed integer linear programming*. Pattern Recognition, 51:481–491, 2016. URL: http://www.sciencedirect.com/science/article/pii/S0031320315003544.

[90] M. Sun, L. Liu, W. Huang, and X. Ren. *Interactive registration for Augmented Reality GIS*. In International Conference on Computer Vision in Remote Sensing (CVRS), pages 246–251, December 2012.

[91] J. Aleotti, G. Micconi, and S. Caselli. *Programming Manipulation Tasks by Demonstration in Visuo-Haptic Augmented Reality*. In IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE), pages 13–18, October 2014.

[92] J. Aleotti, G. Micconi, and S. Caselli. *Object interaction and task programming by demonstration in visuo-haptic augmented reality*. Multimedia Systems, 22(6):675–691, 2015. URL: http://dx.doi.org/10.1007/s00530-015-0488-z.

[93] F. Denaro. *Un sistema aptico in realtà aumentata con simulazione fisica*. Master's thesis, Università degli Studi di Parma, 2009.

[94] J. Aleotti, F. Denaro, and S. Caselli. *Object manipulation in visuo-haptic augmented reality with physics-based animation*. In 19th International Symposium in Robot and Human Interactive Communication, pages 38–43, Sept 2010.

[95] M. Benassi. *Realizzazione e valutazione di ambienti di realtà aumentata per interazione con oggetti virtuali*. Bachelor's thesis, Università degli Studi di Parma, 2011.

[96] G. Micconi. *Un sistema aptico di realtà aumentata per manipolazione robotica*. Bachelor's thesis, Università degli Studi di Parma, 2011.

[97] J. Aleotti, D. Lodi Rizzini, and S. Caselli. *Object Categorization and Grasping by Parts from Range Scan Data*. In IEEE International Conference on Robotics and Automation (ICRA), 2012.

[98] R. B. Rusu, N. Blodow, and M. Beetz. *Fast Point Feature Histograms (FPFH) for 3D Registration*. In IEEE International Conference on Robotics and Automation (ICRA), pages 3212–3217, May 2009.

[99] R. Ugolotti, G. Micconi, J. Aleotti, and S. Cagnoni. *GPU-based point cloud recognition using evolutionary algorithms*. In European Conference on the Applications of Evolutionary Computation, pages 489–500, 2014. Springer Berlin Heidelberg.

[100] J. Kennedy and R. Eberhart. *Particle swarm optimization*. In IEEE International Conference on Neural Networks, volume 4, pages 1942–1948 vol.4, Nov 1995.

[101] R. Ugolotti, Y. S. G. Nashed, P. Mesejo, Š. Ivekovič, L. Mussi, and S. Cagnoni. *Particle Swarm Optimization and Differential Evolution for model-based object detection*. Applied Soft Computing, 13(6):3092–3105, 2013. URL: `http://www.sciencedirect.com/science/article/pii/S156849461200511X`.

[102] Y. S. G. Nashed, R. Ugolotti, P. Mesejo, and S. Cagnoni. *libCudaOptimize: an open source library of GPU-based metaheuristics*. In 14th Annual Conference Companion on Genetic and Evolutionary Computation, pages 117–124, 2012. ACM.

[103] A. Lecuyer, A. Kheddar, S. Coquillart, L. Graux, and P. Coiffet. *A haptic prototype for the simulations of aeronautics mounting/unmounting operations*. In IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), pages 182–187, 2001.

[104] R. Zöllner, M. Pardowitz, S. Knoop, and R. Dillmann. *Towards Cognitive Robots: Building Hierarchical Task Representations of Manipulations from Human Demonstration*. In IEEE International Conference on Robotics and Automation (ICRA), pages 1535–1540, 2005.

[105] J. Aleotti and S. Caselli. *Part-Based Robot Grasp Planning from Human Demonstration*. In IEEE International Conference on Robotics and Automation (ICRA), pages 4554–4560, May 2011.

[106] R. A. Cortez, X. Papageorgiou, H. G. Tanner, A. V. Klimenko, K. N. Borozdin, R. Lumia, and W. C. Priedhorsky. *Smart radiation sensor management*. IEEE Robotics Automation Magazine, 15(3):85–93, September 2008.

[107] G. Micconi, J. Aleotti, S. Caselli, G. Benassi, N. Zambelli, and A. Zappettini. *Haptic guided UAV for detection of radiation sources in outdoor environments*. In 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), pages 265–271, 2015.

[108] J. Aleotti, G. Micconi, S. Caselli, G. Benassi, N. Zambelli, D. Calestani, M. Zanichelli, M. Bettelli, and A. Zappettini. *Unmanned aerial vehicle equipped with spectroscopic CdZnTe detector for detection and identification of radiological and nuclear material*. In IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), pages 1–5, 2015.

[109] G. Micconi, J. Aleotti, and S. Caselli. *Evaluation of a haptic interface for UAV teleoperation in detection of radiation sources*. In IEEE Mediterranean Electrotechnical Conference (MELECON), pages 1–6, 2016.

[110] J. W. MacFarlane, O. D. Payton, A. C. Keatley, G. P. T. Scott, H. Pullin, R. A. Crane, M. Smilion, I. Popescu, V. Curlea, and T. B. Scott. *Lightweight aerial*

vehicles for monitoring assessment and mapping of radiation anomalies. Journal of Environmental Radioactivity, 136(0):127–130, 2014.

[111] K. Boudergui, F. Carrel, T. Domenech, N. Guenard, J.-P. Poli, A. Ravet, V. Schoepff, and R. Woo. *Development of a drone equipped with optimized sensors for nuclear and radiological risk characterization*. In International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA), pages 1–9, June 2011.

[112] K. Kurvinen, P. Smolander, R. Pöllänen, S. Kuukankorpi, M. Kettunen, and J. Lyytinen. *Design of a radiation surveillance unit for an unmanned aerial vehicle*. Journal of Environmental Radioactivity, 81(1):1–10, 2005.

[113] R. Pöllänen, H. Toivonen, K. Peräjärvi, T. Karhunen, T. Ilander, J. Lehtinen, K. Rintala, T. Katajainen, J. Niemelä, and M. Juusela. *Radiation surveillance using an unmanned aerial vehicle*. Applied Radiation and Isotopes, 67(2):340–344, 2009.

[114] P. Neumann, M. Bartholmai, J. H. Schiller, B. Wiggerich, and M. Manolov. *Micro-drone for the characterization and self-optimizing search of hazardous gaseous substance sources: A new approach to determine wind speed and direction*. In IEEE International Workshop on Robotic and Sensors Environments (ROSE), pages 1–6, October 2010.

[115] J. Towler, B. Krawiec, and K. Kochersberger. *Radiation Mapping in Post-Disaster Environments Using an Autonomous Helicopter*. Remote Sensing, 4(7):1995–2015, 2012.

[116] J. Han, Y. Xu, L. Di, and Y. Chen. *Low-cost Multi-UAV Technologies for Contour Mapping of Nuclear Radiation Field*. Journal of Intelligent and Robotic Systems, 70(1-4):401–410, 2013.

[117] P. G. Martin, O. D. Payton, J. S. Fardoulis, D. A. Richards, and T. B. Scott. *The use of unmanned aerial systems for the mapping of legacy uranium mines*. Journal of Environmental Radioactivity, 143(0):135–140, 2015.

[118] Y. Sanada and T. Torii. *Aerial radiation monitoring around the Fukushima Dai-ichi nuclear power plant using an unmanned helicopter*. Journal of Environmental Radioactivity, 139(0):294–299, 2015.

[119] Z. Zivkovic and F. van der Heijden. *Efficient adaptive density estimation per image pixel for the task of background subtraction*. Pattern Recognition Letters, 27(7):773–780, 2006.

[120] G. Sourimant, L. Morin, and K. Bouatouch. *GPS, GIS and Video Registration for Building Reconstruction*. In IEEE International Conference on Image Processing, volume 6, pages 401–404, September 2007.

[121] J. Karlekar, S. Z. Zhou, Y. Nakayama, W. Lu, Z. C. Loh, and D. Hii. *Model-based localization and drift-free user tracking for outdoor augmented reality*. In IEEE International Conference on Multimedia and Expo (ICME), pages 1178–1183, July 2010.

[122] Human Factors and Ergonomics Society. NASA-Task Load Index (NASA-TLX); 20 Years Later, volume 50, October 2006.

[123] F. T. Durso and A. R. Dattel. *SPAM: The real-time assessment of SA*. A cognitive approach to situation awareness: Theory and application, pages 137–154, 2004.

Acknowledgements

Three years can be considered a long time. If I look back at this period, I will see the road I have traveled, the mistakes I made, and what I learnt from them. Although there are still several things that I do not understand, I feel I can say: “now I’m ready to start my PhD!”. The feeling is that I learnt more during the PhD than in the “standard” university career. It was a unique experience and I am glad to have received this possibility. The major acknowledgements for this go to Prof. Stefano Caselli for his willingness to support my PhD, both for the research side and the economic side, and for the fundamental advices for the writing of this thesis. Equal acknowledgements go to Jacopo Aleotti, who followed all the work from close and shared the difficulties and the problems of the field robotics with me. Great acknowledgements for the idea and part of the development of the UAV application go to the guys of the CNR, Andrea Zappettini, Giacomo Benassi, Nicola Zambelli and Manuele Bettelli. I also want to thanks Dario Lodi Rizzini to have shared some of his great knowledge about robotics. A great acknowledgement go to my office mates, Fabjan and Alessandro, for the good days spent in our office with those passionate and, sometimes, absurd talks. A great thanks to all the other mates of the ground floor of the Building 1 of the department. Thanks to my family members, who always support me on this path. Thanks to all my friends for their friendship and for the good moments spent together. Thanks to all my mentors and masters that taught me to not give up and critically analyze every situations I have encountered and the ones I will face in the future. Finally, I thanks the Lord to have given also to me the ability to reach all of this.