Linköping Studies in Science and Technology

Thesis No. 1510

Increasing Autonomy of Unmanned Aircraft Systems Through the Use of Imaging Sensors

by

Piotr Rudol

Linköping University
INSTITUTE OF TECHNOLOGY

Submitted to Linköping Institute of Technology at Linköping University in partial fulfilment of the requirements for degree of Licentiate of Engineering

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2011

ISBN 978-91-7393-034-5
ISSN 0280–7971
Printed by LiU Tryck 2011

URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-71295

Increasing Autonomy of Unmanned Aircraft Systems Through the Use of Imaging Sensors

by

Piotr Rudol

November 2011
ISBN 978-91-7393-034-5
Linköping Studies in Science and Technology
Thesis No. 1510
ISSN 0280–7971
LiU–Tek–Lic–2011:49

ABSTRACT

The range of missions performed by Unmanned Aircraft Systems (UAS) has been steadily growing in the past decades thanks to continued development in several disciplines. The goal of increasing the autonomy of UAS's is widening the range of tasks which can be carried out without, or with minimal, external help. This thesis presents methods for increasing specific aspects of autonomy of UAS's operating both in outdoor and indoor environments where cameras are used as the primary sensors.

First, a method for fusing color and thermal images for object detection, geolocation and tracking for UAS's operating primarily outdoors is presented. Specifically, a method for building saliency maps where human body locations are marked as points of interest is described. Such maps can be used in emergency situations to increase the situational awareness of first responders or a robotic system itself. Additionally, the same method is applied to the problem of vehicle tracking. A generated stream of geographical locations of tracked vehicles increases situational awareness by allowing for qualitative reasoning about, for example, vehicles overtaking, entering or leaving crossings.

Second, two approaches to the UAS indoor localization problem in the absence of GPS-based positioning are presented. Both use cameras as the main sensors and enable autonomous indoor flight and navigation. The first approach takes advantage of cooperation with a ground robot to provide a UAS with its localization information. The second approach uses marker-based visual pose estimation where all computations are done on-board a small-scale aircraft which additionally increases its autonomy by not relying on external computational power.

This work has been supported by the National Aeronautics Research Programs NFFP04-S4202, NFFP05, the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII and the ELLIIT network for Information and Communication Technology project grants.

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Acknowledgements

The process of preparing and writing a thesis can be humorously explained by the Thesis Repulsion Field (TRF)[1]. It is characterized by a vector field directed towards the completion of the thesis but with an intense repulsive singularity at its origin. There are several possible trajectories resulting from the forces involved. A *Complete Repulsion*, an *Infinite Orbit* and a *Periodic Productivity* are the most commonly experienced paths. Although highly accurate, the TRF does not model the inversely proportional external force induced by people helping and supporting the work. I am grateful to many such people for disturbing the repulsion field, at one time or another, throughout my thesis work.

I would like to thank my advisors: Patrick Doherty, Andrzej Szałas and Jonas Kvarnström for maintaining a creative and challenging research environment throughout the years. Additionally, I would like to thank people with whom I have coauthored publications: Mariusz Wzorek, Gianpaolo Conte, Fredrik Heintz, Torsten Merz, Simone Duranti, David Lunström, Maria Hempel, Rafał Zalewski and Patrick Doherty.

I owe my deepest gratitude to Mariusz for sharing much of the professional as well as personal experiences throughout the years, countless discussions and creative disagreements which deeply influenced me and my work. I am also very grateful to all other current and former members of AIICS: Torsten, Gianpaolo, Karol Korwel, Simone, David, Rafał, Łukasz Majewski, Maria and Robert Veenhuizen for all the hard but rewarding work with numerous robotic platforms. Additionally, I would like to thank Tommy Persson, Patrik Haslum, Per Nyblom, Björn Wingman and David Landén and all the others for countless joint experiences and discussions, work related and otherwise.

Moreover I would like to thank Fredrik, Jonas, Mariusz, Gianpaolo and Karol for the invaluable input during the writing of the thesis.

And last but not least, for unconditional and continuous support and patience (often irrationally strong and over-stretched by me) and asking about the thesis progress just the right number of times, I would like to thank Justyna, the person with whom I've shared more than a decade of my life.

[1] Jorge Cham: `http://www.phdcomics.com/comics/archive.php?comicid=1354`

Contents

Abbreviations

DOF	Degree of Freedom
EKF	Extended Kalman Filter
FOV	Field Of View
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LRF	Laser Range Finder
MAV	Micro Air Vehicle
PTU	Pan-Tilt Unit
UA	Unmanned Aircraft
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

Chapter 1

Introduction

In recent years, the potential of using Unmanned Aerial Vehicles (UAVs) in many application areas has become apparent. The goal of this thesis is to investigate the use of passive imaging sensors for increasing the autonomy of such vehicles. For UAVs operating outdoors, the use of cameras and appropriate algorithms allows for widening the range of useful tasks which can be carried out without, or with minimal, human involvement. For UAVs operating indoors, in the absence of positioning systems such as GPS, cameras offer means of facilitating autonomous flight by providing localization information.

An unmanned aerial vehicle is an airborne vehicle without a human crew on-board. Historically, the term and the primary use of UAVs have been in the military area. Continuous development and technology transfer has lowered the cost of accessing the technology outside the military domain. This has allowed for expanding the use of unmanned aerial vehicles to many civilian applications.

In recent years, the term UAV has been replaced with the term UA which stands for *Unmanned Aircraft.* To emphasize that a UA is a part of a complete system including ground operator stations, launching mechanisms and so forth, the term *Unmanned Aircraft System* (UAS) has been introduced[1] and its use is becoming commonplace. In this thesis the terms UA and UAV will be used interchangeably.

The degree of autonomy in a UAS can vary greatly, ranging from teleoperation to fully autonomous operation. At one end of the autonomy spectrum, teleoperation is a mode of operation where a UAV is commanded by a human operator using a remote controller from the ground. This kind of UA is also referred to as a *remotely operated vehicle.* Radio controlled hobby airplanes or helicopters belong to this category as control commands are transmitted wirelessly to the UA. In this mode of operation the degree of

[1] US Army UAS RoadMap 2010-2035, `http://www.rucker.army.mil/usaace/uas/US Army UAS RoadMap 2010 2035.pdf`

autonomy is minimal because continuous input from an operator is required.

Modes of operation from the other end of the autonomy spectrum are of interest in this thesis. Fully autonomous UAVs are characterized by the ability to maintain flight and to carry out complete missions from take-off to landing without any human pilot intervention. Autonomous UAs should be able to execute sequences of actions prepared by an operator and also to achieve so called *high-level* goals. For example, requesting a UAS to search for injured people in a specified area and monitor traffic conditions (e.g. detecting reckless driver behavior) can be considered high-level goals.

In order to carry out such sophisticated tasks a UAS has to be able to autonomously execute and coordinate a wide range of tasks. Making UAS's autonomous requires addressing many problems from different disciplines, such as mechanical design, sensor technologies, aeronautics, control, computer science and artificial intelligence, to name a few. Each of these fields has a community of its own but a combination of all of them can be put under the umbrella of robotics.

The term *robot* first appeared in the 1920 play R.U.R. by the Czech writer Karel Čapek[2]. The term comes from the word *robota* which means labour in many Slavic languages. Contemporarily, the term refers to mechanics, electronics and software of physical entities performing a wide range of tasks. The main aim of robots is to carry out tasks, which are considered Dirty, Dull and Dangerous (D3) for humans. Autonomy means that robots are able to perform a wider range of tasks by interacting and operating safely in different kinds of environments without relying on external help. Even though such help can often be realized using, for example, wireless transmission, it is generally safer for a robot to be self-sufficient. Wireless communication can, for example, easily be interfered with, which may compromise the safety of the operation.

The field of robotics is large and can be categorized in many ways. Mobility is one of the ways to categorize robots. Stationary industrial robots (see figure 1.1A) operate in fixed static environments and most often perform repetitive tasks (e.g. car assembly). Mobile robots, on the other hand, operate in more dynamic domains and require the ability to sense and react to a changing environment. Issues which need to be addressed in mobile robotics involve, for example, localization and mapping. The autonomous car Stanley, developed by the Stanford Racing Team in cooperation with the Volkswagen Electronics Research Laboratory, is an example of a mobile robot (see figure 1.1B) [80]. It was able to autonomously navigate for hundreds of kilometers during the DARPA Grand Challenge in 2005. This was achieved by using sensors to localize the vehicle and avoid obstacles on the way.

Another way to categorize robots is by size. In case of airborne platforms, there exist large UAVs, such as the General Atomics MQ-9 Reaper - Predator

[2] Karel Čapek: `http://capek.misto.cz/english/presentat.html`

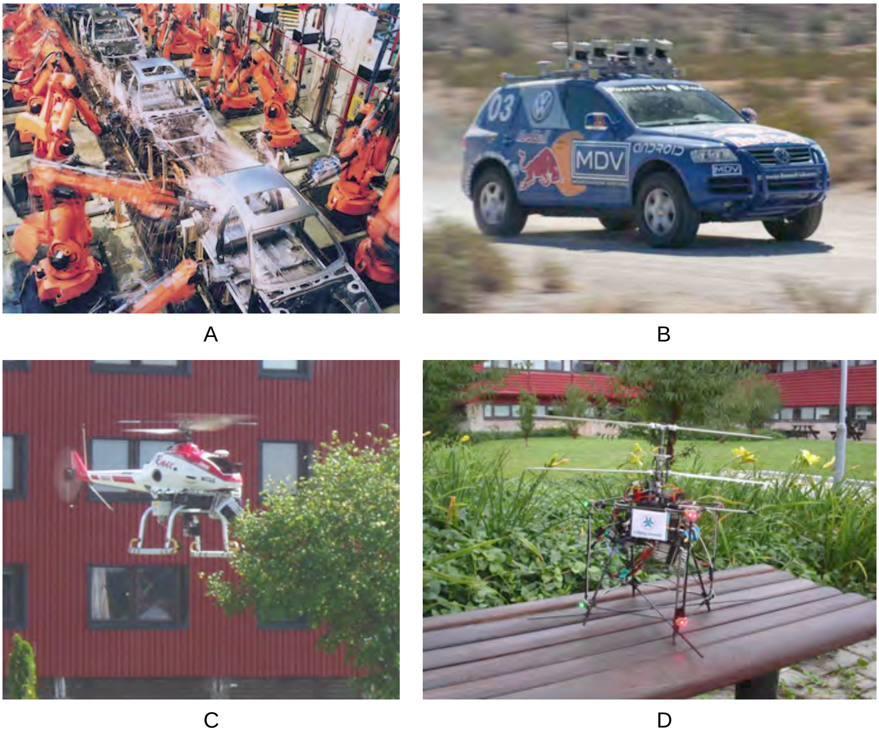

Figure 1.1: A. ABB industrial robots used for car assembly. B. Stanley autonomous car. C. Yamaha RMAX unmanned helicopter. D. LinkMAV small-scale unmanned aerial vehicle.

B (4536 kilograms of weight and 20 meters of wing span)[3]. This thesis, however, focuses on medium and small size UAs. The Autonomous Unmanned Aircraft Systems Technologies Lab (UASTechLab, formerly WITAS) RMAX helicopter [18], presented in figure 1.1C, is an example of a medium size UAV (95 kilograms of weight and length of 3.6 meters). Smaller platforms, also referred to as Micro Air Vehicles (MAVs), which weigh less than 500 grams and whose largest dimension is smaller than 50 centimeters, are mainly used for indoor applications (see figure 1.1D). The smaller the airframe, the narrower the range of suitable sensors and computational hardware that can be used. The main restriction is the smaller payload capability which is further restricted by limited on-board power.

Airborne robots face an additional set of issues compared to ground vehicles. The faster dynamics of flying platforms impose strict requirements on timely execution of control tasks as well as on well-timed decision making

[3] General Atomics Predator B: `http://www.ga-asi.com/products/aircraft/predator_b.php`

which are both dependent on timely sensing. This is especially important for UAs operating indoors where distances to obstacles are small. Moreover, the use of airborne vehicles implies stricter safety requirements which involves high dependability and robustness of UAS's. For example, UAVs normally have larger operating areas compared to ground robots which are therefore easier to contain in a specified safety zone.

Flying robots operating in natural environments face various and often unique challenges. It is important to note that indoor and outdoor environments present different challenges for flying robots. While some techniques for dealing with these problems can be transferred from one to the other, some issues have to be solved independently. The biggest challenge for indoor UAS's is the lack of a ubiquitous positioning system such as the Global Positioning System (GPS). Even though its availability is never guaranteed, it was in fact the enabling technology for outdoor UAVs. Several other commonly used sensors are also not useful indoors. A pressure sensor used for altitude measurement is unreliable due to the influence of, for example, air conditioning systems. Similarly, magnetometers used for measuring the relation between an airframe and the Earth's magnetic field are easily disturbed by metal elements in building structures. These factors make autonomous indoor operation of UAVs an open research issue because the dependence on these sensors has to be relaxed or replacements have to be found.

1.1 Problem statement

The problem which this thesis addresses is the usage of passive imaging sensors, i.e. cameras, to facilitate the autonomy of UAS's operating in both outdoor and indoor environments. In case of the former the input from cameras is used to increase the number of tasks a UAS can perform autonomously. For the latter, a camera sensor is used to provide basic functionality which is the ability to maintain flight without human input.

There are several reasons for focusing on imaging sensors. UAs are almost always equipped with cameras for other purposes (e.g. surveillance) so no additional hardware is generally needed. Moreover, cameras are in general lightweight, low-cost and have low power requirements. A UAS can perform a wider range of tasks by incorporating mainly software solutions.

1.2 Contributions

The thesis investigates the use of imaging sensors for increasing the level of autonomy of unmanned aircraft systems. In case of UAS's operating outdoors the level of autonomy is increased through the use of color and thermal imaging by improving the system's situational awareness. Fusion of these two kinds of image-based input for human body detection and geolocation is part of an autonomous search and rescue mission. It is a mission in which a user selects a region to be scanned for potential survivors. The result of the scanning is a map of victim locations. The generated map can be used by first responders or a robotic system to, for example, deliver food or medical supplies [20, 72]. The car tracking functionality which is also based on the fusion of color and thermal images provides streams of low level events (i.e. geographical coordinates) of vehicles on the ground. It is then transformed into a stream of events that allows for qualitative reasoning about the environment. It allows, for example, for reasoning about vehicles overtaking, entering and leaving crossings. Such a functionality allows for greatly increasing the situational awareness of a robotic system [41, 42].

Additionally, two approaches to the UAV indoor localization problem are presented. Both use cameras as the main sensors in the absence of GPS-based positioning. The first approach takes advantage of cooperation with a ground robot to provide a UAV with its localization information. The aerial vehicle acts as a remote sensor so that the complete system has a greater sensor range and better reach than any of its components [73]. The second approach uses marker-based visual pose estimation where all computations are done onboard the UAV which increases its autonomy by not relying on external computational power [75]. Both approaches have been evaluated and tested in real flight tests with all components fully functional, without tethers or other aids.

1.3 List of publications

This thesis is mainly based on the following peer-reviewed publications:

[72] *P. Rudol and P. Doherty.* Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery. *Proceedings of the IEEE Aerospace Conference, 2008.*

[20] *P. Doherty and P. Rudol.* A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization. *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, 2007.*

[42] *F. Heintz, P. Rudol, P. Doherty.* From Images to Traffic Behavior - A UAV Tracking and Monitoring Application. *Proceedings of the 10th International Conference on Information Fusion, 2007.*

[41] *F. Heintz, P. Rudol, P. Doherty.* Bridging the Sense-Reasoning Gap Using DyKnow: A Knowledge Processing Middleware Framework. *Proceedings of the 30th German Conference on Artificial Intelligence, 2007.*

[73] *P. Rudol, M. Wzorek, P. Doherty.* Micro Unmanned Aerial Vehicle Visual Servoing for Cooperative Indoor Exploration. *Proceedings of the IEEE Aerospace Conference, 2008.*

[75] *P. Rudol, M. Wzorek, P. Doherty.* Vision-based Pose Estimation for Autonomous Indoor Navigation of Micro-scale Unmanned Aircraft Systems. *Proceedings of the IEEE International Conference on Robotics and Automation, 2010.*

The first, third, fifth and sixth publications ([72], [42], [73] and [75], respectively) from the above list are included in the appendix to this thesis. Additionally, the following related publications fall outside the direct scope of this thesis:

[64] *T. Merz, P. Rudol, M. Wzorek.* Control System Framework for Autonomous Robots Based on Extended State Machines. *Proceedings of the International Conference on Autonomic and Autonomous Systems, 2006.*

[90] *M. Wzorek, G. Conte, P. Rudol, T. Merz, S. Duranti, and P. Doherty.* From Motion Planning to Control - a Navigation Framework for an Autonomous Unmanned Aerial Vehicle. *Proceedings of the 21th Bristol International UAV Systems Conference, 2006.*

[23] *S. Duranti, G. Conte, D. Lundström, P. Rudol, M. Wzorek, and P. Doherty.* LinkMAV, a prototype rotary wing micro aerial vehicle. 2007. *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace, 2007.*

[13] *G. Conte, M. Hempel, P. Rudol, D. Lundström, S. Duranti, M. Wzorek, and P. Doherty.* High accuracy ground target geo-location using autonomous micro aerial vehicle platforms. *Proceedings of the AIAA Guidance, Navigation, and Control Conference, 2008.*

[74] *P. Rudol, M. Wzorek, R. Zalewski, and P. Doherty.* Report on sense and avoid techniques and the prototype sensor suite. *National Aeronautics Research Program NFFP04-031, Autonomous flight control and decision making capabilities for Mini-UAVs, 2008.*

1.4 Outline

The first part of the thesis deals with the problem of increasing the autonomy of a UAV through improving its situational awareness. Problems of object detection, mapping and tracking are addressed. Specifically, human body detection and the geolocation problem as well as the issue of vehicle tracking are addressed in chapter 2. These techniques add functional elements to outdoor operating UAS's such as the UASTechLab RMAX autonomous helicopter.

The problem of localization (determining position and attitude i.e. pose estimation) in indoor environments in the absence of a GPS-based positioning is addressed in chapter 3. Two approaches which increase the autonomy of UAS's by allowing for autonomous indoor flight are presented.

The first one (section 3.4) allows for indoor navigation of a UA in cooperation with a ground robot. The computationally intensive task of image processing is delegated to a computer on the ground robot which determines the UAV's pose based on a specially designed cube structure attached to the flying vehicle.

In the second approach (section 3.5), a marker-based visual pose estimation method is used and all computations are carried out on board the UAV. In both cases, a camera is used to provide both the position of a UAV in flight (replacing the GPS and an altimeter) and attitude angles (most importantly heading) replacing a compass.

The thesis concludes and gives directions for future work in chapter 4.

Chapter 2

Fusing thermal and color images for object detection, tracking and geolocation

2.1 Introduction and motivation

The use of autonomous unmanned aircraft systems is slowly but steadily becoming commonplace outside of military application areas. Many of the missions in which UAS's can be employed include tasks such as ground object detection, identification, tracking and calculating its geographical location i.e. geolocation. This chapter presents an algorithm for fusing color and thermal images which is suitable for, for example, detecting and geolocating human bodies as well as for vehicle tracking. Both of these functionalities can be used as parts of a UAS mission, effectively increasing the number of tasks which can be carried out without or with minimal human input.

Human body detection and geolocation is an area which is particularly important and beneficial for society in catastrophe or emergency assistance situations. Rapidly acquiring an awareness of a situation after a man-made or natural disaster can play an important role in relief efforts. It can support rescuers by focusing their work where the help is most needed.

The use of autonomous unmanned aircraft to facilitate the help can be very valuable in such situations since certain types of work can be considered dirty, dull, or dangerous if performed by humans. Specifically, autonomy reduces the involvement of the operator from controlling every movement to supervising the operation. The operator's load is greatly reduced because most of the dull and repetitive work is performed by a robot.

Additionally, performing this task on-board a UAV is beneficial for two reasons. First, it avoids the need for transmitting the video signal. Only the result of the analysis has to be transmitted to an operator. Transmitting these results requires less bandwidth which is preferred in catastrophe situations where the necessary infrastructure for transmitting bandwidth-demanding video signal might be unavailable. Second, the situational awareness is achieved directly during the flight which would not be the case if the gathered data was gathered for later analysis. Partial information can be transmitted before the complete result is available.

This chapter presents a method for fusing thermal and color video streams for creating saliency maps of human bodies as well as vehicle tracking and geolocation. The saliency maps can be directly used by emergency services. The method can also be an intermediary step in a fully autonomous mission which includes generating a plan of supply (e.g. medicine, food, water) delivery followed by the delivery itself. The vehicle tracking functionality, when used with a road system database, allows for detecting high-level traffic events. Counting vehicles entering crossing, and detecting reckless overtakes are examples of such events.

The remainder of the chapter is structured as follows. It starts with presenting an example mission in section 2.2. This is followed by a description of the UAS platform used for experimental validation of the proposed technique. The existing hardware platform and software architecture are presented in section 2.3. An overview of the related work in section 2.4 is followed by a description of an algorithm for building saliency maps in section 2.5. Finally, results of using the algorithm are presented in section 2.6 followed by a description of how these results can be used in a complete autonomous mission in section 2.7. Finally, section 2.8 describes an application of the same algorithm in a slightly modified form used for vehicle tracking and geolocation.

2.2 An emergency situation scenario

In order to understand the potential of using UAS's in emergency situations it is beneficial to briefly describe an example scenario and outline how a robotic system can be useful in such a setting.

After a natural disaster such as an earthquake or a tsunami wave a large area has to be searched for potential survivors. This is a prerequisite for planning the logistics of delivering different kinds of supplies to the survivors. Of course, such a task can be carried out by manned aircraft, but by using a larger number of smaller autonomous UAVs operating in parallel the task can be finished quicker and without risking human pilots' lives.

A complete mission of this type has to incorporate the following subtasks:

- Scanning an area and searching for salient entities such as injured humans, building structures or vehicles.

- Monitoring or surveilling these salient points of interest and continually collecting and communicating information back to ground operators and other platforms to keep them situationally aware of current conditions.
- Delivering supplies or resources to these salient points of interest if required. For example, identified injured persons should immediately receive a relief package containing food, medical and water supplies.

Although quite an ambitious set of tasks, several of them have already been achieved to some extent using the existing experimental helicopter platforms as described in the following subsections.

To be more specific in terms of the scenario, we can assume there are two separate parts to the emergency relief scenario in the context sketched previously:

- Part I:

  In the first part of the scenario, it is essential that for specific geographic areas, the UAVs cooperatively scan large regions in an attempt to identify injured persons. The result of such a cooperative scan is a saliency map pinpointing potential victims, their geographical coordinates but optionally also sensory output such as high resolution photos and thermal images. The resulting saliency map can be used directly by emergency services or passed on to other UAS's as a basis for additional tasks.

- Part II:

  In the second part of the scenario, the saliency map generated in Part I can be used as a basis for generating a logistics plan for several of the UAS's with the appropriate capabilities to deliver food, water and medical supplies to the injured. This would also be done in a cooperative manner among the platforms.

The remainder of this chapter will focus on Part I of the mission. Part II is briefly described in sections 2.7.1 and 2.7.2 and also in Doherty and Rudol [20].

2.3 Experimental platform

This section presents the hardware platform used in the experimental validation of the algorithm. The helicopter UAV has been developed during the WITAS project ([21]) and is now used in the UASTechLab for outdoor experimentation [18]. The following subsections present the main aspects of the UAS hardware as well as the software architecture and the existing system components.

2.3.1 RMAX UAV platform

The UASTechLab platform is a slightly modified Yamaha RMAX helicopter (figure 2.1). It has a total length of 3.6 m (including main rotor) and is powered by a 21 horsepower two-stroke engine with a maximum takeoff weight of 95 kg. Comparing to a standard RMAX helicopter it has a more powerful electric generator (200 W) and has a higher landing gear which allows for placing a camera system underneath the UAV body.

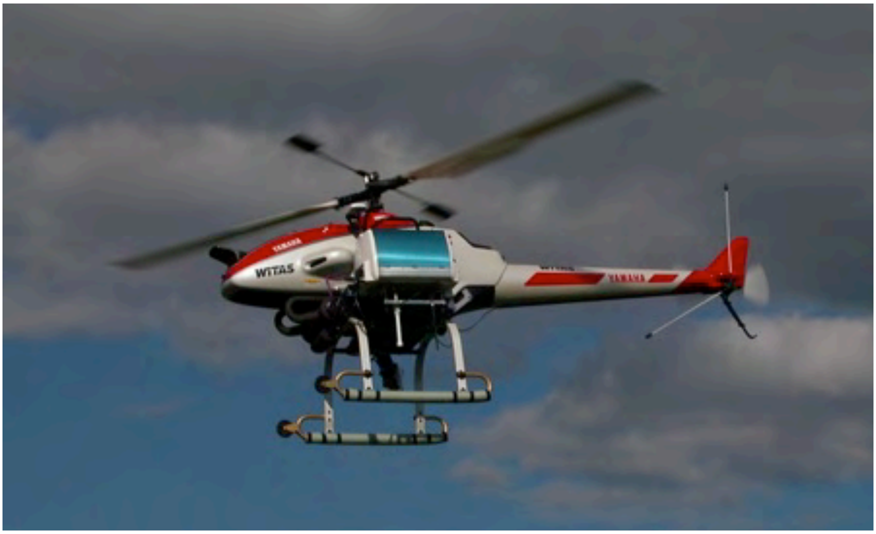

Figure 2.1: UASTechLab RMAX autonomous helicopter platform.

The on-board system consists of three PC104 computers. The general schematic of the system is presented in figure 2.2. The Primary Flight Computer (PFC) hosts a Pentium III 700 MHz processor and interfaces with a GPS receiver, a barometric pressure sensor and a magnetic compass. The PFC also interfaces with the Yamaha RMAX platform through the Yamaha Attitude Control System (YACS) (which contains the Yamaha Attitude Sensor - YAS) and is the only computer required for maintaining autonomous flight capability when using GPS as a positioning sensor.

The main purpose of the PFC is real-time execution of sensor fusion algorithms and control modes such as take-off, hovering and 3D path following [12].

The Image Processing Computer (IPC) hosts a 700 MHz Pentium III processor and is responsible for grabbing images from two video cameras and running image processing algorithms (such as the existing vision-based landing [63] and vision-based localization [11]). The two cameras (Sony CCD

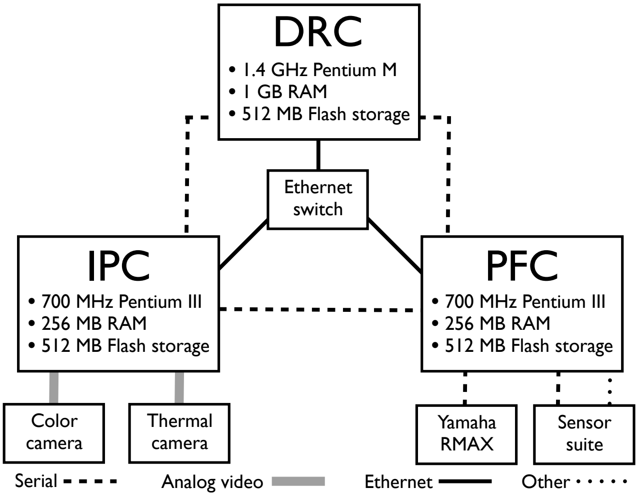

Figure 2.2: Schematic of the computer system and the main hardware components of the UASTechLab RMAX platform.

block camera FCB-780P[1] and a ThermalEye-3600AS[2] miniature infrared camera) are mounted on a pan-tilt unit (PTU-46-17.5W[3]) and are attached under the UA fuselage to a camera platform (see figure 2.3). The platform is vibration isolated by a system of springs.

The video footage from both cameras is additionally recorded at full frame rate by two miniDV recorders to allow post-flight processing. The ability to record the video in this high fidelity format is very useful for image processing algorithm development and debugging.

The third on-board system, the Deliberative-Reactive Computer (DRC), hosts a 1.4 GHz Pentium-M and executes all high-level autonomous functionalities such as mission and path planning.

All three computers are connected with one-to-one serial lines for real-time communication and with Ethernet links through a switch. The communication with the ground station is achieved using a 801.11b/g wireless Ethernet bridge from 3Com.

2.3.2 Software architecture

As the types of missions a UAS can perform become more and more sophisticated, a software architecture has to be able to accommodate and coordinate different types of functional modules. The goal when designing and developing the architecture of the UASTechLab system was to enable so called

[1] Homepage: http://pro.sony.com

[2] Homepage: http://www.l-3com.com

[3] Homepage: http://www.dperception.com

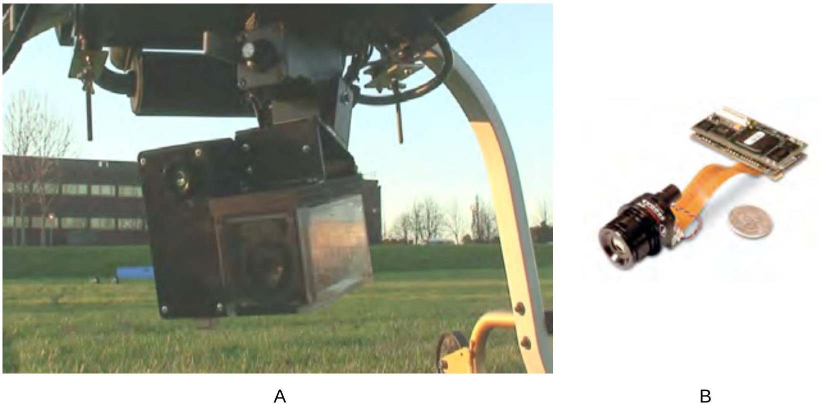

Figure 2.3: A. Thermal and color cameras (left and right respectively) mounted on a pan-tilt unit underneath the UASTechLab RMAX platform. B. ThermalEye-3600AS thermal camera.

push-button missions, in which the UAS autonomously plans and executes tasks from take-off to landing without operator intervention. One such mission is building inspection in which the UAS records video footage of all the facades of all user selected building structures. The software architecture has to support many different functional units, such as control modes (take-off, 3D path following, landing), task and path planners, optimal camera placement algorithms etc.

The software of the UAS is of a hybrid deliberative/reactive type. It is a layered, hierarchical system with deliberative, reactive and control components, although the system can easily support both vertical and horizontal data and control flow. Figure 2.4 presents the functional layer structure of the architecture and emphasizes its reactive-concentric nature. The Control Kernel is encapsulated in a Common Object Request Broker Architecture (CORBA) object. Task Procedures (TPs [67]) are high-level procedural execution components which provide a computational mechanism for achieving different robotic behaviors by using both Deliberative Services (DS) and control components in a highly distributed and concurrent manner. Deliberative services provide high-level functionalities such as task and path planning and Geographic Information System (GIS) databases.

The Control Kernel is distributed over the three previously mentioned on-board computers as presented in figure 2.5. Execution and coordination of various tasks (from device drivers to continuous control laws) is achieved though the use of event-driven hierarchical concurrent state machines (HCSM; similar to the extended state machines described in [64]). An HCSM interpreter running in the real-time Linux environment (RTAI Linux patch [62]) in each of the machines coordinates the task execution as well as communication between computers. By using serial lines for communica-

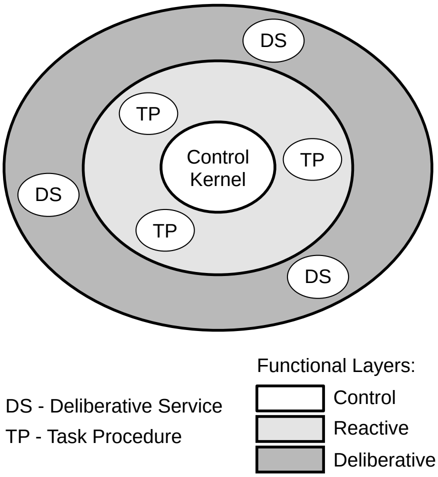

Figure 2.4: Functional structure of the architecture.

tion, the Control Kernel achieves real-time properties despite the physical distribution. It is functionally encapsulated into a CORBA object accessible to TPs (as the Helicopter Server).

The use of a realtime operating system is essential for achieving robust operation of the Control Kernel. The RTAI Linux patch implements the realtime behavior by executing the code as a Linux kernel module. The execution is nonpreemptive and statically scheduled and it has to be periodically suspended to allow the non-realtime part of the operating system to perform its tasks (e.g. keyboard and screen handling, flash memory access and TCP/IP communication). The three PC104 computers on-board the UA have different distribution of processor time between the kernel and user space parts. On one extreme, PFC is mostly realtime as it performs tasks which require strict timing of task execution. On the other extreme, the DRC only uses the realtime execution for timely communication with the other on-board computers. The IPC is somewhere in between as it handles more realtime tasks than the DRC but the execution of tasks is not as time critical as in the case of the PFC.

2.3.3 Existing autonomous functionalities

In order to test the method for fusing thermal and color images, used in the saliency map building and car tracking algorithms, it has been integrated into the UASTechLab UAV helicopter system as a new mission functionality. The following existing autonomous functionalities are part of a complete mission:

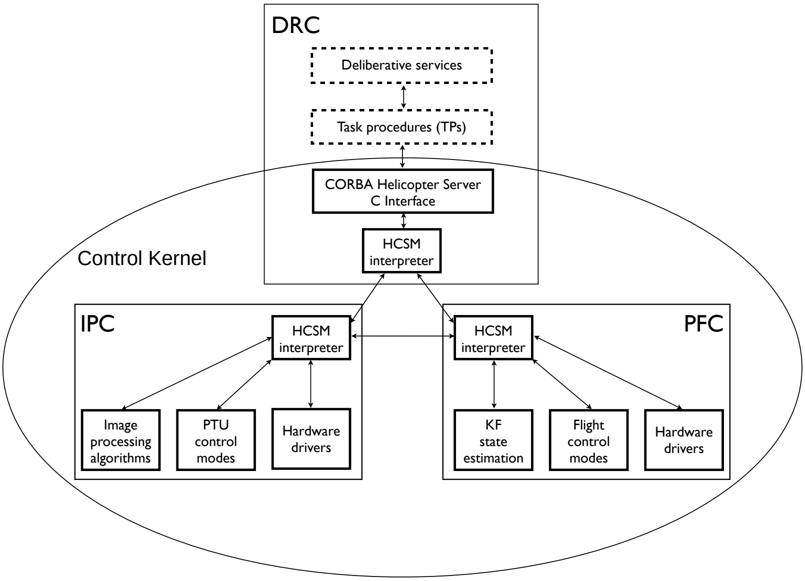

Figure 2.5: Distributed Control Kernel architecture.

- Take-off: this control mode brings the helicopter platform from the ground to a specified altitude.
- Hovering: this is the default control mode of the platform.
- Multiple platform area coverage path planning algorithm: it computes paths for n heterogenous platforms and guarantees complete camera coverage of the specified area. It takes into account sensor properties and platform capabilities when computing the paths.
- 3D path following: this control mode allows for tracking 3D paths defined as splines [12].
- Vision-based autonomous landing (without the use of GPS) using a specially designed landing pattern [63].

2.4 Related work

The task of observing and analyzing human appearance and movement has been of interest to the computer vision community for many years. The existing techniques can be categorized in many ways. One of them is the need for pre-processing, such as background subtraction, which can be achieved by frame differencing [57, 94]. Other factors include the types of features which are needed for describing human appearance e.g. shape, color and contours.

A considerable amount of work is based on the idea of detecting humans by parts. For example, a human body can be modeled as an assembly of parts which are detected separately and represented by co-occurrences of local features [66]. A cascade-of-rejectors with variable size blocks of histograms of oriented gradients as features can also be used. AdaBoost is used as a feature selection technique to choose appropriate blocks from a large set of possible blocks [27]. The use of integral image representation and a rejection cascade allows for 5 to 30 Hz human detection performance (for images of size 320x280 pixels) [95]. Another approach takes advantage of a classifier which is a cascade of boosted classifiers working with Haar-like features. The classifier is learned using boosting [82]. The details of this technique are are presented in section 2.5.5.

A recent work and the state-of-the-art in the area of visual people detection from a UAV is presented in Andriluka et al. [3]. The paper presents results of evaluating five algorithms based on monolithic and on part-based representations for detecting humans using color images. The authors propose combining several types of models for dealing with occlusions as well as complex poses of bodies. Additionally, the authors suggest taking advantage of the knowledge of the UAV state. Both the altitude and attitude help to assess the size of a human body to be detected. The knowledge of attitude additionally makes it possible to rectify images to make the scale of human bodies consistent even if the camera observing a scene is tilted (bodies further away from the camera appear smaller after projecting onto the image plane). The evaluation of the algorithms is performed on a dataset collected on-board a quadrotor UAV in an indoor office environment. Even though the presented results show that the suggested enhancements improve the detection performance, applicability of the method in typical outdoor environments is not obvious. The main aspect to be investigated is the problem that bodies appear smaller when observed from larger distances (order of tens instead of 1-3 meters in indoor settings).

Detecting humans in thermal imagery poses additional challenges such as lower resolution, halos around hot or cold objects and smudging artifacts in case of camera movement. An approach which first performs a fast screening procedure using a template to locate potential person locations, which is then tested using an AdaBoosted ensemble classifier using automatically tuned filters, has been proposed in Davis and Keck [14]. The technique, however, has been tested on footage collected by a stationary thermal camera and therefore the applicability of the technique to a moving camera is unknown.

Techniques using both color and thermal images have been suggested. One example uses color and infrared cameras and a hierarchical scheme to find a correspondence between the preliminary human silhouettes extracted from both cameras using image registration in static scenes. Han and Bhanu [35] also discuss strategies for probabilistically combining cues from registered color and thermal images. A technique for detecting and tracking moving targets in overlapping electro-optical and infrared sensors (EO/IR) by a probabilistic framework for integrating multiple cues from multiple sensors also has been proposed [50]. The method has been tested on footage collected by a UAV but its computational requirements are not discussed and its usability on-board a UAV is unknown.

The use of UAS's has been evaluated in Wilderness Search and Rescue (WiSAR) applications. This application area has been studied in terms of identification of a set of requirements, designing appropriate search algorithms and field tests in realistic settings [30–32]. The results show how to best use fixed wing small-scale UAVs with downward-pointing camera in WiSAR applications. In all the presented work UAVs act as remote sensors providing a live video signal to a ground operator station. No processing of the video is performed on-board. Thus the work of analyzing the data can either be performed on the ground (wireless video transmission can be interfered) or be done by a human.

Another example of using UAVs in search and rescue scenarios is the work presented in Waharte and Trigoni [83]. The paper explores the interplay of some fundamental parameters (e.g. quality of sensor data, energy limitations of a platform, environment hazards and level of information exchange) of search algorithms. Performance of different search algorithms which optimize the time to find a victim is described. Although the paper mentions the use of quadrotor UAV platforms the presented results are based on simulations.

Results of fusion of two different sensor modalities, i.e. thermal and color video, presented in Rasmussen et al. [70] show that the cognitive load of a user can be reduced compared to presenting the two modalities separately. The technique has been evaluated on real video sequences collected using small-scale fixed wing UAS's in a WiSAR scenario.

The problem of object tracking from a UAV and in particular with regard to ground vehicles is an active field of research. Different aspects of the problem are addressed. For example, Helble and Cameron [43] have developed a helicopter-based UAS capable of visual tracking of intelligent targets (which try to hide from an observer) by combining methods for geolocation and automatic target re-acquisition. Other approaches to the vehicle tracking problem can be found in [17, 47, 51, 58, 79, 91]. The related problem of planning a UAV's movements for vehicle tracking are also often addressed [79].

2.5 A saliency map building algorithm

The process of building maps where positions of identified human bodies are the salient points can be divided into several steps. The complete process as well as the results of experimental validations are described in the following subsections.

2.5.1 Overview

An overview of the saliency map building algorithm's pseudocode is presented in Algorithm 1. The algorithm is started and terminated at specific time points during the course of the mission execution. Specifically, the execution starts when the host UAV arrives at the starting position of the area to scan and is terminated when the scanning flight is finished.

Algorithm 1 Saliency map building

```
 1: Initialize data structures
 2: while scanning not finished do
 3:    Simultaneously grab two images: $img_{color}$, $img_{thermal}$
 4:    Analyze $img_{thermal}$ to find potential human body regions
 5:    for each region in $img_{thermal}$ do
 6:       Find corresponding region $r_{color}$ in $img_{color}$
 7:       Compute geographical location $loc$ of $r_{color}$
 8:       Execute human body classifier on $r_{color}$
 9:       if classification positive then
10:          if $loc$ is new then
11:             add location $loc$ to map, initialize certainty factor $p_{body}(loc)$
12:          else
13:             update certainty factor $p_{body}(loc)$
14:          end if
15:       end if
16:    end for
17: end while
```

After initializing the necessary data structures (line 1) the algorithm enters the main loop (line 2), which is terminated when the scanning of the specified area is finished. The main loop begins with simultaneously grabbing two video frames. The thermal image is analyzed first (line 4) to find a set of regions of intensities which correspond to human body temperatures (see the first subsection of 2.5.5). Then (line 6), for each of these subregions a correspondence in the color frame, as well its geographical location *loc*, are calculated (see the second subsection of 2.5.5). The calculated corresponding region in the color frame is analyzed with a human body classifier to verify the hypothesis that the location *loc* contains a human body (see the third subsection of 2.5.5). If the classification is positive and the location *loc* has not been previously identified, then *loc* is added to the map and its certainty factor initialized (line 11). Otherwise, the certainty factor of that location containing a body is updated (line 13, see subsection 2.5.6).

The output of the algorithm is a set of geographical locations loc_i and certainty factors $p_{body}(loc_i)$.

2.5.2 Preliminaries

The saliency map building algorithm for the UAS-assisted emergency services mission has been designed taking into account the following mission requirements and platform properties.

First, the typical distance between an object of interest and a camera is large and the algorithm has to be able to cope with objects of small size. Additionally, an object of interest may remain in the camera Field Of View (FOV) only for a short period of time. Even in the case of platforms capable of low velocity flight (such as helicopters) high flight velocity is preferred in case of search and rescue applications because it allows for finding potential victims faster. Therefore, the rate of image processing is one of the most important parameters.

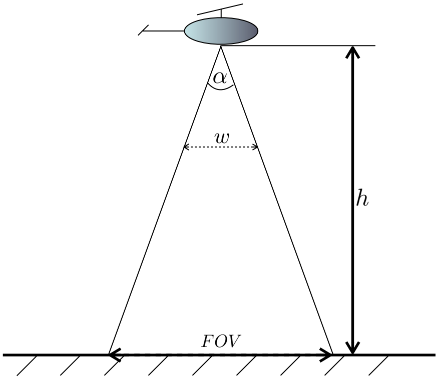

Figure 2.6: Field of view calculation for a downward pointing camera. The image plane is schematically depicted with a dashed line.

The optimal altitude for a scanning mission can be computed given the camera parameters and the requirements of the particular image processing algorithm. Specifically, the minimal required size of an object in an image, measured in pixels. First, the field of view FOV (see figure 2.6 for illustration) is a function of the altitude h and the view angle α:

$$FOV = 2\,h\,tan(\frac{\alpha}{2}). \tag{2.1}$$

Assuming a camera pointing down at a flat ground, the length of the projection of an object S_w on the image plane S_{img} can be calculated as:

$$S_{img} = \frac{S_w\,w}{FOV}, \tag{2.2}$$

where w is the size of the image frame measured in pixels. Both w and α can be different for vertical and horizontal dimensions of a video frame. Figure 2.7 presents the object projection size for two different frame dimensions

w and two example object sizes resembling a human body (S_w of 1.5 m and 1.8 m).

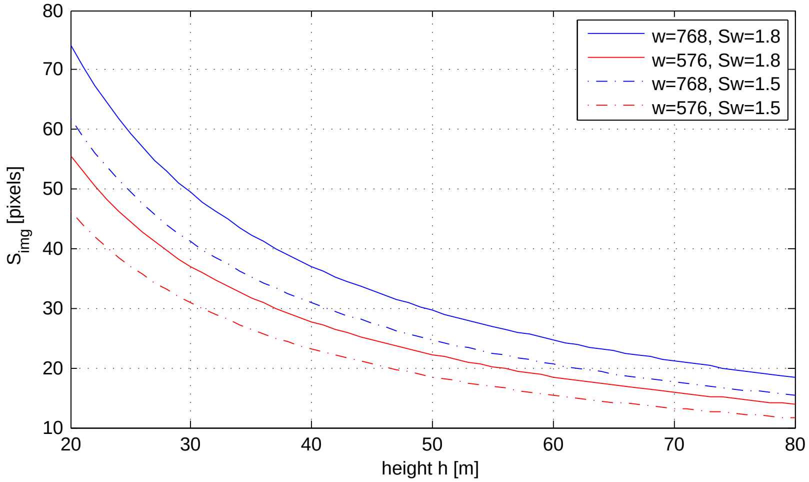

Figure 2.7: The size of an object in an image (S_{img}) depending on the camera height (altitude) when FOV is 50 degrees for two different cameras and two different object sizes.

For example, to obtain object projection sizes of at least 20 pixels in the image for standard PAL resolution[4] and a 50 degrees field of view camera lens, the maximum flight altitude is approximately 50 meters. Higher resolution cameras would allow flight at higher altitudes. Flying at lower altitudes makes the task of image processing easier since a human body appears larger in a video frame. On the other hand, the flight at a lower altitude requires more time to complete since the camera covers a smaller area per video frame.

2.5.3 Image formation

Imaging sensors detect light scattered from viewed scenes and represent the world as two-dimensional projections. In order to relate a real world object and its projection in an image plane a perspective model is commonly used. A simple pin-hole camera model (Equation 2.3) expresses the geometrical relations of a scene and an image where the camera aperture is described as a point and no lenses are used to focus light.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A\,[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{2.3}$$

[4] PAL resolution: 768×576 for square pixels or 720×576 for non-square pixels

where (X, Y, Z) are the coordinates of a point in a three-dimensional scene and (u, v) are the coordinates of its projection in the image plane. The 3×4 matrix $[R|t]$ is the matrix of extrinsic parameters and describes the rotation and translation of the camera in the scene (or vice-versa). Calculating the extrinsic camera parameters R and t is often the task of pose estimation algorithms. The parameter λ is the homogenous scaling factor. The matrix:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

is composed of the intrinsic parameters and is also referred to as the camera calibration matrix. It contains focal lengths f_x and f_y and coordinates of the principal point c_x and c_y. It can also contain a factor which accounts for the skew due to non-rectangular pixels. In most modern cameras, however, this factor can be neglected.

In practice the pin-hole camera model is insufficient because it does not model the effects of distortions introduced by a lens which focuses light on an imaging element of a camera. A more comprehensive image formation model which includes radial and tangential lens distortions and is equivalent to 2.3 can be rewritten in the following way. For $z \neq 0$:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{2.5}$$

$$\begin{aligned} x' &= \tfrac{x}{z}, \\ y' &= \tfrac{y}{z} \end{aligned} \tag{2.6}$$

$$\begin{aligned} x'' &= x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x'y' + p_2(r^2 + 2x'^2) \\ y'' &= y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(x^2 + 2y'^2) + 2p_2 x'y', \end{aligned} \tag{2.7}$$

where $r^2 = x'^2 + y'^2$, k_i and p_i are the radial and tangential distortion coefficients, respectively. Finally, 2.3 becomes:

$$\begin{aligned} u &= f_x x'' + c_x \\ v &= f_y y'' + c_y \end{aligned} \tag{2.8}$$

For efficiency reasons the calculations expressed in Equations 2.7 are often implemented in the form of a lookup table. Such a table maps image coordinates distorted by a lens to the ideal ones. The lookup table is initialized when the application is started. While using such a table requires allocating a certain amounts of memory, the speedup can be significant especially on slower hardware.

Calculating the camera matrix $P = A[R|t]$ as well as determining lens distortion parameters is achieved through a process of camera calibration. Given a number of point correspondences $x_i \leftrightarrow X_i$ between three-dimensional points in a scene X_i and two-dimensional points x_i in an image the camera matrix P can be calculated such that $x_i = PX_i$ for all i. In practice the camera calibration procedure requires capturing images of a known target (e.g. chessboard-like pattern) of known dimensions to obtain point correspondences. Rectangular features are most commonly used in calibration patterns because of the ease of determining pixel coordinates of corner features.

The lens distortion parameters can be calculated together with P or can be estimated by the requirement that straight lines in scenes have to be straight in images. The latter method is used with scenes which include straight line features such as buildings etc. In such a case usage of a calibration pattern is not necessary.

Details of the image formation and camera calibration techniques can be found in Hartley and Zisserman [36].

2.5.4 Thermal camera calibration

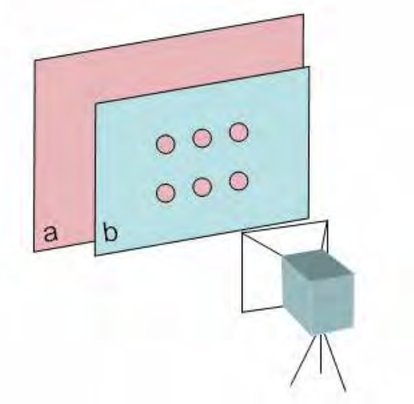

Figure 2.8: Schematic view of the procedure used to calibrate the thermal camera.

While a color camera can be calibrated using the Camera Calibration Toolbox for Matlab [9] the same piece of software cannot be directly used for calibrating a thermal camera. The required chessboard-like calibration pattern is not suitable for use with thermal cameras. The contrast between black and white rectangles which makes it easy to detect corners by visible light cameras is not present when viewing the same pattern with a thermal camera. Additionally, the effects of halos around hot or cold objects typical for this kind of camera makes it very difficult to picture sharp corners.

In order to find the focal lengths, principal point and lens distortion parameters, a custom calibration pattern and an add-on to the toolkit have been used [86]. To obtain images which look like those taken by a standard

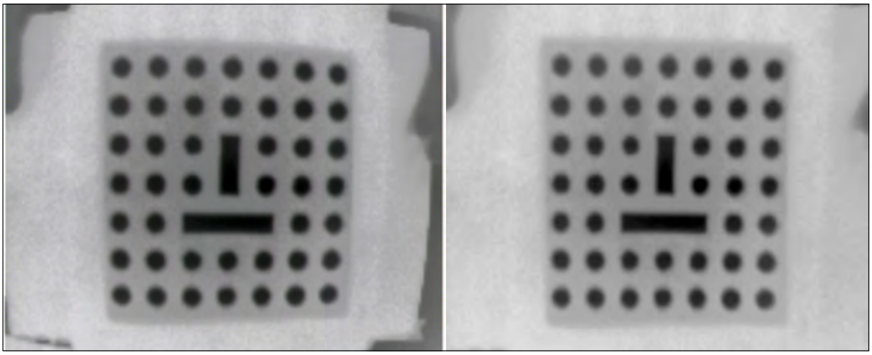

Figure 2.9: The result of image rectification based on the thermal camera parameters.

color camera a special pattern has been fabricated to allow using the calibration procedure normally used for color images. The calibration setup is schematically depicted in figure 2.8. The custom calibration pattern (b) made out of a thin material (e.g. a sheet of overhead plastic) is placed between a warmed up (or cooled down) metal plate (a) and the camera to be calibrated. Black parts of the calibration pattern (which has circular features unlike the standard chessboard pattern) are cut out from the plastic to allow the heat radiation to pass through to produce an image similar to one obtained during the calibration of a color camera.

A number of images of the calibration pattern collected at different positions of the camera are the input to the calibration algorithm. Figure 2.9 presents an example image obtained with the described technique and the result of image rectification based on the parameters obtained during the camera calibration. The radial distortion visible in the left image (a cushion effect) is clearly eliminated thanks to the correctly estimated camera parameters.

2.5.5 Image processing, color-thermal image correspondances and geolocation

The following subsections describe the steps involved in the saliency map building process. Namely, the thermal image processing, calculating pixel correspondences between color and thermal images, and the human body classifier are presented.

Thermal image processing

The algorithm takes a pair of images as input and the processing starts by analyzing the thermal image (see images in the top row of figure 2.10). The image is first thresholded to find regions of certain intensities which correspond to human body temperature. The value of image intensity corresponding to a certain temperature is usually given by the camera manufacturer or can be calibrated by the user. The shapes of the thresholded regions are analyzed and those which do not resemble a human body (i.e. wrong ratio of minor and major axes of the fitted ellipse and incorrect area) are rejected.

Once human body candidates are found in the thermal image, corresponding regions in the color image are calculated.

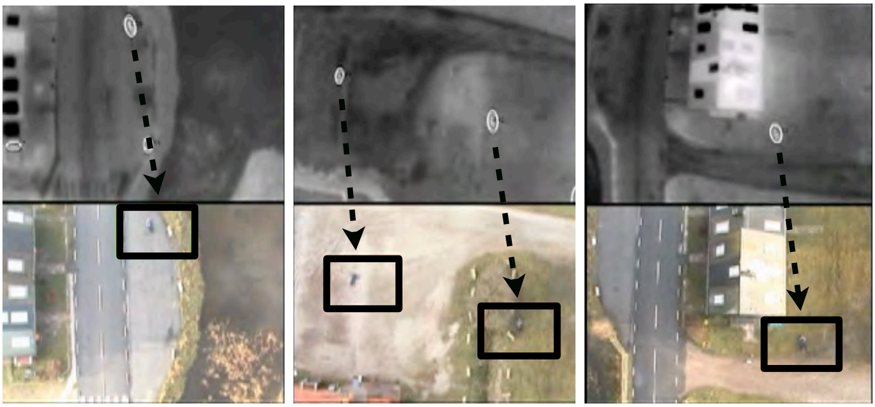

Figure 2.10: Example input to the image processing algorithm. Thermal images in the top row and the corresponding color frames in the bottom row.

Image correspondances and geolocation

Finding corresponding regions using image registration or feature matching techniques is infeasible because of mostly different appearance of features in color and thermal images. For that reason a closed form solution, which takes into account information about the camera pose in the world is preferred. Computing the corresponding region in the color image starts by calculating the coordinates of the target point T ($\vec{v}_T$) whose projection is the pixel in the thermal image $\vec{u}_t$ i.e.

$$\vec{u}_t = P_t \vec{v}_t \quad \vec{u}_t \in \mathcal{P}^2 \quad \vec{v}_t \in \mathcal{P}^3 \tag{2.9}$$

where P_t represents extrinsic and intrinsic parameters of the thermal camera. The general scheme of the problem is shown in figure 2.11. A vector line equation $\vec{g}(d)$ with the direction vector $\vec{v}_{cam}$ which goes through the camera center $\vec{v}_c$ is:

$$\vec{g}(d) = \vec{v}_C + d \cdot \vec{v}_{cam} \quad d \in \mathbb{R} \tag{2.10}$$

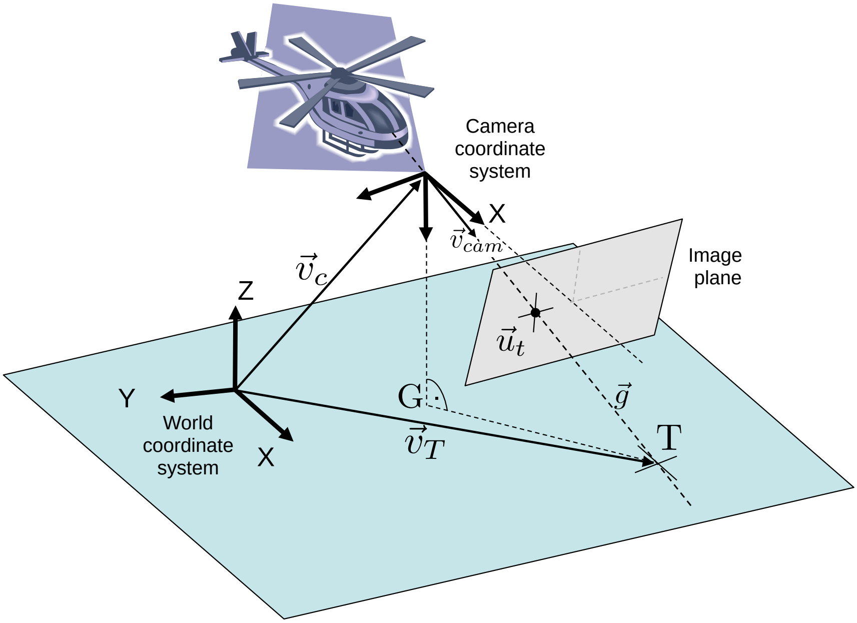

Figure 2.11: Calculating a target's coordinates.

The ground plane is defined by the point $G(\vec{v}_G)$ and the normal vector $\vec{n}$ which is the down component of the North, East, Down (NED) frame:

$$(\vec{p} - \vec{v}_G) \cdot \vec{n} = 0 \tag{2.11}$$

Finally, the vector $\vec{v}_T$ which describes the point of intersection of a ray of light going through the camera center and the target (eq. 2.10) with the ground plane (eq. 2.11) can be calculated according to:

$$\vec{v}_T = \vec{v}_C + \frac{(\vec{v}_G - \vec{v}_C) \cdot \vec{n}}{\vec{v}_{cam} \cdot \vec{n}} \cdot \vec{v}_{cam} \tag{2.12}$$

In order to calculate $\vec{v}_{cam}$ the vector along the X axis of the camera frame must be expressed in the world coordinate frame. This transformation can be expressed as:

$$\vec{v}_{cam} = P_{heli} P_{ptu} P_p \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T \tag{2.13}$$

where P_p describes the transformation depending on the undistorted pixel position $\vec{u}_t$. The matrix P_{ptu} is built to represent the transformation introduced by the pan-tilt unit. P_{heli} represents the attitude of the UAV and is built up from the roll, pitch and yaw angles of the platform.

The method presented can be extended to relax the flat world assumption given the elevation model. The point T can be found by performing ray-tracing along the line described by equation 2.10 to find the intersection with the ground elevation map.

The projection in the color image of the world position given by T can be calculated using the following formula:

$$\vec{u}_c = P_c \vec{v}_T \quad \vec{u}_c \in \mathcal{P}^2 \quad \vec{v}_T \in \mathcal{P}^3 \tag{2.14}$$

where P_c constitutes the matrix encoding intrinsic and extrinsic parameters of the color camera including the transformation describing the relating between the color and thermal cameras. This transformation can be calculated given a set of known image correspondences $u_{t_i} \leftrightarrow u'_{c_i}$ and minimizing the error between u'_{c_i} and u_{c_i} calculated using equation 2.14 for all i. This transformation is constant and is calculated once along with the camera calibration procedure.

The accuracy of the correspondence calculation is influenced by several factors. As can be seen from equation 2.13, all inaccuracies of parameters involved in calculating the position and attitude of cameras in the world contribute to the overall precision of the solution. The evaluation of the accuracy involves investigating the UAV state estimation errors, pan-tilt unit position accuracy, camera calibration errors etc. In the case of the UASTech-Lab RMAX autonomous helicopter platform, during the experimental validation, the corresponding image coordinates were within a subregion of twenty percent of the video frame size (cf. the marked subregions in color images in figure 2.10).

The human body classifier

After calculating the coordinates of the pixel in the color image, a region with the P_c as center (cf. black rectangles in the bottom row images of figure 2.10) is analyzed by an object classifier. The classifier used was first suggested in Viola and Jones [82]. It uses a cascade of classifiers for object detection. The method also includes a novel image representation, the integral image, for quick detection of features. The method was also extended, for example in Lienhart and Maydt [59] by extending the original feature set which is presented in figure 2.12A.

The classifier requires training with positive and negative examples. During the learning process the structure of a classifier is built using boosting. The use of a cascade of classifiers allows for dramatic speed up of computations by skipping negative instances and only computing features with high probability for positive classification. The speed up comes from the fact that the classifier, as it slides a window at all scales, works in stages and is applied to a region of interest until at some stage the candidate is rejected or all the stages are passed (see figure 2.12B). This way, the classifier quickly rejects subregions which most probably do not include features needed for positive classification (i.e. background processing is quickly terminated).

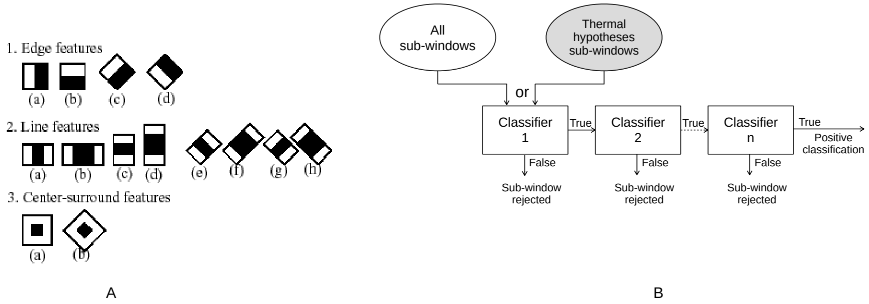

Figure 2.12: A. Leinhart's extended set of features, B. Schematic description of a cascade classifier. Classifiers in the initial stages remove a large number of negative examples. Successive classifiers process a smaller number of sub-windows. The initial set of sub-windows can include all possible sub-windows or can be the result of a previous classification in a thermal image additionally improving processing rate.

The classifier works with features which can be quickly extracted using intermediate representations - integral images. The integral image at position $ii(x, y)$ is derived from the original $i(x, y)$ as follows:

$$ii(x, y) = \sum_{x' \leq x,\, y' \leq y} i(x', y'), \tag{2.15}$$

The integral image can be calculated with one pass over the original image using this pair of equations:

$$s(x, y) = s(x, y - 1) + i(x, y), \tag{2.16}$$

$$ii(x, y) = s(x - 1, y) + s(x, y), \tag{2.17}$$

where the $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$ and $ii(-1, y) = 0$.

The reason for working with features instead of pixel intensities is that features encode knowledge about the domain, which is difficult to learn from raw input data. The features encode the existence of oriented contrasts between regions of an image. The Haar-like features used here can be calculated at any position and any scale in constant time using only eight look-ups in the integral image.

The implementation of the classifier used in this work is a part of the Open Source Computer Vision Library[5] and the trained classifier for upper, lower and full human body is a result of Kruppa et al. [55]. The classifier is best suited for pedestrian detection in frontal and backside views which is

[5] Homepage: `http://opencv.willowgarage.com/wiki/`

exactly the type of views a UA has when flying above the bodies lying on the ground.

The classifier parameters have been adjusted to minimize false negative cases. In case of rescue operations it is better to find more false positives then missing potential victims. The number of neighboring rectangles needed for successful identification has been set to 1 which makes the classifier accept very weak classifications. The factor by which the search window is scaled between the subsequent scans has been set to 1.2 meaning that the search window is increased by 20%.

2.5.6 Map building

Since the body classifier is configured to be "relaxed" it delivers sporadic false positive classifications. To deal with most of them the following method is used to prune the results. Every salient point in the map has two parameters which are used to calculate the certainty of a location being a human body: T_{frame} which describes the amount of time a certain location was in the camera view and T_{body} which describes the amount of time a certain location was classified as a human body. The certainty factor is calculated as follows:

$$p_{body}(loc_i) = \frac{T_{body}}{T_{frame}} \tag{2.18}$$

A location is considered a body if $p_{body}(loc_i)$ is larger than a certain threshold (e.g. 0.5 during the flight tests) and T_{frame} is larger than a desired minimal observation time. Locations are considered equal if geographical distance between them is smaller than a certain threshold (depending on the geolocation accuracy) and the final value of a geolocated position is an average of the observations.

2.6 Experimental validation

The presented technique for geolocation and saliency map building has been integrated into the UASTechLab helicopter system and tested in flight tests. The following subsections present an example mission setup and the results obtained.

2.6.1 Mission setup

An example complete push button mission setup was as follows:

- The flights were performed at the Swedish Rescue Services Agency Test Area (figure 2.13A). This is a city-like closed area with road structures, buildings etc. so that the video streams include different types of textures e.g. grass, asphalt, gravel, water and building rooftops.

- Two RMAX helicopters were used starting from H_1 and H_2 in figure 2.13B.
- A user of the system selected a rectangular area on a map where the saliency map was to be built (figure 2.13B).
- The on-board systems calculated the mission plan which took into account properties of the on-board sensors (e.g. field of view) of both UAVs. The plan consisted of two separate flight plans for the two helicopters.
- The mission started with simultaneous takeoffs and flying to starting positions S_1 and S_2 in figure 2.13B.
- After arriving at the starting positions the execution of the scanning paths autonomously began.
- Upon finishing the scanning paths at positions E_1 and E_2, the UA's flew to the takeoff positions and performed autonomous landings.

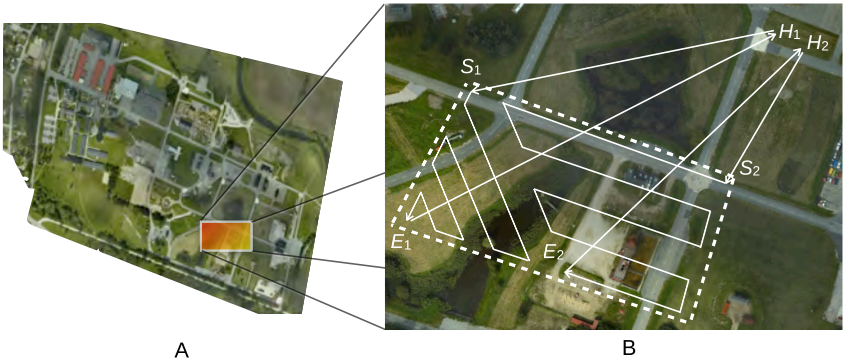

Figure 2.13: A. Map of the Swedish Rescue Services Agency Test Area in Revinge, B. A closeup view of the area where the saliency map was built. Approximate flight paths are marked with solid lines.

The following subsection describes results of the presented algorithm executed on the data collected by the platform starting in position H_2.

2.6.2 Experimental results

The algorithm found and geolocated all eleven human bodies placed in the area. The images of identified objects are presented in figure 2.14. Color and the corresponding thermal images are displayed vertically as pairs. Images 7, 9 and 14 present three falsely identified objects. Erroneous classifications

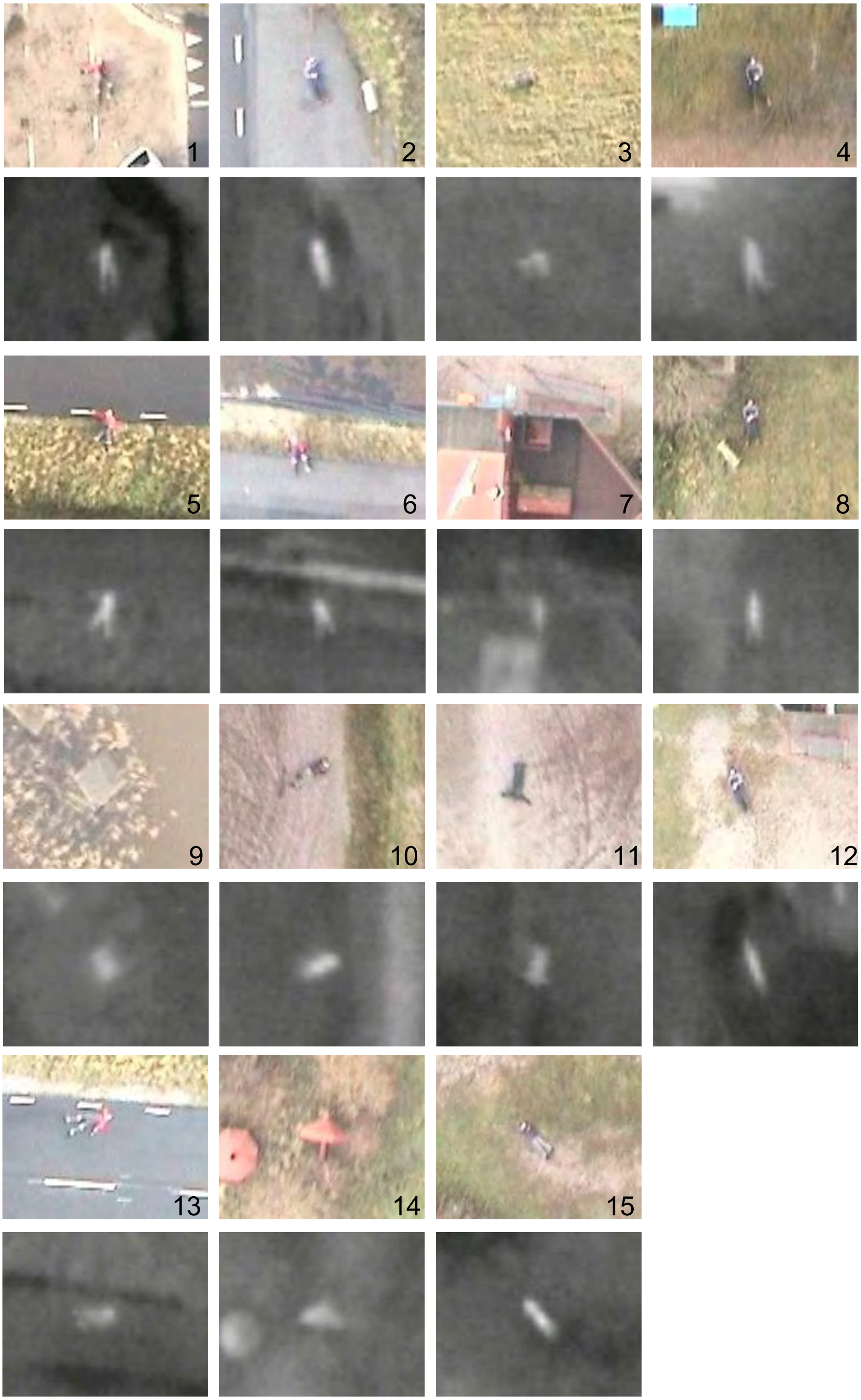

Figure 2.14: Images of classified bodies. Corresponding thermal images are placed under color images.

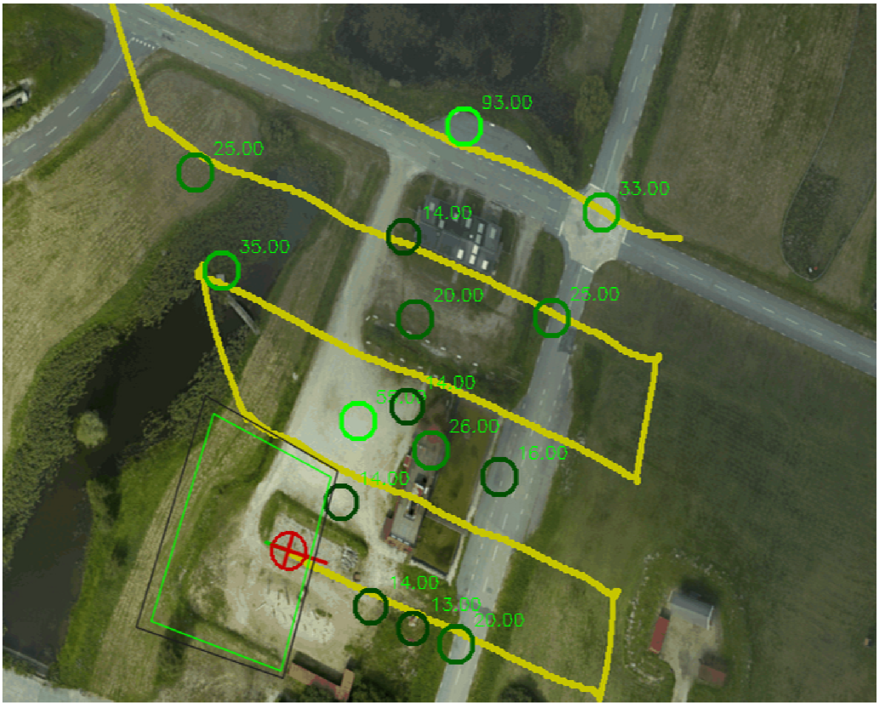

Figure 2.15: The resulting map with salient points marked as circles. The lighter the shade of the color the higher the detection certainty.

were caused by configuring the human body classifier to accept weak classifications. A more restrictive setup could instead result in missing potential victims. The images could additionally be judged by a human operator to filter out the false-positive classifications. Both human bodies and dummies were detected despite the lower temperature of the latter.

Figure 2.15 presents the generated saliency map. The UAV trail is marked with a solid line. It presents the scanning pattern segments flown by the helicopter. It clearly covers one part of the designated search area as presented in figure 2.13B. The UAV ending position is marked with a cross icon. Additionally, fields of view of color and thermal cameras are depicted with light-grey and black rectangles, respectively. They differ slightly as the two cameras have different properties as identified during calibration procedures.

The circles depict the identified body positions. The lighter the shade of the color, the more certain the classification. As can be seen the most certain body positions are objects number 2 and 11 (in figure 2.14). It is due to the fact that these body images are clearly distinguishable from the homogeneous background. Nevertheless, even body images with more cluttered backgrounds were identified.

The accuracy of the body geolocation calculation was estimated by mea-

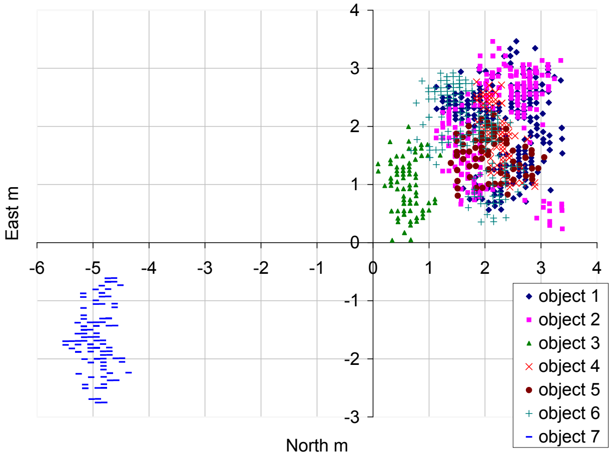

Figure 2.16: Geolocation error for multiple objects.

suring GPS (without differential correction) positions of bodies after an experimental flight. Figure 2.16 presents the error measurement for seven geolocated objects.

The measurement has a bias of approximately two meters in both east and north directions. It is the sum of errors in GPS measurement, accuracy of the camera platform mounting, PTU measurement and camera calibration inaccuracies. The spread of measurement samples of approximately 2.5 meters in both east and north directions is a sum of errors of the UAV's attitude measurement, the system of springs in the camera platform and time differences between UAV state estimate, PTU angle measurement and image processing result acquisition. A detailed analysis is required to accurately measure error contributing factors and improve the precision.

Nevertheless, the current accuracy of the system is sufficient for assessing a victim's position within a 3 meter radius. The large geolocation error of object 7 is most likely caused by an erroneous GPS measurement. Object 7 was located on a metal foot-bridge and the GPS antenna during static measurement was additionally partially occluded by metal railings.The noise of the measurement however is consistent with the rest of the objects.

The presented algorithm requires only a single pair of images for human body classification. In practice, however, the more pairs available the more certain the result of a classification can become (cf. equation 2.18). Additionally, thanks to using the results of the thermal image analysis to focus the classification in the color image subregions, a high rate of processing is achieved (e.g. above 20Hz for the presented results).

2.7 Complete mission

The following subsections briefly present how the generated saliency map can be integrated into a complete autonomous mission. For the full description see Doherty and Rudol [20].

2.7.1 Package delivery

After successful completion of Part I of the scenario, a saliency map with geolocated positions of the injured people has been generated. In the next phase of the mission, the goal is to deliver medical, food and water supplies to the injured. In order to achieve this part of the mission, one would require a task planner to plan for logistics, a motion planner to plan the paths for one or more UAVs to supply and delivery points and an execution monitor to monitor the execution of highly complex plan operators. Each of these functionalities would also have to be tightly integrated in the system. These components are described in subsection 2.7.2.

This part of the mission is developed primarily in simulation with hardware-in-the-loop. The on-board computer system which includes a simulator of the UAV dynamics executes all functionalities necessary for completing the mission on the actual flight tests hardware. Additionally, a physical prototype of a mechanism for carrying and releasing packages has been developed and tested. Figure 2.17 presents two images of the prototype system.

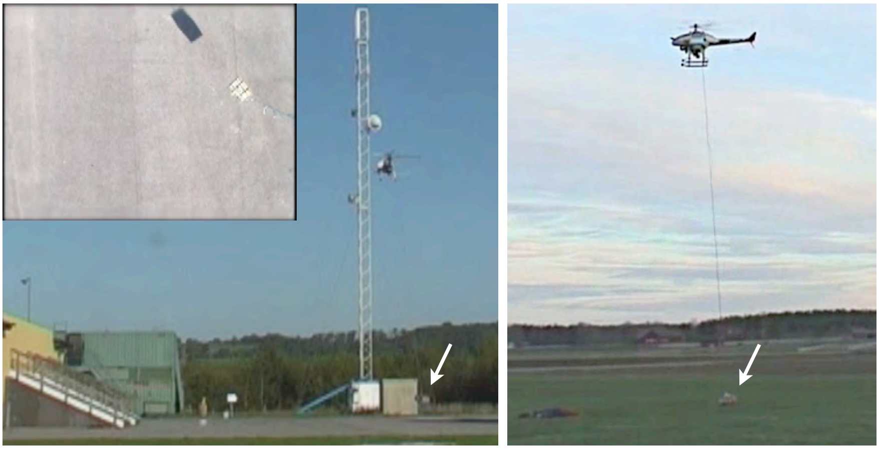

Figure 2.17: Two frames of video presenting the prototype mechanism for carrying and releasing packages using an electromagnet. An arrow points to a package being carried. The top left picture presents the on-board camera view.

For these logistics missions, the use of one or more UAVs with diverse roles and capabilities is assumed. Additionally, n injured body locations, several supply depots and several supply carrier depots are assumed to be given (see figure 2.18).

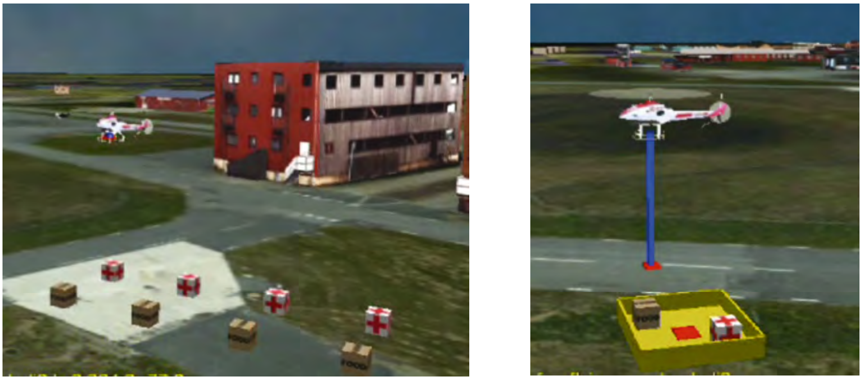

Figure 2.18: A supply depot (left) and a carrier depot (right).

2.7.2 Planning, execution and monitoring

Figure 2.19 shows part of the UAV architecture, with an emphasis on those components that are the most relevant for planning, execution and execution monitoring.

At the top of the center column is the plan executor which given a mission request, calls DyKnow [39, 40], a knowledge processing middleware, to acquire essential information about the current contextual state of the world or the UAV's own internal states. Together with a domain specification and a goal specification related to the logistics scenario, this information is fed to TALplanner [19], a logic-based task planner which outputs a plan that will achieve the designated goals, under the assumption that all actions succeed and no failures occur. Such a plan can also be automatically annotated with global and/or operator-specific conditions to be monitored during execution of the plan by an execution monitor in order to relax the assumption that no failures can occur. Such conditions are expressed as temporal logical formulas and evaluated on-line using formula progression techniques. This execution monitor notifies the plan executor when actions do not achieve their desired results and one can then move into a plan repair phase. The plan executor translates operators in the high-level plan returned by TALplanner into lower level command sequences which are given to the command executor. The command executor is responsible for controlling the UAV, either by directly calling the functionality exposed by its lowest level Flight Command Language (FCL) interface or by using task procedures (TPs) through the TP Executor subsystem. The TP Executor is part of the Modular Task Architecture (MTA) [22], which is a reactive system designed in the procedure-based paradigm and developed for loosely coupled heterogeneous systems. A task

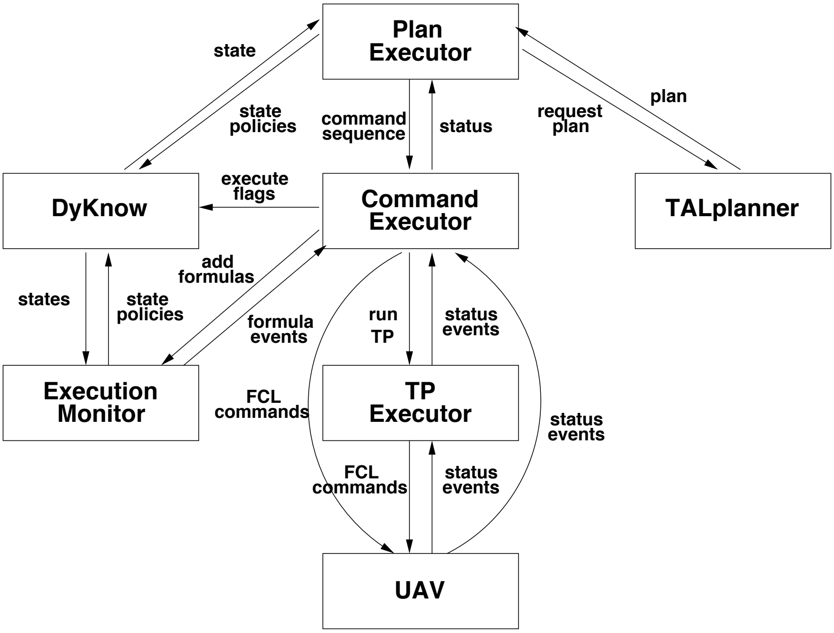

Figure 2.19: System Architecture Overview

is a behavior intended to achieve a goal in a limited set of circumstances. A task procedure is the computational mechanism that achieves this behavior. The TPs have the ability to use deliberative services, such as the task planner described above or motion planners [89, 90], in a reactive or contingent manner and to use traditional control services in a reactive or contingent manner and thereby integrate deliberation and reaction.

During plan execution, the command executor adds formulas to be monitored to the execution monitor. DyKnow continuously sends information about the development of the world in terms of state sequences to the monitor, which uses a progression algorithm to partially evaluate monitor formulas. If a violation is detected, this is immediately signaled as an event to the command executor, which can suspend the execution of the current plan, invoke an emergency brake command, optionally execute an initial recovery action and finally signal new status to the plan executor. The plan executor is then responsible for completing the recovery procedure.

The fully integrated system is implemented on the UAVs and can be used on-board for different configurations of the logistics mission described in Part II of the larger mission. The simulated environments used are in urban areas and quite complex. Plans are generated in the millisecond to seconds range using TALplanner and empirical testing shows that this approach is promising in terms of integrating high-level deliberative capability with lower-level reactive and control functionality.

2.8 A vehicle tracking and geolocation application

Autonomous vehicle tracking and geolocation can play a major role in applications such as traffic behavior analysis which can contribute to increasing road safety. Instead of a human operator trying to maintain situational awareness about a traffic condition in an area, a UAS equipped with appropriate sensors and algorithms can be used. One approach is for a UAS to relay video streams and other data to an operator for human inspection. Another, more scalable approach, is for a UAS to monitor the traffic situations which arise and only report back the high level events observed, such as cars turning in intersections or performing overtakes.

The process of going from images to detection of high-level events can be divided into two parts. First, an image processing algorithm detects and tracks an object in raw video, geolocates the tracked object and produces a stream of world positions. Second, an algorithm which performs temporal and spacial analysis of the data and detects certain types of behaviors (e.g. entering crossings and overtakes) and maintains situational awareness.

This section describes how the method developed in the previous section (specifically the first and second subsections of 2.5.5) can be used to implement the first part of the task. The second part of the task is described in section 2.8.2.

2.8.1 Image processing

The task of image processing is to calculate world coordinates of vehicles tracked in video sequences. First, an object tracker is used to find pixel coordinates of the vehicle of interest based on color and thermal input images. Second, the geographical location of the object is calculated and expressed as world coordinates. The object tracker developed for the purpose of this work can be initialized automatically or manually. The automatic mode chooses the warmest object on a road segment (description of the road system is fetched from a GIS database) within the thermal camera view and within a certain distance from the UAV (cf. the process of calculating the distance to a tracked object described in section 2.5.5). The area around the initial point is checked for homogeneity in thermal and color images. The object is used to initialize the tracker if its area is consistent with the size of a car signature. This method of initialization works with satisfactory results for distances up to around 50m from the tracked object. If the tracker is initialized incorrectly the user can choose an object of interest manually by clicking on a frame of the color or thermal video. The corresponding pixel position (for color and thermal images) is calculated based on the parameters of the cameras, the UAV's position and attitude and the model of the ground elevation.

After initialization, the tracking of an object is performed independently in the color and thermal video streams. Tracking in the thermal image is achieved by finding the extreme value (warmest or coldest spots) within a small (5 percent of the image size) window around the previous result. Object tracking in color video sequences is also performed within such a small window and is done by finding the center of mass of a color blob in the HSI (Hue, Saturation, Intensity) color space. The thresholding parameters are continuously updated to compensate for illumination changes. Tracking in both images is performed at full frame rate (i.e. 25Hz) which allows for compensating for moderate illumination changes and moderate speeds of relative motion between the UAV and the tracked object. If either of the trackers does not find a similar blob in two consecutive frames, the tracking is considered to be lost. The problem of automatic reinitialization as well as more sophisticated interplay between both trackers, is not addressed in this work.

Finding pixel correspondences between the two cameras is difficult to achieved by feature matching commonly used in stereo-vision algorithms since objects generally appear different in color and infrared images. Because of this, the distance to an object whose projection lies in a given pixel must be determined. Given the camera parameters, helicopter pose and the ground elevation model the distance to an object can be calculated (see the second subsection of 2.5.5). It is the distance from the camera center to the intersection between the ground model and the ray going through the pixel belonging to the object of interest. For the environment in which the flight tests were performed the error introduced by the flat world assumption (i.e. ground elevation model simplified to a plane) is negligible. Finally, calculating pixel correspondences between the two cameras can be achieved by performing pixel geolocation using the intrinsic and extrinsic parameters of one of the cameras followed by applying a inverse procedure (i.e. projection of geographical location) using the other camera's parameters.

Using the described object tracker, several data series of world coordinates of tracked vehicles were generated. Two kinds of video sequences were used as data sources. In the first kind (figure 2.20A) the UAV is stationary at altitudes of 50 or 60 meters and observes two vehicles as they drive on a nearby road. In the other kind (figure 2.20B) both the car and the UAV are moving. The ground vehicle drives several hundreds meters on the road system passing through two crossings and the UAV follows the car at altitudes from 25 to 50 meters. For sequences containing two cars, the tracker was executed twice to track both vehicles independently.

A precise measure of the error of the computed world location of the tracked object is not known because the true location of the cars was not recorded during the flight tests. The accuracy of the computation is influ-

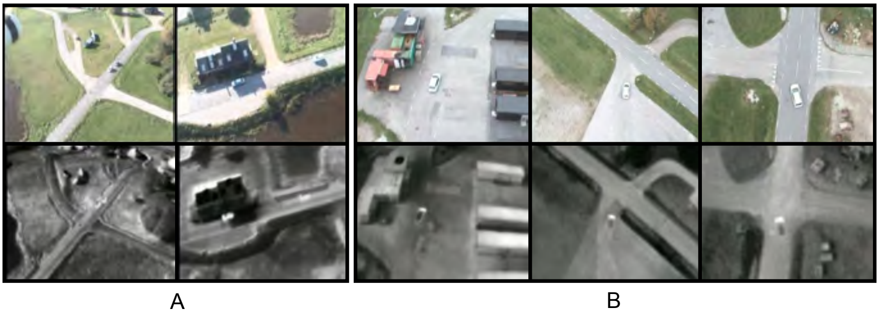

Figure 2.20: A. Two frames from video sequence with the UAV hovering close to a road segment observing two cars performing overtaking maneuver. B. Three frames from video sequence with the UAV following a driving car passing road crossings. Top row contains color images and bottom row contains corresponding thermal images.

enced by several factors, such as error in the UAV position and the springs in the camera platform suspension, but the tracker in general delivers world coordinates with enough accuracy to determine which side of the road a car is driving on. Thus the maximum error can be estimated to be below 4-5 meters for distances to the object of around 80 meters (which is consistent with the accuracy measurement presented in section 2.6.2).

Figures 2.21 and 2.22 present two results of the vehicle tracking. The circle represents the UAV position in the relation to the segmented road system. The tracking trails (red) clearly fall within the boundaries of the road structure.

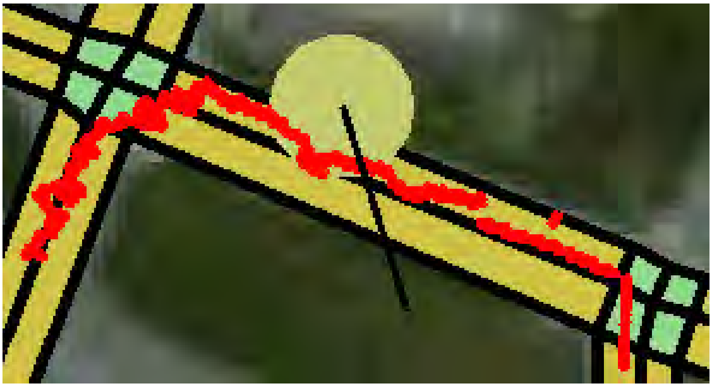

Figure 2.21: Intersection overview.

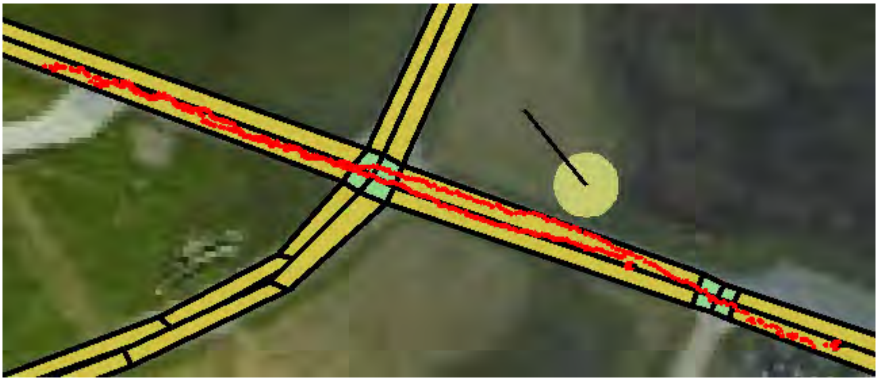

Figure 2.22: Overtake overview.

2.8.2 High level traffic situation awareness

To achieve complex missions an autonomous UAV operating in dynamic environments must create and maintain situational awareness. This can be achieved by continually gathering information from many sources, selecting the relevant information for the current task, and deriving models about the environment and the UAV itself. Often models close to the sensor data, suitable for traditional control, are not sufficient for deliberation. The need for more abstract models creates a sense-reasoning gap. This section briefly presents how DyKnow, a knowledge processing middleware ([39, 40]), can bridge the gap in a concrete UAV traffic monitoring application. Streams of UAV positions generated as described in previous section are used to construct and maintain qualitative object structures modeling the parts of the environment necessary to recognize the traffic behavior of the tracked vehicles in real-time.

In a traffic monitoring scenario a human operator tries to maintain situational awareness about a traffic situation in an urban area using UAVs looking for accidents, reckless driving, or other relevant activities. One approach to this scenario would be for one or more UAVs to relay videos and other data to the operator for human inspection. Another, more scalable approach, would be for the UAVs to monitor the traffic situations which arise and only report back the high level events observed, such as cars turning in intersections and doing overtakes, to reduce the amount of information and help the operator focus her attention. This section describes such a traffic monitoring application where cars are tracked by a UAV platform and streams of observations are fused with a model of the road system in order to draw conclusions about the behavior of the cars in the environment. The input consists of images taken by the color and thermal cameras on the UAV which are fused and geolocated to a single world position. This stream of positions is then correlated with a geographical information system in order to know where in a road system the object is located. Based on this information high level behaviors such as turning in intersections and overtaking are recognized in real-time as they develop using a chronicle recognition system.

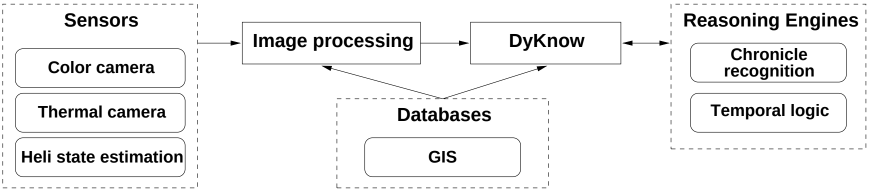

Figure 2.23: Overview of the components of the traffic monitoring application.

An overview of the components of the traffic monitoring application is shown in figure 2.23. The three sensors used, the two cameras and the heli state estimation (which is fused from INS and GPS data), are shown to the left. These provide the primitive information about the environment and the UAV. The next component is the image processing system which tracks objects seen by the cameras. When an object is being tracked the images from the two cameras are fused to provide an estimation of the position in the world of the tracked object. Each time a new frame is analyzed a new position estimate is produced. From this stream of position estimations, the system recognizes high level events such as turning in intersections, overtaking, and so on.

To describe these events a formal representation called *chronicles* is used [28]. A chronicle defines a class of events by a simple temporal network (STN) [16] where the nodes are primitive events and the edges are temporal constraints between the occurrences of the primitive events. If a stream of primitive events contains a set of event occurrences which satisfies all the constraints then an instance of the chronicle is recognized. The chronicles used in this application contain primitive events which capture the structure of the road network, qualitative information about cars such as which road segment they are on, and qualitative spatial relations between cars such as beside and behind. If all occurrences of the primitive events are in the stream used by the chronicle recognition engine then the recognition is complete, meaning that all chronicle instances in the stream will be recognized. Creating this stream of events, accurately representing the environment of the UAV, based on sensor data is a concrete instance of bridging the sense reasoning gap. This is achieved using DyKnow.

DyKnow takes the stream of position observations provided by the image processing system and derives an event stream representation of cars and qualitative spatial relations between cars. DyKnow also derives an event stream representation of the road network from the information stored in the GIS. One issue that must be handled is how to anchor the car symbols used in the chronicles to objects being tracked. Since the image processing system may lose track of cars or start tracking other objects than cars DyKnow has to dynamically estimate and continually monitor the type and identity of objects being tracked. To do this, the normative behavior of different objects and the conditions for assuming that two objects have the same identity are described using temporal logic. When a tracked object is found which satisfies the normative behavior for e.g. a car a new car representation is created and the tracked object is *linked* to the new car representation. From this moment the car representation will be updated each time the tracked object is updated. Since links only represent hypotheses, they are always subject to becoming invalid given additional observations, and therefore the UAV continually has to verify the validity of the links. This is done by monitoring that the normative behavior of the assumed object type is not violated. For example, an object assumed to be a car must not violate the normative constraints on cars, e.g. leaving the road. If it does violate the corresponding link is removed, in other words the object is no longer assumed to be a car. To evaluate temporal logical formulas DyKnow has to derive temporal models representing the value of the variables used in the formulas. Since these models are derived from the sensor data it is another concrete example of how DyKnow can be used to bridge the sense reasoning gap.

Further details are provided in Heintz et al. [41, 42].

2.9 Conclusion

This chapter presented a method for fusing thermal and color video streams for the purpose of increasing the level of autonomy through improving situational awareness for unmanned aircraft systems. Two methods have been described which rely on color and thermal video streams as the main source of information. The first method deals with the problem of building saliency maps where human body locations are marked as points of interest. Such maps can be used in emergency situations to help first responders to quickly focus the help effort in places where it is most needed. The task of acquiring the first overview of the situation can be delegated to autonomously operating UAS's.

The second presented functionality deals with the problem of tracking and geolocation of vehicles. The obtained streams of vehicle positions can be used by a reasoning system to analyze traffic behaviors such as reckless overtakes, speeding and entering crossings. Thanks to an automatic analysis of the traffic situation, the load of an operator can be greatly reduced and the situational awareness improved.

Chapter 3

Towards autonomous indoor navigation of small-scale UAVs

3.1 Introduction and motivation

Enabling autonomous indoor flight and navigation allows a wide range of tasks to be carried out by UAS's with minimal or without operator involvement. A concrete example where the use of indoor flying UAVs would be beneficial is the recent earthquake and the resulting tsunami in Japan. Because of the great devastation caused by the tsunami wave and the radiation caused by a reactor malfunction, the immediate surroundings of the reactor became very dangerous for humans. In such a situation the use of UAS's from a safe distance would help to protect the well-being of the rescuers by allowing them to explore and inspect the inside of the reactor building before eventually sending in humans. This is a typical example of a dangerous task where the use of robots helps saving or protecting human lives.

The progress in the field of autonomous unmanned aircraft has allowed for the use of UAS's in many applications in military, academic and commercial domains. Thanks to advances in several fields, the range of missions that unmanned aerial vehicles can perform today is substantial. The evolution of sensor technologies, state estimation techniques, mission and path planning methods, software frameworks, to name a few, made this progress possible. The number UAS's operating outdoors and the kinds of tasks being performed show a high level of maturity.

Autonomous indoor flight and navigation, on the other hand, are still in their infancy. Even though steady progress is being made, the range of tasks performed autonomously by indoor UAS's is small compared to their outdoor counterparts. There are several reasons for this. Compared to outdoor operating UAS's, the solutions to the indoor flight and navigation problems have to deal with the following issues:

- *Lack of a ubiquitous positioning system.* Even though the GPS is not always reliable it was in fact the enabling technology for outdoor autonomous navigation. Indoors, however, the GPS cannot be used at all.

- *Limited payload* of small airframes suitable for indoor use limits the types and the number of appropriate sensors. Certain types of sensors are not available at all in small scale (e.g. radar) and a platform can often only carry a single sensor of a certain kind.

- *Lower sensor accuracy and/or range* due to miniaturization. For example, inertial navigation systems based on Micro Electro Mechanical Systems (MEMS) sensors drift much faster than systems based on larger Ring Laser Gyroscopes (RLG) used in full size aircraft such as the Airbus A320; miniature cameras often have very simple lightweight plastic lenses instead of professional glass lenses which influences the image quality. Additionally, sensors such as Laser Range Finders (LRFs) have shorter measurement ranges. Even though distances to obstacles in indoor environment are normally smaller compared to outdoor environments, this factor cannot be neglected.

- *Limited computational power* and lack of dedicated hardware components such as Floating Point Units (FPUs) impose restrictions on algorithms. Many of the existing algorithms are inadequate for small platforms because the computational requirements prevent them from delivering timely results.

- *Restricted power* due to limited payload available for the power source. The battery power which is used both for propulsion and avionics systems has to be used efficiently to maximize the flight time. Achieving autonomous flight only for a very short time span is not desirable because it limits the types of missions a UAS can perform.

- *Properties of the indoor environment* introduce several challenges. First, an indoor environment imposes strict requirements on timely and accurate state estimation in relation to obstacles. While in outdoor environments there exists, in the general case, a possibility to "escape" to a safer location (e.g. by increasing the altitude to fly above obstacles) this option is normally not available indoors.

  Additionally, the field of view of sensors has a more "local" character. Outdoors it is often possible to get a *bird's eye view* of the environment. Such a view from the top often allows for identifying landmarks to aid the localization task especially in case of the so-called kidnapped localization problem (i.e. when there is no initial knowledge about the robot location). In such a case a more global view allows for quicker location determination. One solution to this problem for indoor UAVs is the use of wide field of view camera lenses, but this makes the effective resolution of the cameras smaller which negatively influences the accuracy of the sensor data. Another solution could be using a camera on a tilting unit but such a device adds mechanical complexity, weight and consumes electrical power.

All of these issues need to be taken into consideration and addressed to allow indoor UAVs to perform the same wide range of missions as outdoor UA's perform today.

The key problem which needs to be addressed is localization. This means estimating at least three translational and three rotational Degrees of Freedom (DOF) of a vehicle in its environment. This process is also referred to as state or pose estimation. In the case of UAS missions performed in unknown environments, the localization has to be performed along with mapping. This is achieved by means of Simultaneous Localization and Mapping (SLAM) techniques. However, due to the above listed issues, mainly the lower sensor accuracy and lower computational power, both efficient and robust solutions are required. In case of larger airframes, with larger payload capabilities, the available computational power can compensate for sensor inaccuracies or redundant sensor configurations can be used.

The timing requirements of the indoor state estimation solutions depend on the inherent aerodynamical stability of the type of UAV airframes used. Airships or dirigibles, having slow dynamics, allow for longer periods of time when the state is not estimated. This kind of airframe is the most inherently stable and has the least tendency to drift around and possibly collide with obstacles in the environment. Rotor-based platforms in different configurations (e.g. tail-, coaxial-, quad-) require more timely state estimations in order to navigate safely. For this kind of airframe it is especially important to design efficient algorithms which allow control inputs to be computed every few milliseconds.

Like the timing aspect, the accuracy of the state estimation is also very important since distances to obstacles in indoor environments are normally small. The accuracy of the sensors that can be used on a small scale airframe is relatively low and therefore requires optimal usage of available information through sensor fusion techniques, which in turn have to be computationally efficient to be executed on the on-board hardware.

Even though it seems like many solutions could be transferred from the well developed field of indoor navigation of ground robots, indoor localization of UAVs suffers from the major drawback of lacking a direct measure of travelled distance (i.e. odometry). Even though odometers based on wheel encoders are not perfect, obtaining odometry information of equal fidelity on an airborne vehicle is a much more complex task. Instead of reading information delivered by simple wheel encoders, solutions utilizing integration of the Inertial Measurement Units (IMU) or optic flow are required for UAs.

This makes the adaptation and use of many techniques developed for ground robots challenging.

This chapter presents two approaches that enable indoor flight and navigation of small-scale UAS's. The first one is based on cooperating with a ground robot to facilitate the indoor localization of a UAV. The role of the ground robot in this setting is to provide the UAV with location information. This kind of cooperation allows for taking advantage of the well developed techniques for ground robot localization together with the ability of a flying robot to position its sensors in a way normally not available to a ground robotic system. For example, as can be observed in figure 3.1, the UAV can provide a view of what is behind an obstacle. Such a view would not be accessible to the ground robot on its own.

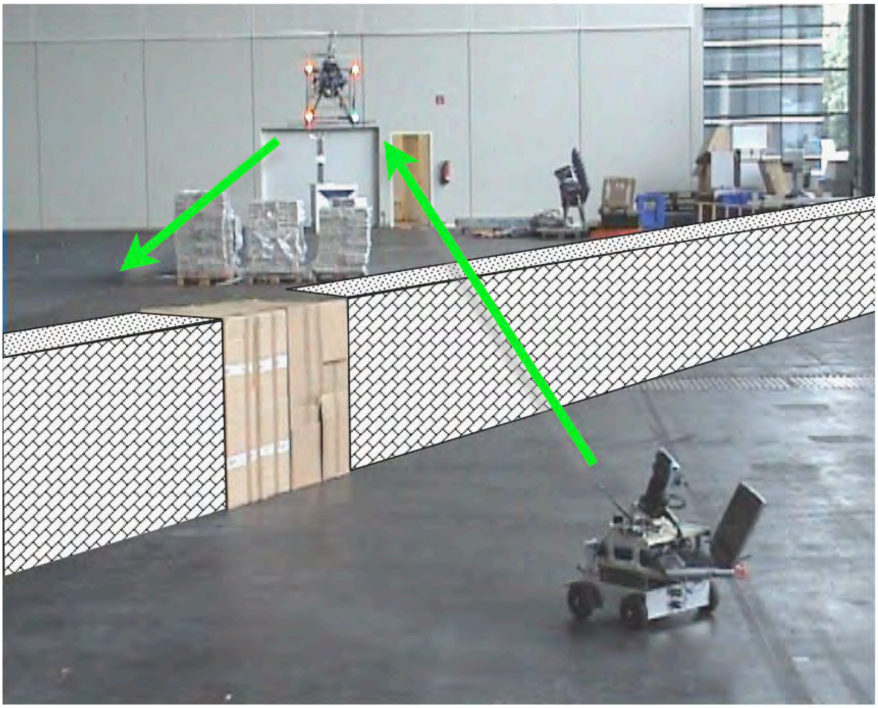

Figure 3.1: The indoor navigation though cooperation. The collaborative system has a greater sensor range and better reach than any of its components.

The second approach is computationally contained on-board a UAV and takes advantage of fusing a vision-based state estimate with inertial data to provide localization information. The method utilizes a marker-based camera pose estimation, often used in augmented reality applications to replace the information normally provided by a GPS and a compass sensor. Thanks to the possibility of using many markers, the indoor flight can be performed in a larger area and the solution can be considered a low-cost indoor localization system which can rapidly enable indoor flight.

Outline

The remainder of this chapter is structured as follows. First, the different platforms and hardware used during validation of the proposed indoor navigation techniques are presented in section 3.2, followed by a description of the related work in section 3.3. The method for cooperative indoor exploration performed by a UAV and a ground robot is presented in section 3.4. Finally, a description of the method for marker-based visual state estimation is presented in section 3.5.

3.2 Experimental platforms

Several hardware platforms and components were used in the process of developing and validating the two indoor UAV navigation solutions. This section describes the micro air vehicles, wheeled ground robots and computer systems used for the validation. First, the platforms involved in the validation of the collaborative navigation (i.e. the LinkMAV and two ground robots) are presented. Second, the hardware elements involved in the marker-based visual navigation approach are described including a high accuracy reference system used for validating the accuracy of the method.

3.2.1 Cooperative navigation

Figure 3.8 presents the schematic view of the system used for experimental validation of the system. The main components of the system are presented in the remainder of this section.

LinkMAV

The LinkMAV, presented in figure 3.3B, is an in-house developed, double-rotor coaxial helicopter platform. This kind of configuration does not require a tail rotor commonly used in standard helicopter designs. The two rotor disks counter each others' torques and the heading control is achieved by differentiating the rotational velocity of the rotors. The LinkMAV weighs around 500 grams without payload, its largest dimension is approximately 500 millimeters and it can stay in the air for around 20 minutes [23].

In 2005 and 2007 the platform took part in the US-European Micro Air Vehicle Competition (MAV05[1] and MAV07[2]) winning the best rotorcraft award and scoring the third place in the indoor competition, respectively.

[1] MAV05: `http://aeromav.free.fr/MAV05/aeromav/aero-mav-garmisch-de.htm`

[2] MAV07: `http://www.recherche.enac.fr/MAV07`

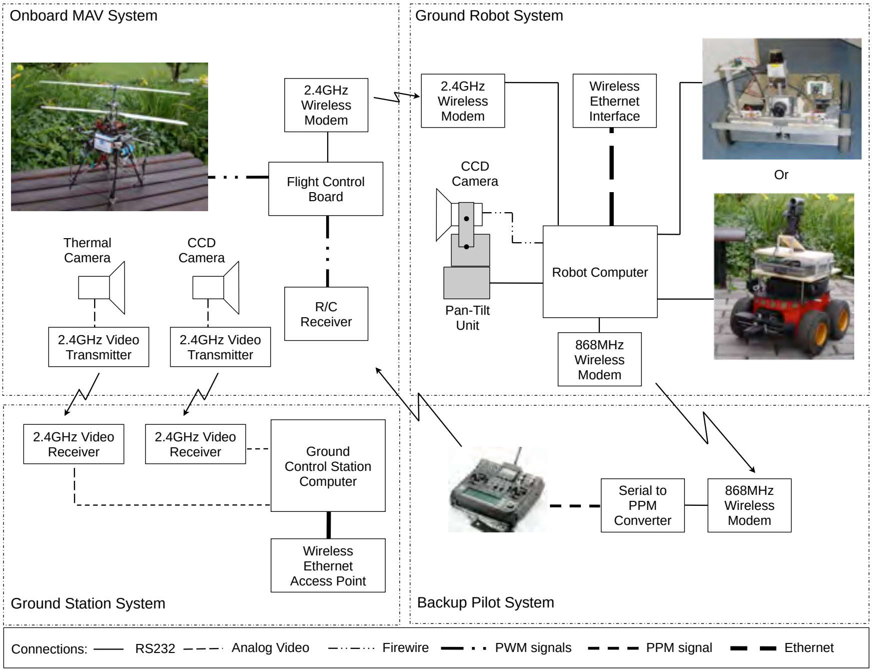

Figure 3.2: The cooperative exploration experimental system components and interconnections between them.

The LinkMAV is equipped with two lightweight cameras: a thermal camera and a color CCD. The thermal camera is a Thermal-Eye 3600AS from L-3 Communications[3] which delivers a PAL resolution analogue video stream. The CCD camera is a miniature high resolution Panasonic KX141 from Black Widow A/V[4]. Both video signals are sent to the ground station using 2.4GHz transmitters and receivers from Black Widow A/V for the ground operator view or on-ground image processing purposes.

Micropilot flight control board

During the experiments, the LinkMAV used a MicroPilot 2028g flight control board[5] for attitude stabilization. All control inputs (i.e. roll, pitch, yaw and thrust) to the MicroPilot board are in the form of Pulse Width Modulation (PWM) signals provided by the onboard R/C receiver.

[3] Homepage: http://www.l-3com.com
[4] Homepage: http://www.blackwidowav.com
[5] Homepage: http://www.micropilot.com

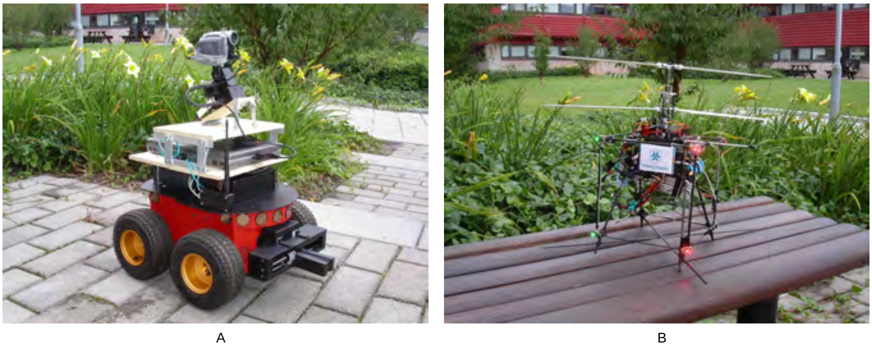

Figure 3.3: A. The Pioneer 3 AT ground robot, B. The LinkMAV.

Ground robots

Two ground vehicles were used for experimentation: a Pioneer 3 AT outdoor robot from MobileRobots Inc.[6], presented in figure 3.3A and a Zerg platform developed at the Freiburg University [54]. A 1.6GHz Pentium Mobile laptop was used to control both robots including the pan-tilt unit (DirectedPerception 46-17.5W[7]) and to host image processing algorithms. A Sony DCR-HC90E[8] video camera was connected to the laptop through a Firewire interface.

Ground station system

The Ground Station (GS) subsystem receives both thermal and color CCD analog video signals which can be used by the GS computer for image processing purposes or ground operator view. The Graphical User Interface (GUI) provides means for controlling the ground robot and the LinkMAV through a wireless Ethernet connection.

3.2.2 Marker-based visual indoor navigation

Two quadrotor UAV platforms were used for the validation of the marker-based positioning system. The Hummingbird-based system as well as the LinkQuad platform are presented in the following subsections.

Hummingbird UAV platform.

The Hummingbird quadrotor UAV (see Figure 3.4A) from Ascending Technologies GmbH [34] has been used as one of the test platforms. The Hummingbird can carry up to 200 grams of payload and has 20 minutes of

[6] Homepage: `http://www.activrobots.com`
[7] Homepage: `http://www.dperception.com`
[8] Homepage: `http://pro.sony.com`

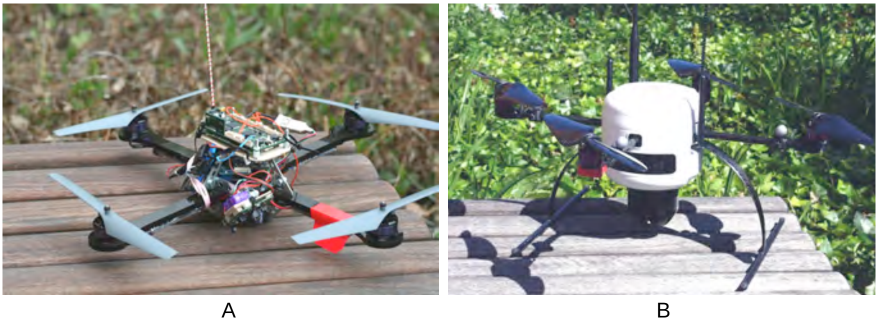

Figure 3.4: The Hummingbird quadrotor UAV with the LinkBoard I flight control board, B. The LinkQuad quadrotor UAV platform.

endurance. Its diameter is approximately 50 cm and the total weight is around 500 grams. The platform's own electronics (X-Base and ResearchPilot boards) implements the inner control loop (attitude stabilization) running at 1 kHz. The UAV can be flown by a human pilot via an RC transmitter but it also accepts control signals (i.e. roll, pitch and yaw angles and thrust commands) via an RS232 connection.

The LinkBoard (revision I) from UAS Technologies Sweden AB[9] has been used as the flight control computer. Thanks to the board's modular design it is capable of performing a wide range of tasks. In this application, the 520 MHz XScale processor running Windows CE 6.0 operating system hosts the vision-based pose estimation algorithm. A Logitech Quickcam Pro 5000 provides a video stream via a USB connection. The result of the pose estimation is delivered via an RS232 link to a second processor (60 MHz ARM7) which implements the sensor fusion algorithm. A second processor of the same type implements four proportional-integral-derivative (PID) loops commanding all the control channels of the UAS.

The flight control board interfaces with the ground control station software using a 2.4 GHz wireless modem from AeroComm[10]. The ground station software is used to monitor the on-board software execution by displaying the telemetry data. The modem can also be used to send commands to the UAS. For example, the hovering position, altitude, or heading can be changed during an autonomous flight.

All the additional components, i.e. the LinkBoard flight control board, the camera, the modem and a 330 mAh battery, have a total weight of approximately 50 grams.

[9] Homepage: `http://www.uastech.com`

[10] Homepage: `http://www.aerocomm.com`

LinkQuad UAV platform.

The LinkQuad quadrotor UAV (see Figure 3.4B) from UAS Technologies Sweden AB[11] has been used for the validation of the marker-based positioning system.

The LinkQuad's airframe is characterized by a modular design which allows for easy reconfiguration to adopt to a variety of applications. Thanks to a compact design (below 70 centimeters tip-to-tip) the platform is suitable for both indoor and outdoor use. It is equipped with custom designed optimized carbon fiber propellers which contribute to an endurance of approximately 30 minutes. Depending on the required flight time, one or two 2.6 Ah batteries can be placed inside an easily swappable battery module. The maximum take-off weight of the LinkQuad is 1.4 kilograms with up to 300 grams of payload.

The platform is equipped with the LinkBoard III flight control system. The LinkBoard has a modular design and this allows for adjusting the required computational power depending on mission requirements. In the basic configuration the LinkBoard is equipped with two Cortex-M3 microcontrollers (72 MHz) but can be extended with two Gumstix Overo modules (up to 720 MHz). The integrated sensors include a 3-axis accelerometer ($\pm 2/\pm 6$ g range), 3 rate gyroscopes, an absolute pressure sensor, a differential pressure sensor and a 4-channel 12-bit 1 MSPS analog to digital converter. The LinkQuad communicates with the ground station using a 2.4 GHz (115200 bits per second data rate) wireless modem from AeroComm. Additionally, the communication can be achieved using wireless Ethernet (802.11b/g) integrated with the Gumstix boards.

The LinkBoard supports external sensors which can be interfaced with using RS232 or USB connections. In the configuration used here a 0.3 megapixel FireFly MV monochrome USB camera from Point Grey[12] was used. It is capable of delivering images of up to 640×480 at a rate of up to 60 Hz. The camera is mounted on a pan-tilt module attached to the LinkQuad.

The main components of the system used during the validation of the marker-based positioning system are presented in figure 3.5. The high accuracy reference data (position, velocity and orientation) obtained using the positioning system described in the following section is sent to the ground station software (LinkGS) and is immediately forwarded to the on-board system using the wireless modem. The marker-based positioning algorithm receives inertial data from the Sensor Microcontroller Unit (Sensor MCU) and uses images delivered by the camera to compute the same set of parameters of the UAV in the environment. The results of the algorithm are then sent to the Control MCU where this data, as well as the reference information are logged at the rate of 500 Hz on a Secure Digital (SD) memory card.

[11] Homepage: `http://www.uastech.com`

[12] Point Grey: `http://www.ptgrey.com`

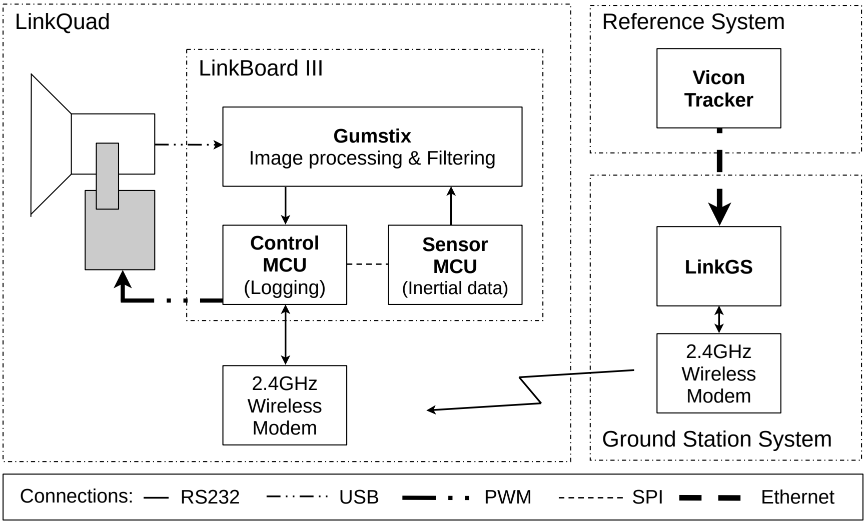

Figure 3.5: The main system components used for validation of the marker-based positioning system.

3.2.3 Indoor positioning reference system

The UASTechLab uses a commercial indoor motion capture system from Vicon[13] both as the source of indoor positioning information for autonomous flight and as a high accuracy reference system for validating other indoor positioning solutions. The system consists of ten T10 cameras. Each camera has the resolution of 1 megapixel and captures 10-bit greyscale images (maximum 1120 × 896 pixels) up to 2000 times per second. In the full resolution, however, the rate is limited to 250 frames per second. The system is installed in a 9.5 × 6 × 2.45 meters room and the camera placement as well as the coordinate system axes used are depicted in figure 3.6. The T10 cameras illuminate the scene with infrared light which is reflected by a set of markers attached to a tracked physical object. The configuration of the reflective markers defines an object and its position and orientation is computed in real-time. The cameras are connected to the Vicon MX Giganet switch which is also responsible for time synchronization of the system. The switch connects to a PC running the software for data collection and analysis. From there the motion tracking data can be logged and also retrieved in real-time to be used for, for example, autonomous flight of a UAV.

The accuracy of UASTechLab indoor positioning system for objects in motion is difficult to estimate as it would require an additional reference system to compare it to. To assess the static accuracy a set of measurements was performed. Figure 3.7 shows plots of the measured positions in X and Y

[13] Vicon: `http://www.vicon.com`

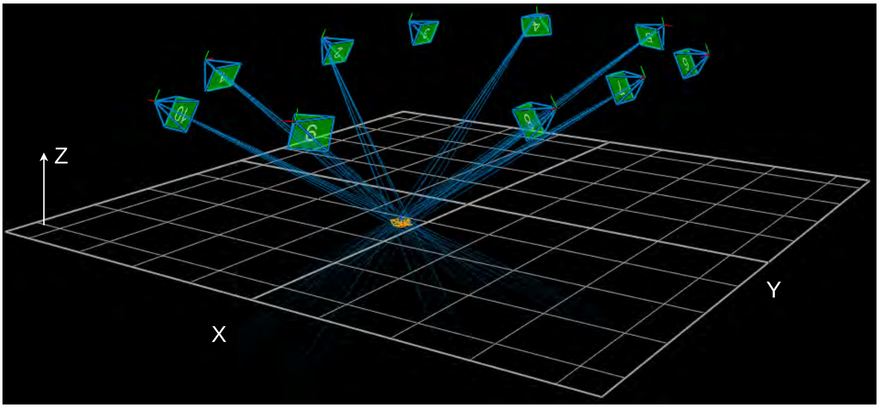

Figure 3.6: The UASTechLab indoor positioning system camera setup.

directions (Z has very similar characteristics as the X direction) and rotation around the X axis (Y and Z rotations have very similar characteristics) of a stationary object placed in the center of the coordinate system. The object had a set of reflective markers placed in a rectangular pattern with a side dimension of 25 cm. Table 3.1 presents maximum values and standard deviations of position and orientation of the same object for all dimensions. The maximum error of position is 0.228 mm and the maximum orientation angle error is 0.0267 degrees in this particular experiment. Even though the X and Z positions exhibit a larger error and apparent oscillations (see figure 3.7 Position X) the static error is minuscule. This makes the system a very high accuracy reference system for evaluating other positioning solutions.

3.3 Related work

Research with small-scale unmanned aircraft systems is being pursued in several different fields. Development of various kinds of airframes is an example of an active research area [6, 23, 34, 71]. Theoretical and prac-

Table 3.1: Position and rotation maximum error and standard deviations for a static measurement.

		Maximum (abs)	Standard deviation
Position [mm]	X	0.2280	0.1253
	Y	0.0849	0.0235
	Z	0.2036	0.1148
Rotation [degree]	X	0.0215	0.0087
	Y	0.0249	0.0059
	Z	0.0267	0.0070

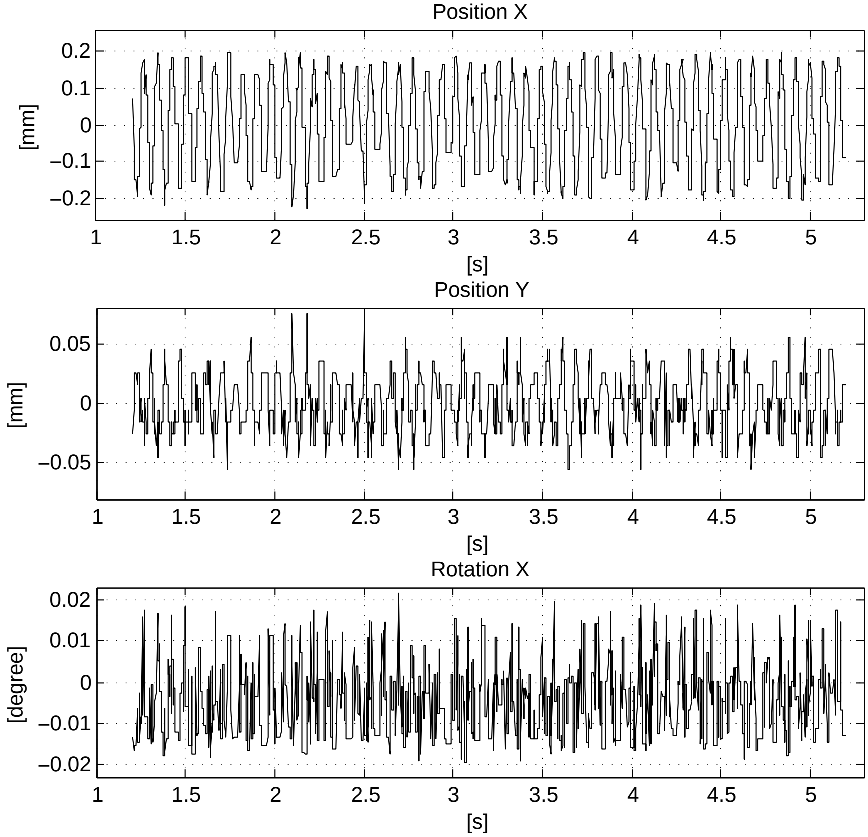

Figure 3.7: Position in X and Y directions and rotation around the X axis of a static object placed in the center of the coordinate system measured using the UASTechLab reference positioning system.

tical solutions to various control problems are also studied by many researchers [7, 25, 44, 69, 85]. Many of the solutions address particular types of control problems, for example aggressive maneuvering and aerobatics with small UAVs [29, 49, 61, 65, 68]. Work on particular problems where UAVs are used for experimentation is also often reported. For example, Hoffmann et al. [45] describes an information-theoretic distributed control architecture based on particle filters to facilitate search by a mobile sensor network. In this work, four quadrotor UAVs are used to search for a rescue beacon. Other types of work with small-scale UAS's involve path planning [37], target tracking [38], mapping [52] and obstacle avoidance [8]. From the perspective of this chapter, however, the most relevant work can be found in the field of indoor localization.

Several approaches to the indoor navigation problem take advantage of the Vicon MX camera system[14], usually used for motion capturing applications. It incorporates a set of cameras which also illuminate the environment with highly efficient and powerful infrared LED light. The system delivers a 6 Degrees of Freedom (6-DOF) solution using lightweight reflective balls attached to a vehicle's structure. A six camera configuration of the Vicon system allows, for example, for simultaneous tracking of four quadrotor MAVs and multiple ground vehicles in a 5 × 5 × 2 meter flight volume in Schweighofer and Pinz [77]. A disadvantage of this technique is the static nature of the environment setup where motion capturing or pose estimation takes place. Once cameras are set up they remain stationary. Exploration of an unknown environment cannot be performed using this method, thus restricting possible applications. Other examples of using motion capturing systems for enabling indoor flight are presented in How et al. [48] and Saad et al. [76].

Other approaches to indoor MAV navigation include using artificial markers commonly used in augmented reality applications [4], or fusing vision information (obtained from light diodes attached to a MAV) and inertial data for a real-time attitude estimation of a quadrotor platform [24]. Alternative solutions take advantage of an artificial beacon on a blimp MAV [15] or use custom low power FPGA (Field-Programmable Gate Arrays) boards for vision aided attitude stabilization for a quadrotor MAV [26].

A number of solutions take advantage of different types of artificial features to enable pose estimation for MAVs. One of them employs two cameras, one mounted on a pan-tilt unit on the ground and one on-board a quadrotor MAV [2]. The two cameras track colored blobs attached both to the UAS and to the ground camera. Besides the need for off-board processing, the disadvantage of this solution is a rather limited flight envelope accessible to a MAV. This method allows for indoor flight, but preferably above the ground camera. This considerably limits the operational range of the MAV. A different method makes use of information obtained from a target which takes advantage of a moiré pattern [81]. The pose of a camera is calculated relative to a novel pattern which requires backlighting. The flight test results presented show the applicability of the method for controlling a quadrotor platform by calculating the position and the yaw angle of the UAS. The disadvantage of this system is a limited operational range because it is not easy to obtain multiple unique instances for this kind of marker. Another approach based on using artificial markers and fusion with the inertial data can be found in Zhang et al. [92].

Several attempts have been made to solve the indoor navigation problem by means of SLAM. For example, a monocular vision SLAM technique for UAS's has been proposed in [10]. It exploits the architectural features of manmade indoor environments, namely corners detectable in corridors. The contribution of this work is "a new absolute range and bearing measurement algorithm using monocular camera". Unfortunately the authors do not pro-

[14] Webpage: `http://www.vicon.com/products/viconmx.html`

vide enough detail to judge the applicability of the method for on-board execution nor any results that use the technique in closed-loop experiments. Recently, a vision-based approach, where the vehicle is localized using a downward looking monocular camera has been proposed by Blösch et al. [5]. A visual SLAM algorithm tracks the pose of the camera, while, simultaneously, building an incremental map of the surrounding region allowing for autonomous indoor flight.

A single camera solution is a basis for a navigation system in Soumelidis et al. [78].The system is capable of 3D localization but it requires an a priori map of the environment. The proposed algorithm is computationally intensive and is executed off-board on a wirelessly transmitted video stream. Similarly, Wendel et al. [84] propose a novel algorithm for monocular visual localization for MAVs based on the concept of virtual views in 3D space. The system takes advantage of a precomputed map and under the assumption that significant parts of the scene do not alter their geometry they serve as natural landmarks.

A number of solutions suggest incorporating biologically inspired methods in order to deal with limited computational capabilities of small-scale UAS's. Utilization of optic flow techniques has received the widest attention. It can be shown that optic flow can constrain the error in velocity and attitude [53]. The technique, however, does not completely eliminate the positional drift and additional information (such as geo-referenced images) is required for an accurate position estimation as described in [11]. Other biologically inspired approaches can be found in Zhang et al. [93] and Zingg et al. [96]

Recently, systems based on ranging sensors have gained a considerable amount of attention. A system based on a laser range finder which uses a particle filter to globally localize a quadrotor platform in a pre-computed map has been suggested [33]. The need for off-board computation, however, makes the system vulnerable to communication interruptions. Another approach taking advantage of a laser range finder as well as stereo-vision can be found in Achtelik et al. [1]. Unfortunately, commercially available small-scale laser range finders are much heavier than miniature video cameras. From a platform's endurance perspective imaging sensors are much more attractive for UAV navigation.

Recently, work based on gaming console's hardware has been reported. Wenzel et al. [87] proposed taking advantage of this kind of low-cost and light-weight commodity consumer hardware. The Nintendo Wii remote infrared (IR) camera is used for obtaining a relative pose between a UAV and a ground robot. The system is capable of taking-off, following and landing on a pad attached to a UGV [87, 88]. Similarly, Lange et al. [56] suggest using a RGB-D sensor (Microsoft Kinect gaming control sensor) for UAV indoor navigation. Thanks to the low-cost of such sensors it is foreseeable that more approaches based on this kind of hardware will be proposed in the future.

3.4 UAV navigation through cooperation with a ground vehicle

This section describes a solution to the UAV indoor navigation problem based on cooperation between a ground robot and a UAV. The remainder of this section is structured as follows. It starts with a general description of the system in section 3.4.1. Section 3.4.2 presents the image processing and the pose estimation technique used as well as the details of the custom LED pattern design. The accuracy evaluation results are presented in section 3.4.3. It is followed by a description of the control system of the UAV in section 3.4.4. The experimental setup and the results of the flight tests are presented in section 3.4.5.

3.4.1 System overview

The system consists of two main elements as presented in figure 3.8 . Namely, a UAV and a ground robot. The backup pilot's system is only required to enhance the safety of operation and is used to relay control commands from the ground system to the UAV.

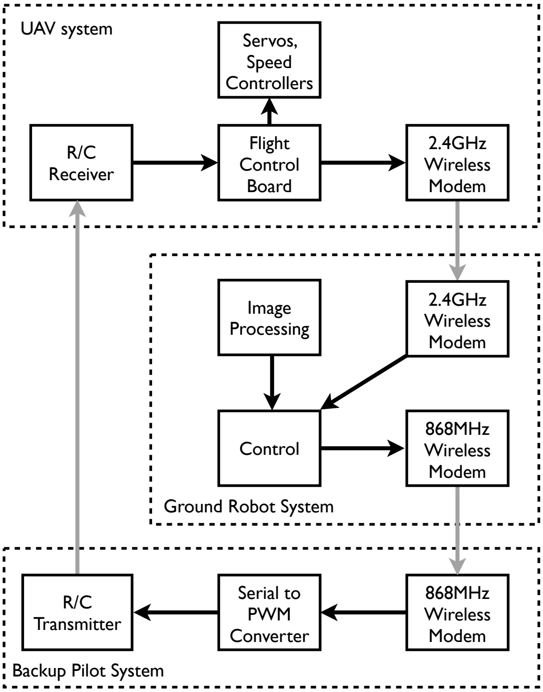

Figure 3.8: Main functional components of the system.

A video camera which is connected to a computer performing the image processing is placed on the ground vehicle. This avoids the need for wireless transmission of the video stream and allows for obtaining interference free images as well as avoiding the problem of image quality degradation due to the camera vibration onboard a UAV. Such a solution greatly improves the robustness of the vision system.

Additionally, the video camera is placed on a pan-tilt unit which allows for navigation of the UAV even if the ground vehicle is stationary. This also makes the system able to maintain control over the MAV in case of flight disturbances, which often occur in indoor environments when passing by a fan or an open door. A camera placed on a pan-tilt unit tracks the flying vehicle and the pose estimation algorithm constantly delivers the pose of the UAV allowing a controlled flight.

The fact that the UAV can fly away from the ground robot (within a certain range) makes the flying vehicle behave as an "external sensor" which provides sensor data normally not accessible from the point of view of a ground robot alone.

3.4.2 Pose estimation method

In order to calculate the 6-DOF pose of the flying vehicle, a vision system has been designed. It includes a custom designed LED cube shaped structure, a video camera mounted on a pan-tilt unit and a computer vision technique which detects colored diodes in the received video stream. A detailed description of all the components is provided in the following subsections.

Pattern design

The pose estimation method relies on a specially designed cube-shaped structure mounted on a UAV (Fig. 3.9A, B). Only one of its faces is required to be visible to the camera for the pose estimation algorithm to deliver a solution. The UAV can perform a full 360 degree yawing motion in order to point its onboard sensors in a desired direction. The fact that side faces of the cube are used for determining the pose of the MAV frees the UAV from the requirement of staying atop the video camera. This makes the flight envelope considerably larger as the UAV can fly away from the ground robot within a certain range. The top and bottom faces of the cube are not considered because they are obscured by the rotor and the landing gear respectively. Including the bottom face would not constitute a problem, except for the requirement of additional diodes. In that case the flight volume would be extended to allow flight directly above the UGV.

There are two high-intensity LEDs (SuperFlux from LumiLeds) in each corner of the cube mounted at 90 degree angles to increase the usable viewing angle to 90 degrees. Colored diodes are used to uniquely code each of the 4 faces. Only red, green and blue colors are used to minimize the possibility of color misclassification in the case of large distances between a diode and

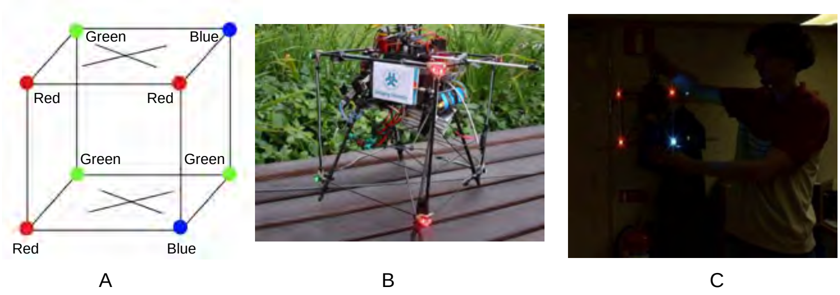

Figure 3.9: A. A schematic view of the LED cube with the actual color diode placement B. The LinkMAV with the LED cube C. Video frame from a camera with a very fast shutter speed.

a camera. Other colors (i.e. magenta and orange) were tested but produced misclassifications, especially at larger distances or steep viewing angles.

The size of the cube was determined based mainly on the properties of the platform used for experimentation, specifically its size and the take-off weight. The cube used in the experiments measured 187x198 mm and was made out of carbon fiber rods. The structure was attached to the UAV frame by a system of springs to cancel the influence of high frequency vibrations generated by the spinning rotors. Its total weight (carbon fiber rods, balsa wood, diodes, resistors and a connector) is approximately 60 grams. It uses a small battery which is matched to the flight endurance of the UAV.

Image processing

In order to filter out as many false positive classifications as possible, the camera operates with a very fast shutter speed (Fig. 3.9C). This makes the process of finding cube corners easier and more robust since most of the background becomes black. To cope with false diode classifications, which often occur in cases when direct sunlight illuminates the background, an additional check has to be performed. It includes examining all possible combinations of the detected diodes in order to find those which belong to a valid configuration. This requires finding configurations of four LEDs with properly ordered colors yielding minimal size and holding appropriate angle relationships between the corners of a pattern.

In case two faces are visible to the camera (around multiples of 45 degrees yaw angle) and six diodes are identified, only one face is chosen based on the distance between corners. The face with maximal distance is preferred in order to minimize the influence of the camera resolution on the final result. The more pixels describing the distance between classified diodes, the more accurate the result.

Image coordinates of four identified diodes of one face are processed by the Robust Pose Estimation from a Planar Target [77] algorithm to extract the pose of a face. Knowing which face is visible and the angles of the pan-tilt unit on which the camera is mounted, the complete pose of the UAV relative to the ground robot is calculated.

3.4.3 Accuracy evaluation

The accuracy of the pose estimation method has been evaluated in a series of static experiments. Both attitude and position precision were assessed. The LED cube pattern was mounted on a pan-tilt unit (DirectedPercpetion 46-17.5W) on a test bench. Distances were measured using a measuring tape. The angles were recorded from the pan-tilt unit which was commanded to perform movements in both axes. To measure the yaw error of the vision system, a scan in range of ±159 degrees (i.e. the standard maximum range of the particular unit) was performed in the pan axis.

Experiments were performed at distances from 2 to 6 meters to determine the maximum range at which the result would allow controlled flight of the UAV. The minimum range of 2 meters stems from the size of the cube pattern and the camera's optical properties. Closer distances would require a smaller pattern in order for it to stay within a usable field of view of the camera. In case of flight disturbances caused, for example, by a fan, an open door, or a close proximity to an obstacle, a certain margin has to be reserved for the MAV to stay within the camera view.

Figure 3.10 presents yaw angle measurements (in range of ±159 degrees) and errors at 2, 4 and 6 meter distances. The vision system experiences difficulty resolving a pose when a face of the cube pattern is close to parallel with the image plane (i.e. ±13, ±20 and ±25 degrees from parallel for the three distances; ranges grayed in the plots). These angle ranges limit the usable extent of allowed yaw angles and were avoided in real flight experiments. For the same reason the flight envelope was limited in distance to approximately 4 meters.

The pitch angle accuracy is approximately the same (small difference in width and height of a face) as for yaw because of the symmetry of these two axes (vertical and horizontal axes of the image plane).

The accuracy of the roll angle (i.e. rotation around the depth axis) was measured in the same fashion as in the case of the yaw axis. The pan-tilt unit was commanded to sweep from 39 to -31 degrees in the tilt axis and the measurements were performed at distances from 2 to 6 meters.

Figure 3.11 presents the roll angle measurements and errors at 2 and 4 meter distances. The error grows slightly with distance. Standard deviations for measured distances increase but are approximately the same (1.3 degree). This stems from the fact that this axis can be resolved from vision without ambiguities. The roll angle is measured with sufficient accuracy for this

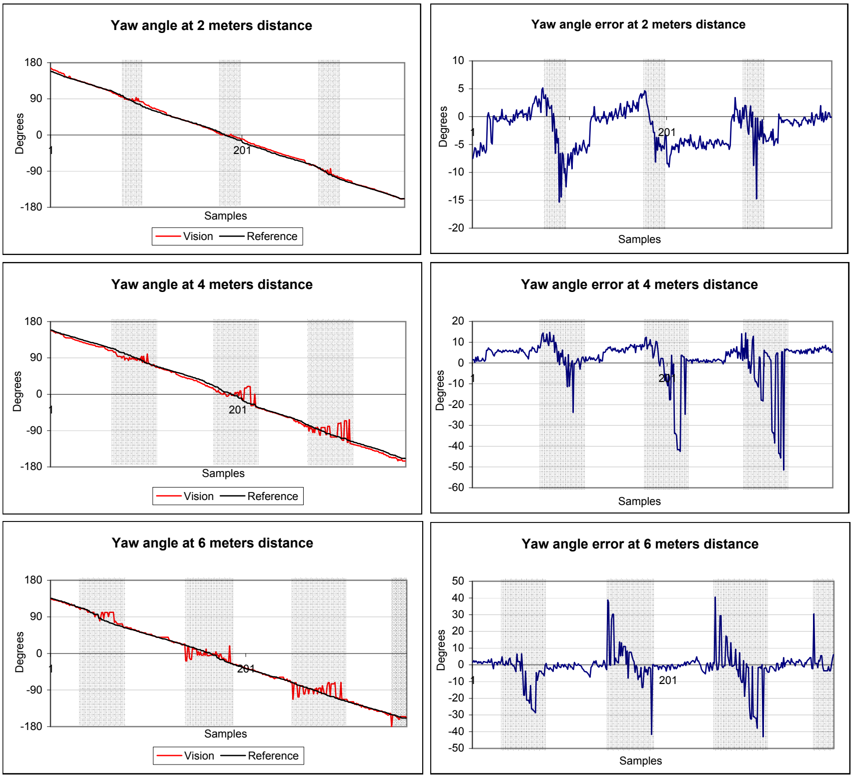

Figure 3.10: Measured values and errors for the yaw axis a three distances.

application.

The distance measurement was performed at distances from 1 to 5 meters. For distances up to 3 meters, the error was smaller than the accuracy of the ground truth measurement (e.g. the exact placement of the CCD element of the camera is unknown). For distances of 4 and 5 meters the absolute error was approximately 13 and 45 centimeters, respectively. The absolute error and its standard deviation grows with distance because of the growing penalty of the camera resolution (the ratio between physical distance between diodes to number of pixels increases). Figure 3.12 presents distance error standard deviations for distances from 1 to 5 meters. It includes the cases where a pattern face is close to parallel with the image plane.

During a real flight, measurements are expected to be worse due to vibrations of the platform and the LED pattern cube. For the flight tests performed, however, it did not pose a noticeable problem. The final vision-only flight envelope is limited to approximately 4 meters distance and by poses where a cube pattern face is parallel to the camera plane as described

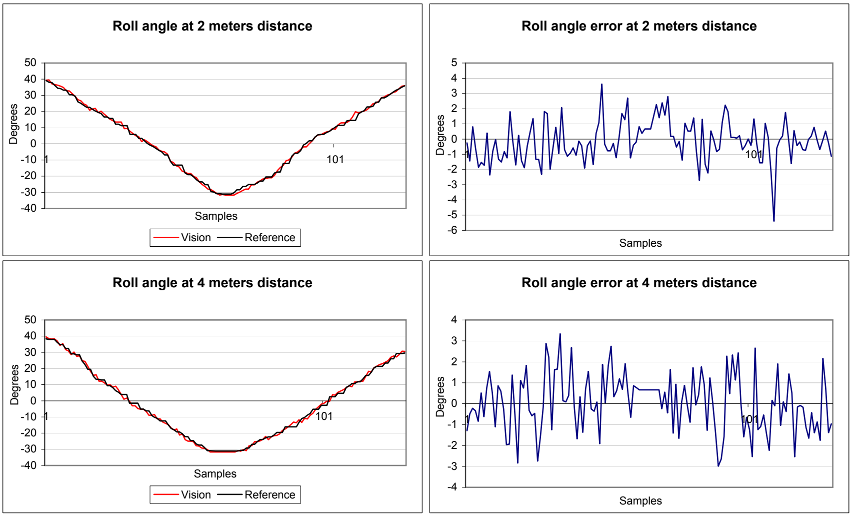

Figure 3.11: Measured values of angles and errors for roll axis.

above. Those poses are avoided during the flight.

3.4.4 Control

The control system signal flow is depicted in figure 3.13. The inner control loop used for the attitude stabilization is closed onboard the LinkMAV by means of the MicroPilot flight control system. The board utilizes a 3 axis accelerometer and MEMS gyroscopes to provide attitude angle estimation for the inner PID stabilization control loop. The autopilot accepts control inputs corresponding to:

- angle in case of roll and pitch channels,
- angular velocity in case of the yaw channel,
- mixed collective pitch and rotors' rotation speed in case of the altitude channel.

The outer loop control (i.e. position stabilization) consists of a set of PID control loops. The loop closure is depicted by the grayed arrow in figure 3.13. The image processing pose estimation output (X, Y, Z positions and the yaw angle) is processed by means of first order low-pass filters.

The control mode was used in two operational modes. One allows the ground operator to change the target position, heading and altitude of the UAV relative to the ground robot's pose. The other mode allows for driving the robot and keeping the relative pose between the robots constant. A combination of the two is also possible.

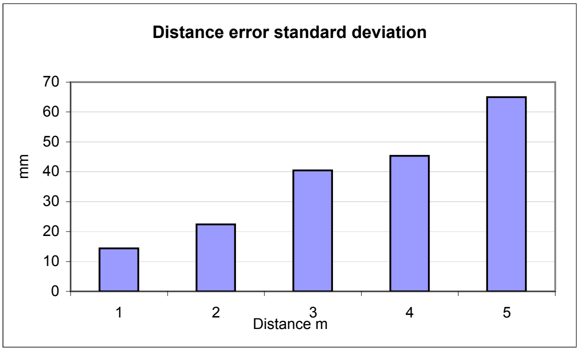

Figure 3.12: Standard deviation of distance error for several distances.

3.4.5 Experimental validation

Several hours of flight tests were performed with the described system. Two kinds of experiments are presented here to demonstrate the suitability of the system for realistic indoor missions. Test flights were performed with all parts of the system fully operational. No tethers or external power supplies were used. The system was operated by a single ground operator who was commanding the ground robot to drive to a certain location and placing the UAV in a desired relation to the ground robot. Autonomous take-off and landing was not implemented and a backup pilot performed those parts of the mission manually. After the UAV entered into the view of the camera and the system was initialized, the control was handed over to the system.

Exploration

This basic flight mode allows for exploring an environment. The ground operator commands the UAV to place itself in a certain pose relative to the ground robot. Then, the operator commands the ground robot to drive to a desired location. During the driving the UAV does not require any direct control input.

”Eye-in-the-sky”

The second flight presented here demonstrates an application of the system as an extended camera which provides video footage from a position not accessible from a ground robot’s point of view. A cardboard box was placed on the path of the ground robot simulating an obstacle. A person was lying behind the box and the task was to provide footage from behind the obstacle. The box was approximately one meter high and anything behind it was out of reach for the ground robot’s video camera.

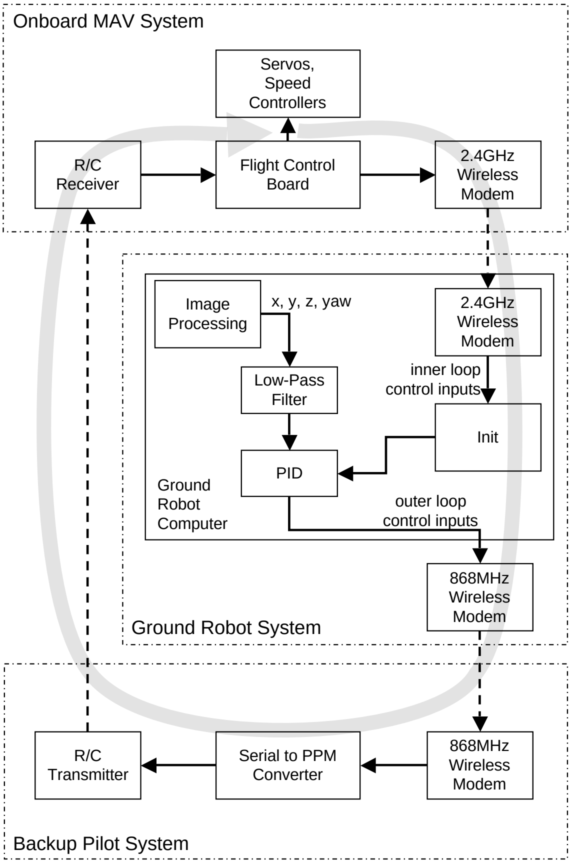

Figure 3.13: Control flow diagram. The grayed arrow depicts the outer loop closure.

Experimental results

Figure 3.14A shows six sample frames of the exploration task video. The ground robot drove approximately 10 meters, stopped and turned 40 degrees left and continued driving forward. Throughout the whole time, the UAV kept a constant relative pose to the ground robot. The second task started with a straight drive of about 7 meters and ended with the ground vehicle arriving close to an obstacle. The LinkMAV was commanded to climb several decimeters above the box. After that, the UAV was commanded to fly 1 meter forward to reach behind the obstacle. Despite turbulence generated by the close distance between the UAV and the obstacle, the flight was autonomous at all times. After the person behind the obstacle was spotted by the ground operator, the ground robot was commanded to return to its starting position. Three sample frames of the video are presented in

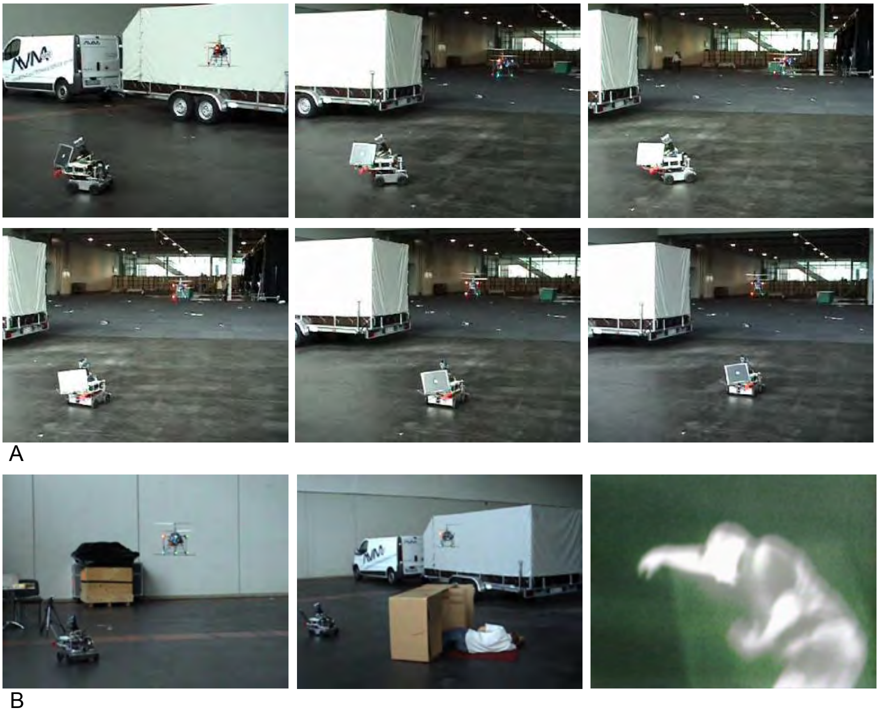

Figure 3.14: Top 6: Frames of video of the system during exploration task. Bottom 3: Frames of video of the system reaching behind an obstacle.

figure 3.14B. The bottom-right image presents a frame from the onboard UAV thermal video with the identified human body.

The pose estimation algorithm runs at a frame-rate of around 20Hz and allows for controlling the UAV purely by vision.

3.4.6 Conclusion

A fully implemented and tested method for indoor UAS navigation has been presented. The system is based on cooperating with a ground robot to facilitate the indoor localization of an aerial vehicle. The role of the ground robot is to provide the UAV with location information. This kind of cooperation allows for taking advantage of the well developed techniques for ground robot localization together with the ability of a flying robot to position its sensors in a way normally not available to a ground robot.

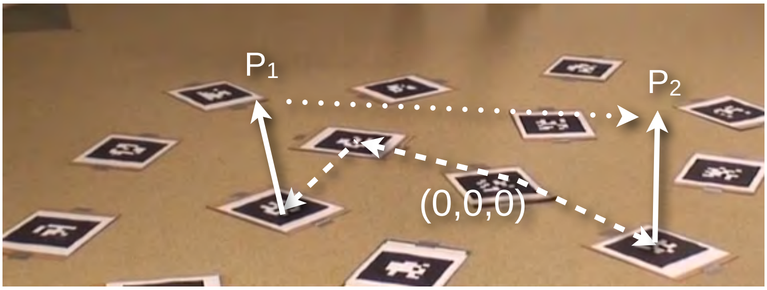

Figure 3.15: Schematic view of the marker-based visual pose estimation solution.

3.5 Marker-based visual pose estimation for indoor navigation

Solutions which require distributing components of the system among several wirelessly communicating hardware entities can be effective and allow for indoor UAV navigation as shown in the previous section. Truly autonomous systems, however, should not rely on external help. This section presents a solution to the indoor navigation problem which is computationally independent of other entities. All computations are done on-board a small scale UAV.

The solution is based on a marker-based visual state estimation algorithm. It delivers a 6-DOF state of a camera on a vehicle in relation (solid lines in figure 3.15) to low-cost markers placed in the environment. The vision-based state estimate is then fused with inertial data to provide the full state estimate which is of higher rate and also includes velocities of the vehicle.

Thanks to incorporating a mapping functionality, the arbitrarily placed markers form a map (relations between markers depicted with dashed lines) of the environment and allow for navigation of a UAV within this map. For example, autonomous flight between two positions (P_1 and P_2 in figure 3.15) can be achieved because the state is calculated along the path (dotted line).

The remainder of this section is structured as follows. First, the main components of the system are introduced in section 3.5.1, followed by the description of the marker-based visual pose estimation method including the technique which allows for mapping of the environment in sections 3.5.2 and 3.5.3, respectively. Second, the sensor fusion method is introduced in section 3.5.4 and is followed by an evaluation of the accuracy of the method in section 3.5.5. The control method used is described in section 3.5.6. Finally, the results of the experimental validation are presented in section 3.5.7 followed by the conclusions in section 3.5.8.

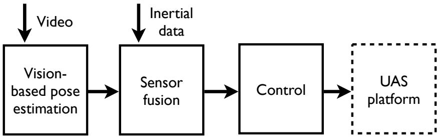

Figure 3.16: The main functional components of the system.

3.5.1 System overview

The main functional components of the system are shown in Figure 3.16. The vision-based pose estimation module makes use of a set of arbitrarily placed artificial markers which form a map of the operational environment. This module delivers the complete 6-DOF pose of a UAV i.e. 3D position and 3 attitude angles. The implementation of the marker-based state estimation module is based on a library used in augmented reality applications, the ARToolkitPlus[15].

The pose information (excluding roll and pitch angles) is fused using a Kalman filter with the inertial data (i.e. three accelerations and three angular rates) to produce the full state of the system. It consists of the pose and 3 velocities (north, east and up). It is used as input to the control module which implements the flight control modes (e.g. hovering and navigation to a given point).

The details of the system components are presented in the remainder of this section.

3.5.2 Pose estimation method

The process of calculating a pose is divided into two major steps, *marker detection* and *pose estimation*. Marker detection is accomplished in the following manner. First, rectangular regions in an image are found as hypotheses of legitimate markers (see Marker boundary in Figure 3.17B). Second, the interior of each hypothesis is analyzed in order to decode the identifying number (see Figure 3.17B ID boundary). The goal of the marker detection phase is to discern the ID and identify a sequence of the marker corners' coordinates in the image. The pose estimation algorithm calculates the relation (rotational and translational) between the camera and the marker coordinate systems. The calculation is based on the coordinates of the detected marker corners (i.e. projection of the marker model on the image plane, cf. figure 3.17A).

Both steps of the vision-based pose estimation process perform differently depending on the relation of the marker and the camera in terms of translations and rotations. For example, for certain distances and camera

[15] Homepage: `studierstube.icg.tugraz.at/handheld_ar/artoolkitplus.php`

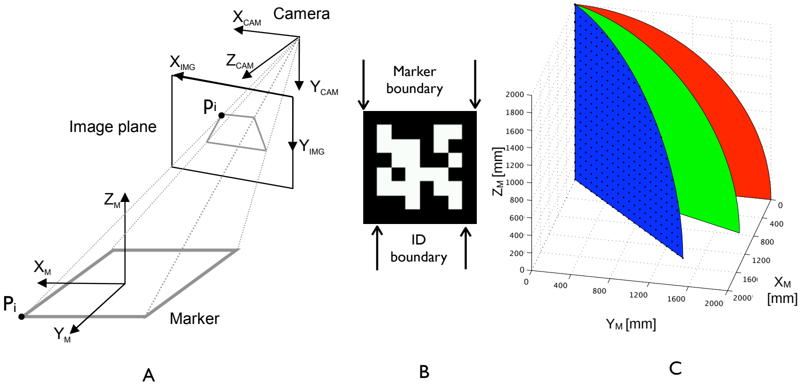

Figure 3.17: A. Vision-based pose estimation coordinate systems B. Example marker C. Camera positions (black dots) for the accuracy estimation experiments.

attitudes estimating the pose is not possible at all or the result has a considerable error. The following subsections describe the process and results of evaluating the algorithm from the perspective of UAS navigation.

Marker detection

A marker ID can be coded in three different ways in the ARToolkitPlus library. The first one uses a template matching technique. This type is the most computationally and memory expensive since the interior of a detected rectangle is compared with an image loaded at startup. For multiple markers all images have to be stored in memory for comparison. The second type, called *Simple ID*, supports 512 predefined markers and relies on decoding the ID from an array of bits (e.g. 6×6) located inside the rectangular boundary of a marker. The third type, called *BCH ID* (depicted in Figure 3.17B), encodes marker IDs using a forward correcting code and supports up to 4096 combinations.

Both Simple and BCH ID coding methods were evaluated to assess the optimal camera placement relative to a marker which assures reliable detection. A set of images was produced by projecting a marker (with linear interpolation) of 100×100 mm size onto the image plane of a camera with image size of 320×240 pixels, focal lengths $f_x = f_y = 386$ and the principal $c_x = c_y = (160, 120)$.

The camera was placed as schematically depicted by black dots in Figure 3.17C i.e. in 50 mm increments in X_M and Z_M directions up to 2 m. Three representative rotations around the Z_M axis were used: 0, 22.5 and 45 degrees (respectively, red, green and blue marked regions). The rotation

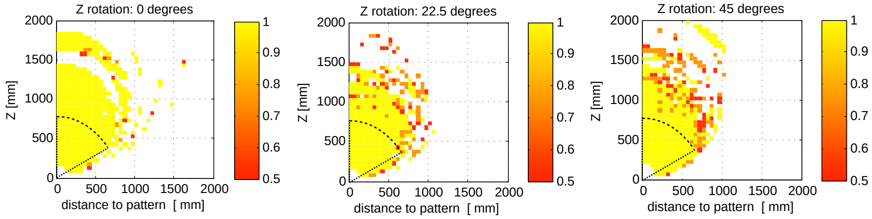

Figure 3.18: BCH Marker detection confidence larger than 0.5 for three rotations around the Z_M axis.

around the X_{CAM} axis was chosen so that the camera was always pointing to the center of the marker which was located at the origin of the coordinate system. Marker detection was performed on the set of generated images.

The confidence of detecting a marker and decoding its ID for the three rotations around the Z_M axis is shown in Figure 3.18. The confidence for detecting BCH IDs is decreased from 1 by 0.25 for every detected and corrected error bit. Results with a confidence factor lower than 0.5 are not included in the plot. The result of detecting Simple IDs was on average 13 percent worse i.e. detection was successful for fewer camera poses. The use of BCH markers adds almost no computational penalty and is preferred for UAV navigation because it allows for the coverage of large operational environments thanks to a larger number of available IDs.

In order to maximize the marker detection rate, the distance of the camera to a marker should not be larger than 75 cm for the given marker size and camera. Additionally, the image plane should not be tilted more than approximately 60 degrees from the marker. The area with high detection rate is depicted by the dashed line in Figure 3.18.

The evaluation did not include the effects of lens distortion. However, with accurate camera calibration this is a minimal error contributor. Additionally, the experimental setup did not include effects of blurring when the camera moves relative to a marker. Depending on the imaging sensor used the effect of blooming or streaking can also influence the detection performance. However, both effects can be minimized by appropriately adjusting the camera parameters as, for example, described in section 3.5.7.

Pose Estimation

After detecting a marker and extracting its four corners $\mathrm{p}_{i=1\ldots4} = \left[p_{i_x}, p_{i_y}\right]^T$ and given four coplanar model points $\mathrm{P}_{i=1\ldots4} = \left[P_{i_x}, P_{i_y}, 0\right]^T$ a pose estimation algorithm calculates the rotation expressed in Euler angles $R = f\left(\phi, \theta, \psi\right)$ and translation $\mathrm{t} = \left[t_x, t_y, t_z\right]^T$ of a camera such that:

$$\mathrm{p}_i \propto R\mathrm{P}_i + \mathrm{t} \tag{3.1}$$

Since p_i is expressed in camera coordinates, it is in practice perturbed by noise and measured in an image as $\hat{\mathrm{p}}_i$.

In general, calculating a pose i.e. finding R and t can be achieved by minimizing one of the commonly used error functions. For example, an *image space* error function in case of bundle-adjustment algorithms or an *object space* error function (E_{os}) used in the *Robust Pose Estimation Algorithm for Planar Targets* (RPP) [77] can be used. In the latter case, a special parameterization allows one to deal with pose ambiguities with up to a 50 percent better success rate over the standard definition. The algorithm uses an initial pose estimate and finds a second minimum of the E_{os} error function. If it exists, the correct pose should yield the lower value of the error function. Due to this property, the algorithm exhibits a considerably lower ”jerkiness” in the calculated pose. While in augmented reality applications this property gives a better visual result, for UAS navigation it gives more stability and robustness.

Initially, the accuracy and the noise of the RPP algorithm were assessed in a series of Monte Carlo simulations. Poses of the camera were the same as for the marker detection experiments and are shown in Figure 3.17C. The model points P_i were projected onto the image plane for the given R and t and the camera model described earlier. The values of $\hat{\mathrm{p}}_i$ were calculated by perturbing p_i with a uniform noise of 2 pixels in 750 iterations. The error in attitude angle estimation was measured as a difference between the nominal values of (α, β, γ) and the average results of the algorithm in the presence of noise. Similarly, the error of position and altitude was measured as the difference between the nominal value of t and the average result in the presence of noise. Figure 3.19 shows the standard deviation of the error in three axes (columns t_x, t_y and t_z) of the camera translation in the marker coordinate system with 2 pixel noise. The three rows correspond to the three rotations around the Z_M axis as described earlier. The range of altitude and distance to a marker were chosen to emphasize the range of marker detection described in the previous subsection.

The RPP algorithm exhibits a mean and standard deviation errors in a range of several centimeters for distances up to 50 cm for the given set of parameters (i.e. 10 x 10 cm pattern size, 320 x 240 pixels image size). The attitude angles are estimated with an average error below 3 degrees with a standard deviation below 6 degrees. In order to assess a usable range of the algorithm, the results can be “scaled”. For example, doubling the size of the pattern cuts the error in half or doubles the range. The same applies to doubling the resolution of the camera. For an image of size 640 x 480 pixels and a pattern of size 50 x 50 cm, the error measured in centimeters can be reached for up to 5 meters.

The usable range can additionally be extended by fusing the vision result with inertial data. In such cases, increased noise from the vision result can be handled by, for example, a Kalman filter.

The minimum distance to a marker depends on the field of view of the

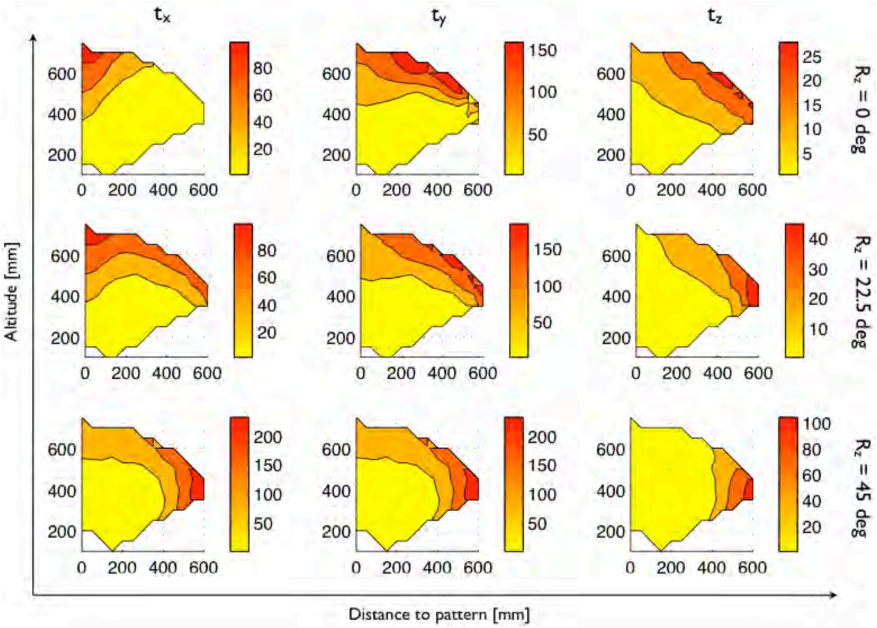

Figure 3.19: Standard deviation in mm of translation error for 0, 22.5 and 45 degrees of camera rotation around Z_M axis of the marker coordinate system with noise of 2 pixels.

camera at hand. Without loss of generality, for a given field of view of α degrees and a rectangular marker width of w mm the minimum distance can be calculated as:

$$d_{min} = \frac{w}{2 \cdot \tan(\frac{\alpha}{2})} \tag{3.2}$$

In practice the distance should be larger to avoid cases when a marker occupies the full frame.

Timing analysis

In order to assess the feasibility of using the proposed vision-based pose estimation on-board a small-scale UAS, a timing analysis of the algorithm was performed. It was made by measuring execution times of the marker detection and the pose calculation implemented in the ARToolkitPlus library on the LinkBoard flight control board. The marker detection phase of the vision-based pose estimation takes approximately 10 percent of the total time of processing of a single frame. Large numbers of markers visible at once in a frame do not add much computational time. Additionally, the detection time is independent of the camera pose.

The pose calculation phase takes the majority of a frame processing

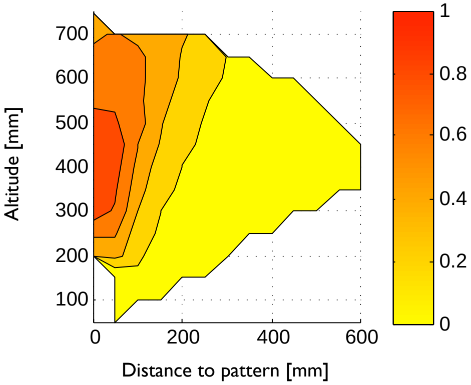

Figure 3.20: Normalized average number of iterations for vision-based pose estimation algorithm depending on the relative camera and marker pose.

time. The amount of time depends on the pose of the camera relative to the marker. Figure 3.20 presents normalized average number of iterations performed by the RPP algorithm. It is approximately the same for all experiments described in section 3.5.2. The number of iterations grows when the image plane and the marker plane become parallel. From a computational efficiency perspective, to achieve the highest frame processing rate, such a relation between the camera and the marker should be avoided during navigation. When the angle between the marker and the image plane is more than approximately 30 degrees the pose calculation takes between 100-200 ms for a single marker (note that the PXA270 microprocessor does not have a floating-point unit). Calculating poses of multiple markers in one frame should be avoided in order to maintain a high processing rate. In fact, it is only required during mapping of the environment. If a map is available and several markers are detected only one of them should be selected to calculate the absolute pose. Selecting the largest (measured in pixels) detected marker has proved in practice to be a very good strategy. This allows for calculating the pose only for this marker, maintaining a high processing rate.

3.5.3 Mapping

Navigation relative to one marker allows only for hovering above it. In order to enlarge the operational range a set of markers is necessary. Thanks to a large number of available IDs (each identified by a BCH code) it is possible to build a map and effectively enlarge the area of navigation. Such a map is a set of marker poses relative to one marker chosen as the origin of the coordinate system. During navigation, when a marker is detected and a relative camera pose is obtained it can be recalculated into an absolute pose in the environment. A map can be built online during environment exploration. A straightforward way to build a map is to iteratively calculate relative poses of newly detected markers in relation to the one chosen as the origin. This requires two markers to be visible in the camera frame at once. The result of the pose estimation of two markers is given by the rotation matrices ${}^{c}_{m}R_i$ and translations ${}^{c}_{m}\mathrm{t}_i, i = 0, 1$. The results describe marker pose relative to the camera coordinate system. The camera pose relative to a marker can be calculated as ${}^{m}_{c}R = {}^{c}_{m}R^T$ and ${}^{m}_{c}\mathrm{t} = {}^{m}_{c}R\ {}^{c}_{m}\mathrm{t}$. The relative pose can be calculated as follows:

$$R_R = {}^{m}_{c}R_0 {}^{c}_{m}R_1 \tag{3.3}$$

$$\mathrm{t}_R = {}^{m}_{c}R_0 \left({}^{m}_{c}\mathrm{t}_1 - {}^{m}_{c}\mathrm{t}_0\right) \tag{3.4}$$

These two values are saved. During navigation, when the i-th marker is observed, the absolute pose (relative to the marker designated as the origin M) can be calculated as follows:

$${}^{c}_{M}R = R_R\ {}^{c}_{m}R_i \tag{3.5}$$

$${}^{c}_{M}\mathrm{t} = R_R\ {}^{m}_{c}\mathrm{t}_i + \mathrm{t}_R \tag{3.6}$$

This method has a very small computational footprint but has a drawback. The error of the pose estimation accumulates in relative poses and grows with the distance from the marker chosen as the origin. In other words, the farther away from the origin, the larger the error of the absolute pose. The error can be minimized by measuring the relative displacement of two markers several times and using an average. This will make the error smaller but will not eliminate it.

One solution to this problem is to employ a method for explicit loop closure which in this case would be a relatively easy task due to the fact that loop detection is solved thanks to using unique IDs.

Another way to solve the problem could be to calculate a globally consistent map given all measurements as in [60]. This algorithm operates on a graph where markers are represented as vertices and relative poses of two markers as edges. It is more computationally expensive than the simple solution and is more suitable for offline calculation.

3.5.4 Sensor fusion

In order to provide a robust navigation solution, the position and heading information delivered by the vision-based pose estimation is fused with the inertial data (from 3-axis accelerometers and 3 rate gyroscopes) by means of a 9 state Extended Kalman Filter (EKF). The use of pitch and roll angles from vision could improve the accuracy of the state estimation but would increase the computational load and was empirically proven not to be necessary to achieve good performance. The state of the system is represented by

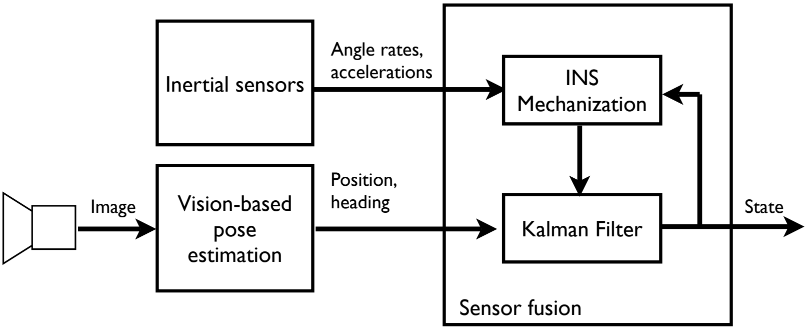

Figure 3.21: The state estimation schematics.

position, horizontal and vertical velocities and attitude angles. The schematics of the sensor fusion technique is presented in Figure 3.21. The EKF uses a linear model for the measurement and process equations. The filter is implemented in the error dynamics formulation and the state represents the errors of the inertial navigation system (INS). The INS mechanization process is responsible for the time integration of the inertial sensor data (i.e. dead reckoning). It provides the full navigation solution but it suffers from a position drift which grows in time. In order to bound this error the output of the EKF (i.e. errors of the position, velocities and attitude) which is aided by the position update is fed to the INS mechanization process. Details of such a formulation of the sensor fusion technique can be found in [11].

Thanks to the sensor fusion technique, the UAS is able to navigate for a limited amount of time even without the position information delivered by the vision system. The inertial sensors provide the complete state estimate with a drift, which is corrected as soon as a vision-based position update is available.

3.5.5 Accuracy evaluation

The absolute accuracy of the marker-based pose estimation has been evaluated. The results of the algorithm running on-board the LinkQuad UAV have been compared with high accuracy reference system (see section 3.2). Figure 3.22 presents an overview of the experiment. UAV has been moved in a circular path with varying altitude in a map composed of three 176 mm rectangular markers. Frame size of 320×240 pixels was used and the camera was pointed at 45 degrees tilt angle.

Figure 3.23 presents the raw vision and Kalman filter results as well as the reference data separately for X, Y and Z axes. Figure 3.24 shows the error between the reference and the filter data. Three marked regions on

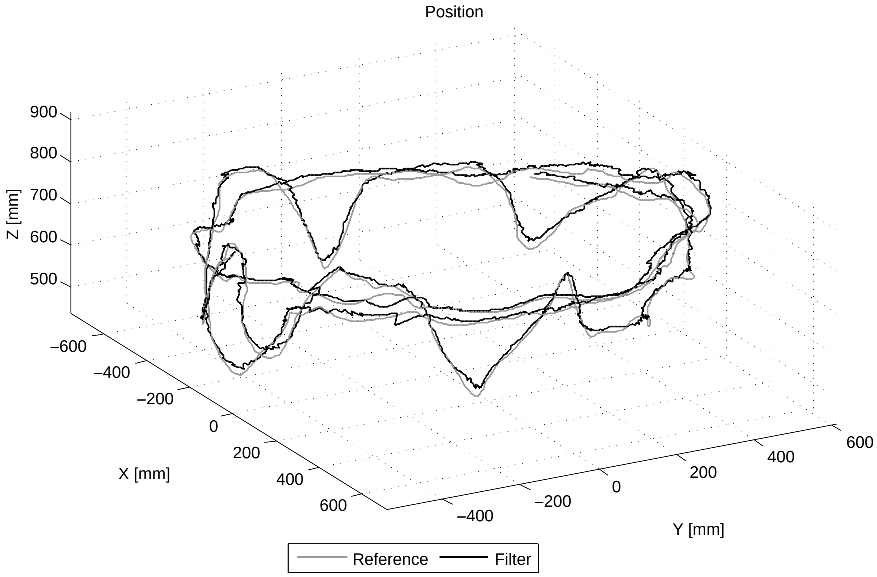

Figure 3.22: Result of the Kalman filter and the reference system of the UAV during the validation experiment.

these plots (at approx. 152, 186 and 227 seconds; see also figure 3.29) show the times when the vision results were not obtained and the filter result is based only on prediction. The maximum position error for the three axes is smaller than ± 40 mm. Position X exhibits a drift of 80 mm when the vision result is not available for 2.6 seconds. Position in the Y and Z direction are not visibly affected by the loss of position update from the vision system. The Root Mean Square Error (RMSE) for X, Y and Z axes is 11.89, 8.10 and 10.69 mm, respectively.

Figures 3.25 and 3.26 present results of comparing the Kalman filter and vision-derived velocities, respectively, with the reference. As can be seen both estimations are close to the reference but pure vision exhibits a larger noise. Since the use of velocity information is crucial for controlling a UAV in flight, the Kalman filter result has a clear advantage over velocities calculated based on the vision data.

The accuracy of attitude angles estimation is presented in figures 3.27 and 3.28. The error for X and Y axes is below ± 1.5 degrees. The maximum error for the Z axis is 2.5 degrees. The RMSE for the three rotations is 0.65, 0.45 and 0.67 degrees, respectively.

The timing of the vision update is presented in figure 3.29. On average the vision-based pose estimation delivers a result every 170 ms. For the same experiment, the average Kalman filter computation time was 43 ms.

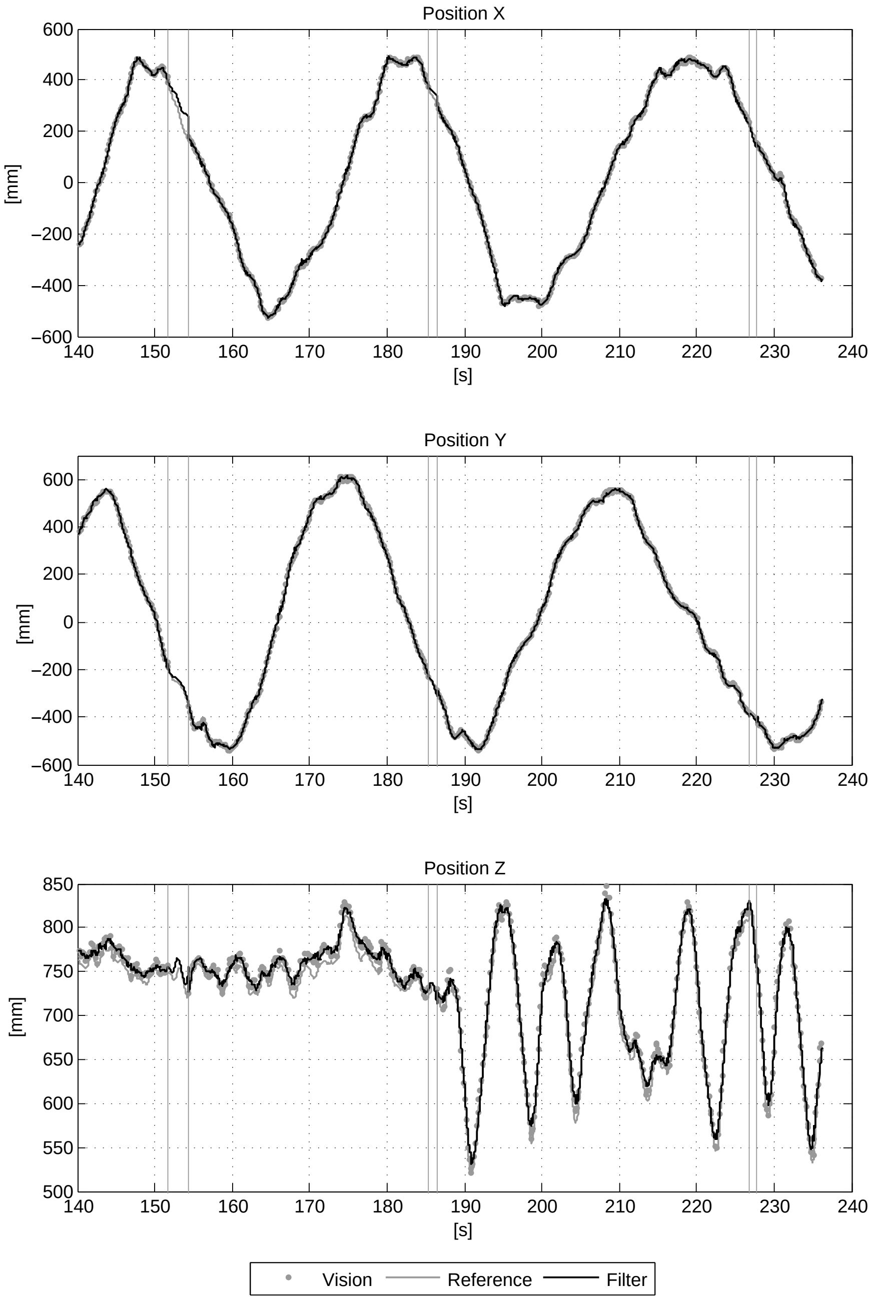

Figure 3.23: Raw vision, Kalman filter and the reference position in three axes. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

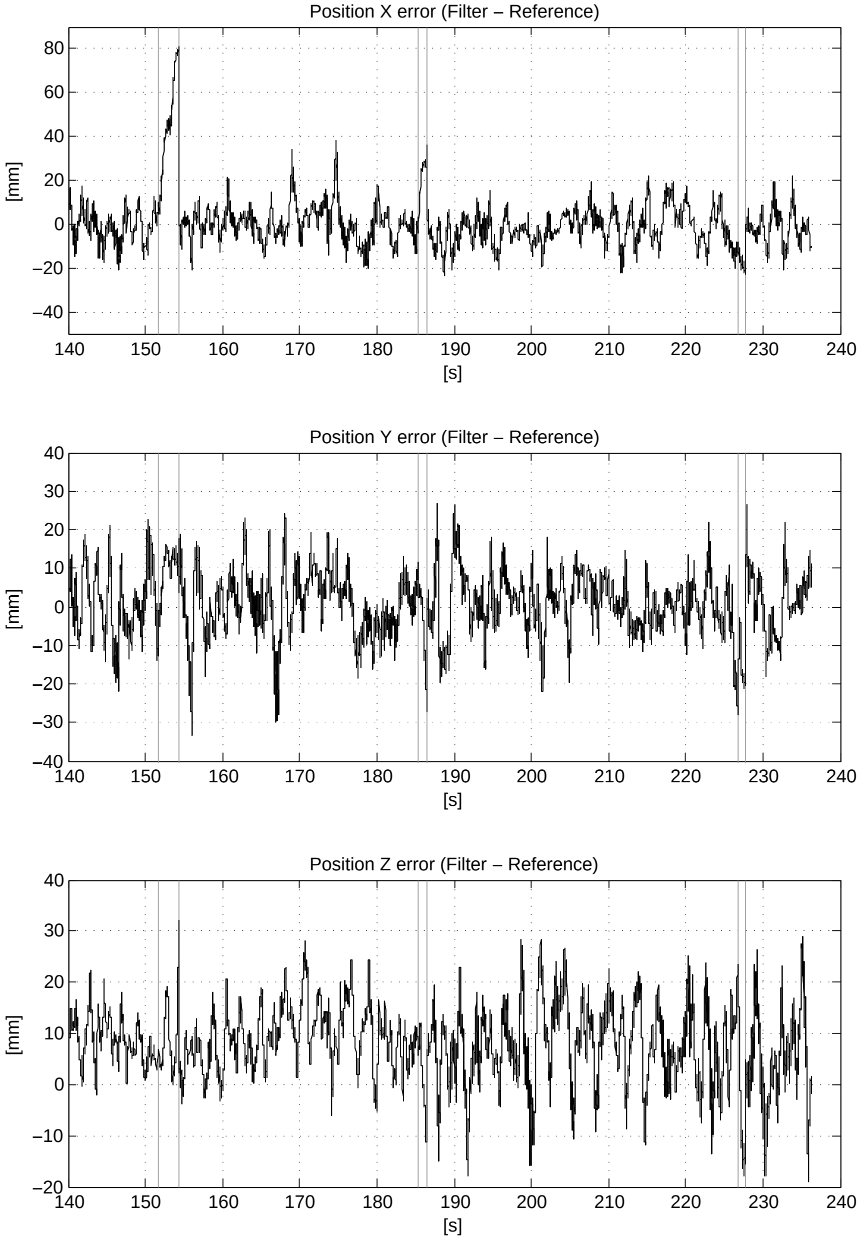

Figure 3.24: Kalman filter and the reference position difference in three axes. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

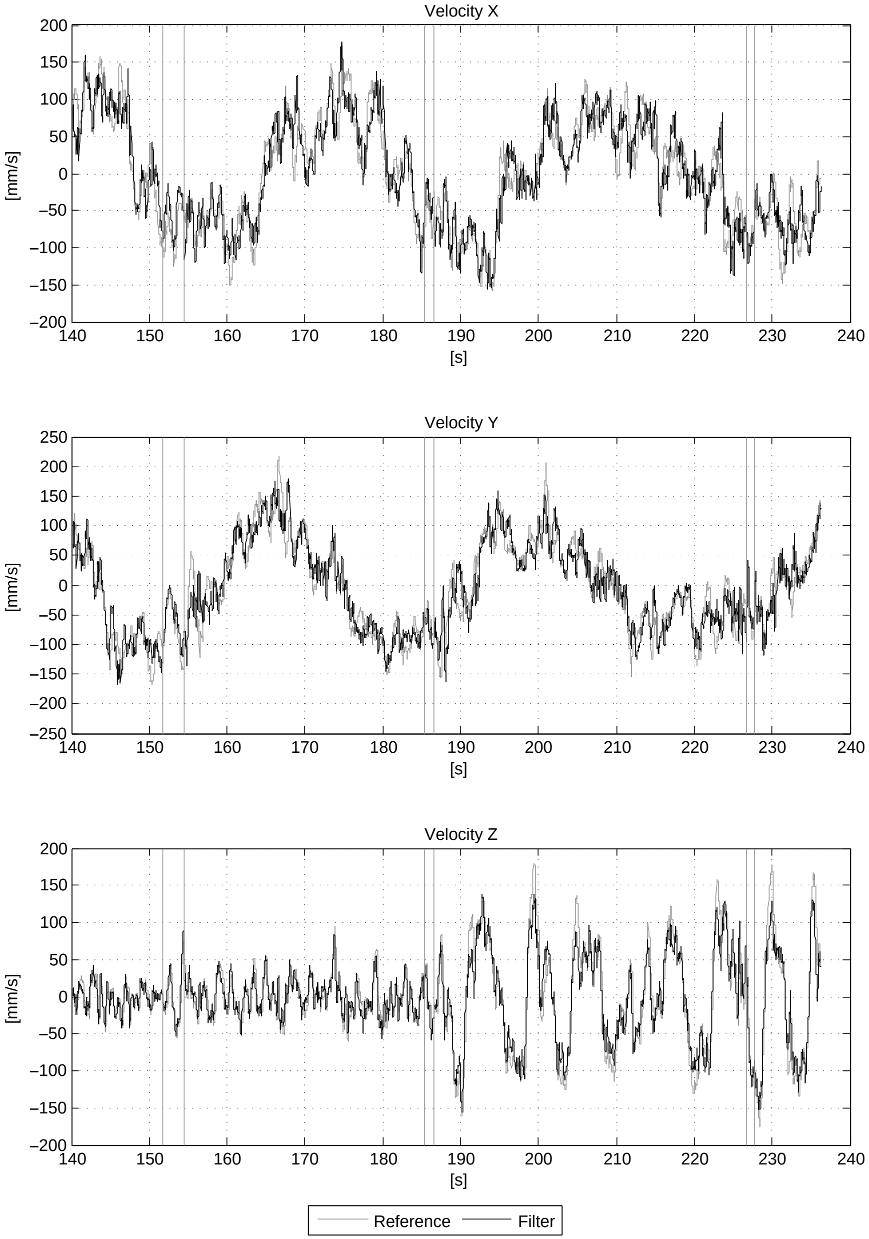

Figure 3.25: Kalman filter and the reference velocities in three axes. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

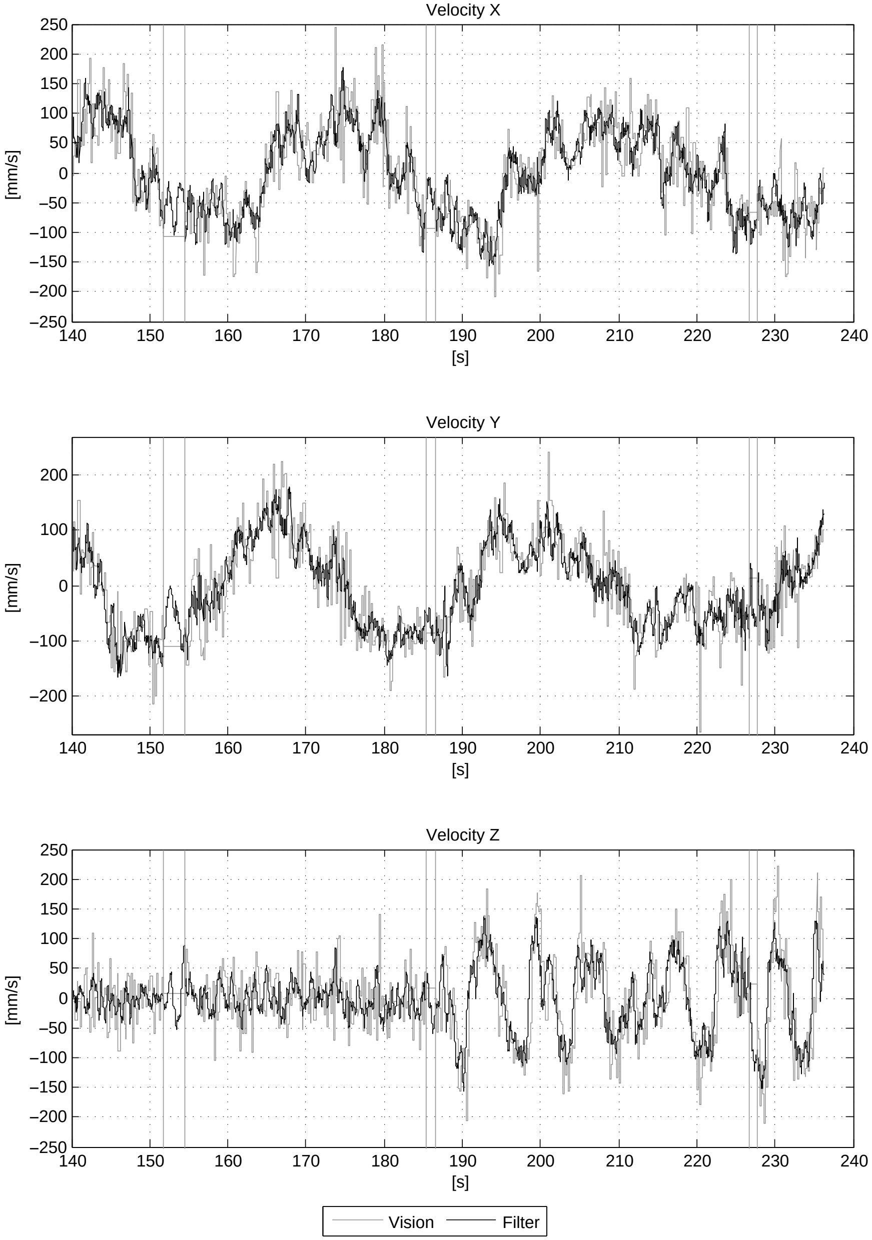

Figure 3.26: Raw vision and the reference velocities in three axes. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

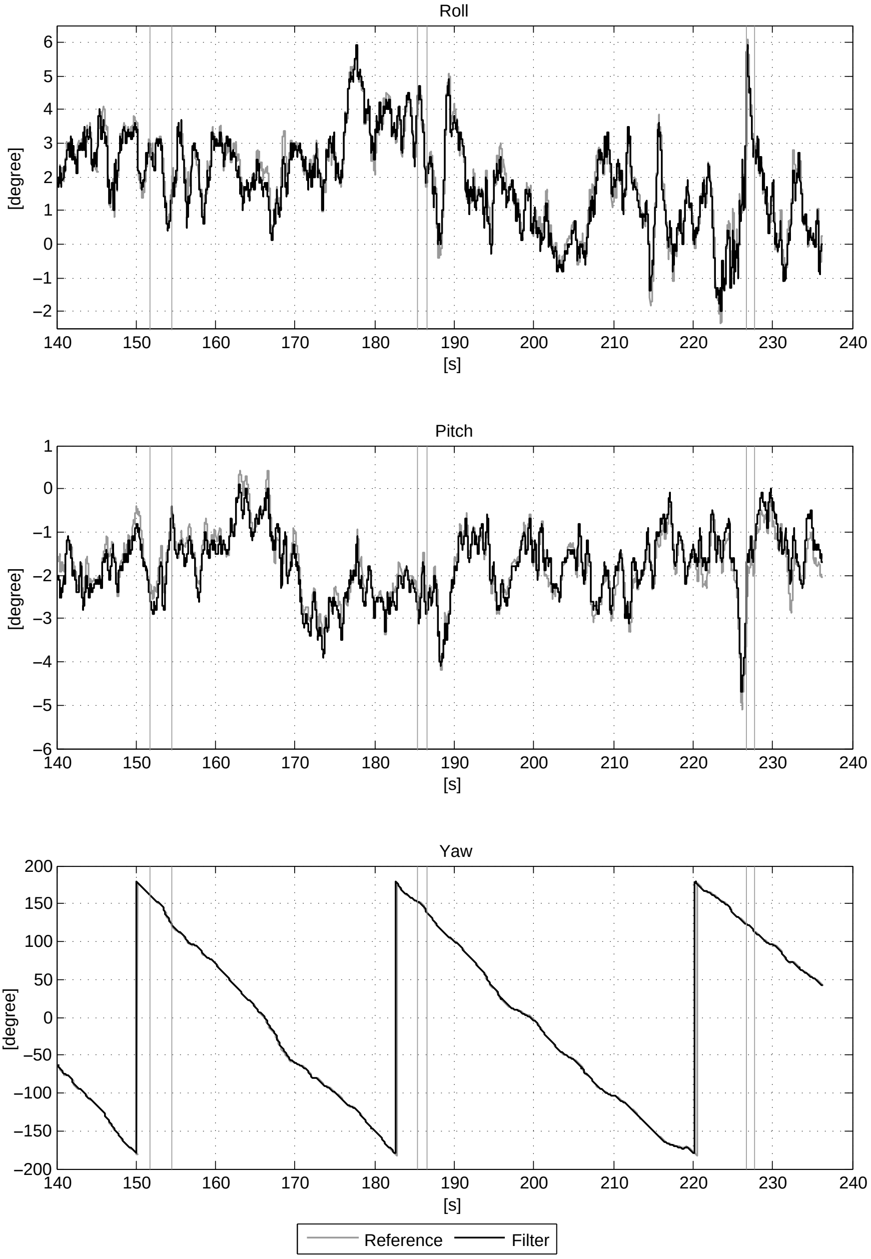

Figure 3.27: Kalman filter and the reference attitude angles. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

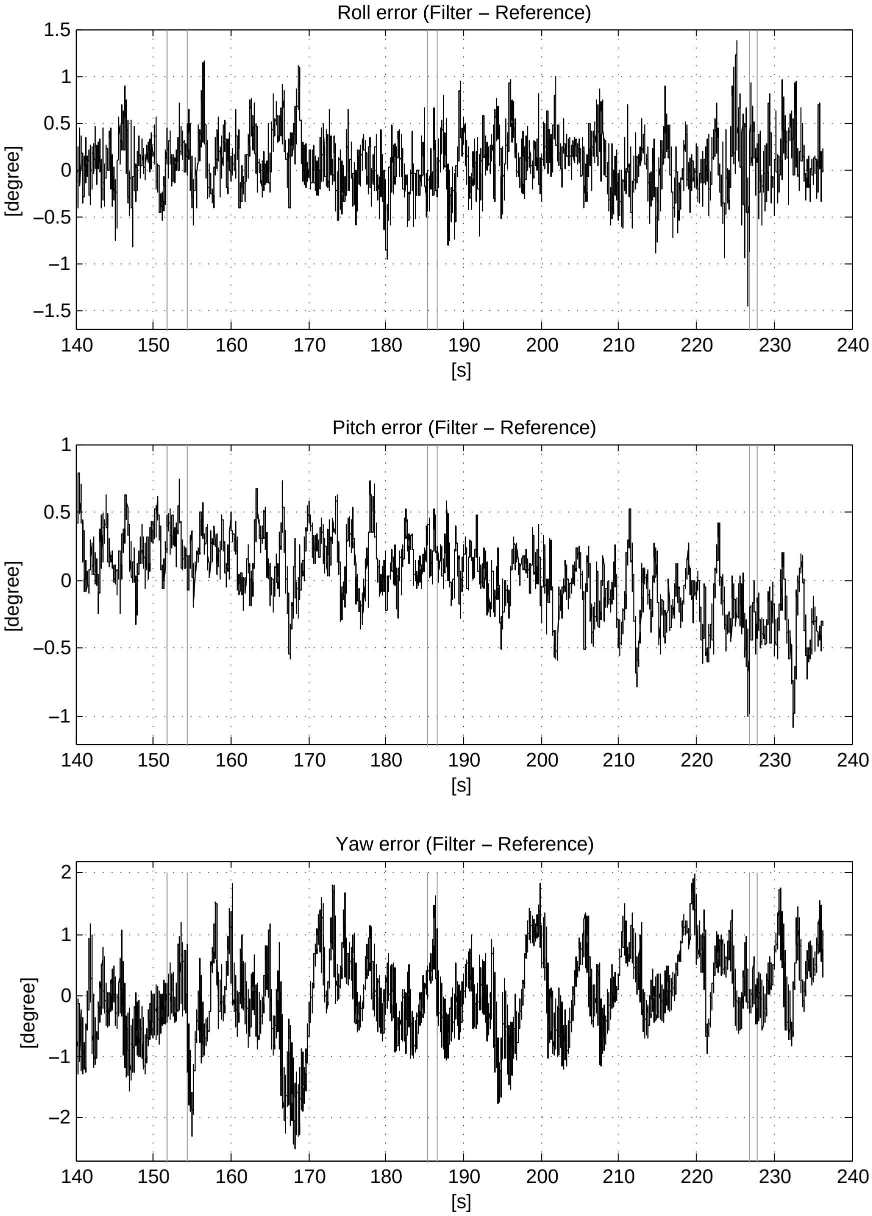

Figure 3.28: Kalman filter and the reference attitude angles difference in three axes. Vertical lines mark regions of vision results not available for longer than 0.9 seconds.

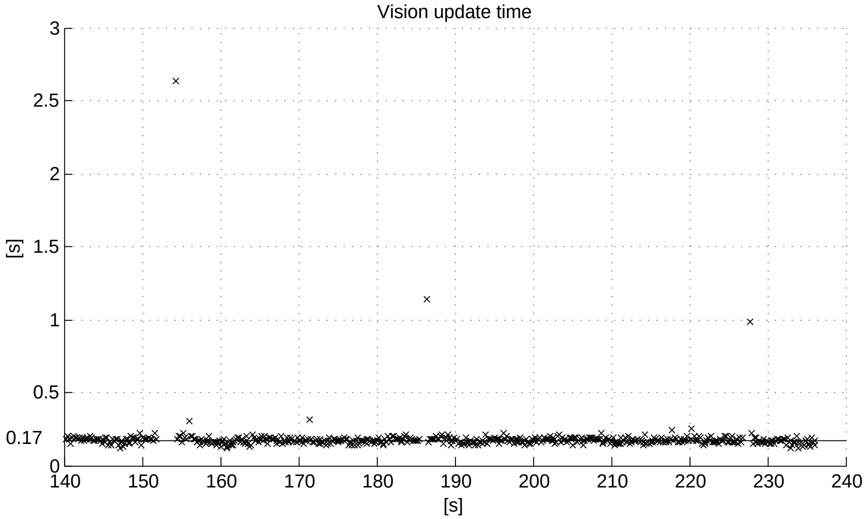

Figure 3.29: Raw vision update timing.

The accuracy of the mapping method described in section 3.5.3 has been evaluated as well. Figure 3.30 presents the X and Y plot of the Kalman filter result and the reference. A set of 12 markers (176 mm) has been placed in a *L* shaped way as presented in figure 3.31. The UAV has been moved from the start position (above a marker placed in the origin of the reference system) along the *L* path of markers. Even though the used mapping method accumulates error the further away from the origin, the drift within the built map of 6 × 3 meters was not visible.

3.5.6 Control

All computations in the system are executed on-board the UAS. The inner control loop (i.e. attitude stabilization) is performed by the Hummingbird quadrotor platform electronics. It accepts input through a serial connection in the form of angles in case of roll and pitch, angular velocity in case of yaw and the thrust value of the altitude channel.

These signals are produced by the outer control loop (i.e. position stabilization) computed on the LinkBoard autopilot. Four PID control loops are responsible for calculating the control signals. During initialization of the outer loop four initial control values are taken from the UAS avionics. They correspond to the values of the sticks of the RC transmitter when the UAS is operated manually. The initialization step is performed when the autonomous flight is engaged and results in a smooth transition between the flight modes.

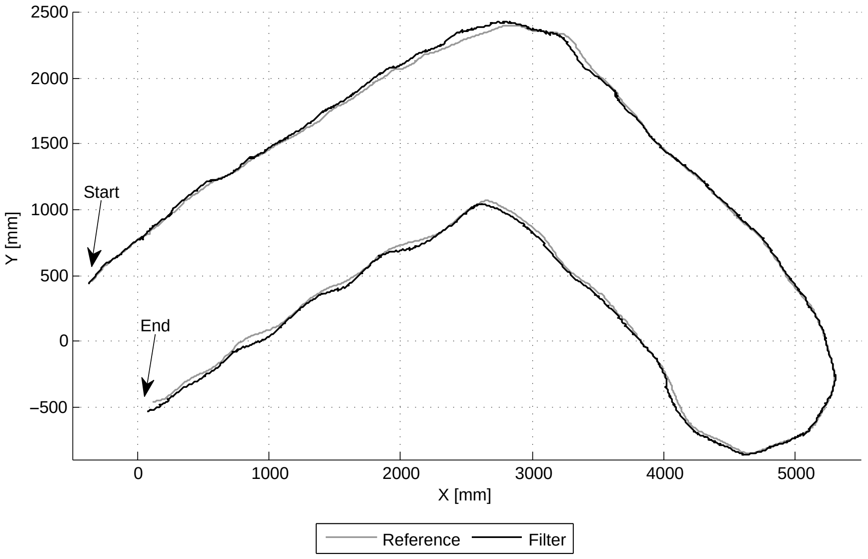

Figure 3.30: Kalman filter and the reference position in X and Y directions for a larger map.

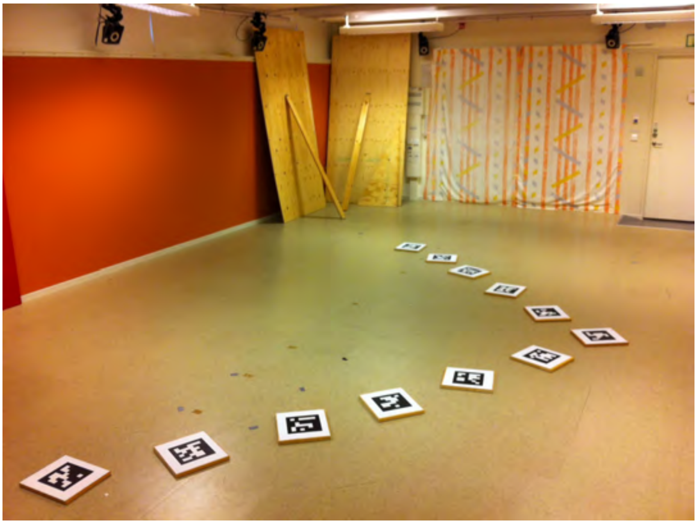

Figure 3.31: The marker setup used for evaluating the accuracy of the map building method. Vicon cameras of the reference system visible at the top of the image.

3.5.7 Experimental validation

The system has been evaluated in real flight tests using the Hummingbird-based UAV platform described in section 3.2 with all the described components operational. No tethers or other aids were used to validate the performance of the system in fully autonomous flight.

Experimental setup

The flights were performed in an office environment. A set of 20 randomly placed artificial markers of size 17×17 cm printed on white paper (see Figure 3.32) were used to compute a map of the environment. The map was constructed on-line during an exploration flight using the procedure described in section 3.5.3 and then saved for successive flights.

In order to minimize the influence of the "rolling shutter" of CMOS sensors the camera parameters were adjusted to use the shortest possible exposure time. To compensate for the darker images the contrast and brightness parameters were appropriately adjusted. Due to this, the typical skew and wobble effects were to a large extent removed from the images.

The camera was pointed 45 degrees down to achieve minimal pose estimation error and a high processing rate (cf. section 3.5.2). The relation between the camera and the UAS coordinate systems was measured with centimeter accuracy. This proved to be sufficient but for the best precision the relation could be fully calibrated using, for example, the method described in Hol et al. [46].

The camera was calibrated to find intrinsic and lens distortion parameters using the Camera Calibration Toolbox for Matlab [9].

Several hours of autonomous flights were performed during development and evaluation of the system. The marker size used allowed for stable hovering up to approximately 1.5 m altitude at different positions and headings in a 4×4 meters region. The results achieved confirm the simulation results presented in section 3.5.2. Since only 20 out of 4096 markers were used, the operational range can be substantially extended in the future.

Experimental results

In order to achieve stable hovering, it was also necessary to include the estimated velocities especially for the horizontal control and to a lesser degree for altitude. The yaw channel did not require velocity in the outer control loop to achieve good performance. Due to battery power depletion over time, it was necessary to include the integral term of the PID controller, especially for the altitude channel. Autonomous flights for up to 20 minutes (i.e. the UAV's endurance) were achieved.

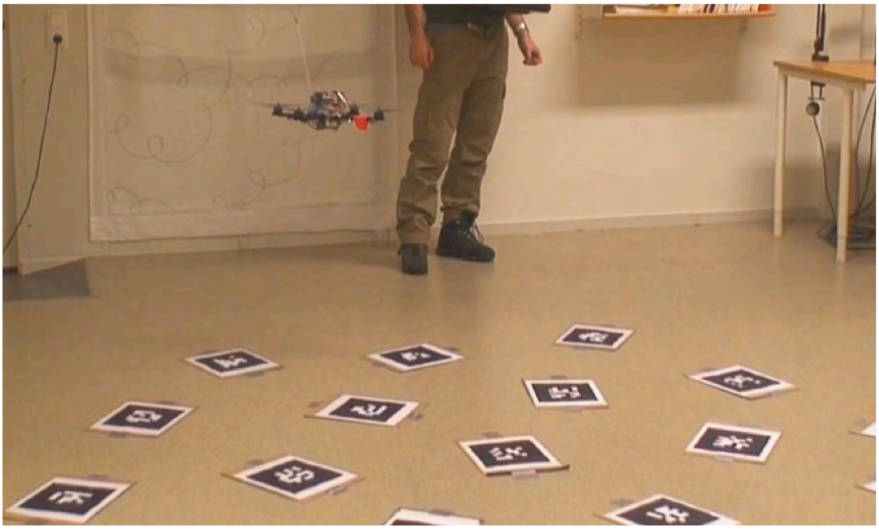

Figure 3.32: Autonomous hovering of the presented system.

3.5.8 Conclusion

An implemented and empirically validated system for autonomous indoor navigation has been presented. The system employs a monocular vision-based pose estimation which uses uniquely coded low-cost rectangular markers. The use of multiple markers (which can be placed at arbitrary poses) and building of a map allow for the operation of a UAS in large areas. The range and accuracy of the pose estimation algorithm has been evaluated and a sensor fusion technique based on a Kalman filter has been presented. Compared to other solutions the system is computationally self-contained. No data or video transmission to other entities is required to achieve autonomous flight. All computations are performed on-board the UAS. This removes the need for continuous wireless communication and increases the robustness and autonomy of the system.

The system has been tested during several hours of flight integrated with a commercially available quadrotor UAS platform. The successful experiments with the existing system show that it is a very promising step towards true autonomous indoor navigation.

Chapter 4

Concluding remarks

The number of missions performed by autonomous UAS's has been steadily growing the past decades. Progress is being made in both military, academic and commercial domains.

Outdoor operating autonomous UAS's have reached a high level of maturity and the types of missions have become relatively sophisticated. Not only do they take off and execute 3D paths with high precision but also plan their trajectories and actions, to finally land without any human involvement.

This thesis presented two functionalities for outdoor operating UAS's which rely on color and thermal video streams as the main source of information. The first functionality deals with the problem of building saliency maps where human body locations are marked as points of interest. Such maps can be used in emergency situations to help first responders to quickly focus the help effort in places where it is most needed. The task of acquiring the first overview of the situation can be delegated to autonomously operating UAS's.

The second functionality deals with tracking and geolocation of vehicles. The obtained streams of vehicle positions can be used by a reasoning system to analyze traffic behaviors such as reckless overtakes, speeding and entering crossings. Thanks to an automatic analysis of the traffic situation, the load on an operator can be greatly reduced and the situational awareness improved.

The use of UAS's in indoor environments suffers from a major drawback, namely the lack of an ubiquitous positioning system such as the GPS. For this reason, the range of missions executed autonomously in indoor settings is quite limited. Before the use of autonomous indoor aerial vehicles becomes commonplace, the problem of indoor pose estimation and localization has to be solved.

This thesis presented two approaches to addressing these problems. In both solutions cameras are used as the main sensor. Their low cost, light weight and low power consumption fit well with typical issues facing small scale UAVs operating in indoor environments: smaller payloads, limited power sources and lower computational power. Both presented solutions have been fully implemented and tested in series of flight tests demonstrating their effectiveness.

The first approach relies on the cooperation between a UAV and a ground robot system. The UGV uses its camera and a computer vision algorithm to compute the pose of the aerial vehicle and is responsible for controlling its flight. The UAV, in turn, acts as a remote sensor and allows the overall system to obtain sensory input not accessible for either of the components on their own. Additionally, the use of a UAV in the presented way allows for building upon the development in the field of ground robotics by extending the sensing capabilities of UGVs. The use of the presented system in emergency situations, such as the recent nuclear power plant accident in Japan, would be very helpful in obtaining information (e.g. video feeds) not accessible to ground robots.

In the second approach all computations are performed solely on the UAV and therefore there is no need for cooperation with other entities. The system makes use of a marker-based visual state estimation and sensor fusion technique to provide a UAV with location information. The presented system constitutes, in fact, a low-cost indoor UAV testbed by solving the localization problem. Building upon the presented system, it is possible to pursue other aspects of UAS use, such as control mode development (e.g. path execution) or applying artificial intelligence techniques (e.g. task and path planning).

4.1 Future work

Future work will involve further investigations of the indoor localization problem. The main objective will be to allow for fully autonomous indoor flight in unmodified environments. The key aspect to building a successful solution will be the use and fusion of different sensors' data (e.g. from laser range finders or optic flow sensors). A combination of several techniques each addressing a particular part of the problem should allow for a robust solution to the indoor localization and navigation of UAVs.

Bibliography

[1] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In Grant R. Gerhart, Douglas W. Gage, and Charles M. Shoemaker, editors, *Unmanned Systems Technology XI*, volume 7332, page 733219. SPIE, 2009.

[2] E. Altuğ, J. P. Ostrowski, and Camillo J. Taylor. Control of a quadrotor helicopter using dual camera visual feedback. *Int. J. Rob. Res.*, 24(5): 329–341, 2005. ISSN 0278-3649.

[3] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele. Vision based victim detection from unmanned aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1740 –1747, oct. 2010.

[4] W. Bath and J. Paxman. UAV localisation & control through computer vision. In *Proc. of the Australasian Conference on Robotics and Automation*, 2005.

[5] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 21 –28, may. 2010.

[6] B. Bluteau, R. Briand, and O. Patrouix. Design and control of an outdoor autonomous quadrotor powered by a four strokes rc engine. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 4136 –4240, nov. 2006.

[7] S. Bouabdallah and A. Noth. PID vs LQ control techniques applied to an indoor micro quadrotor. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2451–2456, 2004.

[8] S. Bouabdallah, M. Becker, and V. De Perrot. Computer obstacle avoidance on quadrotors. In *Proceedings of the XII International Symposium on Dynamic Problems of Mechanics*, 2007.

[9] Jean-Yves Bouguet. Camera calibration toolbox for matlab., 2011. http://www.vision.caltech.edu/bouguetj/calib_doc.

[10] K. Celik, Soon Jo Chung, and A. Somani. Mono-vision corner slam for indoor navigation. In *IEEE International Conference on Electro/Information Technology*, pages 343–348, May 2008.

[11] G. Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*. PhD thesis, Linköping UniversityLinköping University, Department of Computer and Information Science, The Institute of Technology, 2009.

[12] G. Conte, S. Duranti, and T. Merz. Dynamic 3D Path Following for an Autonomous Helicopter. In *Proc. of the IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

[13] G. Conte, M. Hempel, P. Rudol, D. Lundström, S. Duranti, M. Wzorek, and P. Doherty. High accuracy ground target geo-location using autonomous micro aerial vehicle platforms. In *AIAA Guidance, Navigation, and Control Conference, 2008*, volume 26, Honolulu, Hawaii, 2008.

[14] J. W. Davis and M. A. Keck. A Two-Stage Template Approach to Person Detection in Thermal Imagery. In *Workshop on Applications of Computer Vision*, 2005.

[15] L. de Souza Coelho and Mario F. M. Campos. Pose estimation of autonomous dirigibles using artificial landmarks. In *SIBGRAPI '99: Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing*, pages 161–170, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0481-7.

[16] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *AIJ*, 49, 1991.

[17] C.R. del Blanco, F. Jaureguizar, L. Salgado, and N. Garcia. Target detection through robust motion segmentation and tracking restrictions in aerial flir images. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V –445 –V –448, 16 2007-oct. 19 2007.

[18] P. Doherty. Advanced Research with Autonomous Unmanned Aerial Vehicles. In *Proc. of the International Conference on the Principles of Knowledge Representation and Reasoning*, pages 731–732, 2004.

[19] P. Doherty and J. Kvarnström. Talplanner: A temporal logic based planner. *AI MAGAZINE*, 22:2001, 2001.

[20] P. Doherty and P. Rudol. A UAV search and rescue scenario with human body detection and geolocalization. In *AI'07: Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*, pages 1–13, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76926-9, 978-3-540-76926-2.

[21] P. Doherty, G. Granlund, K. Kuchinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund. The WITAS unmanned aerial vehicle project. In *Proc. of the European Conf. on Artificial Intelligence*, pages 747–755, 2000.

[22] P. Doherty, P. Haslum, F. Heintz, T. Merz, P. Nyblom, T. Persson, and B. Wingman. A distributed architecture for autonomous unmanned aerial vehicle experimentation. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[23] S. Duranti, G. Conte, D. Lundström, P. Rudol, M. Wzorek, and P. Doherty. Linkmav, a prototype rotary wing micro aerial vehicle. In *Proceedings of the 17th IFAC Symposium on Automatic Control in Aerospace*, 2007.

[24] M. G. Earl and R. D'Andrea. Real-time attitude estimation techniques applied to a four rotor helicopter. In *43rd IEEE Conference on Decision and Control*, pages 3956–3961, Paradise Island, Bahamas, December 2004.

[25] B. Erginer and E. Altug. Modeling and pd control of a quadrotor vtol vehicle. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 894 –899, june 2007.

[26] S. Fowers, D-J. Lee, B. Tippetts, K. D. Lillywhite, A. Dennis, and J. Archibald. Vision aided stabilization and the development of a quadrotor micro UAV. In *Proc. of the 7th IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 143–148, Paradise Island, Bahamas, June 2007.

[27] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. In *European Conference on Computational Learning Theory*, 1995.

[28] M. Ghallab. On chronicles: Representation, on-line recognition and learning. In *Proceedings of the Fifth Intl Conf on Principles of Knowledge Representation and Reasoning*, 1996. ISBN 1-55860-421-9.

[29] J.H. Gillula, H. Huang, M.P. Vitus, and C.J. Tomlin. Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1649 –1654, May 2010.

[30] M. A. Goodrich, J. L. Cooper, J. A. Adams, C. Humphrey, R. Zeeman, and Brian G. Buss. Using a Mini-UAV to Support Wilderness Search and Rescue Practices for Human-Robot Teaming. In *To appear in Proceedings of the IEEE International Conference on Safety, Security and Rescue Robotics*, 2007.

[31] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey. Supporting wilderness search and rescue using a camera-equipped mini uav: Research articles. *J. Field Robot.*, 25(1-2):89–110, 2008. ISSN 1556-4959.

[32] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. M. Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, 25(1-2):89–110, 2008. ISSN 1556-4959. URL `http://www3.interscience.wiley.com/cgi-bin/fulltext/117865012/PDFSTART`.

[33] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[34] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus. Energy-efficient autonomous four-rotor flying robot controlled at 1 khz. In *IEEE International Conference on Robotics and Automation*, pages 361–366, April 2007.

[35] J. Han and B. Bhanu. Detecting moving humans using color and infrared video. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2003.

[36] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[37] R. He, S. Prentice, and N. Roy. Planning in information space for a quadrotor helicopter in a GPS-denied environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1814 –1820, May 2008.

[38] R. He, A. Bachrach, and N. Roy. Efficient planning under uncertainty for a target-tracking micro-aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, 2010.

[39] F. Heintz and P. Doherty. DyKnow: An approach to middleware for knowledge processing. *Journal of Intelligent and Fuzzy Systems*, 15(1): 3–13, October 2004.

[40] F. Heintz and P. Doherty. A knowledge processing middleware framework and its relation to the JDL data fusion model. In Peter Funk,

editor, *Proceedings of the Swedish Artificial Intelligence and Learning Systems Workshop 2005*, pages 68–77, April 2005.

[41] F. Heintz, P. Rudol, and P. Doherty. Bridging the sense-reasoning gap using dyknow: A knowledgeprocessing middleware framework. In *KI*, pages 460–463, 2007.

[42] F. Heintz, P. Rudol, and P. Doherty. From images to traffic behavior - a UAV tracking and monitoring application. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, July 2007.

[43] H. Helble and S. Cameron. Oats: Oxford aerial tracking system. *Robotics and Autonomous Systems*, 55(9):661–666, 2007.

[44] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *In Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[45] G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin. Mutual information methods with particle filters for mobile sensor network control. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1019 –1024, dec. 2006.

[46] J. Hol, T. Schön, and F. Gustafsson. A new algorithm for calibrating a combined camera and imu sensor unit. In *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, December 2008.

[47] L. Hong, Y. Ruan, W. Li, D. Wicker, and J. Layne. Energy-based video tracking using joint target density processing with an application to unmanned aerial vehicle surveillance. *Computer Vision, IET*, 2(1):1 –12, march 2008. ISSN 1751-9632.

[48] J.P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *Control Systems Magazine, IEEE*, 28(2):51 –64, apr. 2008. ISSN 0272-1708.

[49] H. Huang, G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3277 –3282, May 2009.

[50] J. Kang, K. Gajera, I. Cohen, and G. Medioni. Detection and Tracking of Moving Objects from Overlapping EO and IR Sensors. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 2004.

[51] Jinman Kang, I. Cohen, G. Medioni, and Chang Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 10 – 17 Vol. 1, oct. 2005.

[52] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller. Multiple relative pose graphs for robust cooperative mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3185 –3192, May 2010.

[53] J. Kim and G. Brambley. Dual optic-flow integrated navigation for small-scale flying robots. In *Proc. of the Australasian Conference on Robotics and Automation (ACRA'07)*, 2007.

[54] A. Kleiner, C. Dornhege, R. Kümerle, M. Ruhnke, B. Steder, B. Nebel, P. Doherty, M. Wzorek, P. Rudol, G. Conte, S. Durante, , and D. Lundstr"om. Robocuprescue - robot league team rescuerobots freiburg (germany). In *RoboCup 2006 (CDROM Proceedings), Team Description Paper, Rescue Robot League*, 2006.

[55] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, October 2003.

[56] S. Lange, Sünderhauf N., P. Neubert, S. Drews, and P. Protzel. Autonomous corridor flight of a UAV using a low-cost and light-weight rgb-d camera. In *International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE)*, 2011.

[57] D. J. Lee, P. Zhan, A. Thomas, and R. Schoenberger. Shape-based human intrusion detection. In *SPIE International Symposium on Defense and Security, Visual Information Processing XIII*, 2004.

[58] Y. Li. Vehicle extraction using histogram and genetic algorithm based fuzzy image segmentation from high resolution UAV aerial imagery. In *ISPRS08*, page B3b: 529 ff, 2008.

[59] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. In *Proceedings of International Conference on Image Processing*, pages 900–903, 2002.

[60] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[61] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1642–1648, 2010.

[62] P Mantegazza *et. al.* RTAI: Real time application interface. *Linux Journal*, 72, April 2000.

[63] T. Merz, S. Duranti, and G. Conte. Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In *Proceedings of 2004 International Symposium on Experimental Robotics*, pages 343–352, 2004.

[64] T. Merz, P. Rudol, and M. Wzorek. Control system framework for autonomous robots based on extended state machines. In *Autonomic and Autonomous Systems, International Conference on*, volume 0, page 14, Los Alamitos, CA, USA, 2006. IEEE Computer Society. ISBN 0-7695-2653-5.

[65] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro UAV testbed. *IEEE Robotics and Automation Magazine*, 2010.

[66] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human Detection Based on a Probabilistic Assembly of Robust Part Detectors. In *European Conference on Computer Vision*, 2004.

[67] P. Nyblom. A language translator for robotic task procedure specifications. Master's thesis, Linköping University, 2003.

[68] O. Purwin and R. D'Andrea. Performing aggressive maneuvers using iterative learning control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1731–1736, 2009.

[69] G.V. Raffo, M.G. Ortega, and F.R. Rubio. Backstepping/nonlinear hinf control for path tracking of a quadrotor unmanned aerial vehicle. In *American Control Conference, 2008*, pages 3356 –3361, june 2008.

[70] N.D. Rasmussen, B.S. Morse, M. A. Goodrich, and D. Eggett. Fused visible and infrared video for use in wilderness search and rescue. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 1–8, 2009.

[71] J. F. Roberts, T. Stirling, J-C. Zufferey, and D. Floreano. Quadrotor Using Minimal Sensing For Autonomous Indoor Flight. In *European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, 2007. URL `www.mav07.org`.

[72] P. Rudol and P. Doherty. Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery. In *Proceedings of the IEEE Aerospace Conference*, 2008.

[73] P. Rudol, M. Wzorek, G. Conte, and P. Doherty. Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration. In *Proceedings of the IEEE Aerospace Conference*, pages 1–10, March 2008.

[74] P. Rudol, M. Wzorek, R. Zalewski, and P. Doherty. Report on sense and avoid techniques and the prototype sensor suite. Technical report, National Aeronautics Research Program NFFP04-031, Autonomous flight control and decision making capabilities for Mini-UAVs, 2008.

[75] P. Rudol, M. Wzorek, and P. Doherty. Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems. In *IEEE International Conference on Robotics and Automation, ICRA 2010*, pages 1913–1920, May 2010.

[76] E. Saad, J. Vian, G. Clark, and S. Bieniawski. Vehicle swarm rapid prototyping test-bed. In *Proc. AIAA Infotech@Aerospace Conference and Exhibit*, 2009.

[77] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, pages 2024–2030, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[78] A. Soumelidis, P. Gaspar, G. Regula, and B. Lantos. Control of an experimental mini quad-rotor uav. In *Control and Automation, 2008 16th Mediterranean Conference on*, pages 1252 –1257, june 2008.

[79] P. Theodorakopoulos and S. Lacroix. A strategy for tracking a ground target with a UAV. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1254 –1259, sept. 2008.

[80] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, September 2006.

[81] G. Tournier, M. Valenti, J. P. How, and E. Feron. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In *AIAA Guidance, Navigation and Control Conference*, 2006.

[82] P. Viola and M. J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2001.

[83] S. Waharte and N. Trigoni. Supporting search and rescue operations with UAVs. In *International Symposiom on Robots and Security*, 2010.

[84] A. Wendel, A. Irschara, and H. Bischof. Natural landmark-based monocular localization for mavs. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[85] K. Wai Weng and M.S.B. Abidin. Design and control of a quad-rotor flying robot for aerial surveillance. In *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, pages 173 –177, june 2006.

[86] C. Wengert. A fully automatic camera and hand eye calibration, 2011. URL `http://www.vision.ee.ethz.ch/software/calibration_toolbox/calibration_toolbox.php`.

[87] K. Wenzel, P. Rosset, and A. Zell. Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. *Journal of Intelligent and Robotic Systems*, 57:297–311, 2010. ISSN 0921-0296. 10.1007/s10846-009-9355-5.

[88] K. Engelbert Wenzel, A. Masselli, and A. Zell. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *J. Intell. Robotics Syst.*, 61:221–238, January 2011. ISSN 0921-0296.

[89] M. Wzorek and P. Doherty. A framework for reconfigurable path planning for autonomous unmanned aerial vehicles. *Journal of Applied Artificial Intelligence*, 2007.

[90] M. Wzorek, G. Conte, P. Rudol, T. Merz, S. Duranti, and P. Doherty. From motion planning to control - a navigation framework for an autonomous unmanned aerial vehicle. In *Proceedings of the 21th Bristol International UAV Systems Conference*, 2006.

[91] W.S. Yu, X. Yu, P.Q. Zhang, and J. Zhou. A new framework of moving target detection and tracking for UAV video application. In *International Society for Photogrammetry and Remote Sensing Congress*, page B3b: 609 ff, 2008.

[92] T. Zhang, Ye Kang, M. Achtelik, K. Kuhnlenz, and M. Buss. Autonomous hovering of a vision/imu guided quadrotor. In *International Conference on Mechatronics and Automation (ICMA)*, pages 2870 – 2875, aug. 2009.

[93] T. Zhang, W. Li, M. Achtelik, K. Kuhnlenz, and M. Buss. Multisensory motion estimation and control of a mini-quadrotor in an air-ground multi-robot system. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 45 –50, dec. 2009.

[94] J. Zhou and J. Hoang. Real time robust human detection and tracking system. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, page 149, june 2005.

[95] Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Computer Vision and Pattern Recognition*, 2006.

[96] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. Mav navigation through indoor corridors using optical flow. In *IEEE International Conference on Robotics and Automation (ICRA)*, may 2010.

Avdelning, Institution
Division, Department

AIICS,
Department of Computer
and Information Science
581 83 Linköping

Datum
Date

2011-11-04

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English

☐ ______________

Rapporttyp
Report category

☒ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ______________

ISBN
978-91-7393-034-5

ISRN
LiU-Tek-Lic–2011:49

Serietitel och serienummer
Title of series, numbering

ISSN
0280–7971

Linköping Studies in Science and Technology

Thesis No. 1510

URL för elektronisk version
`http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-71295`

Titel
Title

Increasing Autonomy of Unmanned Aircraft Systems Through the Use of Imaging Sensors

Författare
Author

Piotr Rudol

Sammanfattning
Abstract

The range of missions performed by Unmanned Aircraft Systems (UAS) has been steadily growing in the past decades thanks to continued development in several disciplines. The goal of increasing the autonomy of UAS's is widening the range of tasks which can be carried out without, or with minimal, external help. This thesis presents methods for increasing specific aspects of autonomy of UAS's operating both in outdoor and indoor environments where cameras are used as the primary sensors.

First, a method for fusing color and thermal images for object detection, geolocation and tracking for UAS's operating primarily outdoors is presented. Specifically, a method for building saliency maps where human body locations are marked as points of interest is described. Such maps can be used in emergency situations to increase the situational awareness of first responders or a robotic system itself. Additionally, the same method is applied to the problem of vehicle tracking. A generated stream of geographical locations of tracked vehicles increases situational awareness by allowing for qualitative reasoning about, for example, vehicles overtaking, entering or leaving crossings.

Second, two approaches to the UAS indoor localization problem in the absence of GPS-based positioning are presented. Both use cameras as the main sensors and enable autonomous indoor flight and navigation. The first approach takes advantage of cooperation with a ground robot to provide a UAS with its localization information. The second approach uses marker-based visual pose estimation where all computations are done onboard a small-scale aircraft which additionally increases its autonomy by not relying on external computational power.

Nyckelord
Keywords

UAV, UAS, UAV autonomy, human-body detection, color-thermal image fusion, vehicle tracking, geolocation, UAV indoor navigation

Department of Computer and Information Science
Linköpings universitet

Licentiate Theses

Linköpings Studies in Science and Technology
Faculty of Arts and Sciences

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. (Available at: FOA, Box 1165, S-581 11 Linköping, Sweden. FOA Report B30062E)

No 28 **Arne Jönsson, Mikael Patel:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms, 1984.

No 29 **Johnny Eckerland:** Retargeting of an Incremental Code Generator, 1984.

No 48 **Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching, 1985.

No 52 **Zebo Peng:** Steps Towards the Formalization of Designing VLSI Systems, 1985.

No 60 **Johan Fagerström:** Simulation and Evaluation of Architecture based on Asynchronous Processes, 1985.

No 71 **Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System, 1987.

No 72 **Tony Larsson:** On the Specification and Verification of VLSI Systems, 1986.

No 73 **Ola Strömfors:** A Structure Editor for Documents and Programs, 1986.

No 74 **Christos Levcopoulos:** New Results about the Approximation Behavior of the Greedy Triangulation, 1986.

No 104 **Shamsul I. Chowdhury:** Statistical Expert Systems - a Special Application Area for Knowledge-Based Computer Methodology, 1987.

No 108 **Rober Bilos:** Incremental Scanning and Token-Based Editing, 1987.

No 111 **Hans Block:** SPORT-SORT Sorting Algorithms and Sport Tournaments, 1987.

No 113 **Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge, 1987.

No 118 **Mariam Kamkar, Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs, 1987.

No 126 **Dan Strömberg:** Transfer and Distribution of Application Programs, 1987.

No 127 **Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert Systems, 1987.

No 139 **Christer Bäckström:** Reasoning about Interdependent Actions, 1988.

No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing, 1988.

No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System, 1988.

No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.

No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.

No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models, 1989.

No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment, 1989.

No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager, 1989.

No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition, 1989.

No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems, 1989.

No 187 **Simin Nadjm-Tehrani:** Contributions to the Declarative Approach to Debugging Prolog Programs, 1989.

No 189 **Magnus Merkel:** Temporal Information in Natural Language, 1989.

No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs, 1989.

No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework, 1989.

No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action, 1990.

No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis, 1990.

No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning, 1990.

No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring, 1990.

No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems, 1990.

No 253 **Torbjörn Näslund:** SLDFA-Resolution - Computing Answers for Negative Queries, 1990.

No 260 **Peter D. Holmes:** Using Connectivity Graphs to Support Map-Related Reasoning, 1991.

No 283 **Olof Johansson:** Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge- Bases, 1991.

No 298 **Rolf G Larsson:** Aktivitetsbaserad kalkylering i ett nytt ekonomisystem, 1991.

No 318 **Lena Srömbäck:** Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems, 1992.

No 319 **Mikael Pettersson:** DML-A Language and System for the Generation of Efficient Compilers from Denotational Specification, 1992.

No 326 **Andreas Kågedal:** Logic Programming with External Procedures: an Implementation, 1992.

No 328 **Patrick Lambrix:** Aspects of Version Management of Composite Objects, 1992.

No 333 **Xinli Gu:** Testability Analysis and Improvement in High-Level Synthesis Systems, 1992.

No 335 **Torbjörn Näslund:** On the Role of Evaluations in Iterative Development of Managerial Support Systems, 1992.

No 348 **Ulf Cederling:** Industrial Software Development - a Case Study, 1992.

No 352 **Magnus Morin:** Predictable Cyclic Computations in Autonomous Systems: A Computational Model and Implementation, 1992.

No 371 **Mehran Noghabai:** Evaluation of Strategic Investments in Information Technology, 1993.

No 378 **Mats Larsson:** A Transformational Approach to Formal Digital System Design, 1993.

No 380 **Johan Ringström:** Compiler Generation for Parallel Languages from Denotational Specifications, 1993.

No 381 **Michael Jansson:** Propagation of Change in an Intelligent Information System, 1993.

No 383 **Jonni Harrius:** An Architecture and a Knowledge Representation Model for Expert Critiquing Systems, 1993.

No 386 **Per Österling:** Symbolic Modelling of the Dynamic Environments of Autonomous Agents, 1993.

No 398 **Johan Boye:** Dependency-based Groudness Analysis of Functional Logic Programs, 1993.

No 402 **Lars Degerstedt:** Tabulated Resolution for Well Founded Semantics, 1993.

No 406 **Anna Moberg:** Satellitkontor - en studie av kommunikationsmönster vid arbete på distans, 1993.

No 414 **Peter Carlsson:** Separation av företagsledning och finansiering - fallstudier av företagsledarutköp ur ett agent-teoretiskt perspektiv, 1994.

No 417 **Camilla Sjöström:** Revision och lagreglering - ett historiskt perspektiv, 1994.

No 436 **Cecilia Sjöberg:** Voices in Design: Argumentation in Participatory Development, 1994.

No 437 **Lars Viklund:** Contributions to a High-level Programming Environment for a Scientific Computing, 1994.

No 440 **Peter Loborg:** Error Recovery Support in Manufacturing Control Systems, 1994.

FHS 3/94 **Owen Eriksson:** Informationssystem med verksamhetskvalitet - utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv, 1994.

FHS 4/94 **Karin Pettersson:** Informationssystemstrukturering, ansvarsfördelning och användarinflytande - En komparativ studie med utgångspunkt i två informationssystemstrategier, 1994.

No 441 **Lars Poignant:** Informationsteknologi och företagsetablering - Effekter på produktivitet och region, 1994.

No 446 **Gustav Fahl:** Object Views of Relational Data in Multidatabase Systems, 1994.

No 450 **Henrik Nilsson:** A Declarative Approach to Debugging for Lazy Functional Languages, 1994.

No 451 **Jonas Lind:** Creditor - Firm Relations: an Interdisciplinary Analysis, 1994.

No 452 **Martin Sköld:** Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques, 1994.

No 455 **Pär Carlshamre:** A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development, 1994.

FHS 5/94 **Stefan Cronholm:** Varför CASE-verktyg i systemutveckling? - En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer, 1994.

No 462 **Mikael Lindvall:** A Study of Traceability in Object-Oriented Systems Development, 1994.

No 463 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av Sandviks förvärv av Bahco Verktyg, 1994.

No 464 **Hans Olsén:** Collage Induction: Proving Properties of Logic Programs by Program Synthesis, 1994.

No 469 **Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework, 1995.

No 473 **Ulf Söderman:** On Conceptual Modelling of Mode Switching Systems, 1995.

No 475 **Choong-ho Yi:** Reasoning about Concurrent Actions in the Trajectory Semantics, 1995.

No 476 **Bo Lagerström:** Successiv resultatavräkning av pågående arbeten. - Fallstudier i tre byggföretag, 1995.

No 478 **Peter Jonsson:** Complexity of State-Variable Planning under Structural Restrictions, 1995.

FHS 7/95 **Anders Avdic:** Arbetsintegrerad systemutveckling med kalkylprogram, 1995.

No 482 **Eva L Ragnemalm:** Towards Student Modelling through Collaborative Dialogue with a Learning Companion, 1995.

No 488 **Eva Toller:** Contributions to Parallel Multiparadigm Languages: Combining Object-Oriented and Rule-Based Programming, 1995.

No 489 **Erik Stoy:** A Petri Net Based Unified Representation for Hardware/Software Co-Design, 1995.

No 497 **Johan Herber:** Environment Support for Building Structured Mathematical Models, 1995.

No 498 **Stefan Svenberg:** Structure-Driven Derivation of Inter-Lingual Functor-Argument Trees for Multi-Lingual Generation, 1995.

No 503 **Hee-Cheol Kim:** Prediction and Postdiction under Uncertainty, 1995.

FHS 8/95 **Dan Fristedt:** Metoder i användning - mot förbättring av systemutveckling genom situationell metodkunskap och metodanalys, 1995.

FHS 9/95 **Malin Bergvall:** Systemförvaltning i praktiken - en kvalitativ studie avseende centrala begrepp, aktiviteter och ansvarsroller, 1995.

No 513 **Joachim Karlsson:** Towards a Strategy for Software Requirements Selection, 1995.

No 517 **Jakob Axelsson:** Schedulability-Driven Partitioning of Heterogeneous Real-Time Systems, 1995.

No 518 **Göran Forslund:** Toward Cooperative Advice-Giving Systems: The Expert Systems Experience, 1995.

No 522 **Jörgen Andersson:** Bilder av småföretagares ekonomistyrning, 1995.

No 538 **Staffan Flodin:** Efficient Management of Object-Oriented Queries with Late Binding, 1996.

No 545 **Vadim Engelson:** An Approach to Automatic Construction of Graphical User Interfaces for Applications in Scientific Computing, 1996.

No 546 **Magnus Werner :** Multidatabase Integration using Polymorphic Queries and Views, 1996.

FiF-a 1/96 **Mikael Lind:** Affärsprocessinriktad förändringsanalys - utveckling och tillämpning av synsätt och metod, 1996.

No 549 **Jonas Hallberg:** High-Level Synthesis under Local Timing Constraints, 1996.

No 550 **Kristina Larsen:** Förutsättningar och begränsningar för arbete på distans - erfarenheter från fyra svenska företag. 1996.

No 557 **Mikael Johansson:** Quality Functions for Requirements Engineering Methods, 1996.

No 558 **Patrik Nordling:** The Simulation of Rolling Bearing Dynamics on Parallel Computers, 1996.

No 561 **Anders Ekman:** Exploration of Polygonal Environments, 1996.

No 563 **Niclas Andersson:** Compilation of Mathematical Models to Parallel Code, 1996.

No 567 **Johan Jenvald:** Simulation and Data Collection in Battle Training, 1996.

No 575 **Niclas Ohlsson:** Software Quality Engineering by Early Identification of Fault-Prone Modules, 1996.

No 576 **Mikael Ericsson:** Commenting Systems as Design Support—A Wizard-of-Oz Study, 1996.

No 587 **Jörgen Lindström:** Chefers användning av kommunikationsteknik, 1996.

No 589 **Esa Falkenroth:** Data Management in Control Applications - A Proposal Based on Active Database Systems, 1996.

No 591 **Niclas Wahllöf:** A Default Extension to Description Logics and its Applications, 1996.

No 595 **Annika Larsson:** Ekonomisk Styrning och Organisatorisk Passion - ett interaktivt perspektiv, 1997.

No 597 **Ling Lin:** A Value-based Indexing Technique for Time Sequences, 1997.

No 598 **Rego Granlund:** C^3Fire - A Microworld Supporting Emergency Management Training, 1997.

No 599 **Peter Ingels:** A Robust Text Processing Technique Applied to Lexical Error Recovery, 1997.

No 607 **Per-Arne Persson:** Toward a Grounded Theory for Support of Command and Control in Military Coalitions, 1997.

No 609 **Jonas S Karlsson:** A Scalable Data Structure for a Parallel Data Server, 1997.

FiF-a 4 **Carita Åbom:** Videomötesteknik i olika affärssituationer - möjligheter och hinder, 1997.

FiF-a 6 **Tommy Wedlund**: Att skapa en företagsanpassad systemutvecklingsmodell - genom rekonstruktion, värdering och vidareutveckling i T50-bolag inom ABB, 1997.

No 615 **Silvia Coradeschi**: A Decision-Mechanism for Reactive and Coordinated Agents, 1997.

No 623 **Jan Ollinen:** Det flexibla kontorets utveckling på Digital - Ett stöd för multiflex? 1997.

No 626 **David Byers:** Towards Estimating Software Testability Using Static Analysis, 1997.

No 627 **Fredrik Eklund:** Declarative Error Diagnosis of GAPLog Programs, 1997.

No 629 **Gunilla Ivefors:** Krigsspel och Informationsteknik inför en oförutsägbar framtid, 1997**.**

No 631 **Jens-Olof Lindh:** Analysing Traffic Safety from a Case-Based Reasoning Perspective, 1997

No 639 **Jukka Mäki-Turja:**. Smalltalk - a suitable Real-Time Language, 1997.

No 640 **Juha Takkinen**: CAFE: Towards a Conceptual Model for Information Management in Electronic Mail, 1997.

No 643 **Man Lin**: Formal Analysis of Reactive Rule-based Programs, 1997.

No 653 **Mats Gustafsson**: Bringing Role-Based Access Control to Distributed Systems, 1997.

FiF-a 13 **Boris Karlsson:** Metodanalys för förståelse och utveckling av systemutvecklingsverksamhet. Analys och värdering av systemutvecklingsmodeller och dess användning, 1997.

No 674 **Marcus Bjäreland:** Two Aspects of Automating Logics of Action and Change - Regression and Tractability, 1998.

No 676 **Jan Håkegård**: Hierarchical Test Architecture and Board-Level Test Controller Synthesis, 1998.

No 668 **Per-Ove Zetterlund**: Normering av svensk redovisning - En studie av tillkomsten av Redovisningsrådets rekommendation om koncernredovisning (RR01:91), 1998.

No 675 **Jimmy Tjäder**: Projektledaren & planen - en studie av projektledning i tre installations- och systemutvecklingsprojekt, 1998.

FiF-a 14 **Ulf Melin**: Informationssystem vid ökad affärs- och processorientering - egenskaper, strategier och utveckling, 1998.

No 695 **Tim Heyer**: COMPASS: Introduction of Formal Methods in Code Development and Inspection, 1998.

No 700 **Patrik Hägglund:** Programming Languages for Computer Algebra, 1998.

FiF-a 16 **Marie-Therese Christiansson:** Inter-organisatorisk verksamhetsutveckling - metoder som stöd vid utveckling av partnerskap och informationssystem, 1998.

No 712 **Christina Wennestam:** Information om immateriella resurser. Investeringar i forskning och utveckling samt i personal inom skogsindustrin, 1998.

No 719 **Joakim Gustafsson:** Extending Temporal Action Logic for Ramification and Concurrency, 1998.

No 723 **Henrik André-Jönsson:** Indexing time-series data using text indexing methods, 1999.

No 725 **Erik Larsson:** High-Level Testability Analysis and Enhancement Techniques, 1998.

No 730 **Carl-Johan Westin:** Informationsförsörjning: en fråga om ansvar - aktiviteter och uppdrag i fem stora svenska organisationers operativa informationsförsörjning, 1998.

No 731 **Åse Jansson:** Miljöhänsyn - en del i företags styrning, 1998.

No 733 **Thomas Padron-McCarthy:** Performance-Polymorphic Declarative Queries, 1998.

No 734 **Anders Bäckström:** Värdeskapande kreditgivning - Kreditriskhantering ur ett agentteoretiskt perspektiv, 1998.

FiF-a 21 **Ulf Seigerroth:** Integration av förändringsmetoder - en modell för välgrundad metodintegration, 1999.

FiF-a 22 **Fredrik Öberg:** Object-Oriented Frameworks - A New Strategy for Case Tool Development, 1998.

No 737 **Jonas Mellin:** Predictable Event Monitoring, 1998.

No 738 **Joakim Eriksson:** Specifying and Managing Rules in an Active Real-Time Database System, 1998.

FiF-a 25 **Bengt E W Andersson:** Samverkande informationssystem mellan aktörer i offentliga åtaganden - En teori om aktörsarenor i samverkan om utbyte av information, 1998.

No 742 **Pawel Pietrzak:** Static Incorrectness Diagnosis of CLP (FD), 1999.

No 748 **Tobias Ritzau:** Real-Time Reference Counting in RT-Java, 1999.

No 751 **Anders Ferntoft:** Elektronisk affärskommunikation - kontaktkostnader och kontaktprocesser mellan kunder och leverantörer på producentmarknader, 1999.

No 752 **Jo Skåmedal:** Arbete på distans och arbetsformens påverkan på resor och resmönster, 1999.

No 753 **Johan Alvehus:** Mötets metaforer. En studie av berättelser om möten, 1999.

No 754 **Magnus Lindahl:** Bankens villkor i låneavtal vid kreditgivning till högt belånade företagsförvärv: En studie ur ett agentteoretiskt perspektiv, 2000.

No 766 **Martin V. Howard:** Designing dynamic visualizations of temporal data, 1999.

No 769 **Jesper Andersson:** Towards Reactive Software Architectures, 1999.

No 775 **Anders Henriksson:** Unique kernel diagnosis, 1999.

FiF-a 30 **Pär J. Ågerfalk:** Pragmatization of Information Systems - A Theoretical and Methodological Outline, 1999.

No 787 **Charlotte Björkegren:** Learning for the next project - Bearers and barriers in knowledge transfer within an organisation, 1999.

No 788 **Håkan Nilsson:** Informationsteknik som drivkraft i granskningsprocessen - En studie av fyra revisionsbyråer, 2000.

No 790 **Erik Berglund:** Use-Oriented Documentation in Software Development, 1999.

No 791 **Klas Gäre:** Verksamhetsförändringar i samband med IS-införande, 1999.

No 800 **Anders Subotic:** Software Quality Inspection, 1999.

No 807 **Svein Bergum**: Managerial communication in telework, 2000.

No 809 **Flavius Gruian:** Energy-Aware Design of Digital Systems, 2000.

FiF-a 32 **Karin Hedström:** Kunskapsanvändning och kunskapsutveckling hos verksamhetskonsulter - Erfarenheter från ett FOU-samarbete, 2000.

No 808 **Linda Askenäs:** Affärssystemet - En studie om teknikens aktiva och passiva roll i en organisation, 2000.

No 820 **Jean Paul Meynard:** Control of industrial robots through high-level task programming, 2000.

No 823 **Lars Hult:** Publika Gränsytor - ett designexempel, 2000.

No 832 **Paul Pop:** Scheduling and Communication Synthesis for Distributed Real-Time Systems, 2000.

FiF-a 34 **Göran Hultgren:** Nätverksinriktad Förändringsanalys - perspektiv och metoder som stöd för förståelse och utveckling av affärsrelationer och informationssystem, 2000.

No 842 **Magnus Kald:** The role of management control systems in strategic business units, 2000.

No 844 **Mikael Cäker:** Vad kostar kunden? Modeller för intern redovisning, 2000.

FiF-a 37 **Ewa Braf**: Organisationers kunskapsverksamheter - en kritisk studie av ”knowledge management”, 2000.

FiF-a 40 **Henrik Lindberg:** Webbaserade affärsprocesser - Möjligheter och begränsningar, 2000.

FiF-a 41 **Benneth Christiansson:** Att komponentbasera informationssystem - Vad säger teori och praktik?, 2000.

No. 854 **Ola Pettersson:** Deliberation in a Mobile Robot, 2000.

No 863 **Dan Lawesson**: Towards Behavioral Model Fault Isolation for Object Oriented Control Systems, 2000.

No 881 **Johan Moe:** Execution Tracing of Large Distributed Systems, 2001.

No 882 **Yuxiao Zhao:** XML-based Frameworks for Internet Commerce and an Implementation of B2B e-procurement, 2001.

No 890 **Annika Flycht-Eriksson:** Domain Knowledge Management in Information-providing Dialogue systems, 2001.

FiF-a 47 **Per-Arne Segerkvis**t: Webbaserade imaginära organisationers samverkansformer: Informationssystemarkitektur och aktörssamverkan som förutsättningar för affärsprocesser, 2001.

No 894 **Stefan Svarén:** Styrning av investeringar i divisionaliserade företag - Ett koncernperspektiv, 2001.

No 906 **Lin Han**: Secure and Scalable E-Service Software Delivery, 2001.

No 917 **Emma Hansson:** Optionsprogram för anställda - en studie av svenska börsföretag, 2001.

No 916 **Susanne Odar:** IT som stöd för strategiska beslut, en studie av datorimplementerade modeller av verksamhet som stöd för beslut om anskaffning av JAS 1982, 2002.

FiF-a-49 **Stefan Holgersson:** IT-system och filtrering av verksamhetskunskap - kvalitetsproblem vid analyser och beslutsfattande som bygger på uppgifter hämtade från polisens IT-system, 2001.

FiF-a-51 **Per Oscarsson:** Informationssäkerhet i verksamheter - begrepp och modeller som stöd för förståelse av informationssäkerhet och dess hantering, 2001.

No 919 **Luis Alejandro Cortes:** A Petri Net Based Modeling and Verification Technique for Real-Time Embedded Systems, 2001.

No 915 **Niklas Sandell:** Redovisning i skuggan av en bankkris - Värdering av fastigheter. 2001.

No 931 **Fredrik Elg:** Ett dynamiskt perspektiv på individuella skillnader av heuristisk kompetens, intelligens, mentala modeller, mål och konfidens i kontroll av mikrovärlden Moro, 2002.

No 933 **Peter Aronsson:** Automatic Parallelization of Simulation Code from Equation Based Simulation Languages, 2002.

No 938 **Bourhane Kadmiry**: Fuzzy Control of Unmanned Helicopter, 2002.

No 942 **Patrik Haslum**: Prediction as a Knowledge Representation Problem: A Case Study in Model Design, 2002.

No 956 **Robert Sevenius:** On the instruments of governance - A law & economics study of capital instruments in limited liability companies, 2002.

FiF-a 58 **Johan Petersson:** Lokala elektroniska marknadsplatser - informationssystem för platsbundna affärer, 2002.

No 964 **Peter Bunus:** Debugging and Structural Analysis of Declarative Equation-Based Languages, 2002.

No 973 **Gert Jervan:** High-Level Test Generation and Built-In Self-Test Techniques for Digital Systems, 2002.

No 958 **Fredrika Berglund:** Management Control and Strategy - a Case Study of Pharmaceutical Drug Development, 2002.

FiF-a 61 **Fredrik Karlsson:** Meta-Method for Method Configuration - A Rational Unified Process Case, 2002.

No 985 **Sorin Manolache:** Schedulability Analysis of Real-Time Systems with Stochastic Task Execution Times, 2002.

No 982 **Diana Szentiványi:** Performance and Availability Trade-offs in Fault-Tolerant Middleware, 2002.

No 989 **Iakov Nakhimovski:** Modeling and Simulation of Contacting Flexible Bodies in Multibody Systems, 2002.

No 990 **Levon Saldamli:** PDEModelica - Towards a High-Level Language for Modeling with Partial Differential Equations, 2002.

No 991 **Almut Herzog:** Secure Execution Environment for Java Electronic Services, 2002.

No 999 **Jon Edvardsson:** Contributions to Program- and Specification-based Test Data Generation, 2002.

No 1000 **Anders Arpteg:** Adaptive Semi-structured Information Extraction, 2002.

No 1001 **Andrzej Bednarski:** A Dynamic Programming Approach to Optimal Retargetable Code Generation for Irregular Architectures, 2002.

No 988 **Mattias Arvola:** Good to use! : Use quality of multi-user applications in the home, 2003.

FiF-a 62 **Lennart Ljung:** Utveckling av en projektivitetsmodell - om organisationers förmåga att tillämpa projektarbetsformen, 2003.

No 1003 **Pernilla Qvarfordt:** User experience of spoken feedback in multimodal interaction, 2003.

No 1005 **Alexander Siemers:** Visualization of Dynamic Multibody Simulation With Special Reference to Contacts, 2003.

No 1008 **Jens Gustavsson:** Towards Unanticipated Runtime Software Evolution, 2003.

No 1010 **Calin Curescu:** Adaptive QoS-aware Resource Allocation for Wireless Networks, 2003.

No 1015 **Anna Andersson:** Management Information Systems in Process-oriented Healthcare Organisations, 2003.

No 1018 **Björn Johansson:** Feedforward Control in Dynamic Situations, 2003.

No 1022 **Traian Pop:** Scheduling and Optimisation of Heterogeneous Time/Event-Triggered Distributed Embedded Systems, 2003.

FiF-a 65 **Britt-Marie Johansson:** Kundkommunikation på distans - en studie om kommunikationsmediets betydelse i affärstransaktioner, 2003.

No 1024 **Aleksandra Tešanovic:** Towards Aspectual Component-Based Real-Time System Development, 2003.

No 1034 **Arja Vainio-Larsson**: Designing for Use in a Future Context - Five Case Studies in Retrospect, 2003.

No 1033 **Peter Nilsson:** Svenska bankers redovisningsval vid reservering för befarade kreditförluster - En studie vid införandet av nya redovisningsregler, 2003.

FiF-a 69 **Fredrik Ericsson:** Information Technology for Learning and Acquiring of Work Knowledge, 2003.

No 1049 **Marcus Comstedt:** Towards Fine-Grained Binary Composition through Link Time Weaving, 2003.

No 1052 **Åsa Hedenskog:** Increasing the Automation of Radio Network Control, 2003.

No 1054 **Claudiu Duma:** Security and Efficiency Tradeoffs in Multicast Group Key Management, 2003.

FiF-a 71 **Emma Eliason:** Effektanalys av IT-systems handlingsutrymme, 2003.

No 1055 **Carl Cederberg:** Experiments in Indirect Fault Injection with Open Source and Industrial Software, 2003.

No 1058 **Daniel Karlsson:** Towards Formal Verification in a Component-based Reuse Methodology, 2003.

FiF-a 73 **Anders Hjalmarsson:** Att etablera och vidmakthålla förbättringsverksamhet - behovet av koordination och interaktion vid förändring av systemutvecklingsverksamheter, 2004.

No 1079 **Pontus Johansson:** Design and Development of Recommender Dialogue Systems, 2004.

No 1084 **Charlotte Stoltz:** Calling for Call Centres - A Study of Call Centre Locations in a Swedish Rural Region, 2004.

FiF-a 74 **Björn Johansson:** Deciding on Using Application Service Provision in SMEs, 2004.

No 1094 **Genevieve Gorrell:** Language Modelling and Error Handling in Spoken Dialogue Systems, 2004.

No 1095 **Ulf Johansson:** Rule Extraction - the Key to Accurate and Comprehensible Data Mining Models, 2004.

No 1099 **Sonia Sangari:** Computational Models of Some Communicative Head Movements, 2004.

No 1110 **Hans Nässla:** Intra-Family Information Flow and Prospects for Communication Systems, 2004.

No 1116 **Henrik Sällberg:** On the value of customer loyalty programs - A study of point programs and switching costs, 2004.

FiF-a 77 **Ulf Larsson:** Designarbete i dialog - karaktärisering av interaktionen mellan användare och utvecklare i en systemutvecklingsprocess, 2004.

No 1126 **Andreas Borg:** Contribution to Management and Validation of Non-Functional Requirements, 2004.

No 1127 **Per-Ola Kristensson:** Large Vocabulary Shorthand Writing on Stylus Keyboard, 2004.

No 1132 **Pär-Anders Albinsson:** Interacting with Command and Control Systems: Tools for Operators and Designers, 2004.

No 1130 **Ioan Chisalita:** Safety-Oriented Communication in Mobile Networks for Vehicles, 2004.

No 1138 **Thomas Gustafsson:** Maintaining Data Consistency in Embedded Databases for Vehicular Systems, 2004.

No 1149 **Vaida Jakoniené:** A Study in Integrating Multiple Biological Data Sources, 2005.

No 1156 **Abdil Rashid Mohamed**: High-Level Techniques for Built-In Self-Test Resources Optimization, 2005.

No 1162 **Adrian Pop:** Contributions to Meta-Modeling Tools and Methods, 2005.

No 1165 **Fidel Vascós Palacios:** On the information exchange between physicians and social insurance officers in the sick leave process: an Activity Theoretical perspective, 2005.

FiF-a 84 **Jenny Lagsten:** Verksamhetsutvecklande utvärdering i informationssystemprojekt, 2005.

No 1166 **Emma Larsdotter Nilsson:** Modeling, Simulation, and Visualization of Metabolic Pathways Using Modelica, 2005.

No 1167 **Christina Keller:** Virtual Learning Environments in higher education. A study of students' acceptance of educational technology, 2005.

No 1168 **Cécile Åberg:** Integration of organizational workflows and the Semantic Web, 2005.

FiF-a 85 **Anders Forsman:** Standardisering som grund för informationssamverkan och IT-tjänster - En fallstudie baserad på trafikinformationstjänsten RDS-TMC, 2005.

No 1171 **Yu-Hsing Huang:** A systemic traffic accident model, 2005.

FiF-a 86 **Jan Olausson:** Att modellera uppdrag - grunder för förståelse av processinriktade informationssystem i transaktionsintensiva verksamheter, 2005.

No 1172 **Petter Ahlström**: Affärsstrategier för seniorbostadsmarknaden, 2005.

No 1183 **Mathias Cöster**: Beyond IT and Productivity - How Digitization Transformed the Graphic Industry, 2005.

No 1184 **Åsa Horzella**: Beyond IT and Productivity - Effects of Digitized Information Flows in Grocery Distribution, 2005.

No 1185 **Maria Kollberg**: Beyond IT and Productivity - Effects of Digitized Information Flows in the Logging Industry, 2005.

No 1190 **David Dinka**: Role and Identity - Experience of technology in professional settings, 2005.

No 1191 **Andreas Hansson**: Increasing the Storage Capacity of Recursive Auto-associative Memory by Segmenting Data, 2005.

No 1192 **Nicklas Bergfeldt:** Towards Detached Communication for Robot Cooperation, 2005.

No 1194 **Dennis Maciuszek:** Towards Dependable Virtual Companions for Later Life, 2005.

No 1204 **Beatrice Alenljung:** Decision-making in the Requirements Engineering Process: A Human-centered Approach, 2005.

No 1206 **Anders Larsson:** System-on-Chip Test Scheduling and Test Infrastructure Design, 2005.

No 1207 **John Wilander:** Policy and Implementation Assurance for Software Security, 2005.

No 1209 **Andreas Käll:** Översättningar av en managementmodell - En studie av införandet av Balanced Scorecard i ett landsting, 2005.

No 1225 **He Tan:** Aligning and Merging Biomedical Ontologies, 2006.

No 1228 **Artur Wilk:** Descriptive Types for XML Query Language Xcerpt, 2006.

No 1229 **Per Olof Pettersson:** Sampling-based Path Planning for an Autonomous Helicopter, 2006.

No 1231 **Kalle Burbeck:** Adaptive Real-time Anomaly Detection for Safeguarding Critical Networks, 2006.

No 1233 **Daniela Mihailescu:** Implementation Methodology in Action: A Study of an Enterprise Systems Implementation Methodology, 2006.

No 1244 **Jörgen Skågeby:** Public and Non-public gifting on the Internet, 2006.

No 1248 **Karolina Eliasson:** The Use of Case-Based Reasoning in a Human-Robot Dialog System, 2006.

No 1263 **Misook Park-Westman:** Managing Competence Development Programs in a Cross-Cultural Organisation - What are the Barriers and Enablers, 2006.

FiF-a 90 **Amra Halilovic:** Ett praktikperspektiv på hantering av mjukvarukomponenter, 2006.

No 1272 **Raquel Flodström**: A Framework for the Strategic Management of Information Technology, 2006.

No 1277 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Embedded Systems, 2006.

No 1283 **Håkan Hasewinkel:** A Blueprint for Using Commercial Games off the Shelf in Defence Training, Education and Research Simulations, 2006.

FiF-a 91 **Hanna Broberg:** Verksamhetsanpassade IT-stöd - Designteori och metod, 2006.

No 1286 **Robert Kaminski:** Towards an XML Document Restructuring Framework, 2006.

No 1293 **Jiri Trnka:** Prerequisites for data sharing in emergency management, 2007.

No 1302 **Björn Hägglund:** A Framework for Designing Constraint Stores, 2007.

No 1303 **Daniel Andreasson**: Slack-Time Aware Dynamic Routing Schemes for On-Chip Networks, 2007.

No 1305 **Magnus Ingmarsson:** Modelling User Tasks and Intentions for Service Discovery in Ubiquitous Computing, 2007.

No 1306 **Gustaf Svedjemo**: Ontology as Conceptual Schema when Modelling Historical Maps for Database Storage, 2007.

No 1307 **Gianpaolo Conte**: Navigation Functionalities for an Autonomous UAV Helicopter, 2007.

No 1309 **Ola Leifler**: User-Centric Critiquing in Command and Control: The DKExpert and ComPlan Approaches, 2007.

No 1312 **Henrik Svensson**: Embodied simulation as off-line representation, 2007.

No 1313 **Zhiyuan He:** System-on-Chip Test Scheduling with Defect-Probability and Temperature Considerations, 2007.

No 1317 **Jonas Elmqvist:** Components, Safety Interfaces and Compositional Analysis, 2007.

No 1320 **Håkan Sundblad:** Question Classification in Question Answering Systems, 2007.

No 1323 **Magnus Lundqvist**: Information Demand and Use: Improving Information Flow within Small-scale Business Contexts, 2007.

No 1329 **Martin Magnusson:** Deductive Planning and Composite Actions in Temporal Action Logic, 2007.

No 1331 **Mikael Asplund:** Restoring Consistency after Network Partitions, 2007.

No 1332 **Martin Fransson:** Towards Individualized Drug Dosage - General Methods and Case Studies, 2007.

No 1333 **Karin Camara:** A Visual Query Language Served by a Multi-sensor Environment, 2007.

No 1337 **David Broman:** Safety, Security, and Semantic Aspects of Equation-Based Object-Oriented Languages and Environments, 2007.

No 1339 **Mikhail Chalabine:** Invasive Interactive Parallelization, 2007.

No 1351 **Susanna Nilsson:** A Holistic Approach to Usability Evaluations of Mixed Reality Systems, 2008.

No 1353 **Shanai Ardi:** A Model and Implementation of a Security Plug-in for the Software Life Cycle, 2008.

No 1356 **Erik Kuiper:** Mobility and Routing in a Delay-tolerant Network of Unmanned Aerial Vehicles, 2008.

No 1359 **Jana Rambusch**: Situated Play, 2008.

No 1361 **Martin Karresand:** Completing the Picture - Fragments and Back Again, 2008.

No 1363 **Per Nyblom:** Dynamic Abstraction for Interleaved Task Planning and Execution, 2008.

No 1371 **Fredrik Lantz:**Terrain Object Recognition and Context Fusion for Decision Support, 2008.

No 1373 **Martin Östlund:** Assistance Plus: 3D-mediated Advice-giving on Pharmaceutical Products, 2008.

No 1381 **Håkan Lundvall:** Automatic Parallelization using Pipelining for Equation-Based Simulation Languages, 2008.

No 1386 **Mirko Thorstensson:** Using Observers for Model Based Data Collection in Distributed Tactical Operations, 2008.

No 1387 **Bahlol Rahimi:** Implementation of Health Information Systems, 2008.

No 1392 **Maria Holmqvist:** Word Alignment by Re-using Parallel Phrases, 2008.

No 1393 **Mattias Eriksson**: Integrated Software Pipelining, 2009.

No 1401 **Annika Öhgren:** Towards an Ontology Development Methodology for Small and Medium-sized Enterprises, 2009.

No 1410 **Rickard Holsmark:** Deadlock Free Routing in Mesh Networks on Chip with Regions, 2009.

No 1421 **Sara Stymne:** Compound Processing for Phrase-Based Statistical Machine Translation, 2009.

No 1427 **Tommy Ellqvist**: Supporting Scientific Collaboration through Workflows and Provenance, 2009.

No 1450 **Fabian Segelström:** Visualisations in Service Design, 2010.

No 1459 **Min Bao**: System Level Techniques for Temperature-Aware Energy Optimization, 2010.

No 1466 **Mohammad Saifullah**: Exploring Biologically Inspired Interactive Networks for Object Recognition, 2011

No 1468 **Qiang Liu**: Dealing with Missing Mappings and Structure in a Network of Ontologies, 2011.
No 1469 **Ruxandra Pop**: Mapping Concurrent Applications to Multiprocessor Systems with Multithreaded Processors and Network on Chip-Based Interconnections, 2011.
No 1476 **Per-Magnus Olsson**: Positioning Algorithms for Surveillance Using Unmanned Aerial Vehicles, 2011.
No 1481 **Anna Vapen**: Contributions to Web Authentication for Untrusted Computers, 2011.
No 1485 **Loove Broms:** Sustainable Interactions: Studies in the Design of Energy Awareness Artefacts, 2011.
FiF-a 101 **Johan Blomkvist:** Conceptualising Prototypes in Service Design, 2011.
No 1490 **Håkan Warnquist:** Computer-Assisted Troubleshooting for Efficient Off-board Diagnosis, 2011.
No 1503 **Jakob Rosén:** Predictable Real-Time Applications on Multiprocessor Systems-on-Chip, 2011.
No 1504 **Usman Dastgeer:** Skeleton Programming for Heterogeneous GPU-based Systems, 2011.
No 1506 **David Landén:** Complex Task Allocation for Delegation: From Theory to Practice, 2011.
No 1509 **Mariusz Wzorek:** Selected Aspects of Navigation and Path Planning in Unmanned Aircraft Systems, 2011.
No 1510 **Piotr Rudol:** Increasing Autonomy of Unmanned Aircraft Systems Through the Use of Imaging Sensors, 2011.