DOI:10.15774/PPKE.ITK.2014.008

Visual Sense-and-Avoid for Small Size Unmanned Aerial Vehicles

Tamás Zsedrovits

A thesis submitted for the degree of
Doctor of Philosophy

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
Multidisciplinary Technical Sciences Doctoral School

Scientific Advisor:
Ákos Zarándy, D.Sc.

Supervisor:
✝ Tamás Roska, D.Sc.

Budapest, 2014

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

To my lovely wife Kriszti and my sweet daughter Julcsi.

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

Acknowledgment

First of all I would like to thank and acknowledge the support of my scientific advisor and supervisor Ákos Zarándy. He helped me a lot to find my way at the beginning of the PhD, guided me throughout my work and motivated me with his enthusiasm. I would like to thank also Tamás Roska for his great support and fatherly guidance during my studies.

I am grateful to Péter Szolgay, Judit Nyékyné Gaizler, Ferenc Kovács and Ágnes Bércesné Novák for their continuous support and encouragement during the years.

I am also thankful to my closest collaborators for their help, and for all the discussions we had: Bálint Vanek, Zoltán Nagy, András Kiss, Péter Bauer, Tamás Péni, Borbála Pencz, Máté Németh and Arnold Leitli.

I am very grateful to my colleagues especially to those with whom we shared an office: Miklós Koller, Gábor Tornai, Mihály Radványi, Tamás Fülöp, Ádám Rák, András Horváth, Attila Stubendek, Domonkos Gergelyi, Csaba Józsa, László Füredi, Kálmán Tornai, Bence Borbély, István Reguly, Csaba Nemes, László Laki, Ádám Balogh, Norbert Sárkány, Zoltán Tuza, János Rudan, Balázs Varga, Dávid Tisza, Tamás Pilissy, Zoltán Kárász, László Kozák, Andrea Kovács, Dániel Szolgay, Ádám Fekete, András Gelencsér, Dóra Bihary, Zsolt Gelencsér, Antal Hiba, Endre László, Borbála Siklósi, András Rácz, Balázs Oláh, Anna Horváth, András Laki, Imre Juhász and Mátyás Jani for their help in any ways.

I am also thankful to our postdoctoral fellows András Oláh, György Cserey, Kristóf Karacs, Kristóf Iván, Zoltán Gáspári and Miklós Gyöngy for their inspiring work and for their encouraging worlds.

I acknowledge also the kind help of the personnel of Students' Office, in particular Katinka Tivadarné Vida, the Dean's Office, the Financial Department and the IT Department. I am also very thankful to Viktória Sifter from the Library.

I am especially grateful to my wife Kriszti who has encouraged me and believed in me all the time. I am very grateful to my mother, my father and my sister and to my whole family who always believed in me and supported me in all possible ways. I am very grateful to my daughter Julia, who let me work close to the deadlines.

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

Contents

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

List of figures

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

List of tables

DOI:10.15774/PPKE.ITK.2014.008

List of Abbreviations

ACAS	Airborne Collision Avoidance System
ATC	Air Traffic Control
CCD	charge-coupled device
CEO	Chief executive officer
CMO	Close-Minus-Open morphological filter
CUDA	Compute Unified Device Architecture
CV	Collision Volume
DCM	direction cosine matrix
EASA	European Aviation Safety Agency
EKF	Extended Kalman Filter
ELOS	equivalent level of safety
EO	Electro-optical
EU	European Union
FAA	Federal Aviation Administration
FOV	Field of View
FPGA	Field Programmable Gate Array
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	graphical processing unit
HIL	hardware-in-the-loop
HMM	hidden Markov model
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infra-red
ISO	International Organisation for Standardisation
KF	Kalman filter
MAV	Micro Air Vehicle
NAS	National Airspace System (USA)
NED	North-East-Down
NextGen	Next Generation Air Transportation System
OF	Optical Flow
PC	personal computer
RADAR	Radio Detection and Ranging
RC	Radio Controlled
RER	relative entropy rate
ROI	Region of Interest

DOI:10.15774/PPKE.ITK.2014.008

RPAS	Remotely Piloted Aircraft Systems
SAA	Sense-and-avoid
SARSE	Subtended Angle Relative State Estimation
SSD	Solid State Drive
SVM	support vector machine
TCAS	Traffic Collision Avoidance System
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
USA	United States of America
USB	universal serial bus

DOI:10.15774/PPKE.ITK.2014.008

DOI:10.15774/PPKE.ITK.2014.008

Chapter 1

Introduction

1.1 Preface

In the last decade Unmanned Aircraft Systems (UAS) – beforehand Unmanned Aerial Vehicles (UAVs) – technology has evolved considerably. Besides the military applications now there is a great opportunity to use UAS in commercial applications as well [15], [16], [17]. More and more companies start to develop applications and services based on the UAS platform. According to many aviation experts pilotless aircrafts are going to revolutionize air transport in the near future. As written in the cover story of December 2011 issue of IEEE Spectrum Magazine: “A pilotless airliner is going to come; it's just a question of when,” said James Albaugh, the president and CEO of Boeing Commercial Airlines [18]. Recent examples are from the field of agriculture [19], wildlife and eco conservation [20], search and rescue [21], firefighting [22], delivery of small packages [23], tv broadcasts [24] and even a “Flying Companion” for automobiles [25]. Additionally, there are many great ideas which can improve the quality, reliability or cost effectiveness of a service.

1.1.1 UAS challenges

Nevertheless, in order to use UAS in these fields their reliability needs to be increased as well as their capabilities need to be extended further, their ease of use needs to be improved and their cost have to be decreased. At the same time the regulatory challenge of integrating UAS into national and international air space has to be solved [26]. One of the most important problems which has to be solved is the collision avoidance or sense-and-avoid capability [27], [28]. These functions have to be run on-board even if the connection between the aircraft and the control station is lost or some of the on-board sensors fail.

DOI:10.15774/PPKE.ITK.2014.008

Furthermore, the absence of standards, regulations and procedures to govern the safe integration of the UAS into civilian air space are against the broader civilian use. The organizations involved in the regulatory process of the air traffic, like Federal Aviation Administration (FAA), or European Aviation Safety Agency (EASA), do not want to decrease the safety of the air traffic systems currently in use. Thus the UAS have to provide an “equivalent level of safety” (ELOS) to manned aircraft while not negatively impacting the existing infrastructure and manned Traffic Alert and Collision Avoidance System (TCAS) [29], [30]. Additionally, privacy concerns have also been raised about the widespread use of UAS by government and business [31].

1.1.2 UAS integration

Currently the routine operation of UAS with a certification for a civilian task at low altitude or in areas where there is little traffic is allowed only in Japan, Australia, South Africa and some other countries. It is important to remark that in regulations UAS means not only the unmanned aircraft but it also refers to the ground station, all the communication devices and services, all the sensors and computers involved and one or more aircrafts too [32].

In the United States FAA has got a roadmap for integration of UAS into the Next Generation Air Transportation System (NextGen), which should be finished no later than 2025, but requires that UAS could operate with no more than one catastrophic failure in a billion flight hours [33].

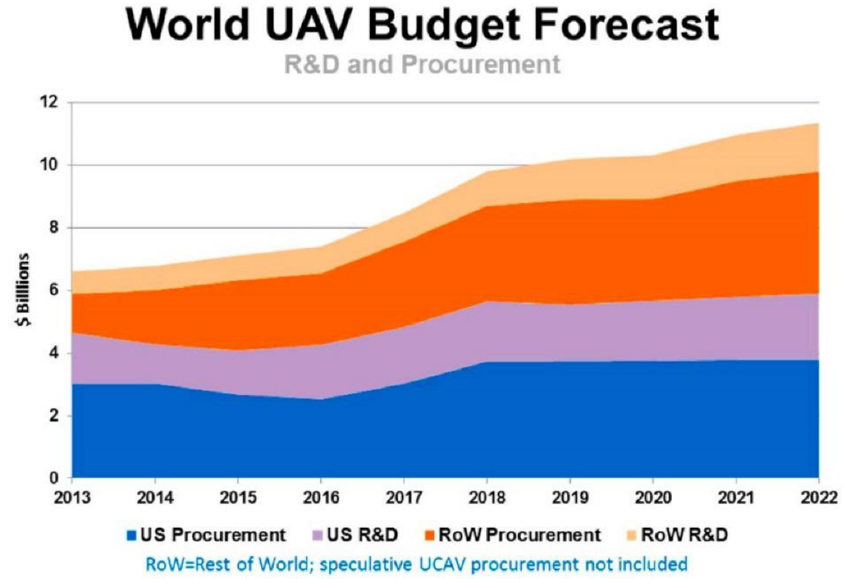
In the European Union the plan is first to integrate small UAS (less than 150 kg) into the non-segregated airspace no later than 2023 with several line-of-sight conditions (including UAVs connected to the remote pilot station through communication services beyond radio line-of-sight) [34].

In Hungary the preliminary negotiations started last year between the academic, industrial partners and legislators about the necessary steps [35]. Currently the Hungarian regulations cannot distinguish between the RC planes used by hobbyists and the commercial UAS [36]. Both can be used outside of populated areas in visual line-of-sight range without any special permission and they can be used elsewhere with the permission of the local authorities. The Hungarian regulations will be harmonised with the EU regulations later on.

DOI:10.15774/PPKE.ITK.2014.008

1.1.3 UAS economic effects

According to the forecasts, the research and development expenditures on UAS are growing in the next decade [37]. Although the main driving force are still the military agencies, the development and manufacture of UAS for use by public entities (i.e., federal, state, or local governments, and by public universities) and commercial users are expected to grow over the next few years [38]. Unfortunately, numerous regulatory and technical issues remain to be resolved before government agencies or commercial operators can begin routine flight operations in the national airspace.



RoW=Rest of World; speculative UCAV procurement not included

Figure 1.1 World UAV Budget Forecast [39]

1.2 Aims and motivations

As stated in the roadmap for remotely piloted aircraft systems (RPAS) in the EU, first the small aircrafts will be integrated [34]. Provided that the size and the energy consumption of the UAV are limited, a camera based avoidance system would provide cost and weight advantages against the systems currently in use on bigger aircrafts, like cooperative systems for example TCAS [29]. Furthermore near airfields, because of a great density of aircrafts and the limited frequency resources of air traffic controllers the camera-based approach seems to be more feasible then others.

Although the camera based solutions has limitations on weather conditions in which they can be used as well, at this first stage of research these are less important than the size and power limitations. Furthermore most of the weather limitations can be handled with

DOI:10.15774/PPKE.ITK.2014.008

supplementary infra-red sensors in the near future, as these sensors are cheaper and cheaper and their resolution is developed a lot in the last decade. Thus the goal was that the developed algorithms work fully in good weather conditions, when the sky is blue or when there are low or medium contrast clouds. These methods work partially, when there are high contrast clouds, which means some additional data on the actual scene is needed.

Today’s kilo-processor chips allow us to implement complex algorithms in real time with low power consumption, so even if we use more sophisticated algorithms for the collision avoidance task, it has a smaller impact on the maximum flight time of the system than the special manoeuvre we have to run in some cases for the position and path estimation of the other aircraft, we want to avoid.

My work was done in a research group funded by the Office of Naval Research (ONR) and ONR Global within the framework of the grant N62909-10-1-7081. The main goal of this research is to develop an autonomous mid-size fixed-wing safety critical UAV for civil applications. The development of the actual hardware went parallel with the development of the algorithmic framework which included the research on vision based UAVs. At the time of this research there were no complete, visual SAA system for mid-size and small UAS and the properties of this kind of systems had been not described yet.

The aim of this work is to introduce and analyse visual methods for the UAS SAA problem. In particular, what kind of information can be extracted from the image flow if the intruder airplane is close enough that is its shape and position can be calculated? This information will be used in the position and path estimation of an aircraft which can be seen on our camera. We want to avoid the run of a special extracting manoeuvre which is used during the estimation, but consumes a significant amount of fuel. On the other hand we wanted to know what can we expect from various visual space reconstruction algorithms in the case of own aircraft’s attitude estimation. This is important because the positions and paths of another aircraft are estimated relative to our camera. The more accurate our own attitude estimation, the better and quicker our estimation process.

In [1]-[14] a camera-based autonomous on-board collision avoidance system and its implementation aspects on kilo-processor architectures are introduced which is the main framework and application of the results are shown in this thesis.

DOI:10.15774/PPKE.ITK.2014.008

1.3 Framework of the dissertation

The organization of the dissertation is as follows. In Chapter 2 the concept and recent developments of the collision avoidance systems for UAS are introduced giving special importance to sense and avoid (SAA) systems. Although these systems are comparable with the results of the whole research project which include the results of this dissertation, this introduction gives a broader perspective in which my work can be put. In Chapter 3 the base ideas and the most important principles are shown which are used in the development of the UAV SAA system, including the used simulation environment and an image processing algorithms for the aircraft detection. This system was the result of the joint work of our research group and formed the basis of my research. In Chapter 4 the relative direction angle estimation algorithm is introduced and the capabilities of the algorithm are shown, which are summarized in the first thesis group in Chapter 6, as this new algorithm is one of the scientific results of my work In Chapter 5 four camera pose estimation algorithms are investigated in simulations. The aim of the investigation is to show the strengths and weaknesses of these algorithms in the aircraft attitude estimation task. The results are summarized in the second thesis group in Chapter 6. Thus Chapter 6 summarizes the new scientific results in this dissertation. Finally, in Chapter 7 the developed UAV platform as the main application target of the results is shown.

DOI:10.15774/PPKE.ITK.2014.008

Chapter 2

UAS Collision Avoidance

In this chapter the concept and recent developments of the collision avoidance systems for UAS are introduced giving special importance to sense and avoid (SAA) systems. The collision avoidance capability is one of the most important features that UAS must have before they are let in the common airspace, for example the National Airspace System (NAS) in the USA. This task has to be run autonomously and on board even if the connection between the aircraft and the base station is lost.

2.1 Collision avoidance

In air traffic management the rules of the safe flight operations are given. In order to reduce the risk of mid-air collisions and prevent accidents caused by wake turbulence, aircrafts have to keep a separation distance (separation minima) from another aircrafts [40]. This separation is well defined in the regulations and maintained by the air traffic controllers (ATC). The given rules take into account different types of aircrafts, different types of safety equipment, as well as different scenarios.

Besides the traffic management rules, there are airborne collision avoidance systems (ACAS). The objective of ACAS is to provide a backup collision avoidance system for the existing conventional air traffic control system without the need of any ground services and to minimize the false alarms, in encounters for which the collision risk does not warrant escape manoeuvres [41]. These methods are considered as cooperative collision avoidance, because the ACAS of the aircrafts, which are participating in the scenario are communicating with each other.

DOI:10.15774/PPKE.ITK.2014.008

However, in general only bigger and most expensive aircrafts are equipped with ACAS. On smaller and cheaper aircrafts for the collision avoidance mainly the pilot is in charge. Most of the time, the safe operation is possible this way as well, because the operation altitude and the maximum speed of these smaller aircrafts is much smaller than the bigger aircrafts. If the two aircrafts are not communicating with each other, the aircrafts have to run non-cooperative collision avoidance. In the case of a human pilot the concept called see and avoid, as in the case of a UAS it called sense and avoid. The different kind of collision avoidance systems form a layered approach, which can be seen in Figure 2.1.

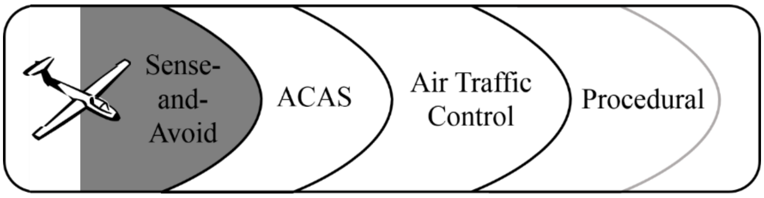


Figure 2.1 The layered concept collision avoidance

In Figure 2.1. the scenario is shown where a manned and an unmanned aircraft come close to each other. The manned aircraft called intruder as it is crossing the path of the UAS. As a typical situation, the manned aircraft is in cruise mode that is a straight path can be assumed. The intruder first has to be detected in some way, and after that its path has to be estimated. To be able to avoid the separation minima, the intruder should be detected from a distance, which is not smaller than the traffic avoidance threshold. If the intruder is not detected before crossing the traffic avoidance threshold, but detected before the collision avoidance threshold, the collision can be still avoided. Because of the small size of the UAV, we can assume that the pilot of the intruder cannot see our aircraft early enough to run an appropriate avoiding manoeuvre.

For human pilots the minimum reaction time from the first time they discovered an object is 12.5 seconds including the recognition of the object, the recognition of the collision risk, the decision, the muscular reaction and the aircraft lag. It means that for a human pilot 12.5 before collision is the last time instant, when collision can be avoided [42]. Naturally, to avoid scaring the pilots and the passengers of the other aircraft, and to increase the safety level, earlier initialization of the avoidance manoeuvre is required, which certainly assumes earlier detection.

It would be better to give the separation minima for a given aircraft category as a requirement for the UAVs, but this is out of the scope of this thesis. Most of the time UAV systems have smaller lag times and are capable of running manoeuvres with higher accelerations, as there is no human pilot on-board.

DOI:10.15774/PPKE.ITK.2014.008

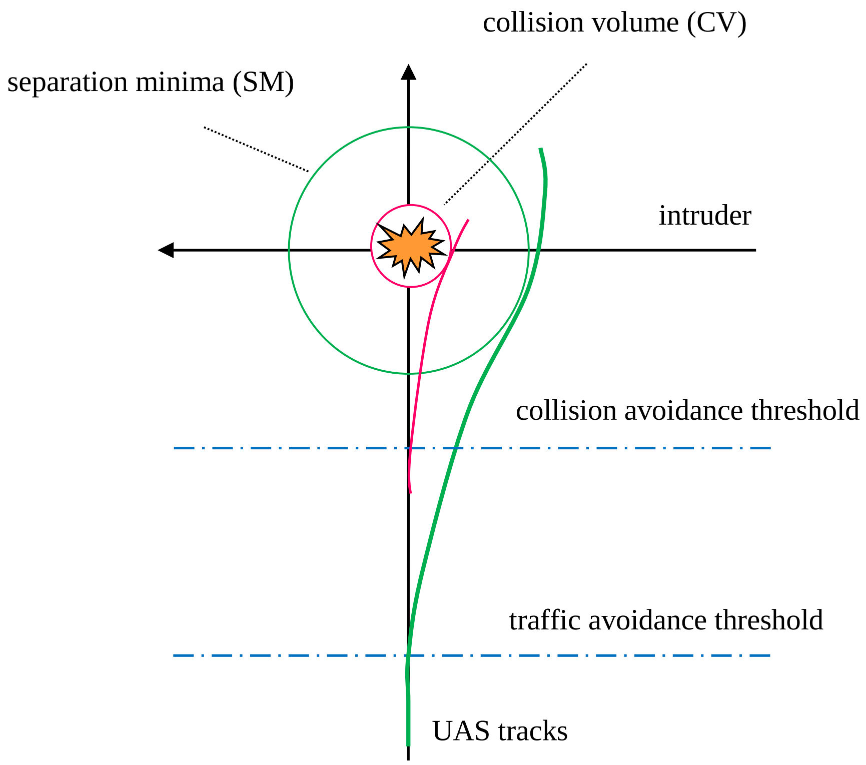


Figure 2.2 Traffic and collision avoidance.

As an example a small or medium size UAS is presented. Since the tracks of the small and medium size UAS usually do not interfere with streamliners, or high speed jets, they have to be prepared for other UAS and small sized manned aircrafts, like the Cessna 172. This means that the expected maximal joint approaching speed is 100 m/s, therefore they should be detected from 2000 meters (20 seconds before collision), to be able to safely avoid them.

2.2 Sense and avoid systems

In the literature there are many approaches to address the SAA problem. The SAA systems are at different levels and the method of solving the problem differs a lot as well. There are partial solutions, which address some aspects of the whole SAA task, like detection, segmentation, tracking or the detection and control. Each of these methods varies with the type (fixed, rotary or flapping wing) and the size of the UAV, as well as with the available sensors and the environment in which the application is run. In [43] several sensor technologies were examined to determine which on can be a good candidate for the main sensor of a UAV SAA system. The tested sensors are: Visual/Pixel, Infrared, Microwave RADAR, LASER RADAR and Bistatic RADAR. Although the visual sensor had the best score among them, the LASER and the Microwave RADAR had similar performance.

DOI:10.15774/PPKE.ITK.2014.008

In the next subsections examples from the literature are shown also with remarks upon the strengths and weaknesses of the particular solution. In 2.2.1 the RADAR based solutions are introduced, and in 2.2.2 the related papers from the bio-motivated SAA are shown, and finally in 2.2.3 the EO based solutions are presented.

2.2.1 RADAR based SAA

In [44] the concept of a RADAR based collision avoidance system for smart UAS is introduced. The main idea of the cooperative and non-cooperative collision avoidance is shown with the current collision avoidance systems for manned aircrafts. The requirements for this system are different from the requirements of our system as the UAV, which is used here is capable of flying with 440 km/h (~122 m/s) and the SAA system can be 25 Kg while in our case the expected joint speed of the intruder and our UAV is 100 m/s (the speed of our UAV is around 40 m/s) and the size of the whole control system is less than 1 kg. The authors show the desirable small-sized and light-weighted RADAR design and the capabilities. This paper shows the feasibility of the solution as the performance meet the ELOS criteria.

The performance of the system is calculated considering the sensor detection ranges and speed and the mean reaction time of a pilot. The work is continued in [45] with simulation of typical scenarios. Simulations show that the probability of the detection is 90% at the given detection range and that the probability of the collision avoidance is more than 85% in the presence of error. The main advantage of this system that it is scalable according to the requirements and the detected objects range information is available. Furthermore, the distance from the intruder can be detected is bigger compared to the EO sensor based systems. Also these systems can be used all time and all weather conditions. The main drawbacks are the size, weight, power consumption and relatively slow data rate (2 Hz).

More recently in [46] and [47] a miniature RADAR system is introduced for miniature unmanned air vehicles (MAV). The system design and concepts are shown in [46]. The system is lightweight (only 230g) and is capable of detecting and identifying aircrafts of many type and size, which meet with our requirements. This first paper shows an indoor test for the system, where the RADAR is put on board of a small rotorcraft and the MAV is fixed to the ground. In this indoor test a conventional type miniature helicopter is detected and identified from 3m. The identification is done by comparing the detected Doppler pattern to a signature database through Sum of Absolute Differences (SAD). The SAD can provide real-time identification, because it is easy to compute. The signature vector is based on the frequencies generated by the target

DOI:10.15774/PPKE.ITK.2014.008

aircraft’s propulsion system. In the paper 3 target vehicles are identified. The main problem with this is that if the database contains more vehicles a more complex algorithm is needed, which has negative effect on the real-time capability of the system. Another drawback is that the RADAR beam should be focused in order to have this high resolution, so it cannot cover the entire area needed for the detection.

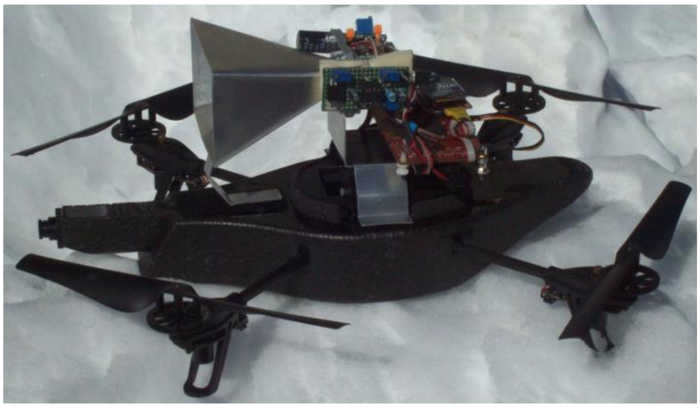


Figure 2.3 Quadrotor equipped with RADAR sensor

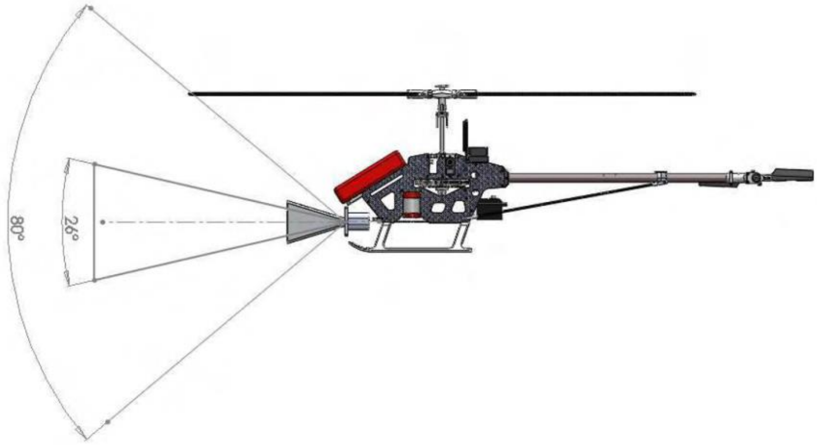


Figure 2.4 RADAR coverage

In [47] the results of outdoor tests are shown. First the indoor tests were repeated in outdoor environment that is the two vehicles stayed on the ground 7m from each other but the engines were operating. In this case the detection rate is 100% as before. This prove the authors hypothesis that the random frequencies produced by the environment does not disturb the measurement significantly. In the final test both vehicles were airborne. In this case the accuracy is dropped significantly due to the fact that it is very difficult to keep the two vehicles in the right position, which shows again that the focused RADAR beam is covers a relatively small area.

DOI:10.15774/PPKE.ITK.2014.008

Another concept is shown in [48]-[51] where the system uses information from RADAR as well as from EO sensor. This way the all-time, all-weather conditions operation can be provided because of the RADAR as well as the desired angular resolution because of the image sensor. The main sensor is a Ka-band pulsed RADAR and the aiding sensors are IR and conventional EO cameras.

In [48] the system architecture for collision avoidance is shown. This system consists of 2 IR and 2 regular EO cameras and a RADAR next to the conventional guidance navigation and control (GNC) system. The paper focuses on the tracker algorithm for the collision avoidance task. It is stated that it is not the accuracy what is important but the reliability of the tracker at short distances, because at long distances the probability of the collision scenario is lower. Different type of Kalman filters (KF) are tested in numerical Monte Carlo simulations, and the Extended Kalman Filter (EKF) based solution is selected as the best compromise between reliability, computational load and accuracy.

In [49] and [50] a multi-sensor-based fully autonomous non-cooperative collision avoidance system for UAS is introduced. This system is developed for a High-Altitude Long-Endurance (HALE) UAV. The size and weight is comparable to a lightweight commercial aircraft. The system is tested on a TECNAM P-92 with wingspan of 8.7m and weight of 450Kg. For the detection Optical flow (OF) and feature point matching was tested. Because of the resolution limitations and the computational cost of the OF, the feature point matching was selected.

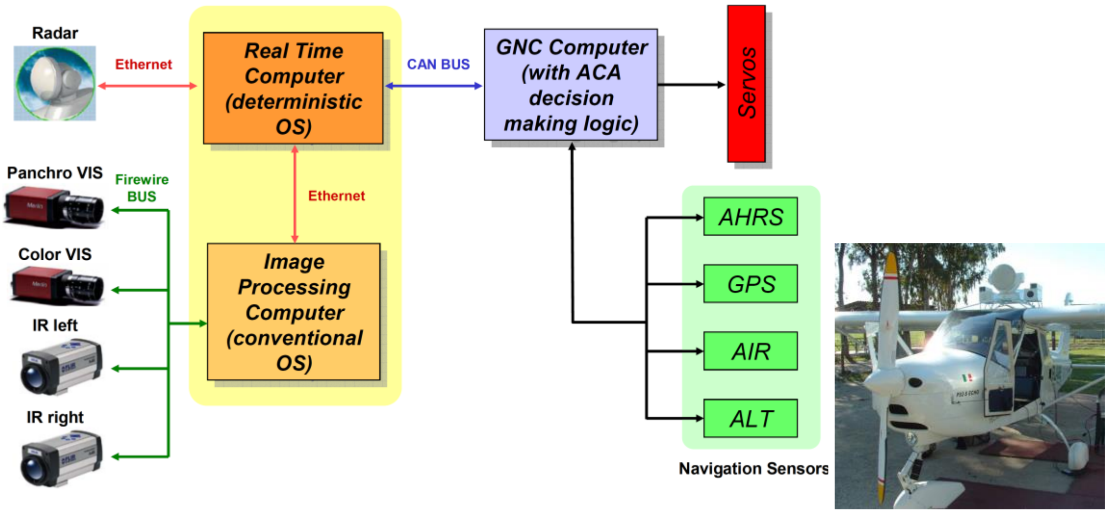

Radar
Ethernet
Real Time Computer (deterministic OS)
CAN BUS
GNC Computer (with ACA decision making logic)
Servos
Panchro VIS
Firewire BUS
Ethernet
Color VIS
Image Processing Computer (conventional OS)
IR left
IR right
AHRS
GPS
AIR
ALT
Navigation Sensors

Figure 2.5 Multi-sensor-based fully autonomous non-cooperative collision avoidance system and the system placed to TECNAM P-92

For the sensor fusion a central-level fusion architecture was selected with decentralized detection. It means that the detection is performed on each sensor separately to avoid the high

DOI:10.15774/PPKE.ITK.2014.008

communicational burden caused by raw data exchange, but the object tracking is run in a unique central-level tracking module. For the tracking an EKF is used with Cartesian coordinates. The concept was tested in numerical simulations and the performance met the requirements. The system was built and it was calibrated. Also preliminary flight tests were performed, where they recorded different scenarios for offline processing.

Finally in [51] flight tests for the RADAR component of the developed system was run. The performance of the tracker was measured by accuracy of the estimated the closest point of approach (CPA). This study showed that the used RADAR is capable of detecting the intruder aircraft reliably. The ranges were compared with GPS measurements. It was shown that on low altitudes there is a significant noise due to the clutter from the ground. This system provided reliable situational awareness at 10 Hz.

It is stated that the detection unit needs a decent navigational unit as the performance of the detection is depends on the accuracy of the navigation, which coincide what we have seen during our work and also confirms that the controls system can benefit from additional visual information. The authors mention that the angular velocity biases did not cause any problem in this case, because the misalignment of the RADAR sensor to the aircraft's body axis did not change with time. On the other hand it is a real problem for the fused system, because the different sensors will have different biases.

The main advantage of this system is that it is capable of running the SAA in all-time all-weather conditions. Due to the camera sensor it is more reliable and more accurate than other RADAR systems. The main drawback of the system is the problem caused by the fusion of different sensors. The system cannot be cheap because of the used sensors, and it is heavy as well, so it cannot be used on a mid-size or small UAV. The computational costs are high as well because of the image processing and the sensor fusion. Furthermore, as it is stated in the last paper, the biases caused by the navigation measurements have significant effect on the performance of the system.

2.2.2 Bio-motivated SAA

The bio-motivated systems focus more on the control and attitude estimation of the UAS. These results can be a good starting point towards a complete SAA system. The main advantages of the bio-motivated systems will be the low power consumption, the small size and the robustness. The main downside can be that the integration of these components into conventional systems is not straightforward. In the following four examples are shown.

DOI:10.15774/PPKE.ITK.2014.008

In [52] a biomimetic visual sensing and control system for MAVs are introduced. In the paper two models are introduced for visual navigation in insects: an optic flow based approach, when the insect uses its compound eye for depth and range sensing and collision avoidance, and another visual sensing based on the ocelli organ for flight stabilization. In the paper it is shown how insects use these sensing information in different tasks, for example for landing or hovering. The available OF sensor chips and artificial ocelli sensor is introduced with the control algorithms. At the time the system was capable of flying at low altitudes (some meters) and following a shallow (±10°) terrain. The development of the system is still in progress. This system is designed for micro air vehicles and for flapping-wing, insect-like robots, which typically fly at low altitudes. The main advantage of the system is that it will be cheap and extremely lightweight. The main drawback is that because of the OF algorithm it cannot be scaled up for a bigger UAS and that it needs special hardware elements (OF chip).

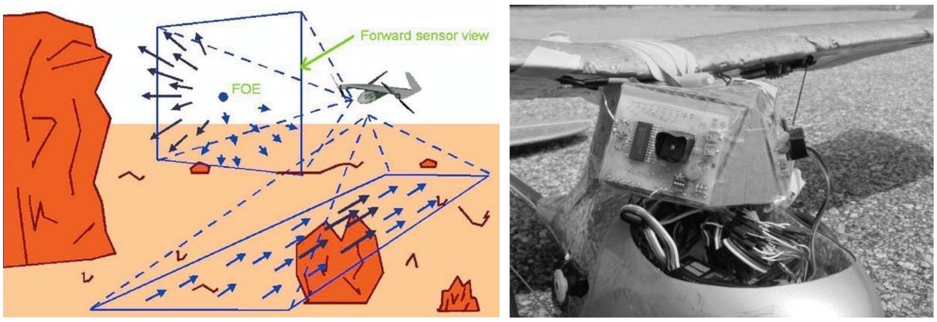


Figure 2.6 Concept of collision avoidance based on OF, and the mounted camera system on a fixed-wing UAV

A biomimetic flight control system for blimp-based UAS is shown in [53]. The system consists of two forward looking CCD cameras with wide angle optics, providing 180° horizontal field of view (FOV). The recorded images are processed at the ground control station. The stabilization and collision avoidance are derived from insect neuronal models. The image processing uses the photoreceptor’s logarithmic rule and the centre-surround antagonism in order to introduce robustness in the system and reduce redundancy. After that two independent processing streams are run parallel to calculate stabilization and collision cues and at the control the collision sues have preference. This system can be used indoor environments and with slowly moving vehicles (blimp) only. In the tests the some black and white patterns were used in order to enhance the contrast, because it needs objects with enough contrast for the robust operation which is another drawback.

DOI:10.15774/PPKE.ITK.2014.008

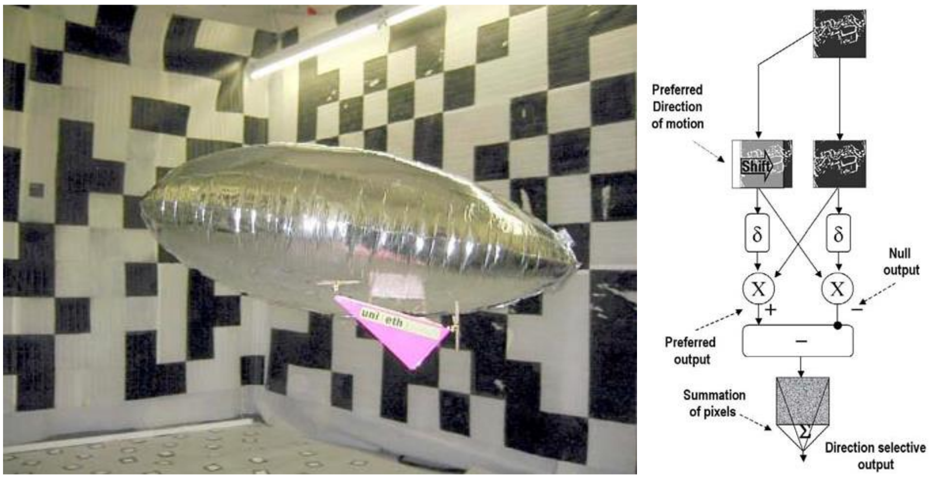


Figure 2.7 Blimp-based UAV in the test environment and biomimetic image processing system

In [54] again an optic flow based lateral collision avoidance is used on a MAV. This MAV is a fixed-wing aircraft, but with hovering capabilities, so it is well suited for low altitude flights and applications like homeland security applications, or search and rescue. The authors uses again models found investigating flying insects. The main problem with the solely optic flow based collision avoidance was that it performs badly when the vehicle was flying directly at low textured obstacles, for example walls. The hovering mode is the authors answer for this problem. The hovering allows the MAV to avoid imminent collisions and also to manoeuvre through tight spaces.

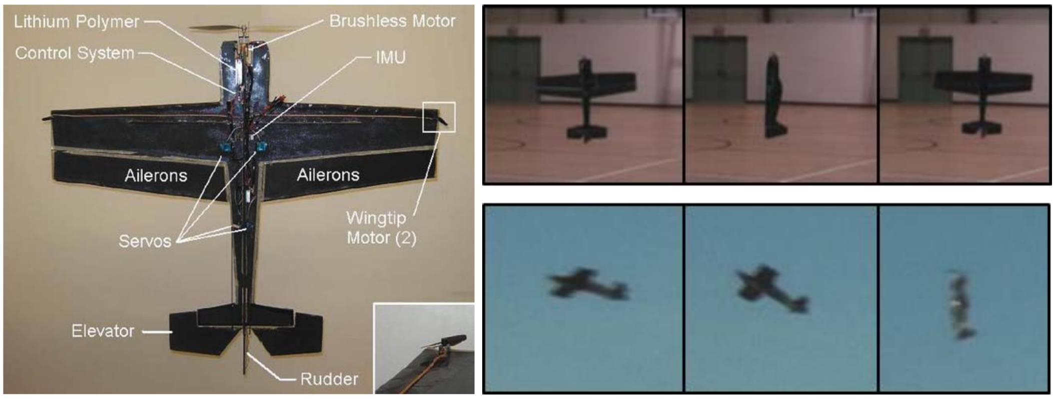


Figure 2.8 Fixed –wing MAV with hovering capability and OF based collision avoidance, autonomous hover and transition from cruise to hover

Besides the concepts and models the developed MAV is also introduced. It has got 1m wingspan, 600g weight and a speed range from 0 to 20 m/s. For the hovering mode roll stabilization additional wingtip motors are installed. The MAV uses an IMU outputting direction quaternions with 100 Hz. It is capable of autonomously hover and autonomously switch between cruise and hover. The authors hope that with an additional proximity sensor (for example

DOI:10.15774/PPKE.ITK.2014.008

ultrasonic distance sensor) installed on the nose the aircraft can automatically switch from cruise to hover when it flies to a wall. The main advantage of this system that it can cruise and hover as well. Although the collision avoidance algorithm is not suited for higher altitude flights than a couple of meters. Another drawback is that additional motors are needed on the wingtips for the hover mode.

The next paper shows the development of biomimetic attitude and orientation sensors [55]. The orientation sensor is based on the polarization of light changes caused by Rayleigh scattering. The polarization is measured by three cameras each of them with different polarization filter. This mimics the function of the dorsal rim area of dragonflies. The developed device was calibrated and tested in static and flight tests. The accuracy of the device is comparable to the accuracy of a conventional magnetic compass sensor. The attitude sensor is based on the ocelli. The function of the ocelli is flight stabilization that is the precise control of the roll and pitch angles. The artificial ocelli consists of four pairs of miniature cameras. Each pair has got a green and an ultra-violet sensor. The tests showed that the roll angle can be controlled by this sensor but the pitch angle was inconsistent. The roll angle error during flight test was less than 2°.

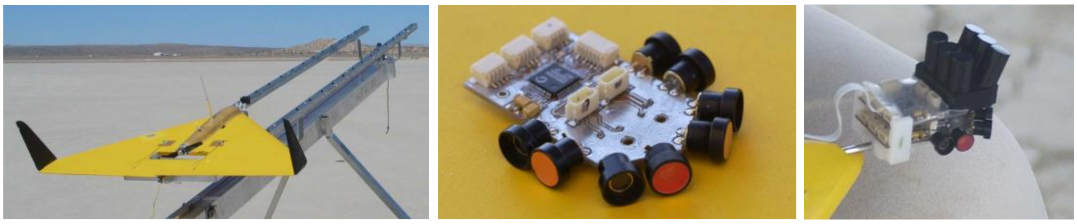


Figure 2.9 UAV housing artificial ocelli sensor and light polarization based compass

2.2.3 EO based SAA

The main advantages of the EO based SAA systems are that they are lightweight and have affordable price. The drawbacks are the relatively high computational cost of the processing algorithms and the restricted weather conditions and range. As the examples show, despite the drawbacks these systems can be a good choice for small UAS.

In [56] the available algorithms and ideas in 2004 are reviewed and a new SAA algorithm is introduced. According to the authors the RADAR sensors were not feasible for the task because of their size and power consumption as well as LASER. SONAR sensors have only a few meter detection range and suffer from multipath propagation and other noise causing effects. They found monocular camera systems as a good candidate for UAS applications.

DOI:10.15774/PPKE.ITK.2014.008

Authors reviewed the state-of-the art image processing algorithms as well. Because of the large depth of field requirements and the fast attitude changes OF algorithm is not good for the purpose. They found feature tracking methods not feasible, because of the fast attitude changes and high computational need and focus of expansion algorithms are not suitable as well, because of the same reasons.

The authors propose a new algorithm, which uses feature density and distribution analysis. The algorithm uses edge and corner features and calculates the time-to-impact based on the expansion rate of feature density and distribution. According to the tests the algorithm is robust to low image quality. On the other hand the algorithm was sensitive to the aircraft's attitude changes. Furthermore the target had to be sufficiently large (bigger than 40% of the image), in order to get good expansion rate, and only one target could be tracked at a time.

In the papers [57]-[63], the development of a computer vision based collision avoidance system is shown. This system is developed at the Queensland University of Technology, Australia, as a part of the Australian Research Centre for Aerospace Automation's Smart Skies research project. The main advantage of this research project is that they have access to various types of aircrafts, sensors and computational resources, and have a big database of flight videos collected in various situations.

In [57] the feasibility study of the vision based collision avoidance system is presented. The system uses a monocular camera as the main sensor for the detection. In this first stage the camera had a 1024x768 resolution and a 17°x13° FOV. For the detection a Close-Minus-Open (CMO) morphological filter is used. This approach finds both bright and dark objects on grayscale images using the grayscale version of the close and the open filter. The output of the CMO still contains significant amount of false targets due to image noise. In order to filter out most of the false targets a dynamic programming algorithm is used.

The algorithm was tested on image sequence contains a distant aircraft and heavy cloud clutter in the background. The results showed that the method is feasible for the collision avoidance. Problems caused by moving platform are not addressed in this stage, but the authors propose of the use of the inertial sensor measurements for supressing the effect of the camera motion later.

In [58] the CMO based algorithm is compared with another morphological filter, the so called Preserved-Sign (PS) filtering. The PS is very similar to the CMO except it preserves the sign of the features, and this way the image noise can be characterised with a zero mean

DOI:10.15774/PPKE.ITK.2014.008

Gaussian function, which improves the performance of the subsequent noise filtering. In the paper the performance of the CMO approach is compared to a human observer.

It is shown that the algorithm performed better than the human observer even in the cloudy situation. The consistent target detection of the algorithm was 19% further than the human detection distance. The test of the two different filtering approach showed that the PS performs slightly better, but the additional computational cost is too high. The ego motion compensation is still mentioned as a problem for the future development.

In [59] a new hidden Markov model (HMM) temporal filtering for the detection is introduced with the addition of relative bearing and elevation estimation capabilities. Additionally, the algorithm is implemented on graphical processing unit (GPU) and a benchmark on different GPUs is shown. Furthermore, a control strategy for collision avoidance based on target dynamics and estimation of target relative bearing/elevation angles is described.

For the HMM two complementary hypotheses are considered, the first one is, when there is one target and the second, when there is no target on the image plane. They used four independent HMM filters on the same preprocessed image, which means after the CMO filtering step. The dynamics of the target is extracted using a standard projective model, using pinhole camera. They developed a new control law for the collision avoidance task as well based on the calculated relative angles and the camera motion model (the optical flow equation). This new control law was under testing at that time.

The performance of the different GPU architectures are introduced in this paper as well. The implementation used the CUDA C language and the GTX280, the 8800GTS and the 9600GT chips from nVIDIA were running it. The computation speed was compared to a naïve C implementation on a Pentium IV based PC running Linux. The improvement was x20, x7 and x1,5 respectively. For the final implementation the 9600GT was used, because its power consumption is the smallest from these three GPUs, it is 59 W. It was capable of doing the computation with 11Hz. After further code optimizations the authors expected 30Hz image processing rate with the 9600GT.

In the next paper [60] besides the HMM, a Viterbi-based filtering method is evaluated in realistic situations. The test videos are recorded using two UAVs simulating a collision situation. In the tests the Viterbi-based filtering had a slightly bigger detection range, but the SNR for the HMM was much better. The computational cost of the two algorithms is very similar.

DOI:10.15774/PPKE.ITK.2014.008

The authors built a GPU based system for the detection, and according to the paper it is suitable for UAV integration. The power consumption of the GPU itself is 59W and there is a host computer next to it, which seems to be too much for a small size UAS.

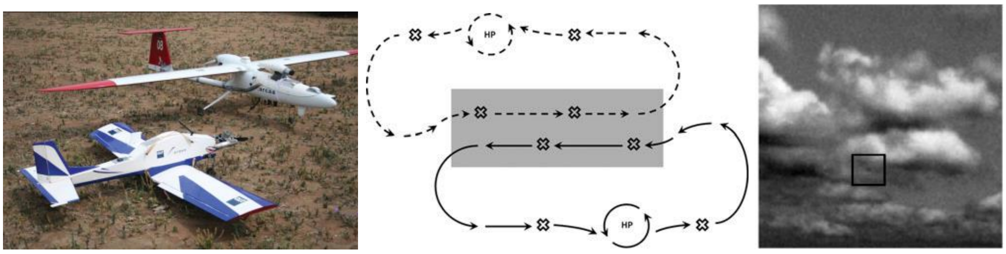


Figure 2.10 Fixed wing UAVs for data collection, with the planned trajectory and a frame from the recorded video, with the target aircraft

Due to the fixed-wing aircraft platform and the autonomous flight mode for the UAVs they had difficulties during the data collection. That is why they decided to switch to a manned, full sized airframe (Cessna 172) to collect a big database of video data. This data is used to further test the algorithms.

In [61] the authors propose a visual-spectrum image-based method of obtaining supplementary bearing angle rate information that exploits CMO preprocessing, HMM temporal filtering, and relative entropy rate (RER) concepts. The main contribution of this paper is the proposal of an online vision-based heading angle and speed estimator for airborne targets using these concepts. In particular targets that appear as small features in the image measurements without distinct texture or shape are considered. A possible connection between RER and probabilistic distance measures are considered first. Then a mean heading angle and speed estimator (or pseudobearing rate estimator) that exploits this connection is proposed.

The tests for this algorithm are run on computer-generated image data, real ground-based image data, and real air-to-air image data. The simulation studies demonstrated the superiority of the proposed RER-based velocity estimation methods over track-before-heading-estimation approaches, and the study involving real air-to-air data demonstrated application in a real airborne environment.

In [62] and [63] the extensive experimental evaluation of the sky-region, image-based, aircraft collision-detection system introduced in the previous publications is shown, with the description of a novel collection methodology for collecting realistic airborne collision-course target footage in both head-on and tail-chase engagement geometries. Under blue sky conditions, the system achieved detection ranges greater than 1540m in three flight test cases with no false-alarm events in 14.14 h of non-target data (under cloudy conditions, the system achieved detection ranges greater than 1170 m in four flight test cases with no false-alarm events in 6.63 h of non-target data).

The new methodology for flight video collection is remarkable as well. In all flight experiments, the camera aircraft was a custom modified Cessna 172 and the target aircraft was a Cessna 182. In order to avoid dangerous situations and to provide reliable data, they followed ISO standards for the data collection experiments. This way they could test the algorithms on the basis of a uniquely large quantity of airborne image data. The image data was analised before the test based on the target range, the SNR and the cloudiness.

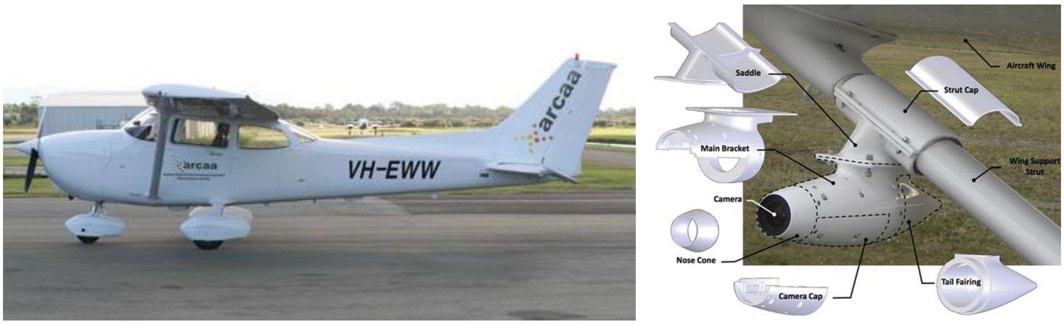


Figure 2.11 Modified Cessna 172 aircraft and the used camera frame

On the test data the detection range versus the false-alarm rate is calculated with both the Viterbi and the HMM algorithm. These curves are treated as system operating characteristics (SOC), even if the dataset is still small for a proper statistical analysis. The empirically determined SOC curves were able to demonstrate that morphological–Viterbi-based approaches seem very unlikely to be a practical solution to this collision detection problem (due to high false-alarm rates). Conversely, a morphological–HMM-based approach was shown to be able to achieve reasonable detection ranges at very low false-alarm rates (in both blue sky and cloudy conditions).

It seems that these methods are well thought out and extensively tested in real situations. The detection range and false alarm rates are very impressive, and the authors have the biggest known airborne video database as well, with a real target aircraft. The main drawback seems to be the power consumption of the proposed system due to the computationally extensive preprocessing and temporal filtering steps. The algorithm is capable of detecting aircrafts in the sky region and only the videos with dark targets are involved in the tests.

In [64] an obstacle detection method for small autonomous UAV using sky segmentation is introduced. The proposed algorithm uses a support vector machine (SVM) on

YCrCb color space to separate sky and non-sky pixels. The recorded images are first filtered with a Gaussian filter, and then segmented with the SVM. The horizon is determined according to the sky and non-sky pixels using Hough transformation. The objects are formed of those non-sky pixels which are in the sky region. The algorithm is real-time and was tested in hardware-in-the-loop (HIL) simulations, as well as in real flight tests. The main disadvantage of the algorithm is that it can only detect obstacles above the horizon that are viewed with sky in the background. In our system besides the detection on the sky region, the detection below the horizon will be included as well.

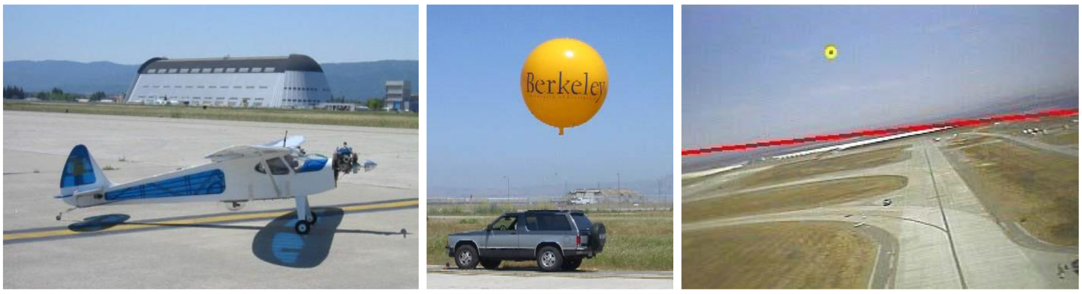


Figure 2.12 Test aircraft, the target balloon and a frame from the processed flight video

In [65] and [66] the development of a SAA system is shown. According to the paper the system has the potential to meet the FAA’s requirements. This system uses 3 CCD cameras as sensors and FPGAs for the processing. Detection and tracking algorithms characterize global scene motion, Sense objects moving with respect to the scene, and classify the objects as threats or non-threats. Detection algorithms operate directly on sensor video to extract candidate features. Tracking algorithms operate on the candidate features (“detections”) to correlate them over time forming “tracks.’ Declaration algorithms operate the track set to classify them as threat or non-threat based on their temporal behaviour.

A total of 27 collision scenario flights were conducted and analysed. The average detection range was 11.6 km and the mean declaration range was 8 km. There were many false tracks first due to the sensor vibration, but later on an improved sensor mount was developed which helped to lower the false alarm rate significantly. The number of false alarms per engagement has been reduced to approximately 3 per engagement. This shows the importance of a good anti-vibration system. In our approach, as we are using a five camera system we had to handle the cross vibration of the cameras as well. Unfortunately, because this system was developed for US Air Force, the details are not provided for the algorithms or the system.

In [67] and [68] a system with 3 nested KF for OF computation, UAV motion estimation and obstacle detection is introduced. The system is used as a vision based autopilot for small

UAVs, flying close to the ground, in cluttered, urban environment. They use a monocular camera as the main sensor. The three KF are exchanging information about the UAV’s motion and the estimated structure of the scene.

The OF calculation is using block matching and differential method. The block matching uses motion constraints based on the INS module, and uses an adaptive shape for the matching. The rough estimates given by the block matching are then refined by the differential algorithm. The results are filtered with the first KF in order to select features for the structure computation and to determine the angular velocity. For the ego motion estimation, the results from this module and the measurements from the INS are fused with the second KF. And the third KF is used to estimate the pure translational motion of the UAV.

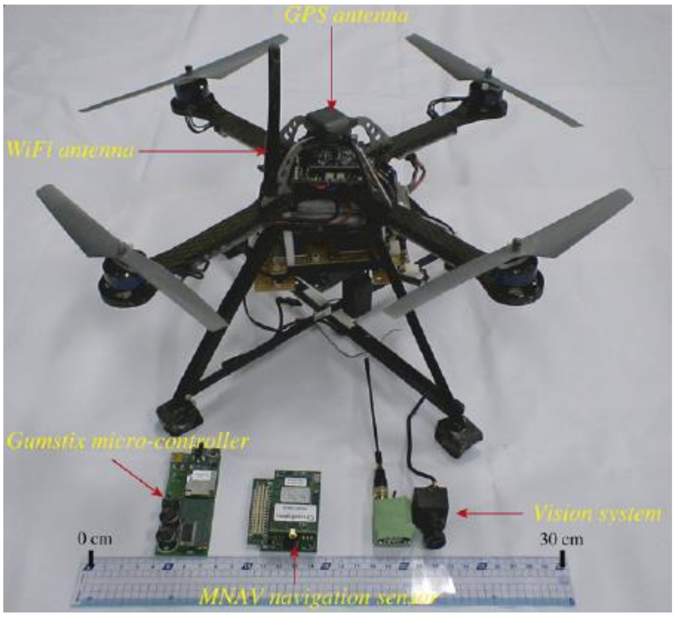


Figure 2.13 Quadrotor for the flight tests

The algorithm is tested in numerical simulations and in real environment. A quadrotor is used with a low cost IMU and a downward looking camera with 320x240 px resolution @25Hz. The quadrotor has a 400g weight and can carry a 300g payload. The scale ambiguity introduced by the camera is resolved with a static pressure sensor. The efficiency and robustness of the proposed vision system were demonstrated for indoor and outdoor flights. The problem with this approach is that the computations are run on a ground control station, and the obstacle detection was not tested. In this way the UAV is not capable of doing the collision avoidance if there is a lost connection in between the aircraft and the base station. In our system all processing is done on-board.

In [69], [70] and [71] introduces a visual collision and detection system based on a monocular camera. A new method called expansion segmentation is shown, which simultaneously detects “collision danger regions” of significant positive divergence in inertial aided video, and estimates maximum likelihood time to collision (TTC) within the danger regions. The algorithm was tested in simulations and a real video as well. The algorithm was implemented in C and was run on a Core 2 Duo PC @0.2 Hz. The main drawback of this concept is that the size of the intruder has to be big enough in order to determine the expansion rate. It means that the range of the detection is small or the camera sensor has to have a very big resolution.

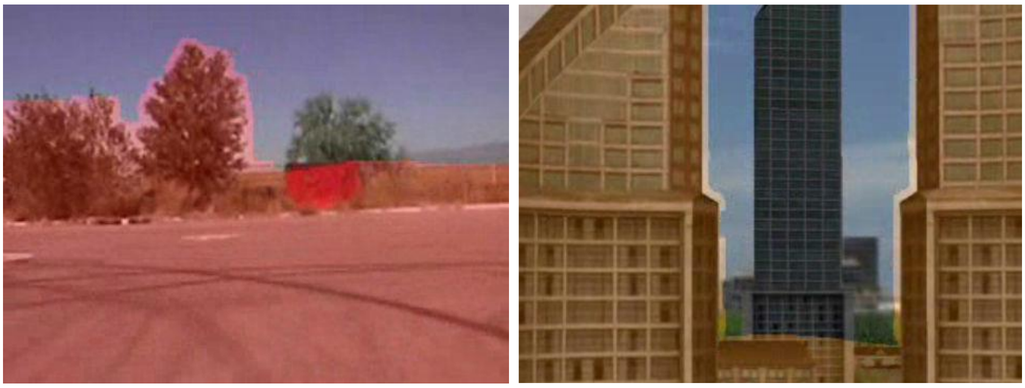


Figure 2.14 Processed video frames (left: real flight, right: simulation)

DOI:10.15774/PPKE.ITK.2014.008

Chapter 3

UAV SAA Test Environment

In this chapter the base ideas and the most important principles are shown which are used in the development of the UAV SAA system. In order to develop and test new methods and algorithms for UAS SAA, a test environment was built. This setup consists of three main parts, the sensors, the image processing part and the control part.

The goal of our research is to create a complete, autonomous flight control system for UAS. This is a closed loop flight control system with the collision avoidance capability based on visual detection of the approaching object (Figure 3.1). The organization of the system is as follows.

The first part contains the sensors. The input images are recorded by the *Camera* and the own position and inertial data measured by the on-board *INS/GPS* (Inertial Navigation System/Global Positioning System).

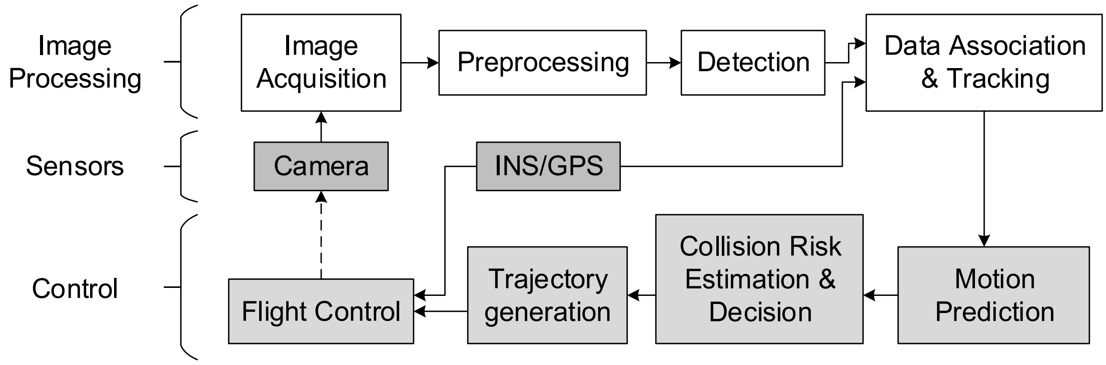


Figure 3.1 Flowchart of the closed-loop SAA system

The second part is the image processing. The recorded pictures are transmitted by the *Image Acquisition* to the *Pre-processing* block by which the pictures are filtered.

DOI:10.15774/PPKE.ITK.2014.008

The next step of the processing is the *Detection*. The images are processed by image processing algorithms to detect the approaching objects. The *Data Association & Tracking* is responsible for the combination of the orientation and angle of attack data of the approaching object calculated by the *Detection*.

The third part is the flight control. According to the combined data the relative motion of the approaching object is predicted by *Motion Prediction*. If a risky situation is identified by *Collision Risk Estimation & Decision* a modified trajectory is generated by the *Trajectory generation*. The avoiding manoeuvre is passed to the *Flight Control*, which is responsible for the autonomous control of the aircraft.

3.1 Coordinate Systems

In most applications a small UAV flies only short distances (about several kms of range). This allows considering the North-East-Down (NED) frame as an inertial (non-moving, non-rotating) frame (earth frame) [32]. The NED frame is defined as follows: the Z axis is the normal vector of the tangent plane of Earth at aircraft starting position pointing into the inner part of the ellipsoid. The X axis points to north and the Y axis forms a right-handed system with the other two. The NED is referenced later on as the earth coordinate system as well.

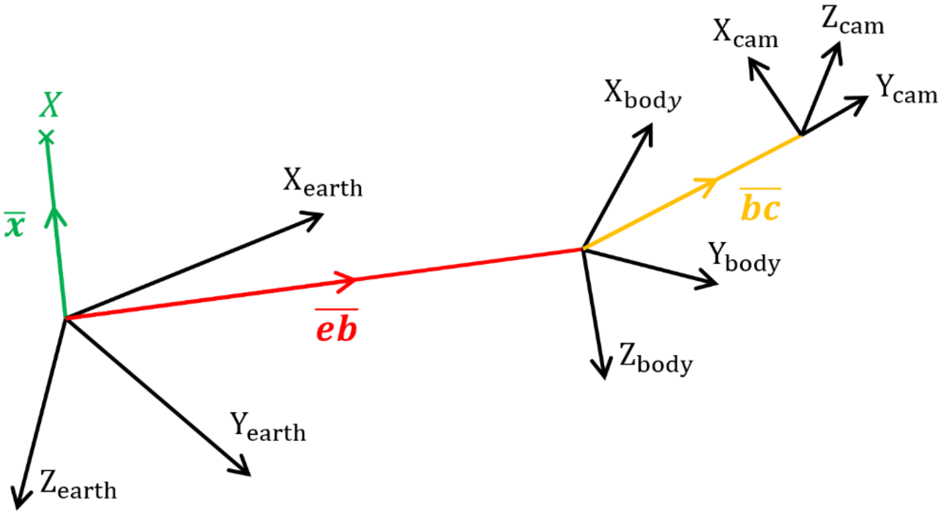


Figure 3.2 The earth, the body and the camera coordinate systems in general .
$(X_{earth}, Y_{earth}, Z_{earth})$ earth (NED), $(X_{body}, Y_{body}, Z_{body})$ body
and $(X_{cam}, Y_{cam}, Z_{cam})$ camera coordinate systems.
$\overline{eb}$ the position of aircraft centre of gravity in earth coordinate system,
$\overline{bc}$ the position of camera in body coordinate system and
$\bar{x}$ the position of a feature point (X) in earth coordinate system

The other two applied coordinate systems are the body and camera systems. The body frame is fixed to the aircraft centre of gravity with Z axis pointing downward, X axis pointing forward and the Y axis forms a right-handed system with the other two.

The axes of the camera system are in general nonparallel with the axes of the body system (see Figure 3.2). In the considered set up the axes of the camera and body coordinate systems are parallel but the camera coordinate system is rotated in the body frame (Figure 3.3).

In Figure 3.2 X is a feature point in the earth coordinate system characterized by vector $\bar{\boldsymbol{x}}_{\text{earth}}$ (the $\overline{()}_{\text{earth}}$ means a vector with coordinates in earth coordinate system). $\overline{\boldsymbol{eb}}_{\text{earth}}$ gives the position of the body frame relative to earth while $\overline{\boldsymbol{bc}}_{\text{body}}$ gives the position of the camera frame relative to body. The coordinates of point X in the camera frame can be calculated as follows:

$$\bar{\boldsymbol{x}}_{\text{cam}} = \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}} - \overline{\boldsymbol{bc}}_{\text{earth}}\right) =$$

$$= \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}} - \overline{\boldsymbol{bc}}_{\text{earth}} - \overline{\overline{\mathbf{EB}}}\,\overline{\boldsymbol{bc}}_{\text{body}}\right) \tag{3.1}$$

Here, $\overline{\overline{\mathbf{F_2F_1}}}$ defines a transformation matrix from frame F_1 to F_2. In our special case the origins of the body and camera system are assumed to coincide (see Figure 3.3) and so, $\overline{\boldsymbol{bc}} = 0$ can be considered:

$$\bar{\boldsymbol{x}}_{\text{cam}} = \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right) \tag{3.2}$$

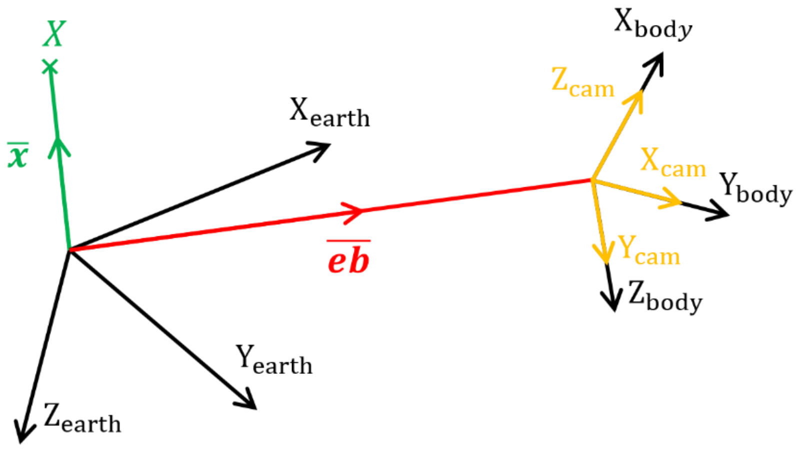

X
$\bar{\boldsymbol{x}}$
$\mathrm{X}_{\text{earth}}$
$\overline{\boldsymbol{eb}}$
$\mathrm{Y}_{\text{earth}}$
$\mathrm{Z}_{\text{earth}}$
X_{body}
Z_{cam}
X_{cam}
Y_{body}
Y_{cam}
Z_{body}

Figure 3.3 The earth, the body and the camera coordinate systems in this specific scenario when the origins of body and camera system coincide and so $\overline{\boldsymbol{bc}} = 0$

DOI:10.15774/PPKE.ITK.2014.008

3.2 Camera model

The electro optical sensor is modelled as a special case of a projective camera [72]. The camera matrix $\bar{\bar{\mathbf{P}}}$ consists of the internal and external parameters of the camera and can be decomposed as follows:

$$\bar{\bar{\mathbf{P}}} = \bar{\bar{\mathbf{K}}}\left[\,\bar{\bar{\mathbf{R}}}\,\middle|\,\bar{\boldsymbol{t}}\,\right] \tag{3.3}$$

where $\bar{\bar{\mathbf{R}}}$ and $\bar{\boldsymbol{t}}$ are the rotation and translation of the camera, which are the extrinsic parameters. $\bar{\bar{\mathbf{K}}}$ contains the intrinsic parameters: the focal length f in pixels (it can be different in the x and y directions) and the position of camera principal point $\bar{\mathrm{p}}$ in the image plane as follows:

$$\bar{\bar{\mathbf{K}}} = \begin{bmatrix} f_{\mathrm{x}} & 0 & p1 \\ 0 & f_{\mathrm{y}} & \mathrm{p2} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

3.3 Measured and estimated variables

We assume there is only one intruder to be detected. The detection of the intruder is formulated as a state estimation problem, where the dynamics are the relative motion of the intruder to our aircraft. The motion of the intruder is described as a linear motion of a point mass driven by an external force.

The measured output contains all information that can be extracted from the camera images. Since the camera projects the 3D view onto a 2D plane, which is a nonlinear mapping, the measured outputs are nonlinear functions of the states. Even if the motion of the aircrafts is modelled by a linear system, the nonlinearity of the output equation makes it necessary to apply Extended (EKF) or Unscented Kalman Filters (UKF) to estimate the intruder's data [9].

To simplify the filter design the vehicles (intruder and own aircrafts) are modelled in the NED frame by simple point mass dynamics. The relative position of the target, as the function of time $\bar{\boldsymbol{x}}_{\mathrm{cam}}(t)$, can be expressed in the camera frame as follows:

$$\begin{gathered}\bar{\boldsymbol{x}}_{\mathrm{cam}}(t) = \bar{\bar{\mathbf{CB}}}(t)\,\bar{\boldsymbol{x}}_{\mathrm{body}}(t) = \\ =[\,x1_{\mathrm{cam}}(t)\;\; x2_{\mathrm{cam}}(t)\;\; x3_{\mathrm{cam}}(t)\,]^{\mathrm{T}}\end{gathered} \tag{3.5}$$

DOI:10.15774/PPKE.ITK.2014.008

Assuming pinhole camera model the location of the target on the image plane can be computed as follows:

$$\bar{\mathrm{x}}_{\mathrm{cam}}(t) = \frac{f}{x1_{\mathrm{cam}}(t)} \begin{bmatrix} x2_{\mathrm{cam}}(t) \\ x3_{\mathrm{cam}}(t) \end{bmatrix} = \begin{bmatrix} x1_{\mathrm{image}}(t) \\ x2_{\mathrm{image}}(t) \end{bmatrix} \tag{3.6}$$

where f is the focal length of the camera. The details can be seen in Figure 3.4. By locating and tracking the intruder on the image plane the image processing unit can determine:

- the direction unit vector

$$\bar{\boldsymbol{u}}(t) = \frac{\bar{x}_{\mathrm{cam}}(t)}{\|\bar{x}_{\mathrm{cam}}(t)\|} \tag{3.7}$$

- and the subtended angle

$$\phi(t) = 2\tan^{-1}\left(\frac{b}{2\|\bar{x}_{\mathrm{cam}}(t)\|}\right) \tag{3.8}$$

under which the target is seen. (The constant b in the formula is the unknown wingspan of the target, which is also to be estimated by the filters). These parameters are the inputs of the estimation.

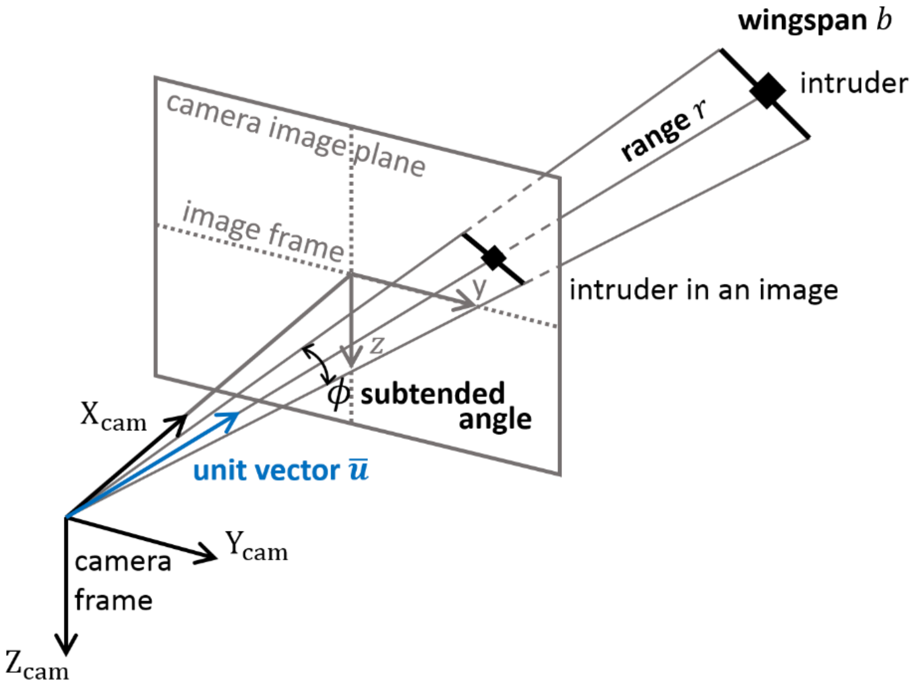


Figure 3.4 Subtended Angle Relative State Estimation (SARSE) methods

DOI:10.15774/PPKE.ITK.2014.008

3.4 Simulation environment

Before the actual flying we have to prove the operability of our system. Based on the planned closed-loop flight control system, is shown in Figure 3.1 we developed a simulation environment. The block diagram of the simulation environment is shown in Figure 3.5 and the photograph of the system is shown in Figure 3.6.

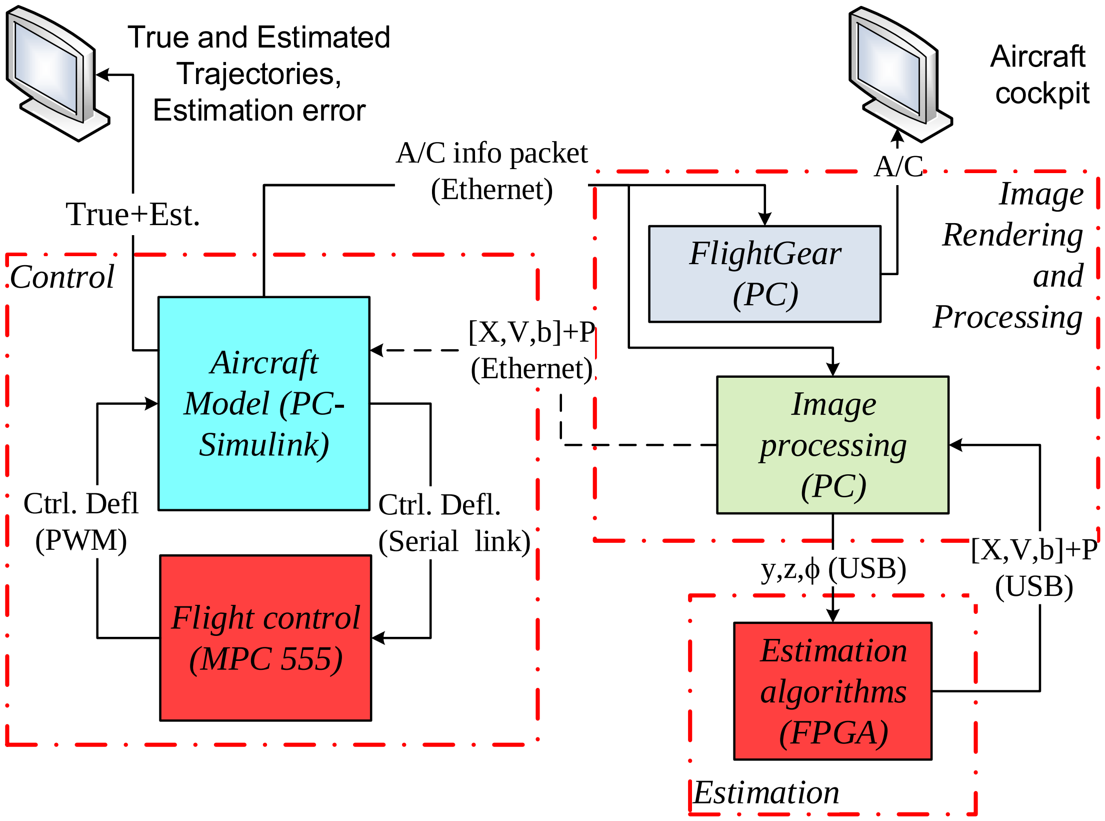

True and Estimated Trajectories, Estimation error
Aircraft cockpit
A/C info packet (Ethernet)
A/C
True+Est.
Image Rendering and Processing
FlightGear (PC)
Control
Aircraft Model (PC-Simulink)
[X,V,b]+P (Ethernet)
Image processing (PC)
Ctrl. Defl (PWM)
Ctrl. Defl. (Serial link)
y,z,ϕ (USB)
[X,V,b]+P (USB)
Flight control (MPC 555)
Estimation algorithms (FPGA)
Estimation

Figure 3.5 Block diagram of the HIL simulator

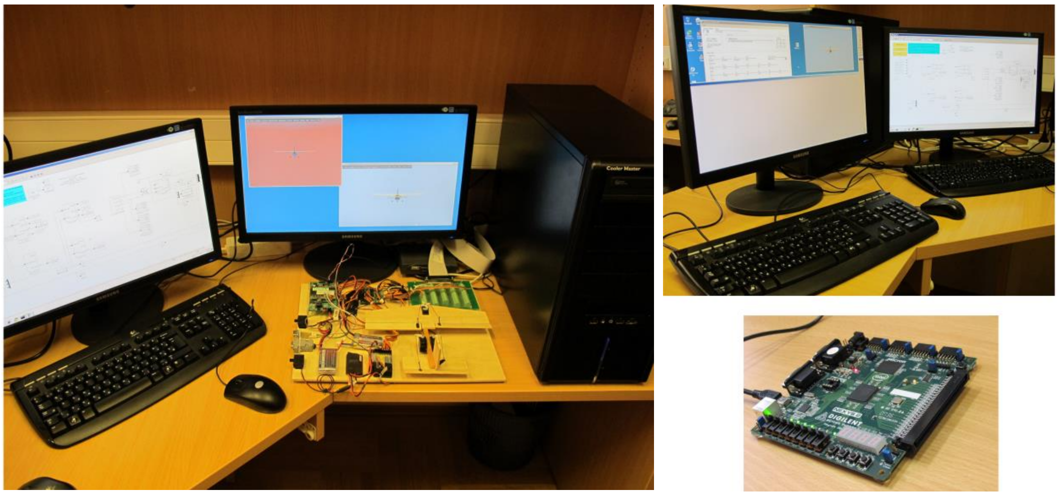


Figure 3.6 The HIL simulator

DOI:10.15774/PPKE.ITK.2014.008

The flight control is running on hardware-in-the-loop system, shown at the upper left corner in Figure 3.5 and Figure 3.6. The aircrafts are simulated in MATLAB/Simulink using the Aerospace blockset. For the own aircraft a high fidelity mathematical model has been identified using the measurement data collected from the Ultrastick unmanned aircraft [73]. The intruder is modelled as a simple double integrator. For the own aircraft a trajectory tracking controller has been designed, which runs on an MPC555 embedded microprocessor. The flight simulator PC communicates with the image rendering and processing computer via Ethernet.

The rendering is done by the FlightGear simulator program. The FlightGear cooperative flight simulator is an open-source, multi-platform program. It can fetch real weather conditions, it contains more than 100 3D aircraft models and it contains the real geographical data of the half globe. We used this program to visualize our aircrafts and the environment and do the image processing on the rendered pictures. I modified the FlightGear program in order to save the rendered images for offline processing with MATLAB routines or to run the image processing algorithm real-time with the functions based on the OpenCV library.

For the sake of calculating precise input data for the estimation algorithm the FlightGear program has to be calibrated. First the FOV and the aspect ratio settings are measured. For the measurements a Cessna 172P aircraft model was used because this is a very popular light weight airplane. UAV share airspace with this type of aircrafts and most of them have no radar and use visual sensing for collision avoidance.

The wingspan of Cessna 172P is 11m. The FOV of the rendered image from the following model is calculated:

$$FOV = \frac{2\tan^{-1}(5.5/r)}{w_a} \cdot w \tag{3.9}$$

where FOV is in degree, r is the distance of the two aircrafts in meters, w_a is the measured width of the aircraft in pixels, w is the width of the rendered image in pixels.

From the measurements it turned out that two regions can be defined from the rendering point of view: a far region (r >20m), where this model can be used and a close region (r <20m), where distortions of this model are observed. The images can be used without additional compensation, since the far region is of interest in our case, because we are not dealing with the emergency situation yet. We have to detect the other aircraft far enough to do the avoiding manoeuvre.

DOI:10.15774/PPKE.ITK.2014.008

The next figure (Figure 3.7) shows the resulting images of the FlightGear program’s rendering tests. It turned out that the FlightGear does not take care about the aspect ratio parameter. If geometry is not 1:1, the FOV is set to the bigger size and the image is cut by FlightGear from the 1:1 ratio image. According to the measurements that are not detailed here, it can be asserted that the geometry used by FlightGear is linear perspective.

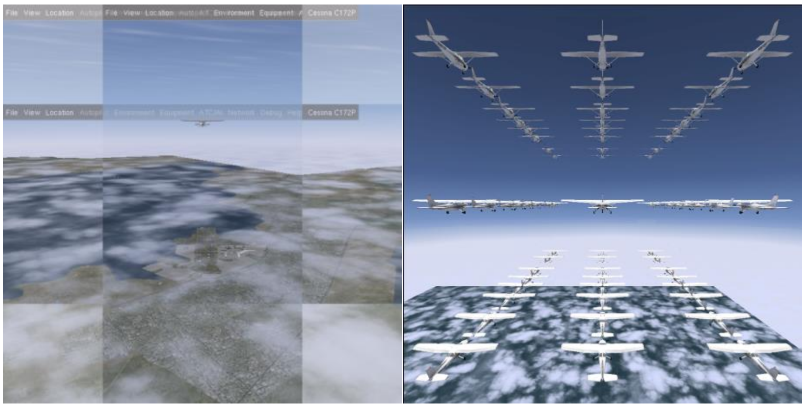


Figure 3.7 FlightGear rendering test; On the left, the test of the aspect ratio change; On the right the demonstration of the camera projection

The modified FlightGear sends the results of the image processing (the subtended angle and the size) to an FPGA via USB, which runs an EKF in order to predict the relative 3D position of the intruder. In the current system a Spartan 3 FPGA runs the motion prediction task. The results are sent back to the control part where the risk estimation and the trajectory generation take place.

3.5 Image processing algorithm

In this section an image processing algorithm is presented which was designed to operate in daylight with clear or cloudy sky, when the contrast of the clouds is small or medium. When the contrast of the cloud is high (sunrise, sunset or storms), this vision algorithm cannot detect the intruder airplane robustly, however these situations can be predicted very well in advance. In our experimental environment the camera is fixed to the NED co-ordinate system.

From the very beginning of the algorithm design, we kept in mind the strict power, volume and other constraints of an airborne UAV application. To be able to fulfil these constraints, we decided to use many-core cellular array processor, implemented in ASIC or FPGA. Therefore we selected topographic operators, which well fit in this environment.

DOI:10.15774/PPKE.ITK.2014.008

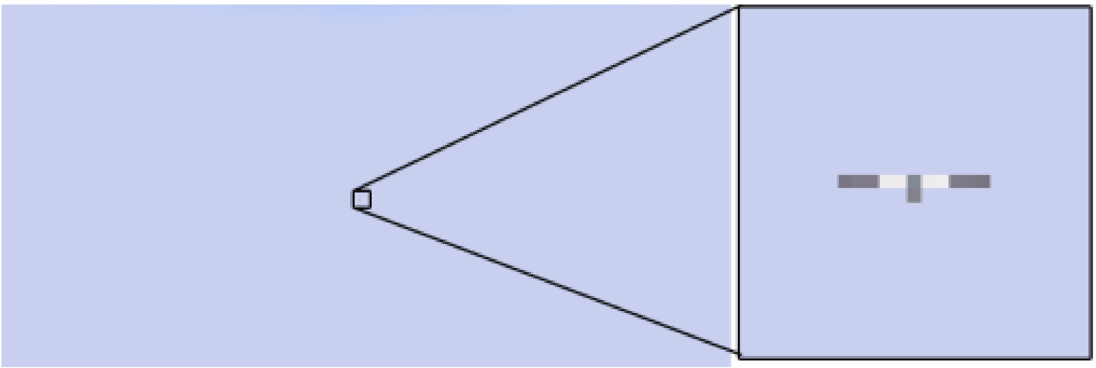


Figure 3.8 Input image (2200x1100 pixel) from the simulator;
the square shows the location of the intruder, on the right side the enlarged image of the intruder

On Figure 3.9 the flowchart of the image processing algorithm is shown. The input images of the algorithm are at least 1 megapixel (Figure 3.8).

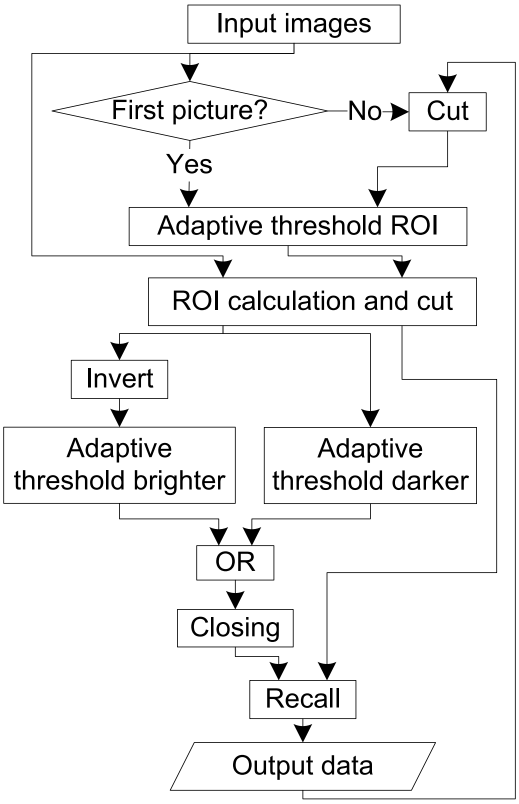


Figure 3.9 Diagram of the image processing algorithmAs shown in Figure 3.9 the first step is a space variant adaptive threshold [74] to filter out the slow transitions in an image. This can be the applied to the entire raw image if the position of the intruder is not known. If the location is already known, we track the intruder in a smaller window to reduce the data size and speed up the computation. The adaptive threshold results a binary image containing some of the points of the aircraft.

DOI:10.15774/PPKE.ITK.2014.008

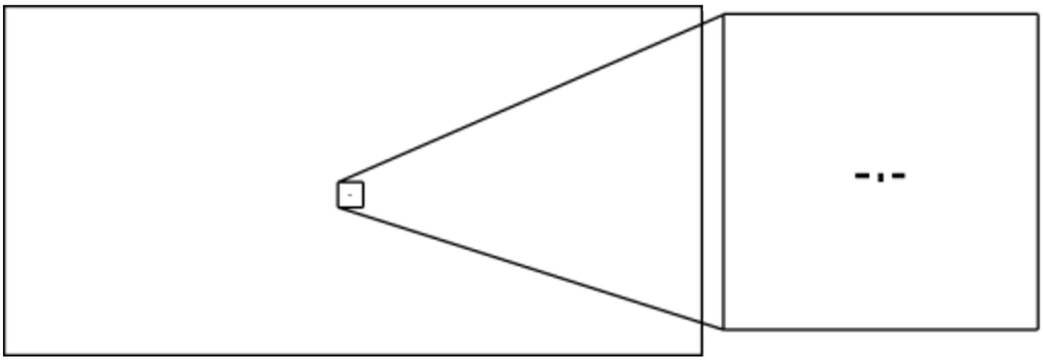
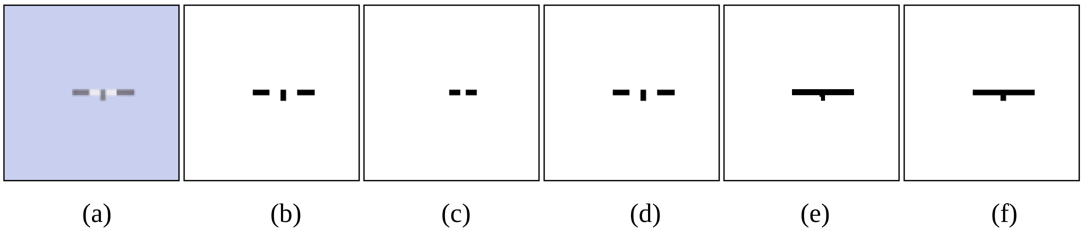


Figure 3.10 Result of the first adaptive threshold on a raw 2200x1100 input image; on the right side the enlarged image of the intruder aircraft

On this binary image a centroid calculation [74] is applied, which gives the co-ordinates of the central pixel of the object. This co-ordinate will be the central pixel of the Region of Interest (ROI). The size of the ROI is determined by the previously calculated wing size plus 20 pixels in each direction. In that way two images are cut: one from the original picture (colored ROI image: Figure 3.11 a) and one from the result of the adaptive threshold (binary ROI image: Figure 3.11 b). The aircraft is composed of darker and brighter pixels than the intensity mean value of the original picture (background) (Figure 3.8). On the coloured ROI image two adaptive threshold operators are calculated. The first one is calculated on the inverse picture of the grayscale image created from coloured ROI image. With this threshold the pixels brighter than the intensity mean value of the original picture are found (Figure 3.11 c). The result is a binary image with the brighter pixels.

The other threshold is calculated on the coloured ROI image and with this the darker pixels are extracted (Figure 3.11 d). A logic OR is applied for the two threshold images. The result is a binary picture with the found pixels of the aircraft and with some other pixels (Figure 3.11 e).



(a) (b) (c) (d) (e) (f)

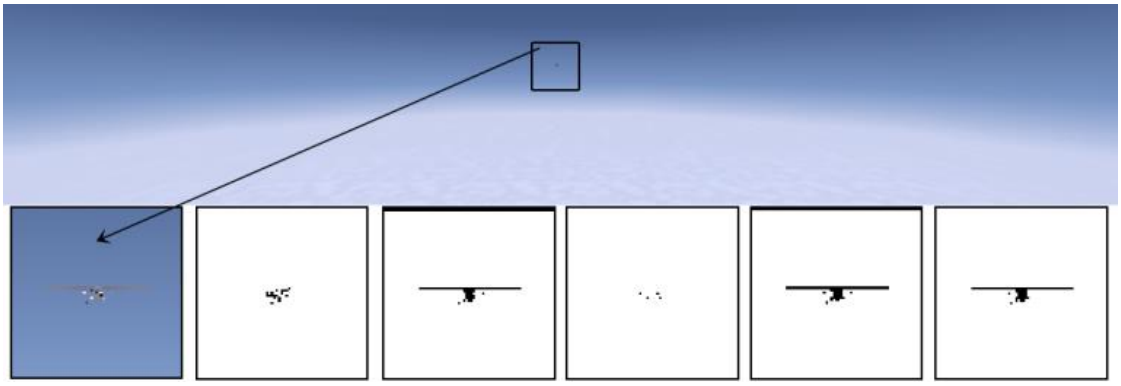
Figure 3.11 The steps of the segmentation; from left to the right: a) coloured ROI, b) binary ROI, c) brighter pixels, d) darker pixels, e) OR operation and closing, f) segmented shape of the intruder aircraft

DOI:10.15774/PPKE.ITK.2014.008

In some cases the parts of the airplane are not connected. A closing operation [74] is applied to connect the components. From the binary ROI picture we have an approximation for the aircraft and from the previously calculated picture we have the pixels of the whole airplane with some noise. As a last step, a recall operator [75] is applied, because the two adaptive threshold (darker and brighter) may find other objects from the background, which are not extracted with the first adaptive threshold. This way these false objects can be filtered out.

The silhouette of the airplane is obtained this way. In this picture the centroid in pixels is determined. Based on the co-ordinate of the centre of the silhouette direction $\overline{\boldsymbol{u}}(t)$ and the subtended angle $\phi(t)$ of the intruder aircraft in radians can be determined accurately.

In the previous example, the intruder aircraft was at 1 km distance (60° view angle, 1200 pixels horizontal resolution, 1.02m/pixel), hence the extracted silhouette was very coarse. Here another example is shown, where the intruder aircraft is only 300m to the camera (Figure 3.12). It is observable in this snapshot that the first adaptive threshold does not find all the pixels of the intruder (Figure 3.12 c) and the whole algorithm is needed to extract the entire aircraft.



(a) (b) (c) (d) (e) (f)

Figure 3.12 Steps of the image processing: up the input image, down the outputs of each step: a) color ROI, b) adaptive threshold, c) darker pixels, d) brighter pixels, e) OR operation and closing, f) segmented aircraft

3.5.1 Detection performance

In our experimental settings, the intruder can be detected from 3.3km. In Figure 3.13 the farthest detectable intruder is shown. In this case the size of the intruder aircraft is 2 pixels only.

DOI:10.15774/PPKE.ITK.2014.008

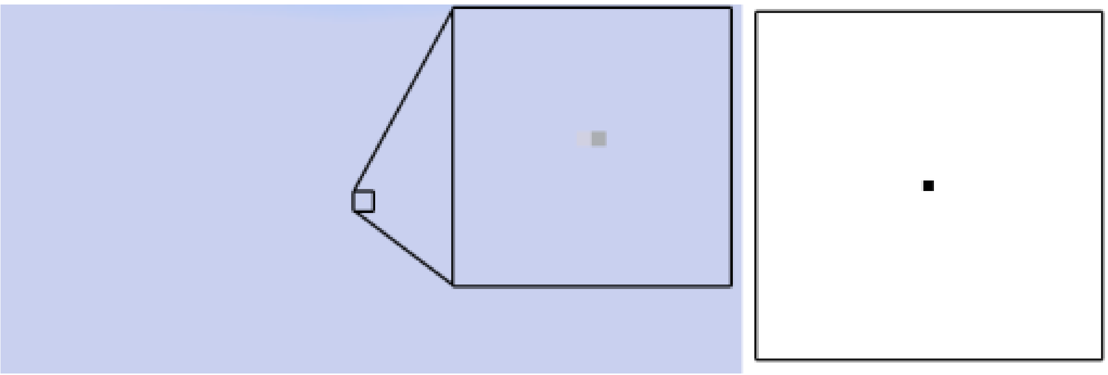


Figure 3.13 Farthest detectable position of the intruder C172p aircraft (wingspan=11m), the distance is 3.3 km; on the left is the input image from FlightGear flight simulator, on the right the result of the segmentation

In Figure 3.14 an example is shown with real image with cloudy background, when the contrast of clouds is medium. In Figure 3.14 on the upper right corner the result of the first adaptive threshold is shown, from which the position of the intruder aircraft can be calculated.

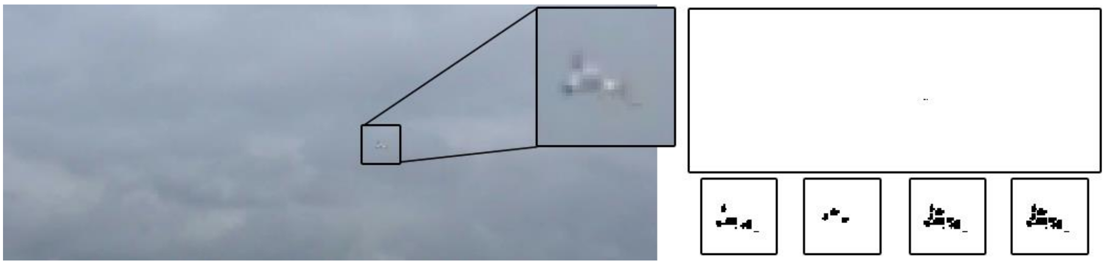


Figure 3.14 Example of the situation with medium contrast clouds: on the left the original image with the enlarged aircraft; on the upper right the result of first adaptive threshold, on bottom right from left to the right the darker pixels, brighter pixels, OR operation and the segmented aircraft

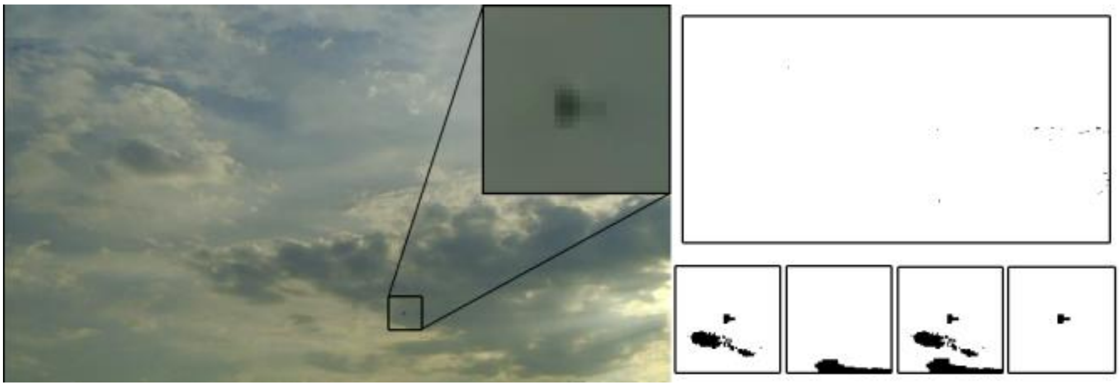


Figure 3.15 Example of the situation with high contrast clouds: on the left the original image with the enlarged aircraft, on the upper right the result of first adaptive threshold, on bottom right from left to the right the darker pixels, brighter pixels, OR operation and the segmented aircraft

DOI:10.15774/PPKE.ITK.2014.008

In Figure 3.15 we can see a typical situation during sunset, when the contrast of the clouds is high. In this case the position of the intruder can be determined only if we have prior information about it. In Figure 3.15 on the upper right corner not only the points belonging to the intruder aircraft are detected by the first adaptive threshold but some cloud points also. On the bottom right the situation is shown when there is prior information about the position. This prior information may come from tracking or from a dispatcher. On the other hand, high contrast cloudy situations are known in advance (hence can be avoided), because it happens during sunrise, sunset, and in case of an approaching storm.

3.6 Distant airplane detection

In section 3.5 the tests of the image processing algorithm in simulations were presented. It was shown that with this algorithm in the described environment (for detecting one intruder aircraft in daylight with clear or cloudy sky when the contrast of clouds are low or medium) the intruder can be detected at 3.3 km maximum. In this section the improved algorithm and tests on real videos in long-range situations are presented. On Figure 3.16 the flowchart of the improved image processing algorithm is shown. The input images of the algorithm are at least 2 megapixels.

3.6.1 Pre-processing

As shown in Figure 3.16 the first step is a Gaussian low pass filter to filter out high frequency noise. 2D Gaussian filter preserves the position of the edges which is important in this application. In this case a 3x3 Gaussian filter is sufficient. The coefficients are calculated according to (3.10).

$$\mathrm{h_g}(n_1,n_2) = e^{\frac{-\left(n_1^2+n_2^2\right)}{2\cdot\sigma^2}} \quad \mathrm{h}(n_1,n_2) = \frac{\mathrm{h_g}(\mathrm{n_1},\mathrm{n_2})}{\sum_{n_1}\sum_{n_2}\mathrm{h_g}(n_1,n_2)} \qquad (3.10)$$

where n_1 and n_2 are the coordinates and σ is the standard deviation. The next step is a space variant adaptive threshold to filter out the slow transitions in an image (Figure 3.17 b). This adaptive threshold is either executed on the entire raw image or on a smaller sub-image of it, depending on whether we have good position estimate or not. To reduce the input image size and speed up the computation, a foveal approach is implemented, that is a window containing the intruder airplane according to the previous results is cut. The adaptive threshold results a binary image containing some of the points of the aircraft (Figure 3.17 b, plus other points coming from clouds, ground objects, or noise.

DOI:10.15774/PPKE.ITK.2014.008

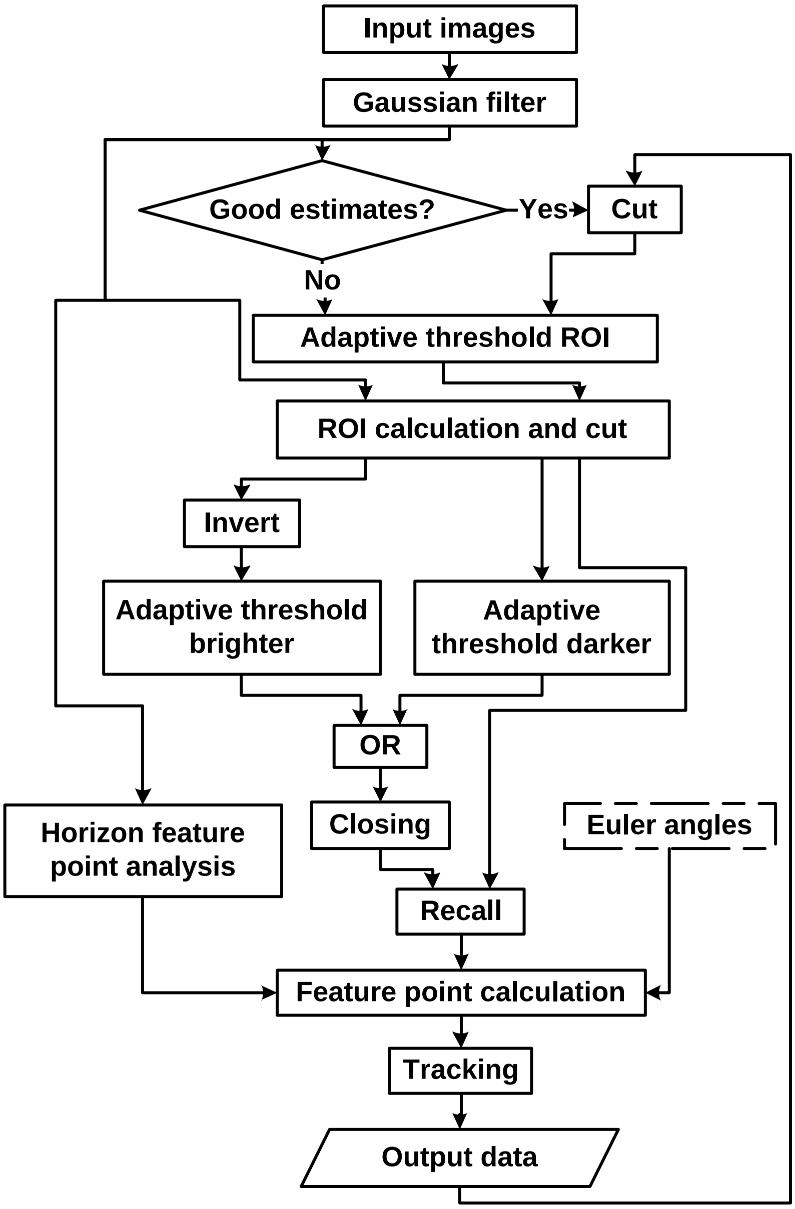


Figure 3.16 Diagram of the improved image processing algorithmSegmentation

On the adaptive threshold image the centroid coordinates of the objects are calculated. The calculated centroid coordinates are the centre points of the Region of Interest (ROI) windows. There are two types of ROIs one on the adaptive threshold image (binary ROI, Figure 3.17 c) and one on filtered input image (coloured ROI, Figure 3.17 d). The size of the ROI is determined by the previously calculated wingspan size plus 20 pixels in each direction. The next steps of the algorithm are calculated only on ROI images to speed up the calculation and lower the power consumption.

The approaching aircraft is composed of darker and brighter pixels than the background. Therefore, two adaptive thresholds are used to get the pixels of the aircraft (Figure 3.17 e, f). After the combination of the two results with the binary OR operation, a binary closing [8] is run to connect the found pixels (Figure 3.17 g). After the closing a binary reconstruction operation is applied based on the binary ROI image to filter out noise remaining after the adaptive thresholds. The shape of the detected aircraft is given by the result of the reconstruction (Figure 3.17 h).

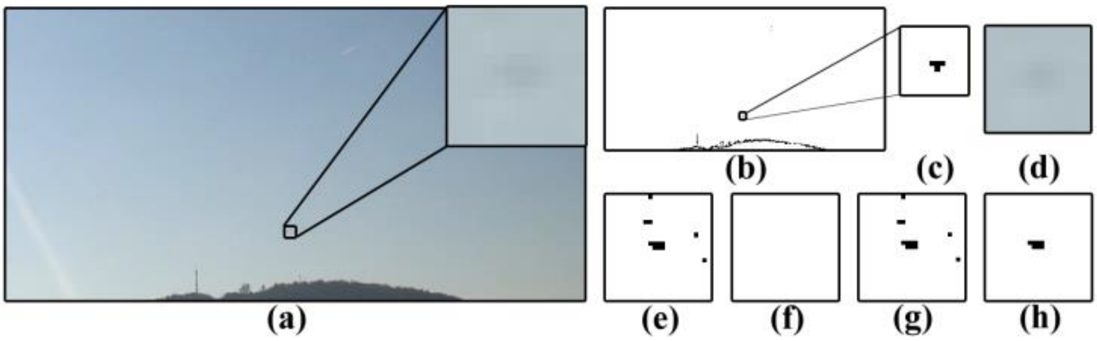


Figure 3.17 The steps of the segmentation (ROI size = 24),
(a) the central part of the original 1440x1080 pixels image with the enlarged area contains the aircraft, (b) result of the adaptive threshold (c) the enlarged area contains the aircraft on the adaptive threshold image (binary ROI) (d) coloured ROI (e) darker pixels (f) brighter pixels (g) OR operation and closing (h) segmented aircraft

3.6.3 Tracking

Our camera is attached to the nose of the Unmanned Aerial Vehicle (UAV). If our plane is carrying out some manoeuvre the calculated position values have to be corrected to eliminate the effect of our ego motion. Euler angles [76] are provided by the *INS/GPS* module can be used to calculate these corrections, but these Euler angles are often imprecise and in some cases they are not provided at all. The position and the orientation of the horizon is used by *Horizon feature point analysis* to correct the calculated position coordinates.

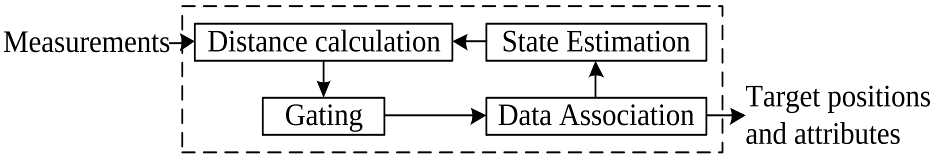


Figure 3.18 Diagram of the tracking algorithm

After this step the positions according to each ROI are collected and are given to the *Tracking*. Multi Target Tracking Library from Eutecus Inc. is used [77]. The algorithm consists of four main steps (Figure 3.18): 1) *State Estimation*: Using the track data gathered previously,

DOI:10.15774/PPKE.ITK.2014.008

the set of measurements are estimated, 2) *Distance calculation*: the distances in the proper metrics between the estimate and the input measurements are calculated, 3) *Data association/ gating*: the measurements and estimates are assigned to each other with a given threshold, 4) *Correction/ track management*: the estimated variables are corrected based on the measurement assigned to them, non-assigned tracks become subject to deletion and new tracks are initialized using non-assigned measurements.

For the estimation the library provides first, second and third order steady state KF methods. We used second order 4D KF with optimal filter parameters and transient handling. The state variables were the two coordinates of the centroid of the object and the two sides of the bounding box of the object with a given weight. Based on the found tracks, the position coordinates and the subtended angles are calculated.

3.6.4 Detection performance

The detection performance is demonstrated through an example, by detecting a remote Cessna. The camera was on ground and was fixed. We had estimated the relative position of the Cessna based on the landmarks. According to this estimate the Cessna was 3.7 km to the camera. In the video this aircraft was only 3.5 pixels and the size of the aircraft coincides with our range estimate (3.12).

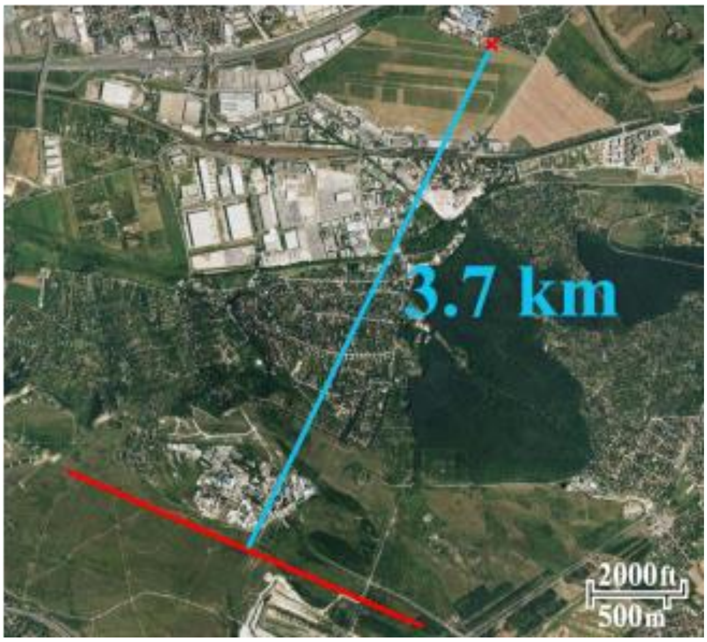


Figure 3.19 Distant aircraft trajectory and camera position; In the image we marked the position of the camera with a red x, the route of the recorded aircraft with a red line, and the distance with blue.

DOI:10.15774/PPKE.ITK.2014.008

The resolution of the camera was 1440 x 1080 pixels, the size of the sensor was 4.8mm (1/3 inch), the focal length was 5.1 mm and the field of view was 50.4°.

$$FOV = 2 \cdot \tan^{-1}\left(\frac{\text{sensor width}}{2 \cdot f}\right) \tag{3.11}$$

The length of the aircraft was about 8m and the wingspan was 11m. From the size data, the field of view and the resolution we can get the estimated distance.

$$\text{distance} = \frac{8}{\tan\left(\frac{\text{size in pixels}}{1440} \cdot FOV\right)} \tag{3.12}$$

In Figure 3.20 the central part of one video frame is shown and the detected aircraft is enlarged. We tracked 16 tracks with gate of 30 pixels, so the maximum distance of the estimated and the measured point in Euclidian norm was 30 pixels. The average velocity of the detected aircraft is 60m/s, from 3.7km it is around 27 px/s, so it is 1 px/frame and we could have some estimation error too.

The fade in time was 8 frames so for a given track in 8 consecutive frames the tracker has to assign a corrected estimate value to say it is a valid track. The fade out time was 20 frames, because of the noisy measurements, so if in 20 consecutive frames there is not any estimate which is assigned to a given track, the track is deleted.

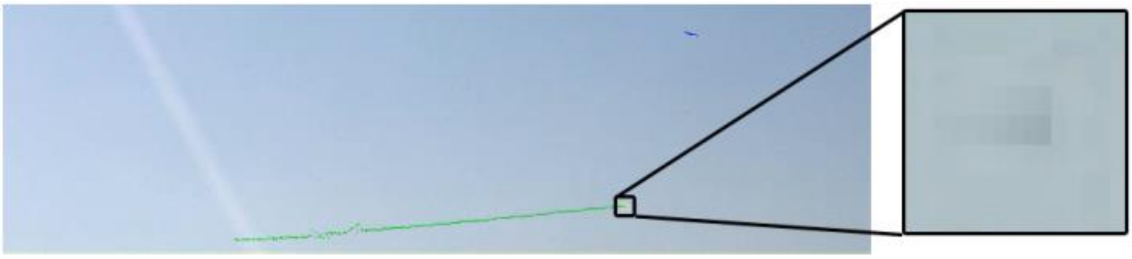


Figure 3.20 Central part of processed video frame with track of intruder (dotted green line) and the enlarged pixels of the intruder

The intruder was tracked successfully during the whole video. Besides the intruder other objects are identified as well, like a jet and some cars on the ground. The two aircrafts can be separated by their speed and size and the ground objects can be filtered out based on their position.

DOI:10.15774/PPKE.ITK.2014.008

Chapter 4

Relative Direction Angle Estimation

In the previous chapters a camera-based autonomous on-board collision detection part of the closed loop SAA system was introduced. This SAA system is capable of avoiding a single target as long as the lighting conditions are good, or the sky is nearly homogenous. If the intruder is far from our camera, less information can be obtained with image processing, but from a given distance the shape of the intruder is distinct, thus shape analysis can be used to get more information [74].

Provided that the intruder aircraft is close enough to our UAV its wing can be seen, the relative angle of attack can be obtained and can be used to estimate its trajectory. In this chapter the automatic estimation process is introduced and the precision in miscellaneous situations is studied. The automatic solution is compared to the ground truth and to the theoretically computed values in each situation. For the measurements realistic images rendered by FlightGear flight simulator is used.

4.1 Geometrical description

In this section the geometrical description of the studied situation is introduced. Let us assume that we have one intruder aircraft and it is on a colliding trajectory with our UAV. In this case the position of the intruder on the image plane is almost constant (given no attitude change).

This situation is unobservable, because of the camera projection with our KF based estimation algorithm [9], which estimates the 3D position of the intruder from the change of the coordinates of the intruder in the image plane. Thus, additional information is required in order to determine the relative position of the intruder aircraft. This information can be achieved with running an excitatory manoeuvre [78], which consumes fuel, which is a limited resource on a UAV.

On the other hand, if wingtips of the intruder aircraft can be distinguished on the image, the relative direction angle can be estimated.

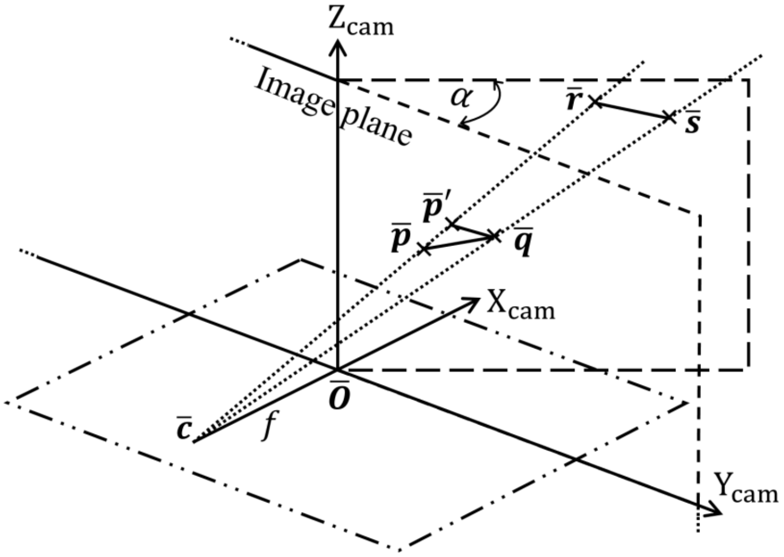


Figure 4.1 Diagram of the relative direction angle (α) calculation:
$\bar{\boldsymbol{c}}$ is the camera centre; f is the focal length; $\bar{\boldsymbol{O}}$ is the centre of the image plane (YZ plane) and the origin;
$\overline{\overline{\boldsymbol{rs}}}$ line segment is the model of the wing of the intruder aircraft in space;
$\overline{\overline{\boldsymbol{pq}}}$ is the wing in image plane; $\bar{\boldsymbol{p}}'$ is the projection of $\bar{\boldsymbol{p}}$ to the horizontal line that goes through $\bar{\boldsymbol{q}}$

Provided that the intruder is coming towards us, it grows in the image. In the beginning this growth is slow and later it accelerates. The relative bank angle of the intruder in the picture, using the coordinates of the wingtips, is measurable.

As shown in Figure 4.1 the wing of the intruder in the image plane is projected to $\overline{\overline{\boldsymbol{pq}}}$ and in space it is $\overline{\overline{\boldsymbol{rs}}}$. It is assumed that the wing of the intruder is horizontal, that is parallel with Y, assuming straight level flight. The centre of our coordinate system is the central point of the recorded image and the YZ plane is the image plane. It is assumed that the images are transformed into the NED frame.

If the intruder is not in XY plane, that is none of its wingtip coordinates are 0 in the camera coordinate system, the line going through the two wingtips includes an angle with Y, introduced by the Z axis offset. Assuming $\overline{\bar{\boldsymbol{q}}\bar{\boldsymbol{p}}'}$ is parallel with Y, from this $\overline{\bar{\boldsymbol{p}}\bar{\boldsymbol{q}}\bar{\boldsymbol{p}}'}$ angle we would like to estimate the intruder's relative angle in 3D (α) that is its direction, which can be used to enhance the estimation. Consequently this $\overline{\bar{\boldsymbol{p}}\bar{\boldsymbol{q}}\bar{\boldsymbol{p}}'}$ depends on the angle α and the subtended angle in which it is seen. This subtended angle (ϕ) is calculated as follows:

$$\phi = 2 \cdot \tan^{-1}\left(\frac{\|\bar{\boldsymbol{p}}-\bar{\boldsymbol{q}}\|}{f}\right) \tag{4.1}$$

If the intruder is on the XY horizontal plane, $\bar{\boldsymbol{p}}$ equals $\bar{\boldsymbol{p}}'$ and the α angle cannot be estimated with this algorithm. The altitude of our UAV can be easily changed with acceleration or deceleration, which consumes less fuel than the complex excitatory manoeuvre mentioned before. The angle α can be calculated as follows. From the measurement we have:

$$\bar{\boldsymbol{p}}(0, p2\ , p3)\ \ \bar{\boldsymbol{q}}(0, q2, q3)\ \ \bar{\boldsymbol{c}}(-f, 0, 0) \tag{4.2}$$

where $\bar{\boldsymbol{c}}$ is the camera centre and f is the focal length. Vectors pointing form the camera centre to wingtips are:

$$\vec{\boldsymbol{v}} = \bar{\boldsymbol{p}} - \bar{\boldsymbol{c}},\ \vec{\boldsymbol{w}} = \bar{\boldsymbol{q}} - \bar{\boldsymbol{c}}. \tag{4.3}$$

The lines on these points are:

$$\bar{\boldsymbol{l}} = \bar{\boldsymbol{c}} + t \cdot \vec{\boldsymbol{v}},\ \bar{\boldsymbol{m}} = \bar{\boldsymbol{c}} + u \cdot \vec{\boldsymbol{w}}. \tag{4.4}$$

Thus parameters t and u are computed that

$$\langle\, \bar{\boldsymbol{l}} - \bar{\boldsymbol{m}}\,;\ \bar{\boldsymbol{0}}\,\rangle = 0. \tag{4.5}$$

Let us assume that

$$t\ := 1, \text{so } u = \frac{p2}{q2},\ \text{if}\ q2 \neq 0 \tag{4.6}$$

Now $\bar{\boldsymbol{r}}$ and $\bar{\boldsymbol{s}}$ are the following:

$$\bar{\boldsymbol{r}} = \bar{\boldsymbol{c}} + t \cdot \vec{\boldsymbol{v}} = \begin{pmatrix} p1 \\ p2 \\ 0 \end{pmatrix};\ \bar{\boldsymbol{s}} = \bar{\boldsymbol{c}} + u \cdot \vec{\boldsymbol{w}} = \begin{pmatrix} \frac{p2}{q2} \cdot q1 \\ p2 \\ f \cdot \left(\frac{p2}{q2} - 1\right) \end{pmatrix}. \tag{4.7}$$

DOI:10.15774/PPKE.ITK.2014.008

The angle of horizontal projection of $\overline{\overline{\boldsymbol{pq}}}$ and $\overline{\overline{\boldsymbol{rs}}}$ is the angle α. The horizontal projection means that the second coordinates of $\overline{\boldsymbol{p}}$ and $\overline{\boldsymbol{q}}$ are equalized so

$$\overline{\boldsymbol{p}}' \coloneqq \begin{pmatrix} p1 \\ q2 \\ p3 \end{pmatrix}. \tag{4.8}$$

Thus

$$\cos\alpha = \frac{\langle \overline{\boldsymbol{p}}' - \overline{\boldsymbol{q}}\,;\overline{\boldsymbol{r}} - \overline{\boldsymbol{s}} \rangle}{\|\overline{\boldsymbol{p}}' - \overline{\boldsymbol{q}}\|\|\overline{\boldsymbol{r}} - \overline{\boldsymbol{s}}\|}. \tag{4.9}$$

In this model the instances rotated by 180° are equal and the $\alpha = \cos^{-1} X$ function gives good solution in $\alpha = [0°; 180°]$ range. The relative angle α should be in the $[-90°; 90°]$ range, so it is transformed according to the following rules. If $\alpha > 90°$, then $\alpha = 180° - \alpha$, if $\alpha < -90°$, then $\alpha = -180° - \alpha$. With these calculations the expected results are obtained consistently.

4.2 Measurement situations

The accuracy of the calculation is studied with given image resolution and position. Four kinds of situations are examined:

1) With pinhole camera model, the given centroid point of the intruder is projected back from image plane to space to several distances. The wingspan of the intruder is 11m (36 ft. 1 in), which is the wingspan of Cessna 172, a typical light aircraft that shares the airspace with our UAV. Thus the wing is represented by an 11m line segment and is rotated in the previously calculated point. The field of view and resolution of the camera and the distance along x axis is required for the calculation. The fuselage of the aircraft is neglected, which gives an initial error. With these calculations the lower bound of the error is approximated. Two kinds of points are used:

    a. calculated points without rounding to determine the error induced by the limited numerical precision

    b. calculated points with rounding to determine the error induced by the discretization in space

2) With the calculated centroid points in space according to situation 1) images are taken from FlightGear flight simulator. The wingtip coordinates are taken by a human expert from these simulated images and the angle values are calculated from these coordinates.

DOI:10.15774/PPKE.ITK.2014.008

3) Similarly to the above, the intruder points are extracted from the simulated images rendered by FlightGear with our image segmentation algorithm. After that, from intruder pixel coordinates the wingtip coordinates are calculated with the following simple algorithm. The wingtip coordinates are determined by the extremes of the *y* and *z* coordinates in the appropriate order. In order to reduce the error induced by the image formation, the calculated coordinates are refined according to the image pixel values with the following expression:

$$p2_{\mathrm{corrected}} = \frac{\sum_{i=p2-s}^{p2+s} i \cdot \mathrm{G}\left(p2^{i}\right)}{\sum_{i=p2-s}^{p2+s} \mathrm{G}\left(p2^{i}\right)} \tag{4.10}$$

where $p\mathrm{N}_{\mathrm{corrected}}$ is the refined coordinate value, $p\mathrm{N}$ is the original coordinate value, s is the radius, $\mathrm{G}\left(p\mathrm{N}^{i}\right)$ is the grayscale value of the i$^{\mathrm{th}}$ point. This way the original wingtip coordinates which were calculated in the binary image are refined according to the grayscale pixel values from the surrounding region.

4) In this measurement setup the images are recorded by a full HD interlaced video camera with 50° field of view, in an outdoor environment. The background is clear blue sky. The intruder is placed according to the previous measurements. The shape of the intruder is correctly segmented from the images. Images are noisy because of the video compression, the interlaced camera and wind effects. In this situation an aircraft Matchbox is used as the intruder.

4.3 Precision determination

In this section the measurements are described in situations introduced in chapter 4.2. The position dependence of the error and the effect of the discretization are shown.

4.3.1 Pinhole camera

First the pinhole camera model is used. Provided that the points are calculated without rounding, this approach should come close to the theoretical limits and the computation error has to be near zero. The measurements are done with double precision and the error of the angles is in the range of picodegree as shown in Figure 4.2, which is the range of the error introduced by the numeric representation. Indeed this error can be seen as zero in the point of the computation part.

In Figure 4.2 a) the real rotation angles versus the calculated angle values are shown, and the part b) depicts the error of the estimated angle, which is the difference between the two

DOI:10.15774/PPKE.ITK.2014.008

angles. The distance along the x axis to the image plane is 2 km (1.24 miles) and the intruder is seen in 7° azimuth and elevation angle offset.

Let us assume that a typical HD camera is used to record the scene. This camera is calibrated and the recorded pictures are undistorted, thus the pinhole camera model can be a valid approximation. The difference between this measurement scenario and the one stated above is that here the image coordinates are discrete integer values and the image plane is finite.

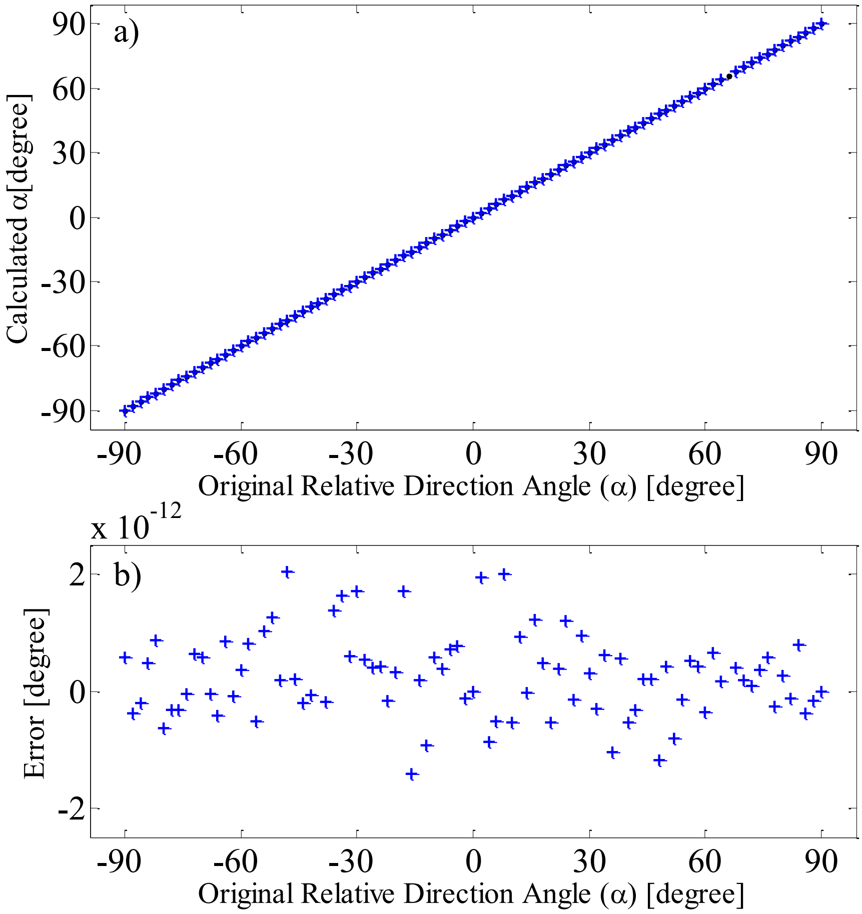


Figure 4.2 α angles calculated from pinhole model and their error to ground truth;
a) the original angles with black dots (covered by calculated angles) and the calculated angles with blue plus signs; b) the error for each calculated angle

DOI:10.15774/PPKE.ITK.2014.008

According to the measurements, the precision of the estimation with a given camera depends on the subtended angle and the relative distance along the X axis. The reasons are that the larger the distance the smaller the intruder in the image and the bigger the altitude difference the more you observe the wing of the intruder.

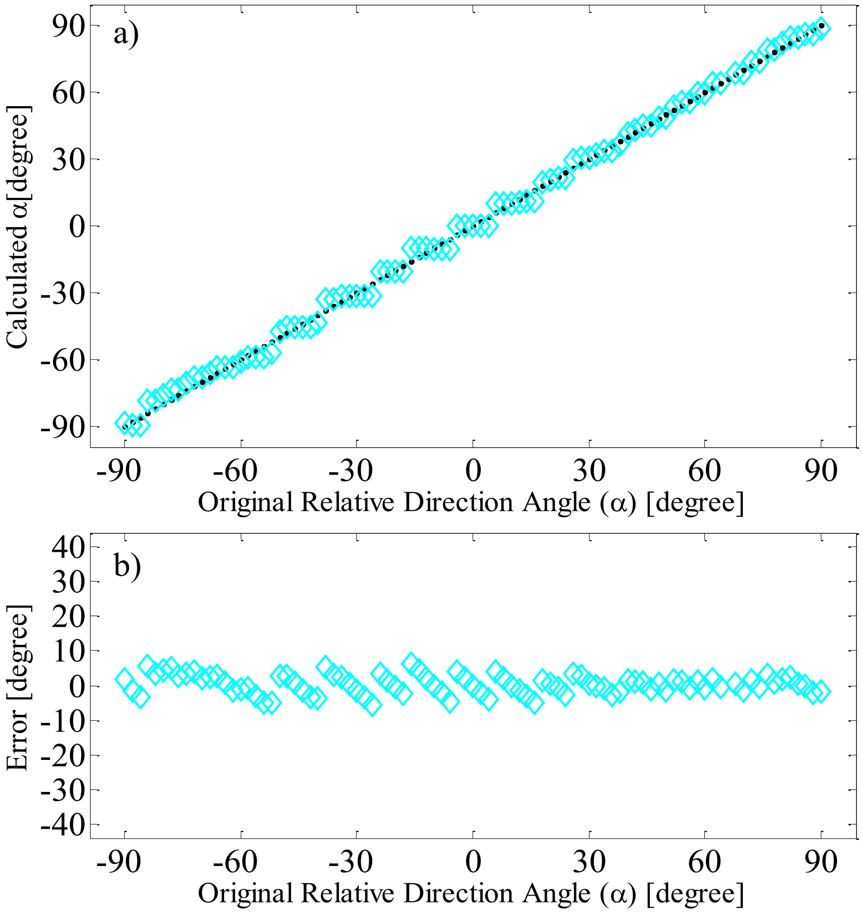


Figure 4.3 α angles calculated with rounding and their error to original rotation angles;
a) the original angles with black dots and the calculated angles with cyan diamonds;
b) the error values for each calculated angle (max ±6°);
the intruder is seen in (24°, 14°) direction and the distance along X axis is 1km

The three figures (Figure 4.3, Figure 4.4, Figure 4.5) show examples where the relative distance along the x axis is 1 km (0.62 miles), the resolution is 1920x1080 pixels, the horizontal field of view is 50° and the pixels are squares. The wingspan of the intruder is 11m (36 ft. 1 in), which is the wingspan of Cessna 172.

The size of intruder in the image plane is between 15 and 20 pixels, depending on the rotation angle and the position. The intruder is seen in 14°, 7° and 3.5° elevation successively, and it is seen constantly in 24° azimuth.

DOI:10.15774/PPKE.ITK.2014.008

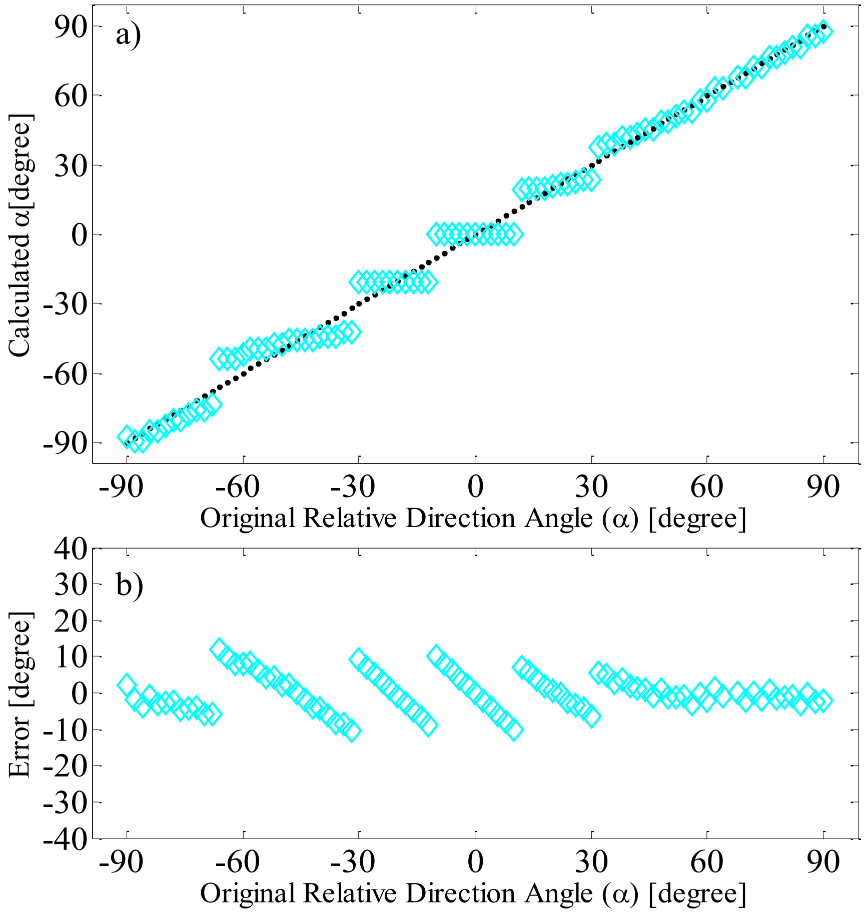


Figure 4.4 α angles calculated from pinhole model with rounding;
same as before, the subtended angle is (24°, 7°) and the maximum error is ±11°;
the asymmetry in the error function is caused by the position of the intruder

DOI:10.15774/PPKE.ITK.2014.008

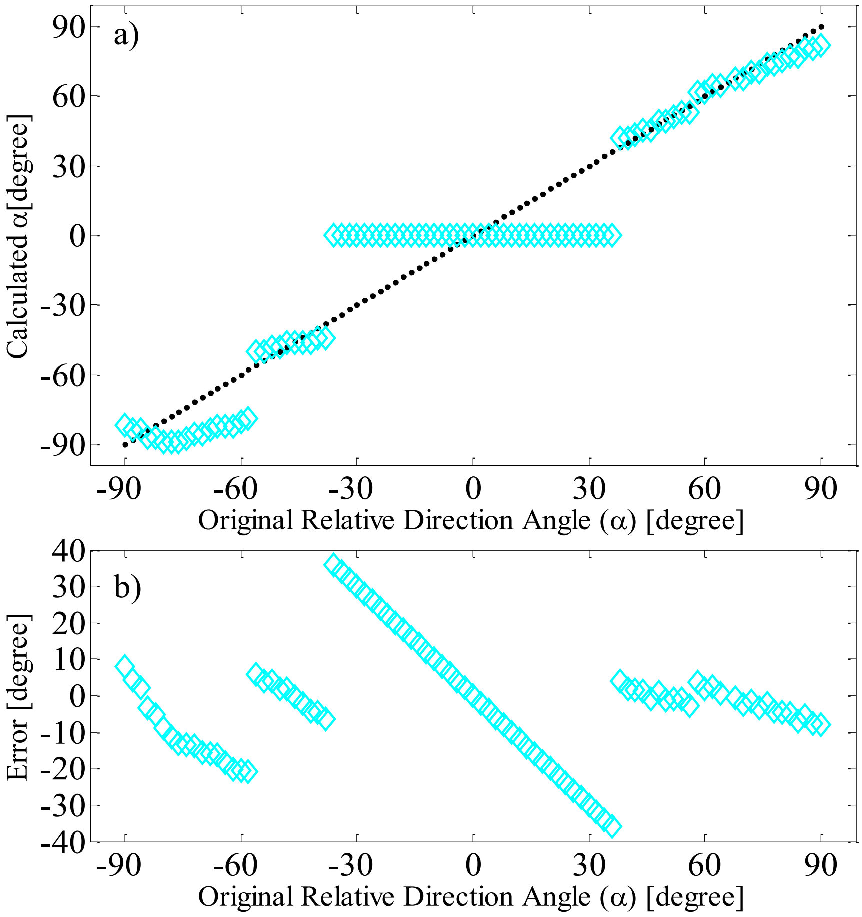


Figure 4.5 α angles calculated from pinhole model with rounding; same as before the intruder is seen in (24°, 3.5°) direction and the maximum error is ±37°

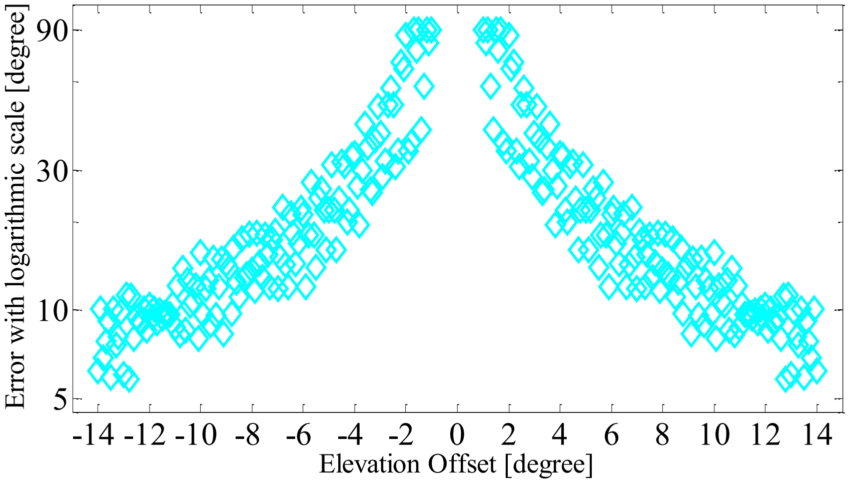


Figure 4.6 Maximum of absolute value of the errors of the rounded α calculated with pinhole camera model in different positions and from 1 km distance along the X axis; on the horizontal axis the elevation offset; on the vertical axis the error in degree with logarithmic scale

DOI:10.15774/PPKE.ITK.2014.008

Figure 4.6 shows the maximum error values in each α with constant azimuth of 24° and with changing elevation from -14° to 14°. In each position the intruder is rotated with angles from -90° to 90° and the maximum of the absolute of the error is chosen. This shows the position dependence of the calculated α. Figure 4.6 depicts that the initial error is ±6° and the closer the intruder is to the horizontal axis the bigger the error we get.

Similarly, the bigger the distance along the X axis the smaller the intruder is in the image, therefore the spatial discretization gives higher error value, as shown in the figures Figure 4.7. and Figure 4.8. Furthermore, the proximity to Y has a greater effect on the error than in the smaller distance case (Figure 4.8).

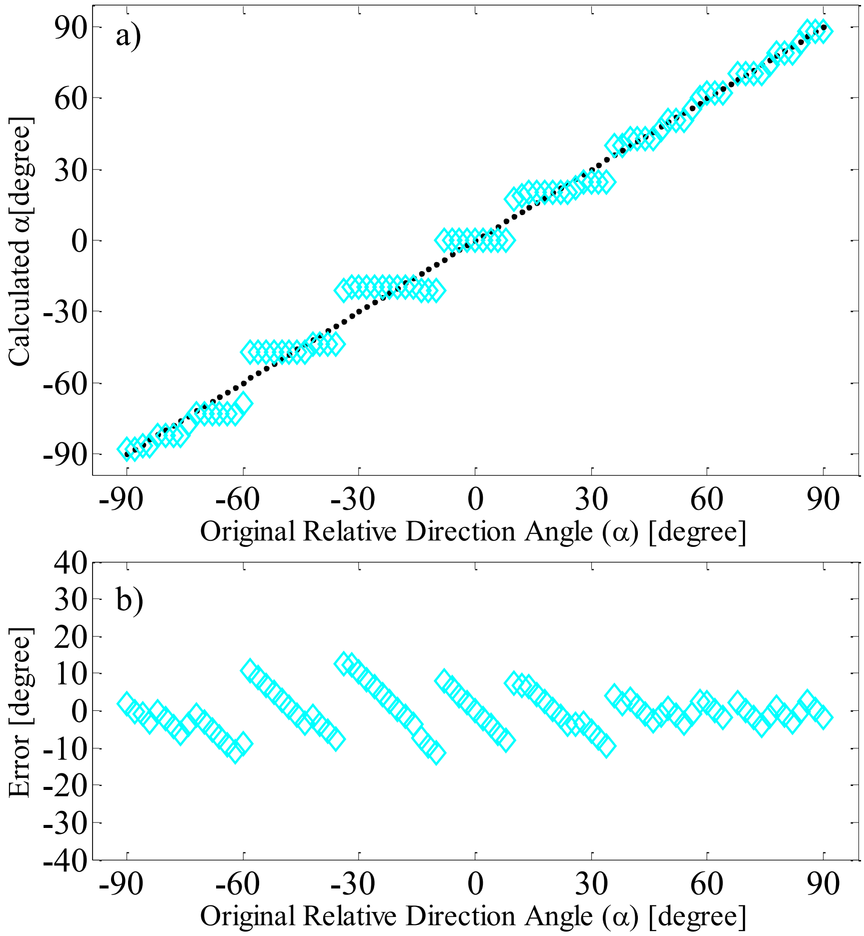


Figure 4.7 α angles calculated from pinhole model with rounding;

same as before the intruder is seen in (24°, 14°) direction and it is 2km to the camera

b) the error values for each calculated angle (max ±13°);

DOI:10.15774/PPKE.ITK.2014.008

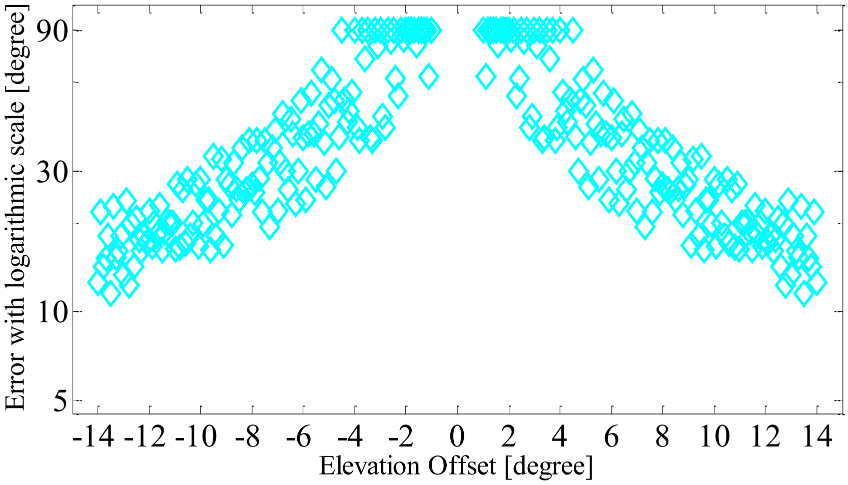


Figure 4.8 Maximum of absolute value of the errors of the rounded α angles calculated with pinhole camera model in different positions and from 2 km distance along the X axis; on the horizontal axis the elevation offset; on the vertical axis the error in degree with logarithmic scale

4.3.2 Points by human expert on simulated images

In our simulation environment pictures are taken and the wingtip pixel coordinates are selected by a human expert. The intruder is placed in space according to section 4.2. 1) and in every position it is rotated by specific angles in the XY plane. The resolution is 1920x1080 pixels and the horizontal field of view is 50° and the pixels are squares, similarly to the previous case in 4.3.1.

In Figure 4.9 a) the ground truth α values are with black (covered). The angles calculated from pinhole camera model are shown with blue plus signs; the values calculated from rounded coordinates are shown with cyan diamonds and the angles calculated from points selected by hand are shown with green asterisks. On Figure 4.9 b) the error values are shown and the colours are similar to previous. The figure depicts only the result of the measurement in one specific distance. The intruder was placed in 9 different positions and was rotated with 9 different angles (-80°, 80°, -40°, 40°, -10°, 10°, -5°, 5°, 0°). The other results obtained from another distances are similar to that are described previously in section 4.3.1, thus the altitude difference is in inverse ratio to the error.

DOI:10.15774/PPKE.ITK.2014.008

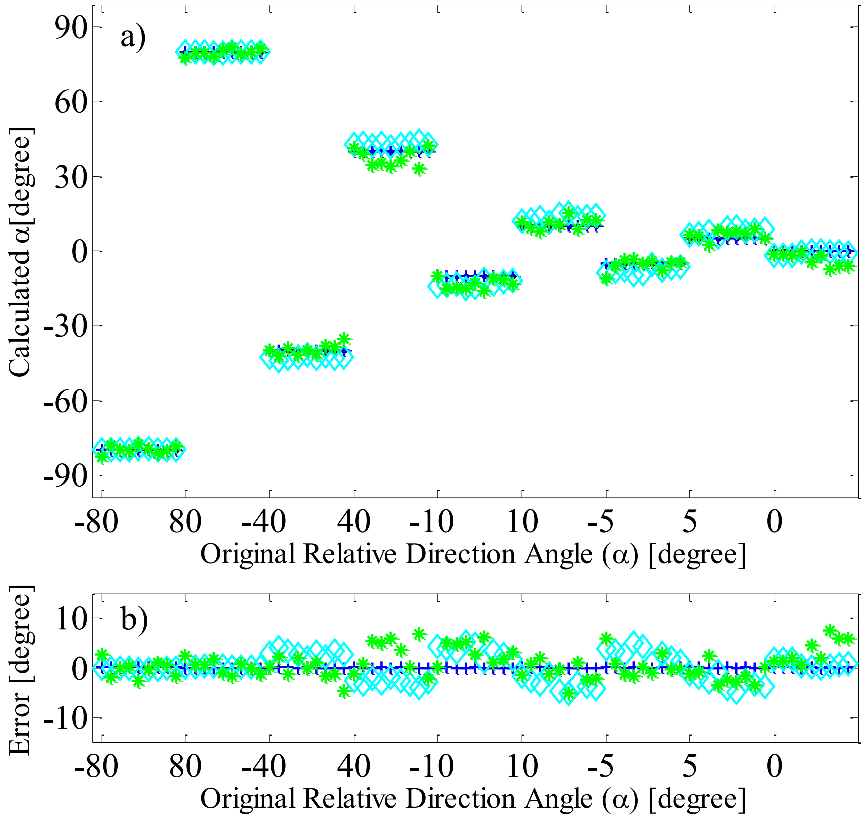


Figure 4.9 α angles calculated from coordinates selected by a human expert on simulated images; a) angles in different elevation, on the vertical axis the angle values, on the horizontal axis the real rotation angles in 9 different positions; b) the error; original angles with black dots, angles calculated from pinhole model with blue plus signs, angles calculated from pinhole model with rounding with cyan diamonds, angles calculated from coordinates selected by hand with green asterisks

The measurements above shows that with good wingtip coordinates in realistic situation the error can be close to the theoretical minimum.

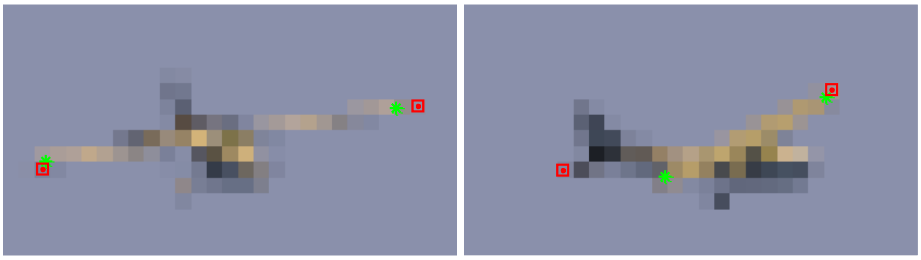


Figure 4.10 Images of wingtip points selected by a human expert and by the algorithm on images generated by FlightGear simulator; on the left an example when the algorithm gives good points, on the right when the algorithm make a mistake; with green asterisks the points given by human expert, with red squares points given by the algorithm

DOI:10.15774/PPKE.ITK.2014.008

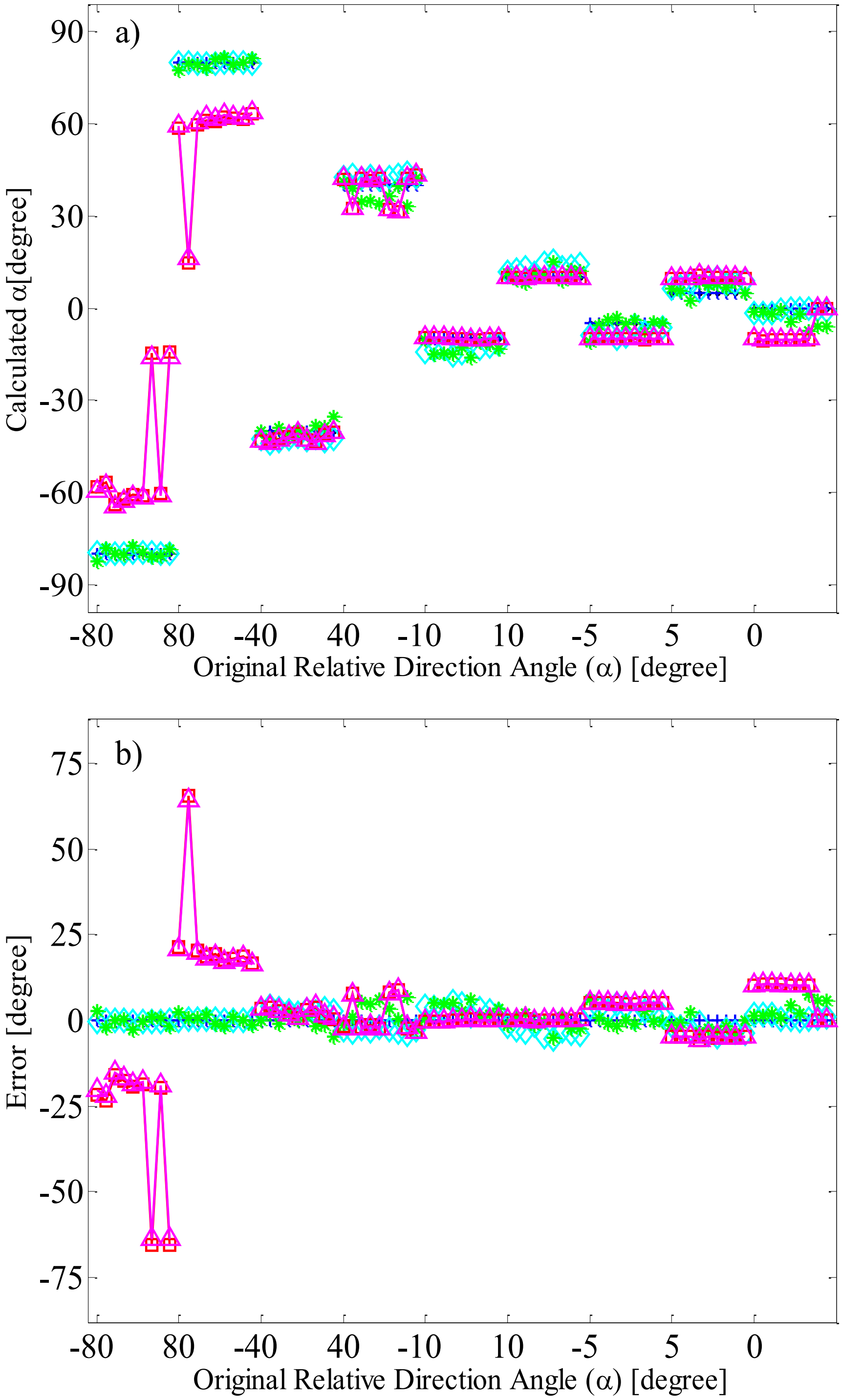


Figure 4.11 α angles calculated from coordinates calculated by the automatic algorithm simulated on images; a) angles in different vertical positions; b) the error; original angles with black dots (covered), angles calculated from pinhole model with blue plus signs, angles calculated from pinhole model with rounding with cyan diamonds, angles calculated from coordinates selected by hand with green asterisks, angles calculated automatically with red squares and the corrected values with magenta triangles

4.3.3 Points by automatic algorithm on simulated images

The error of the automatic wingtip detection algorithm running on simulated images has been measured. The simple algorithm determines the wingtip coordinates from the segmented images. The extreme of Y and Z coordinates are used in appropriate order to get the coordinates (Figure 4.10).

Figure 4.11 depicts one example, where similarly to section 4.3.2, the intruder had been placed in a specific locations in space and then it was rotated with specific angles (same as before). In the figure the ground truth is with black dots (covered); the values from pinhole camera model are with cyan asterisks and blue plus signs; the values form points selected by human expert are green asterisks; the values from automatic algorithm are with red squares and the values calculated from corrected points are with magenta triangles.

In this case when the intruder had been rotated with 80° and with -80° angles, the error of the estimation is bigger, because the simple algorithm could not distinguish between the pixels of the wing and the pixels of the tail. In contrast, in the mid-range the performance of this really simple algorithm is almost the same as the performance of the human expert (close to the theoretical limit).

4.3.4 Points by automatic algorithm on images from real video data

In this case images are taken from video data recorded with a full HD video camera. The resolution is 1920x1080 pixels and the approximate horizontal field of view is 50° and the pixels are squares, like in the previous cases. A frame from the video is shown on Figure 4.12.

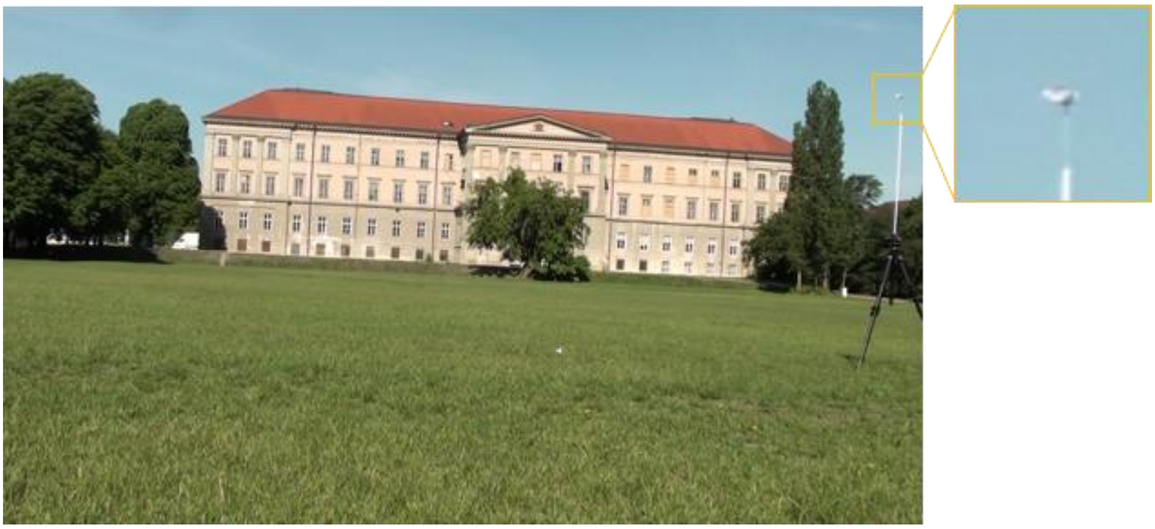


Figure 4.12 One frame from a recorded video; the intruder is shown in the enlarged picture

DOI:10.15774/PPKE.ITK.2014.008

Figure 4.12 depicts one example, where an airplane matchbox is used as an intruder. The intruder has a wingspan of 10 cm, so it is placed 454 cm to the camera to have the same size on the image plane as a Cessna 172 from 500 m.

As shown in Figure 4.13, with the automatic algorithm in this situation the theoretical precision can be reached. The results of the automatic algorithm are with magenta and the results from the discretized real coordinates are with cyan. The black is the ground truth. The noise introduced by the video camera and the environment is suppressed with a simple averaging in time, the calculated α angles are averaged for 25 frame (1s).

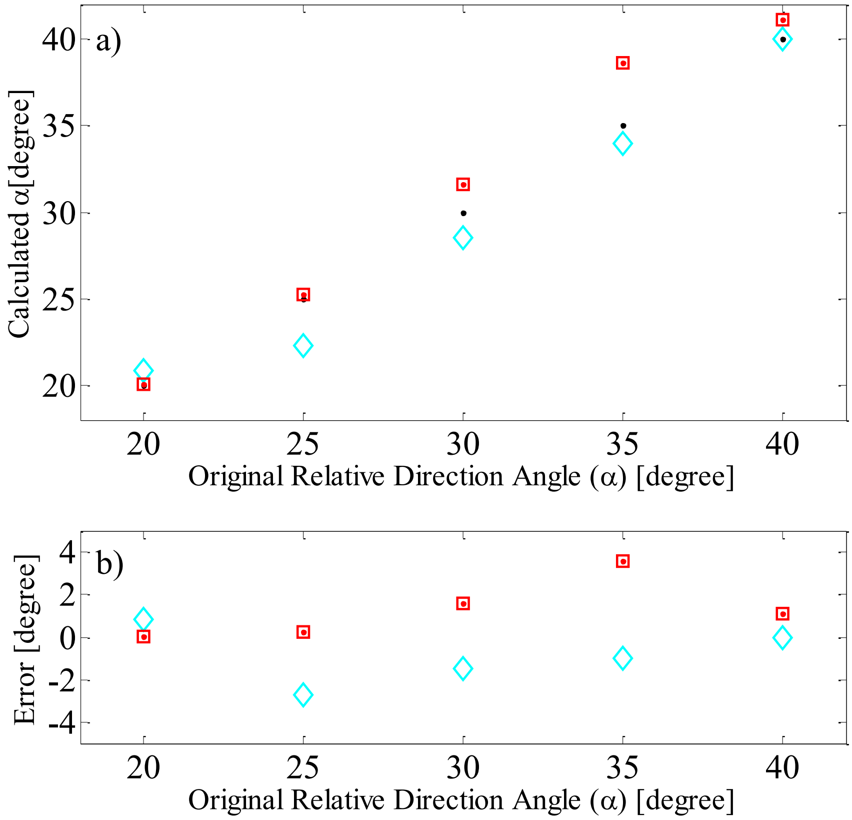


Figure 4.13 α angles calculated from coordinates calculated by the automatic algorithm on images from real video; the intruder is seen in (24°, 14°) direction and the equivalent distance along X axis is 500m; b) the error values for each calculated angle (max ±4°); original angles with black, angles calculated from pinhole model with rounding with cyan, angles calculated automatically with red

DOI:10.15774/PPKE.ITK.2014.008

4.4 Conclusion

The reachable accuracy of the orientation calculation of visually detected remote airplanes was studied. The orientation calculation was based on the detection of the wingtips. As it turned out the relative orientation of the remote aircraft (depicted by α) can be calculated if it is on a straight course, and its level differs from the observer. Naturally, the orientation measurement is more accurate when the level difference is higher, and the airplane is closer. The exact reachable accuracy figures are shown in charts, and their calculation methods are given. The acquired measurements will be used to enhance the estimation accuracy of the currently existing EKF based sense and avoid system.

DOI:10.15774/PPKE.ITK.2014.008

Chapter 5

Error Analysis of Camera Rotation Estimation Algorithms

In this chapter four camera pose estimation algorithms are investigated in simulations. The aim of the investigation is to show the strengths and weaknesses of these algorithms in the aircraft attitude estimation task. Two main issues are addressed with these measurements, one is the sense-and-avoid capability of the aircraft and the other is sensor redundancy. Both parts can benefit from a good attitude estimate. Thus, it is important to use the appropriate algorithm for the camera rotation estimation. Results show that many times even the simplest algorithm can perform at an acceptable level of precision for the sensor fusion and outperform more sophisticated algorithms.

The sense-and-avoid task has to be run in critical situations as well, for example when one or more sensor fails. One solution is redundancy in the sense of the number of similar sensor modules or in different sensor modalities. In this case the use of our camera can be broadened to localisation task besides its main function in collision avoidance.

On the other hand with an IMU/Camera fusion better accuracy can be achieved in the ego motion as shown in [7]. With these more accurate results our SAA algorithm can be speed-up which provides even higher separation distance or the avoidance of aircrafts with higher speed.

In [76] performance comparison of tight and loose (KF based), INS-Camera integration is studied by Chu et al. through simulations. The paper shows that tight coupling can provide

DOI:10.15774/PPKE.ITK.2014.008

higher accuracy but it is less stable due to the linearization methods of the filters. Thus loose integration is favourable in low cost systems.

In [79] a monocular camera, INS and GNSS integration is presented for ground vehicles by Chu et al. This system is validated through a real drive test and results show that the system based on camera-INS fusion outperforms the conventional INS-GNSS systems. However the GNSS measurements are not included in the camera-INS system. As stated in the paper this step can further improve the performance of the system. Furthermore, the real-time functionality is a challenging task because of the image processing algorithms involved.

For aircraft attitude estimation many different image processing algorithms can be used from a simple homography based calculation to the more complicated five point algorithm. The question is how these algorithms can be ranked based on their performance and computational complexity in realistic simulations.

The inventors of these algorithms provide information about their accuracy [80], [81], and there are other papers which assemble and compare different algorithms from some perspective [82]. To the best of my knowledge there is no analysis for these algorithms for GPS/IMU/Camera fusion which can easily show the strength and weaknesses of a specific algorithm in this scenario.

The error analysis of the four given algorithms is done with realistic flight paths generated by the HIL simulator. The camera model is based on the real calibration matrix of the camera, used on board of our test aircraft. These results can give a general idea that in which situation which algorithm can be used effectively. As an application example simulation and measurement results from our camera-IMU (including GPS) sensor integration are shown in Chapter 7.

5.1 Algorithmic background

In this section the basics of used camera pose calculation algorithms are introduced. For the measurements four feature point based relative pose estimation algorithms are chosen. A homography based solution as a basic algorithm with small computational need but with less accuracy. The eight point algorithm, as standard algorithm in epipolar geometry. The five point algorithm, as one of the state of the art algorithms with higher computational need, but with promising stability over the various scenes. Finally, MLESAC, as an iterative, stochastic solution. Other algorithms can be tested in the future with the same framework. The coordinate frames and the transformations are defined in section 3.1.

DOI:10.15774/PPKE.ITK.2014.008

5.1.1 IMU models

Our IMU consists of sensors which are required for outdoor waypoint navigation. In our system the conventional accelerometer, rate gyro, differential and absolute pressure sensor and magnetometer are completed with a GPS unit [14].

5.1.2 Camera measurements

The electro optical sensor is modelled as a projective camera. The camera matrix $\bar{\bar{\mathbf{P}}}$ consists of the internal and external parameters of the camera and can be decomposed as follows:

$$\bar{\bar{\mathbf{P}}} = \bar{\bar{\mathbf{K}}} \left[\bar{\bar{\mathbf{R}}} \,\middle|\, \bar{\boldsymbol{t}} \right] \tag{5.1}$$

where $\bar{\bar{\mathbf{R}}}$ and $\bar{\boldsymbol{t}}$ are the rotation and translation of the camera, which are the extrinsic parameters. $\bar{\bar{\mathbf{K}}}$ contains the intrinsic parameters: the focal length f in pixels (it can be different in the x and y directions) and the position of camera principal point $\bar{\mathrm{p}}$ in the image plane as follows:

$$\bar{\bar{\mathbf{K}}} = \begin{bmatrix} f_{\mathrm{x}} & 0 & p1 \\ 0 & f_{\mathrm{y}} & \mathrm{p2} \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

Here the resolution of the camera is interesting as well, because the effect of pixelization and spatial resolution is studied. A projective camera can be characterized by the angular resolution of the central pixel (or CPAR), which is defined as follows:

$$CPAR = \tan^{-1}\frac{1}{f} \tag{5.3}$$

where f is the focal length of the camera. With this measure cameras with different resolution and field of view can be compared.

5.1.3 Feature extraction and matching

On the consecutive frames a modified Harris corner feature extraction is used [74]. Corner features are extracted but two constraints are used:

1) the feature points should be farther to each other in the image than a given threshold and

2) feature points should be in the ground region, below the horizon.

DOI:10.15774/PPKE.ITK.2014.008

The latter constraint can be satisfied by an adaptive threshold, which is applied before the corner detection. With these two constraints the number of the feature points is limited. The first constraint can assure in most cases that degenerate feature point combinations are avoided.

Our UAV will be used mainly in rural environment, where there are only a few tall buildings (if any). It means that static features according to the NED frame are located on the ground. That is why feature points are searched for on the ground. This is viable, because except the take-off and a few manoeuvres, the ground can be seen by the camera.

5.1.4 Homography

As a basic solution for the problem of camera pose estimation a scene homography based algorithm is tested. In this case the assumption is made that the movement of the camera is so small that the effect of the translational motion can be neglected thus only the camera rotation is calculated. The basic equations of the calculation are used for planar panoramic mosaicking as well and also known as inhomogeneous DLT [72]. The equations are as follows:

$$\bar{\bar{\mathbf{A}}} = \begin{bmatrix} 0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\ x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \end{bmatrix}$$

$$\bar{\bar{\mathbf{A}}} * \bar{\boldsymbol{h}} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix} \tag{5.4}$$

where $x_i \leftrightarrow x_i'$ and $y_i \leftrightarrow y_i'$ are the coordinates of the corresponding feature points on the consecutive frames, and the elements of $\bar{\boldsymbol{h}}$ vectors are the elements of the homography matrix up to an unknown scale. This scale is given by w_i and w_i' for each frame and each feature point. An optimal solution for the homography can be yielded with the SVD of the $\bar{\bar{\mathbf{A}}}$ matrix. And again the optimal rotation can be calculated from the SVD of the resulting homography matrix. More details about the calculation can be found in [72].

5.1.5 Eight point algorithm

As a more promising variant the normalised eight point algorithm is tested [72]. From feature point pairs the fundamental matrix $\bar{\bar{\mathbf{F}}}$ can be calculated. $\bar{\bar{\mathbf{F}}}$ is defined by the epipolar constraint as follows:

$$\bar{\boldsymbol{x}}'^{\mathrm{T}}\, \bar{\bar{\mathbf{F}}}\, \bar{\boldsymbol{x}} = 0 \tag{5.5}$$

If one has a calibrated camera the essential matrix $\bar{\bar{\mathbf{E}}}$ can be obtained from $\bar{\bar{\mathbf{F}}}$ by multiplying with the camera matrix $\bar{\bar{\mathbf{K}}}$ such as:

$$\bar{\bar{\mathbf{E}}} = \bar{\bar{\mathbf{K}}}'^{\mathrm{T}}\, \bar{\bar{\mathbf{F}}}\, \bar{\bar{\mathbf{K}}} \tag{5.6}$$

Here we have only one camera, so $\bar{\bar{\mathbf{K}}}' = \bar{\bar{\mathbf{K}}}$.

5.1.6 Five point algorithm

In the case of calibrated cameras the $\bar{\bar{\mathbf{E}}}$ matrix can be computed directly from five point correspondences because it has only five degrees of freedom. In [81] and [84] an efficient algorithm is presented, which is numerically more stable than other methods. Furthermore, the five point algorithm should be accurate in the case of pure rotational or pure translational movement as well.

5.1.7 MLESAC

As the member of the RANSAC family, the MLESAC algorithm is tested [82]. This is a more advanced RANSAC variant where the fundamental matrix is robustly calculated based on probability features.

This algorithm is not the best with respect to accuracy as stated in [83] but the computational complexity of the algorithm is reasonable and the implementation is available online.

5.1.8 Camera rotation and translation from epipolar matrices

With the eight point algorithm, the MLESAC and the five point algorithm the E matrix can be calculated from point correspondences. From $\bar{\bar{\mathbf{E}}}$ the two camera matrices can be calculated in canonical form (that is the first camera matrix is $\bar{\bar{\mathbf{P}}} = \left[\, \bar{\mathbf{I}} \mid \bar{\mathbf{0}} \,\right]$ and the second is $\bar{\bar{\mathbf{P}}}' = \left[\, \bar{\bar{\mathbf{R}}} \mid \bar{\boldsymbol{t}} \,\right]$), because $\bar{\bar{\mathbf{E}}}$=$[\, \bar{\boldsymbol{t}}\,]_{\times}\bar{\bar{\mathbf{R}}}$, where $[\, \bar{\boldsymbol{t}}\,]_{\times}$ is a skew symmetric form of translation vector t representing vector cross product. For the calculation E has to be decomposed with SVD as follows:

$$\bar{\bar{\mathbf{E}}} = \bar{\bar{\mathbf{U}}}\, \mathrm{diag}(1{,}1{,}0)\, \bar{\bar{\mathbf{V}}}^{\mathrm{T}} \tag{5.7}$$

From that four solutions can be constructed for the second camera. Only one of them satisfy the chirality constraint [85] that is in only one arrangement are the reprojected feature points in front of both cameras [72] for example:

$$\overline{\overline{\mathbf{P}}}' = \left[\, \overline{\overline{\mathbf{U}}}\,\overline{\overline{\mathbf{W}}}^{\mathrm{T}}\,\overline{\overline{\mathbf{V}}}^{\mathrm{T}} \mid \bar{\boldsymbol{u}}_{\mathbf{3}} \,\right] \tag{5.8}$$

where $\bar{\boldsymbol{u}}_{\mathbf{3}}$ is the 3rd column of $\overline{\overline{\mathbf{U}}}$ and

$$\overline{\overline{\mathbf{W}}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.9}$$

5.1.9 Reconstruction of aircraft attitude change from camera rotation matrix

From the matched feature points in two consecutive camera frames the camera rotation matrix $\overline{\overline{\mathbf{R}}}$ and translation vector $\bar{\boldsymbol{t}}$ (with scale ambiguity) can be reconstructed assuming canonical cameras. Here, normalised coordinates and calibrated cameras are considered as stated before, but the effect of normalization will be considered only in the next section.

This way the $\bar{\boldsymbol{x}}_{\mathrm{cam}}$ (not normalized) vector can be transformed into the first frame as (using homogenous coordinates):

$$\bar{\boldsymbol{x}} = \overline{\overline{\mathbf{P}}} \begin{bmatrix} \bar{\boldsymbol{x}}_{\mathrm{cam}} \\ 1 \end{bmatrix} = \begin{bmatrix} \overline{\overline{\mathbf{I}}} & \bar{\mathbf{0}} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}}_{\mathrm{cam}} \\ 1 \end{bmatrix} = \bar{\boldsymbol{x}}_{\mathrm{cam}} \tag{5.10}$$

The same $\bar{\boldsymbol{x}}_{\mathrm{cam}}$ vector can be transformed into the second frame considering the transformation between the two frames which is the $\overline{\overline{\mathbf{P}}}'$camera matrix:

$$\bar{\boldsymbol{x}}' = \overline{\overline{\mathbf{P}}}' \begin{bmatrix} \bar{\boldsymbol{x}}_{\mathrm{cam}} \\ 1 \end{bmatrix} = \begin{bmatrix} \overline{\overline{\mathbf{R}}} & \bar{\mathbf{t}} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{x}}_{\mathrm{cam}} \\ 1 \end{bmatrix} = \overline{\overline{\mathbf{R}}}\,\bar{\boldsymbol{x}}_{\mathrm{cam}} + \bar{\mathbf{t}} \tag{5.11}$$

$\bar{\boldsymbol{x}}'$ is the image of point X in the second (rotated and translated) camera frame which means the rotation and translation of the aircraft body frame. This way $\bar{\boldsymbol{x}}'$ can be also constructed by considering the changed $\overline{\overline{\mathbf{BE}}}'$ matrix and $\overline{\boldsymbol{eb}}'_{\mathrm{earth}}$ vector:

$$\bar{\boldsymbol{x}}' = \bar{\boldsymbol{x}}_{\mathrm{cam}}{}' = \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}'\left(\bar{\boldsymbol{x}}_{\mathrm{earth}} - \overline{\boldsymbol{eb}}'_{\mathrm{earth}}\right) \tag{5.12}$$

From the two representations of $\bar{\boldsymbol{x}}'$ and the original expression for $\bar{\boldsymbol{x}}_{\mathrm{cam}}$ by considering

$$\overline{\overline{\mathbf{BE}}}' = \overline{\overline{\Delta}}\,\overline{\overline{\mathbf{BE}}} \text{ and} \tag{5.13}$$

$$\overline{\boldsymbol{eb}}'_{\mathrm{earth}} = \overline{\boldsymbol{eb}}_{\mathrm{earth}} + \overline{\boldsymbol{\Delta eb}}_{\mathrm{earth}} \tag{5.14}$$

one gets:

$$
\begin{gathered}
\bar{\boldsymbol{x}}' = \bar{\bar{\mathbf{R}}}\,\bar{\boldsymbol{x}}_{\text{cam}} + \bar{\mathbf{t}} = \bar{\bar{\mathbf{R}}} \cdot \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}'\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}{}'\right) + \bar{\mathbf{t}} \\
\bar{\boldsymbol{x}}' = \overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}} - \overline{\boldsymbol{\Delta eb}}_{\text{earth}}\right) \\
\bar{\boldsymbol{x}}' = \overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right) \underbrace{-\overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}\,\overline{\boldsymbol{\Delta eb}}_{\text{earth}}}_{\bar{\mathbf{t}}} \;\Rightarrow \\
\bar{\bar{\mathbf{R}}} \cdot \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right) = \overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right)
\end{gathered}
\tag{5.15}
$$

From the last equation above, the aircraft attitude change T_Δ results as follows:

$$
\begin{gathered}
\bar{\bar{\mathbf{R}}} \cdot \overline{\overline{\mathbf{CB}}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right) = \overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}\left(\bar{\boldsymbol{x}}_{\text{earth}} - \overline{\boldsymbol{eb}}_{\text{earth}}\right) \\
\bar{\bar{\mathbf{R}}} \cdot \overline{\overline{\mathbf{CB}}} = \overline{\overline{\mathbf{CB}}}\,\bar{\bar{\Delta}} \\
\bar{\bar{\Delta}} = \overline{\overline{\mathbf{CB}}}^{\mathrm{T}} \cdot \bar{\bar{\mathbf{R}}} \cdot \overline{\overline{\mathbf{CB}}}
\end{gathered}
\tag{5.16}
$$

In the application example the $\overline{\overline{\mathbf{CB}}}$ transformation matrix changes the order of axes from body to camera coordinate system (see Figure 3.3):

$$
\overline{\overline{\mathbf{CB}}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}
\tag{5.17}
$$

5.2 Simulation Methods

In this section the methodology of the error analysis of image processing are introduced. In particular, the simulation environment with the real flight paths used for the measurements is shown and the error measures used for the analysis are defined. Furthermore, an empirical correction term for the homography algorithm is described with which the error introduced by the translation can be reduced.

5.2.1 Simulation environment

The simulation environment is based on the MATLAB EGT toolbox [86]. This toolbox was developed at Siena Robotics and Systems Lab and it provides wide a set of functions for multiple view geometry. It can plot the whole scene with feature points and cameras as well as the projected frames. It handles camera calibration matrices, so it is possible to use realistic camera projections.

DOI:10.15774/PPKE.ITK.2014.008

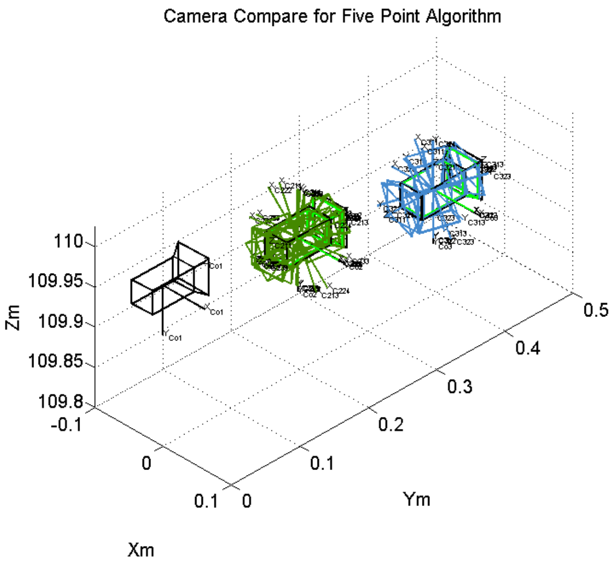


Figure 5.1 Cameras in the EGT frame

For the tests realistic flight paths are used, which are generated by a HIL simulator. The test were run on two flight paths: 1) a sinusoidal path with almost constant altitude and 2) a zigzag path with also nearly constant altitude. The resulting error figures show similar phenomena, that is why only one of them is shown as an example.

For the tests 350 feature points are placed randomly with uniform distribution in a right prism which is 2000m wide, 3000m long and 30m tall. The point coordinates are between -1000 and 1000 in the Y direction and from 0 to 3000 in the X direction. The maximum altitude of the points is 23 m and the Z coordinate starts from 3 m beyond the ground level to simulate small holes.

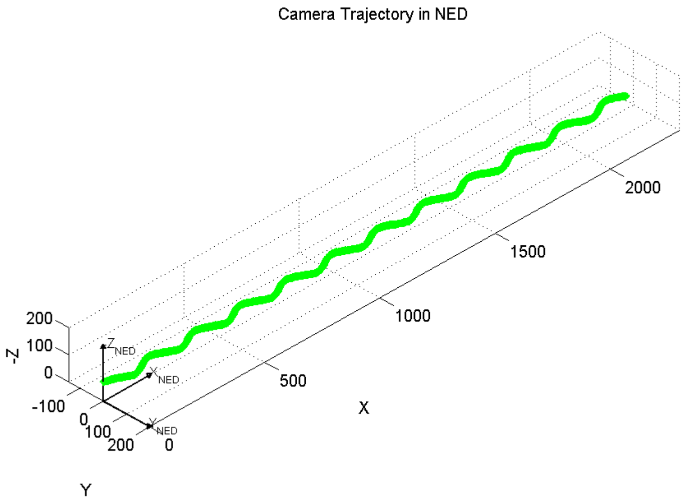


Figure 5.2 Sinusoidal path in the NED frame

DOI:10.15774/PPKE.ITK.2014.008

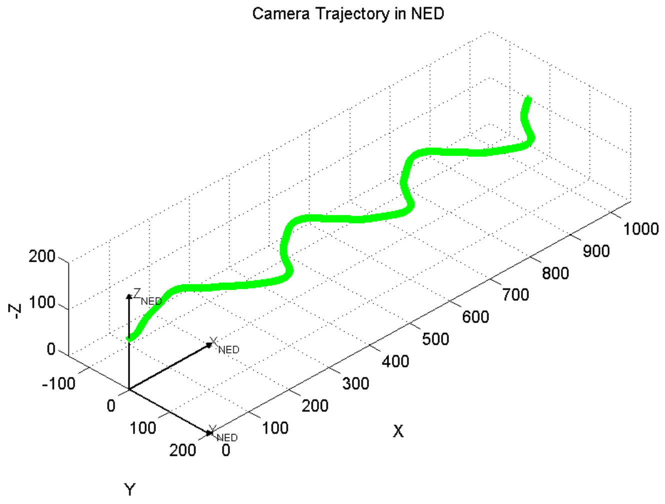


Figure 5.3 Zigzag path in the NED frame

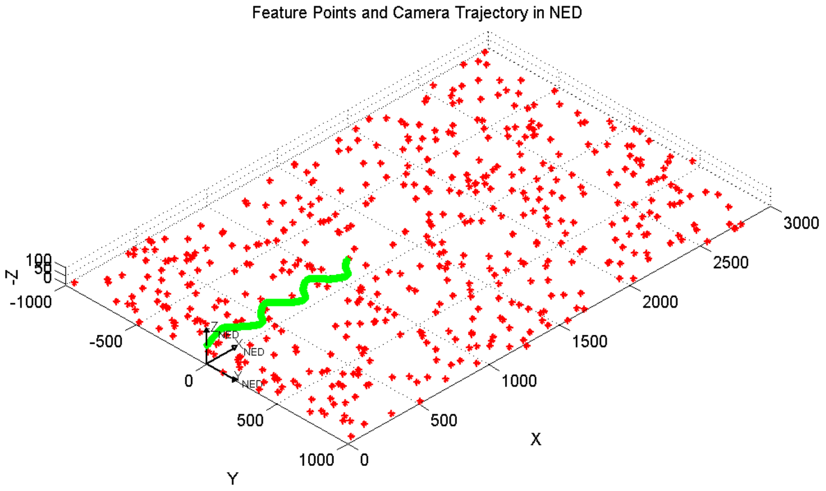


Figure 5.4 Camera trajectory and feature points in NED frame

The camera can see only feature points which are closer than 800m. This way the dense feature point cloud can be avoided on the images near the horizon level. This is important, because in the real images feature points near the horizon cannot be extracted because the blurring effect of the distant objects.

For the camera projection the calibration matrix of one of our miniature camera is used. The calibration was obtained using the Camera Calibration Toolbox in MATLAB [87]. The resolution is 752×480 pixel and the FOV is ~63°×~43°. Based on this calibration matrix 5 virtual cameras are generated with the same FOV and different resolution, that is with different CPAR as shown in Table 1.

DOI:10.15774/PPKE.ITK.2014.008

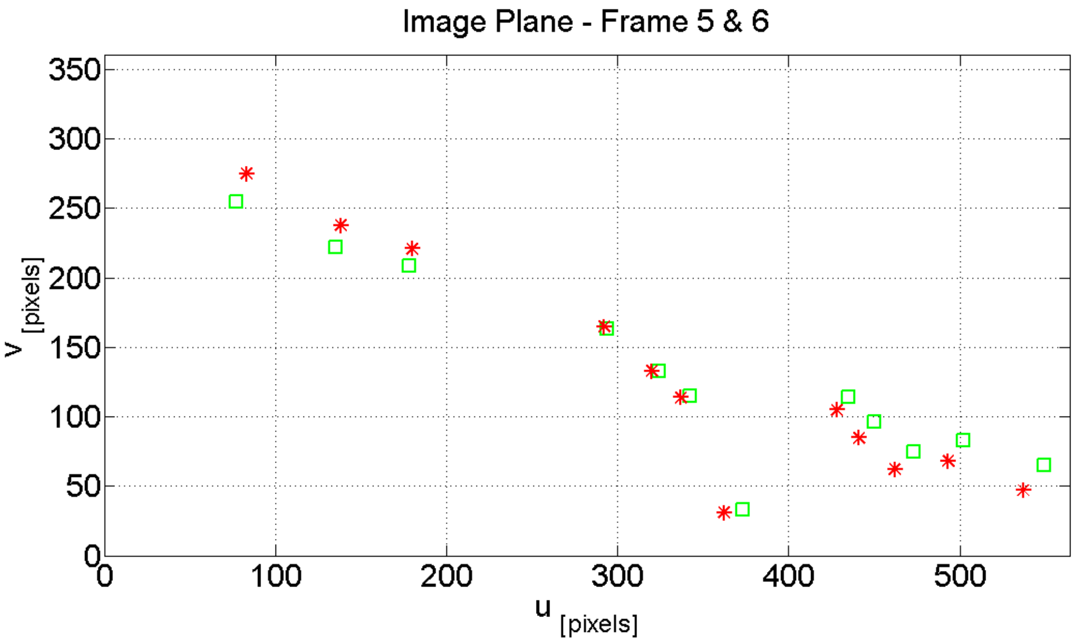


Figure 5.5 Feature points of two consecutive frames on the image plane; with green squares feature points of frame 5 and with red stars the feature points for frame 6; the camera resolution is 752×480

Resolution [px]	564×360	752×480	1017×649	1280×817	1540×960	1692×1080
CPAR [°/px]	0.12	0.093	0.068	0.055	0.046	0.041

Table 1 Resolution and CPAR of cameras

The simulations are run with different sampling frequencies. As in our test bed, the camera is running at its maximum with 56Hz. In the simulation this is approximated with 50Hz base sampling frequency that is with 20ms sampling time. Due to the processing steps or if we change the camera for another with bigger resolution, the frame rate can be dropped. The effect of the sampling frequency that is the effect of the translation on the different algorithms, is investigated in ten steps from 20 ms sampling time (50Hz) to 200 ms (5Hz).

Standard implementations of the aforementioned algorithms are used. The eight point algorithm and the MLESAC is implemented in the EGT toolbox and the implementation of the five point algorithm is from its authors' website [88]. The homography algorithm was implemented in house according to [72].

5.2.2 Error measures

In each and every step the direction cosine matrix (DCM) between the two frames is extracted which describes the rotation from one camera orientation to another. Based on this DCM the Euler angles are calculated (with an algorithm from [89]) and these are compared to the ground truth. To characterize the performance of each algorithm the absolute error of the three Euler angles are used.

(5.18)

where α_i is the ground truth angle for the i$^{\text{th}}$ frame (roll, pitch or yaw) and α_i^{calc} is the calculated angle. Additionally, for each run also the mean, the median and the corrected standard deviation of the absolute error are calculated.

5.2.3 Homography algorithm correction

To handle that the homography neglects the translation a simple correction algorithm is introduced based on the sampling time, the measured velocity and the altitude. Most of the time the error introduced by the translation has a bigger effect on the pitch and it has a smaller effect on the yaw angle, but the error is distributed proportionally to the roll angle. Thus the correction term is as follows:

$$pitch_{\text{correction}} = \frac{\cos(roll)+\sin(roll)}{\cos(roll)} \cdot \mathrm{f}(\tau, alt, \overline{\boldsymbol{v}}) \qquad (5.19)$$

$$yaw_{\text{correction}} = \frac{\cos(roll)-\sin(roll)}{\cos(roll)} \cdot \mathrm{f}(\tau, alt, \overline{\boldsymbol{v}}) \qquad (5.20)$$

where the correction terms are added to the calculated angle values and $\mathrm{f}(\tau, alt, \overline{\boldsymbol{v}})$ is an empirical function based on the linear interpolation of the measured error term for different τ (sample time), altitude and velocity values.

DOI:10.15774/PPKE.ITK.2014.008

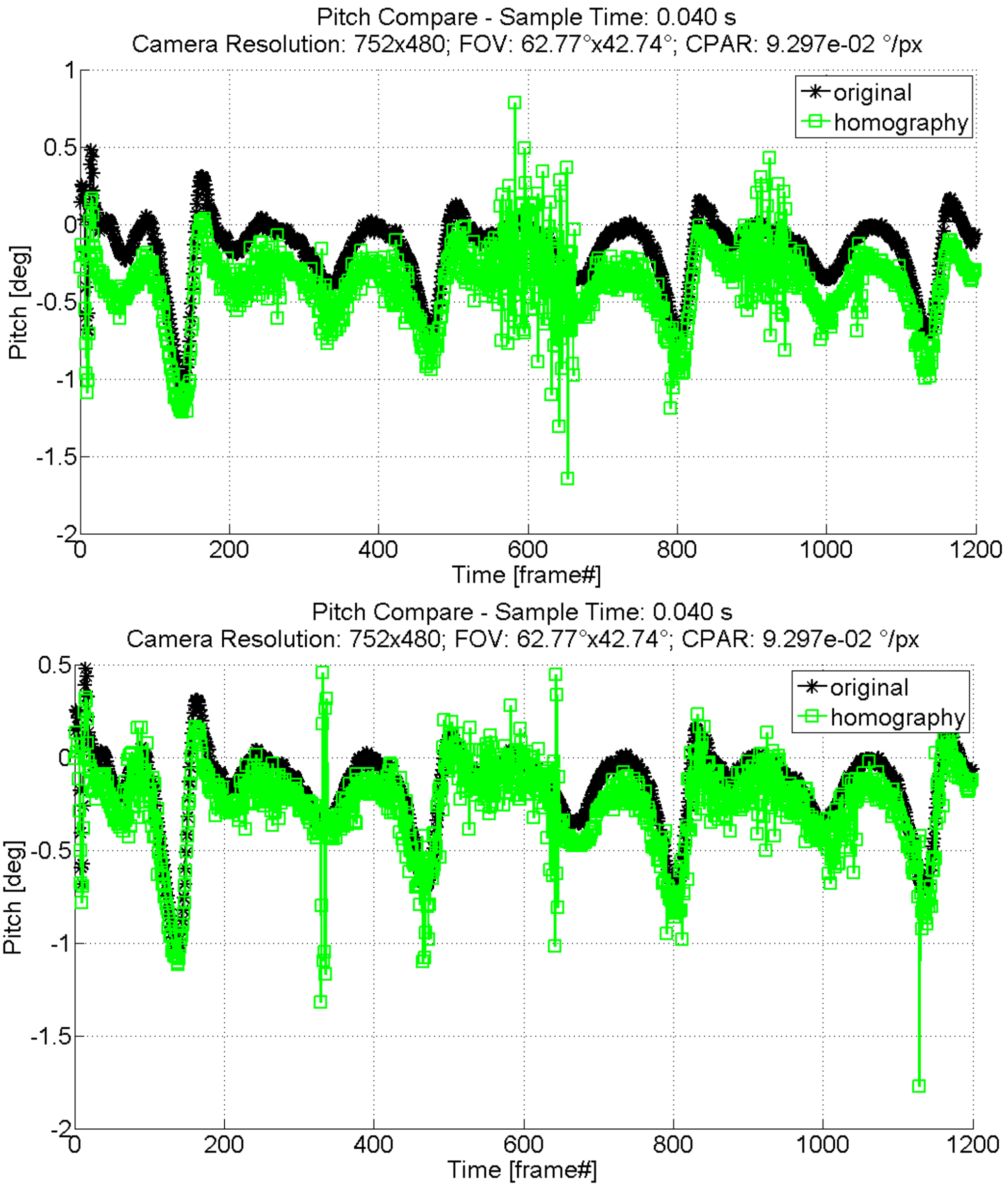


Figure 5.6 Pitch compare for homography on sinusoidal path; with black stars the ground truth, with green squares the homography results; top without correction, bottom with correction

As an example, in Figure 5.6 the correction of the pitch angle is shown. On the upper part, the pitch values are compared to the original values without correction and on the lower part with correction. As it can be seen in Figure 5.7 the error is almost twice without the correction. In this case the original camera matrix is used and the sample time is 40 ms.

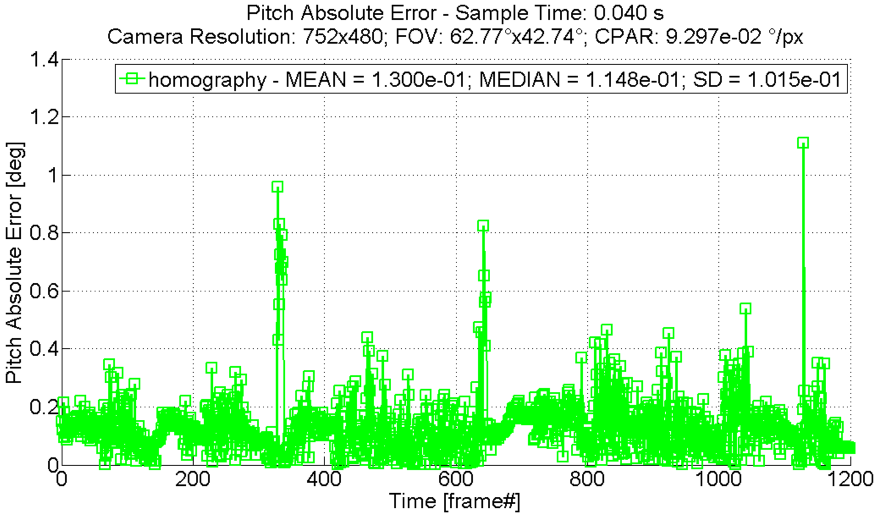
DOI:10.15774/PPKE.ITK.2014.008

5.3 Results of the Error Analysis

In this section the results of the error analysis of image processing are introduced. The pose estimation algorithms introduced in the previous section are analysed in a realistic simulation environment. The algorithms are tested with different image resolutions and sampling time. This way the tendencies can be pointed out for each algorithm as well as the performance of these algorithms can be compared.

5.3.1 Results with absolute feature point precision

First, tests with absolute feature point precision are run. In this case the best achievable results are obtained because there is practically no spatial discretization, the effect of the temporal resolution change can be investigated independently.



y

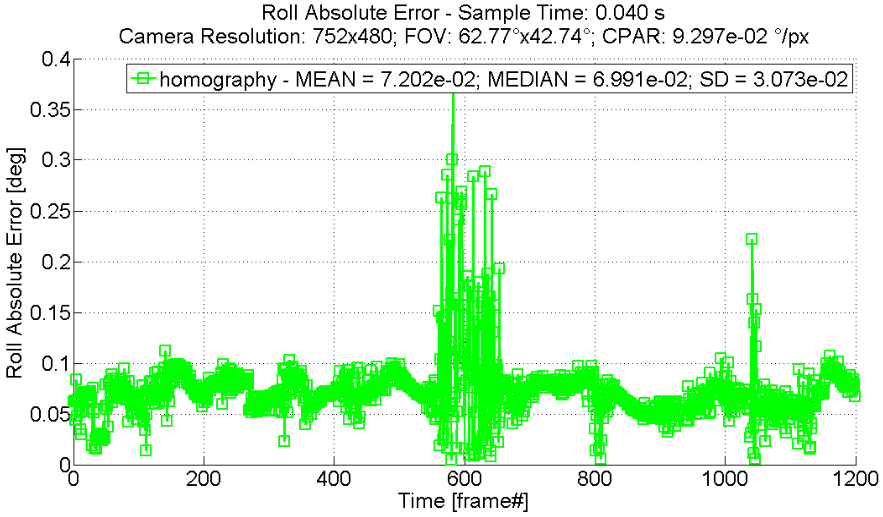


Figure 5.7 Pitch absolute error for homography on sinusoidal path;
top without correction, bottom with correction

DOI:10.15774/PPKE.ITK.2014.008

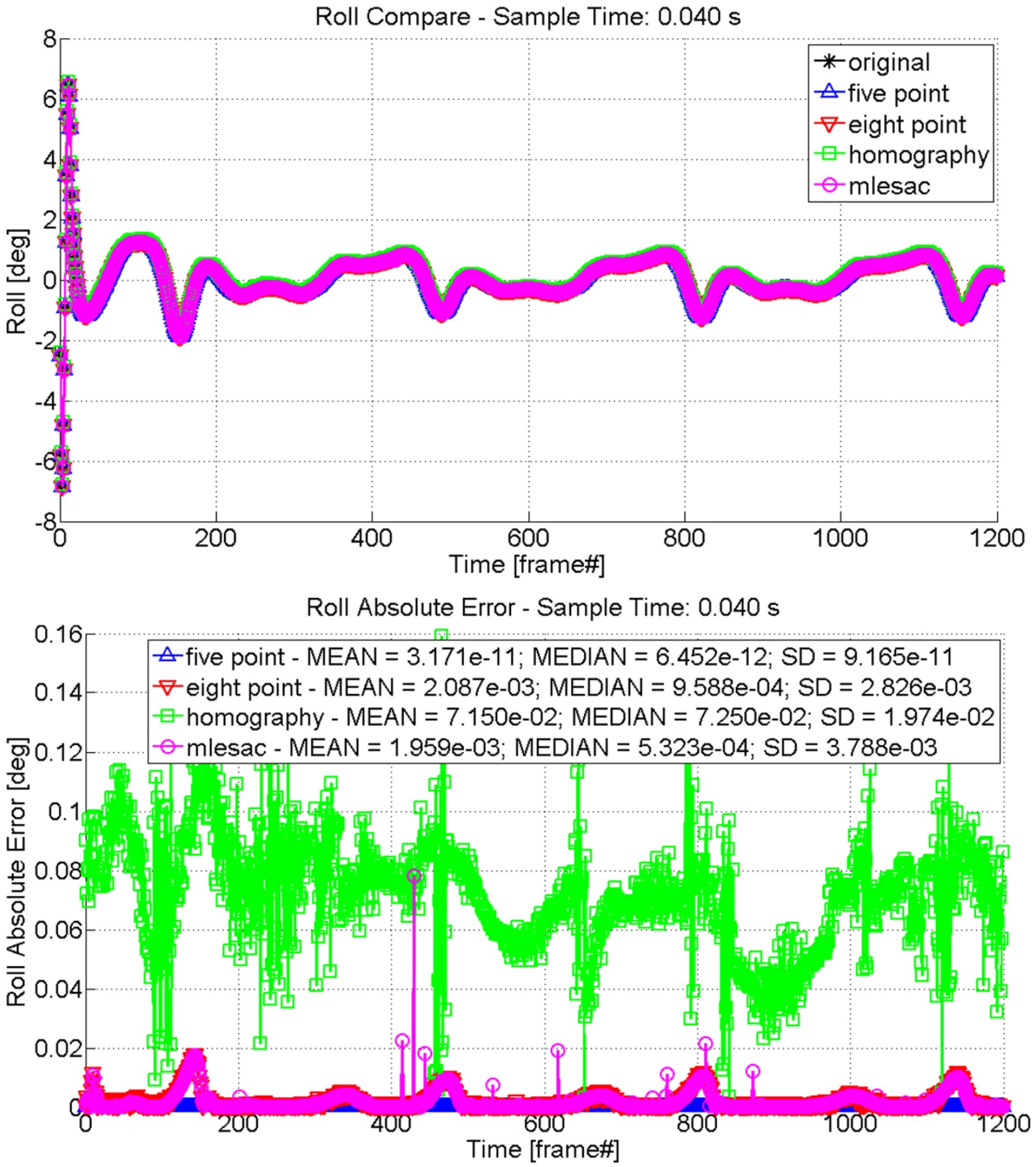


Figure 5.8 Compare of the four different algorithm with absolute feature point precision on sinusoidal; top the roll angle, bottom the error of the roll angle; with black star the original, with blue triangle the five point, with red triangle the eight point, with green square the homography and with magenta circle the MLESAC results

As shown in Figure 5.8, without any feature point coordinate error the five point algorithm is the best. The error of the five point algorithm is close to the numerical precision of the calculations. The errors of other two epipolar geometry based solutions are also at least one order of magnitude smaller than the 1 pixel angular resolution. And the homography has got an error that remains below 1 pixel.

DOI:10.15774/PPKE.ITK.2014.008

The effect of the translation is shown in the next figure with the pitch angle, which is most affected. Theoretically due to the bigger baseline separation bigger translation between the two frames could be advantageous for the three algorithms which are based on the epipolar constraint (5 point, 8 point and MLESAC). It can be seen in the figure practically this is not true, the error is bigger as the step is bigger in between the frames except for the five point algorithm in some situations. One possible explanation is that the number of the feature points which can be seen in both frames is reduced and the feature points are more drifted to the side of the image.

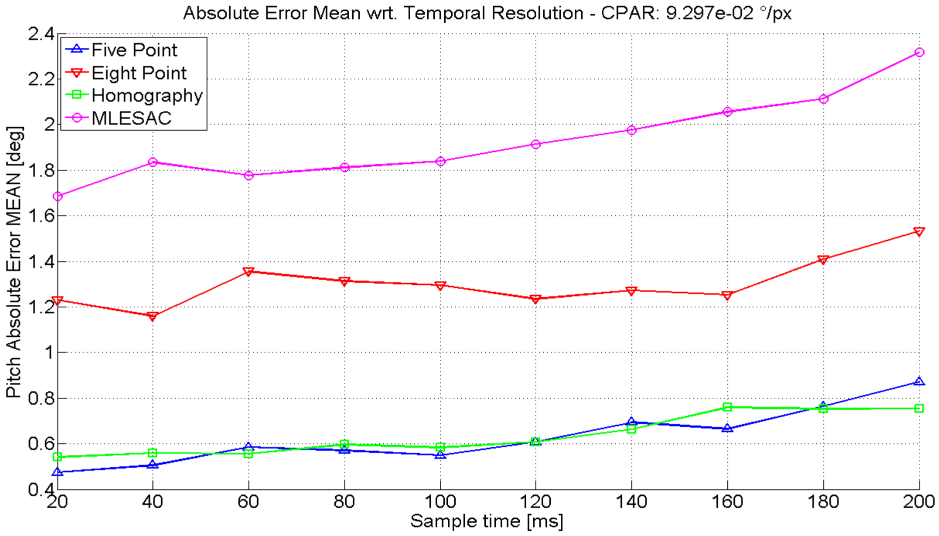


Figure 5.9 Effect of the translation through the sample time change on the pitch angle error; on sinusoidal; the pitch angle is most affected by the translation effect

5.3.2 Results with sub pixel precision

As mentioned before, the sub pixel feature point extraction is simulated by random, normal distribution noise (0 mean and 0.5 pixel standard deviation) on absolute precise feature point coordinates.

Surprisingly, the five point algorithm cannot benefit from the subpixel resolution (Figure 5.10). The eight point algorithm and the MLESAC have lower mean error values, but the median of the error of the five point algorithm is closer, which shows that the problem might be caused by specific feature point and noise arrangement. The effect of the temporal resolution change is similar to the previous case and the standard deviation shows similar features (Figure 5.11).

DOI:10.15774/PPKE.ITK.2014.008

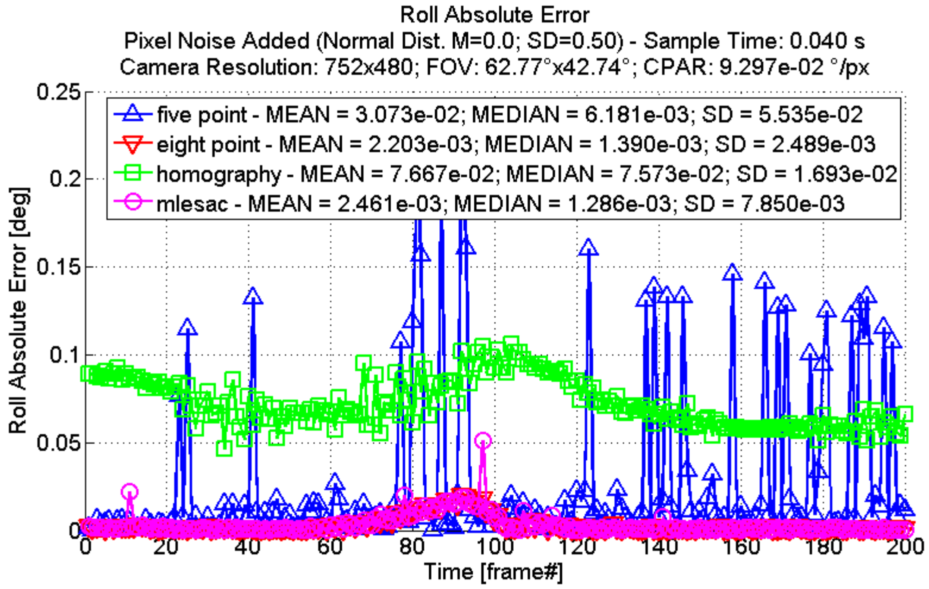


Figure 5.10 Roll error with subpixel resolution on sinusoidal; the five point algorithm performance is worse than expected

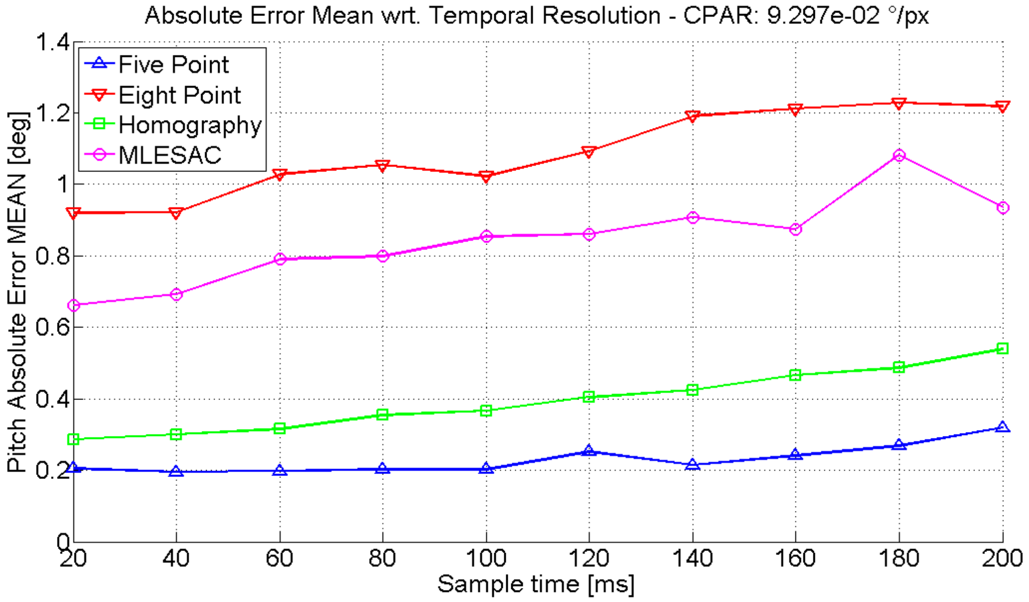


Figure 5.11 The mean error with low resolution on the pitch angle on sinusoidal

5.3.3 Results with pixelized coordinates

In this case the performance of the algorithms changed again. The best performing algorithm is the five point, but most of the time the homography can keep up with its performance (see Figure 5.12).

DOI:10.15774/PPKE.ITK.2014.008

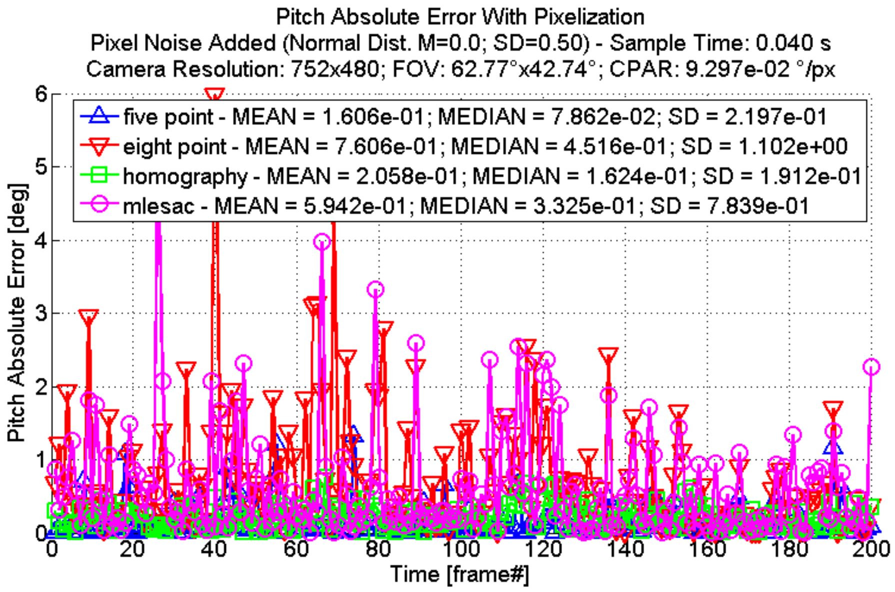


Figure 5.12 Pitch error with pixelization on sinusoidal path;
the homography is almost as good as the five point algorithm

This is important because the computational need of the homography algorithm is much less than the others. The pixelization has got a smaller effect on the homography algorithm than on the others. An extreme example is the roll mean error of homography which is almost independent of the CPAR (see Figure 5.13).

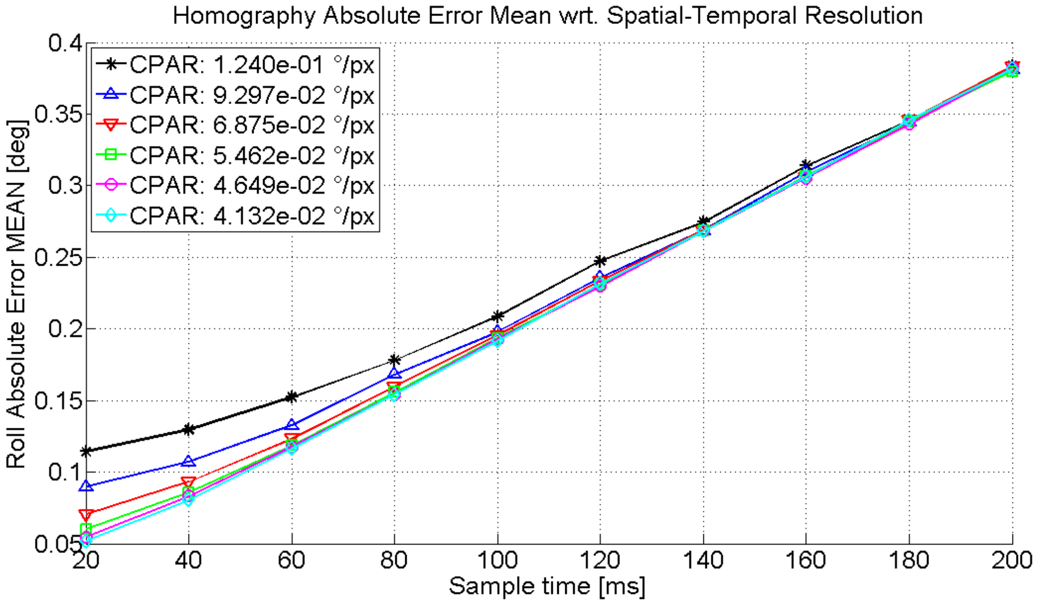


Figure 5.13 Roll error mean with pixelization on sinusoidal;
the roll error mean of the homography is almost independent of the camera resolution

DOI:10.15774/PPKE.ITK.2014.008

5.3.4 Results with pixelized coordinates and noise

The noise added to the pixelized coordinates causes only a slightly higher error level compared to the pixelization. The results here are very similar to the results in the previous section. For example the yaw error change of the homography can be seen in Table 2. Furthermore, most of the time the noise on the images can be filtered out effectively.

	mean	median	sd
Absolute precision	$4.422 \cdot 10^{-2}$	$3.119 \cdot 10^{-2}$	$4.670 \cdot 10^{-2}$
Subpixel	$4.609 \cdot 10^{-2}$	$3.278 \cdot 10^{-2}$	$5.065 \cdot 10^{-2}$
Pixelized	$6.036 \cdot 10^{-2}$	$3.845 \cdot 10^{-2}$	$6.924 \cdot 10^{-2}$
Pixelized & noise	$7.002 \cdot 10^{-2}$	$4.051 \cdot 10^{-2}$	$9.379 \cdot 10^{-2}$

Table 2 Yaw error of homography changing with different feature point precision for the CPAR=0.055°/px camera

5.4 Conclusion

In this chapter the error analysis of four image processing algorithms targeting the reconstruction of camera orientation change is introduced. It is shown how the change of the spatial or temporal resolution as well as random noise affects these algorithms. It can be stated that the homography algorithm can be used in those situations where the computational power is restricted. If the precision is important than either the five point algorithm and the homography can be used keeping in mind the effect of translation and the pixelization.

Chapter 6

Summary of New Scientific Results

1 Thesis: *Development of a relative direction angle estimation algorithm for visual sense and avoid system for autonomous unmanned aerial systems: I have introduced a new algorithm for relative direction angle estimation and shown the reachable accuracy in various situations. The algorithm is based on the assumption that the two approaching aircrafts are on a straight path and we have calibrated camera. I have also shown a simple algorithm for the extraction of the aircraft's wingtip points on the images. The accuracy of the relative direction angle is measured in pure simulation, on rendered frames and on recorded videos as well. Furthermore, the accuracy of the algorithm on wingtip coordinates marked by a human expert and extracted by the algorithm is shown.*

1.1 I have introduced a new algorithm for relative direction angle estimation for autonomous UAV visual SAA systems in the case when the two approaching aircrafts follow a straight path. I have shown that the accuracy of the algorithm in pure simulations when there is no noise or rounding to coordinates added is comparable with the numerical precision.

The relative direction angle, α can be calculated from the following formula:

$$\cos\alpha = \frac{\langle \bar{\boldsymbol{p}}' - \bar{\boldsymbol{q}}\ ;\ \bar{\boldsymbol{r}} - \bar{\boldsymbol{s}} \rangle}{\|\bar{\boldsymbol{p}}' - \bar{\boldsymbol{q}}\|\,\|\bar{\boldsymbol{r}} - \bar{\boldsymbol{s}}\|} \tag{6.1}$$

where p_{p3} and p_4 are measured on the image plane and P_1 and P_2 are estimated based on the camera matrix, and the assumptions made on the two aircraft's path. In this model the instances rotated by 180° are equal and the $\alpha = \cos^{-1} X$ function gives good solution in $\alpha = [0°; 180°]$ range.

DOI:10.15774/PPKE.ITK.2014.008

The relative angle α should be in the $[-90°; 90°]$ range, so it is transformed according to the following rules. If $\alpha > 90°$, then $\alpha = 180° - \alpha$, if $\alpha < -90°$, then $\alpha = -180° - \alpha$. With these calculations the expected results are obtained consistently.

If the intruder is on the xy horizontal plane, p_{p3} equals p_4 and the α angle cannot be estimated with this algorithm. The altitude of our UAV can be easily changed with acceleration or deceleration.

With pinhole camera model, the given centroid point of the intruder is projected back from image plane to space to several distances. The wingspan of the intruder is 11m (36 ft. 1 in), which is the wingspan of Cessna 172, a typical light aircraft that shares the airspace with our UAV. Thus the wing is represented by an 11m line segment and is rotated in the previously calculated point. The field of view and resolution of the camera and the distance along x axis is required for the calculation. The fuselage of the aircraft is neglected. With these calculations the lower bound of the error is approximated.

1.2 I have investigated through simulations how the relative position of the intruder changes the accuracy. I have shown experimentally that the closer the intruder is to the horizontal (y) axis the bigger the error of the α. And similarly the bigger the distance along the x axis the smaller the intruder is in the image, therefore the spatial discretization gives higher error value. Furthermore, the proximity to y has a greater effect on the error than in the smaller distance case.

The measurements was made with the same pinhole camera and airplane model that I used in the first case, except that the calculated points are rounded, like in the case of a real camera.

The relative distance along the x axis is 1 km (0.62 miles), the resolution is 1920x1080 pixels, the horizontal field of view is 50° and the pixels are squares. The wingspan of the intruder is 11m (36 ft. 1 in), which is the wingspan of Cessna 172. The size of intruder in the image plane is between 15 and 20 pixels, depending on the rotation angle and the position.

I have shown that the azimuth angle has technically no effect on the accuracy, but the change in the distance of the intruder to the camera and in the elevation change the accuracy. The reason is that the larger the distance the smaller the intruder in the image and the bigger the altitude difference the more you observe the wing of the intruder.

DOI:10.15774/PPKE.ITK.2014.008

1.3 I have investigated the reachable accuracy of the algorithm on wingtip coordinates extracted from rendered images and from real videos. I have shown experimentally that the accuracy can be close to the theoretical value with wingtip points selected by a human expert and extracted with a simple algorithm on rendered images. I have also shown experimentally that on real videos with a simple time average the noise introduced by the wind can be filtered out.

In our simulation environment pictures are taken and the wingtip pixel coordinates are selected by a human expert. With pinhole camera model, the given centroid point of the intruder is projected back from image plane to space to several distances and in every position it is rotated by specific angles in the xy plane. The resolution is 1920x1080 pixels and the horizontal field of view is 50° and the pixels are squares. The measurements have shown that with good wingtip coordinates in realistic situation the error can be close to the theoretical minimum.

The wingtip points were also extracted with a simple algorithm, which determines the wingtip coordinates from the segmented images. The extreme of y and z coordinates are used in appropriate order to get the coordinates. In this case when the intruder had been rotated with 80° and with -80° angles, the error of the estimation is bigger, because the simple algorithm could not distinguish between the pixels of the wing and the pixels of the tail. In contrast, in the mid-range the performance of this really simple algorithm is almost the same as the performance of the human expert (close to the theoretical limit).

DOI:10.15774/PPKE.ITK.2014.008

2 Thesis: *Error analysis of the 4 chosen camera pose estimation algorithms in the case of UAV SAA application for the rotation calculation: I have chosen four feature point based relative pose estimation algorithm. A homography based solution as a basic algorithm with small computational need but with less accuracy. The eight point algorithm, as standard algorithm in epipolar geometry. The five point algorithm, as one of the state of the art algorithms with higher computational need, but with promising stability over the various scenes. Finally, MLESAC, as an iterative, stochastic solution. The aim of the investigation is to show the strengths and weaknesses of these algorithms in the aircraft attitude estimation task.*

2.1 I have investigated the performance of the four chosen algorithms in simulations using two different real flight paths and synthetized images with randomly placed feature points and taken into account the model of the camera used on board with different resolution. I have experimentally shown that without any feature point coordinate error the five point algorithm is the best. The error of the five point algorithm is close to the numerical precision of the calculations. The errors of other two epipolar geometry based solutions are also at least one order of magnitude smaller than the 1 pixel angular resolution. And the homography has got an error that remains below 1 pixel.

For the tests 350 feature points are placed randomly with uniform distribution in a right prism which is 2000m wide, 3000m long and 30m tall. The point coordinates are between -1000 and 1000 in the Y direction and from 0 to 3000 in the X direction. The maximum altitude of the points is 23 m and the Z coordinate starts from 3 m beyond the ground level to simulate small holes.

For the camera projection the calibration matrix of one of our miniature camera is used. The internal calibration matrix is scaled in order to simulate cameras with different resolutions.

First, tests with absolute feature point precision are run. In this case the best achievable results are obtained because there is practically no spatial discretization, the effect of the temporal resolution change can be investigated independently.

2.2 I have investigated the effect of the translation on the performance of the four chosen algorithm. I have experimentally shown that the error is bigger as the time step is bigger in between the frames except for the five point algorithm in some situations.

I have shown the results of the pitch angle, which is most affected. Theoretically due to the bigger baseline separation bigger translation between the two frames could be advantageous for the three algorithms which are based on the epipolar constraint (5 point, 8 point and MLESAC). It can be seen in the figure practically this is not true, the error is bigger as the step is bigger in between the frames except for the five point algorithm in some situations. One possible explanation is that the number of the feature points which can be seen in both frames is reduced and the feature points are more drifted to the side of the image. On the other hand, the integral error altogether is smaller for the whole path.

2.3 I have investigated the possibility of the use of feature extraction algorithms with subpixel capability with the four algorithms. I have experimentally shown that except the five point algorithm, the pose estimation can benefit from the subpixel feature point calculation.

The sub pixel feature point extraction is simulated by random, normal distribution noise (0 mean and 0.5 pixel standard deviation) on absolute precise feature point coordinates. Surprisingly, the five point algorithm cannot benefit from the subpixel resolution. The eight point algorithm and the MLESAC have lower mean error values.

	Five point	Eight point	Homography	MLESAC
Absolute precision	$3.171 \cdot 10^{-11}$	$2.087 \cdot 10^{-3}$	$5.065 \cdot 10^{-2}$	$1.323\ 10^{-3}$
Subpixel	$1.234 \cdot 10^{-1}$	$1.080 \cdot 10^{-2}$	$7.150 \cdot 10^{-2}$	$1.959\ 10^{-3}$
Pixelized	$9.371 \cdot 10^{-2}$	$5.476 \cdot 10^{-1}$	$1.169 \cdot 10^{-1}$	$3.240\ 10^{-1}$

Table 3 Roll error of the four algorithms changing with different feature point precision for the CPAR=0.093°/px camera

2.4 I have investigated the performance of the algorithms in more general case, when the feature point coordinates are rounded, or are rounded and contain noise as well. I have experimentally shown that the five point algorithm performs the best with mean error value around 1 pixel. I have experimentally shown that the homography algorithm can perform almost as good as the five point, with mean error around 1.5 pixels. The computational need of the homography algorithm is 2 orders of magnitude smaller than the computational demand of five point algorithm in the number of the multiplications. I have experimentally shown that the pixelization has got a smaller effect on the homography algorithm than on the others. It can be stated that the homography algorithm can be used in those situations where the computational power is restricted.

DOI:10.15774/PPKE.ITK.2014.008

Chapter 7

Applications

As it is stated at the introduction, the aim of the research project is to develop an autonomous UAS with SAA capability. The UAS which is developed is the main application area of the results of this dissertation. In this chapter the developed system is introduced and then from section 7.2 the results with the GPS/IMU/Camera attitude estimator are shown, in which the results from Chapter 5 are used.

7.1 Mid-size fixed wing UAS

In this section the hardware components of our system are introduced. The system consists of two main components, the IMU (including GPS) and a visual sensor-computer system. As a data recorder an SSD drive was used thus the measured data can be evaluated not only on board but offline as well after the flight.

7.1.1 The aircraft

The airframe used in the flight tests is an upper wing, two engine foam aircraft with 1.85m length, 3.2m wingspan and about 10kg loaded weight.

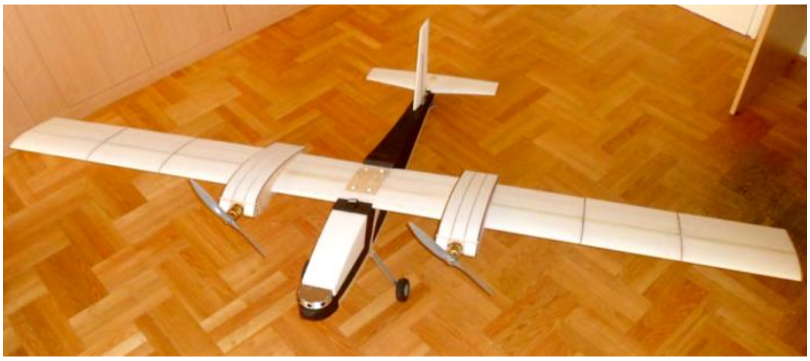


Figure 7.1 The aircraft called Orca, the five camera system can be seen on the nose of the fuselage

DOI:10.15774/PPKE.ITK.2014.008

7.1.2 Inertial Measurement Unit (IMU)

The integrated inertial and satellite navigation system is developed in house (MTA SZTAKI) to comply with the overall system architecture requirements. The sensor suite includes the conventional set of sensors required for autonomous outdoor waypoint navigation, with 3 axis gyro, 3 axis accelerometer, 3 axis magnetic sensor, static and differential pressure sensors, GPS unit with raw data capability (ublox LEA-6T), as shown in Figure 7.2 The aim is to have a compact module, with small size and consumption, but accurate enough for the control of aerial vehicles.

The dimensions of the unit are 57 × 53mm, the same footprint as the flight control computer, with a weight of 20g without the GPS antenna. The navigation unit is built of digital MEMS sensors with digital interfaces; hence the analogue interfaces are omitted from the design.

The control unit of the navigation system is a 32 bit AVR microcontroller. It contains a hardware floating point unit, which is able to perform preliminary calculations, such as sensor calibration. This microcontroller communicates with sensors, with GPS module and with external devices through digital interfaces. To suit the needs of the safety critical architecture the unit is directly connected to the flight control computer via CAN communication bus.

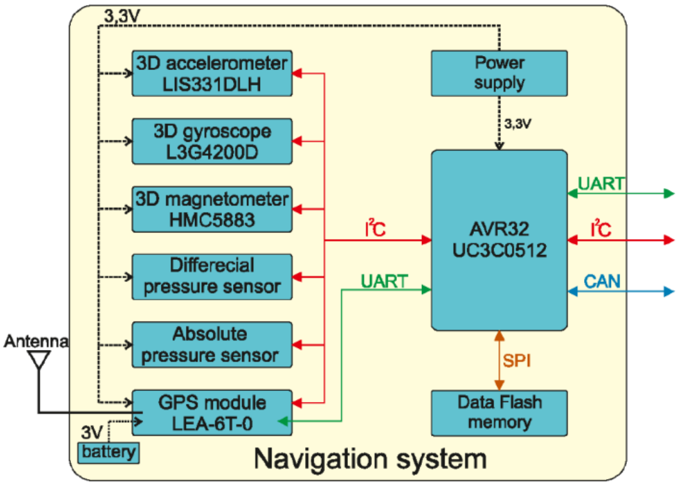


Figure 7.2 Block diagram of the integrated inertial and satellite navigation system

7.1.3 Visual sensor-processor system

The block diagram of the remote airplane detector system is shown in Figure 7.3 It contains the cameras, an interface board, an FPGA board, and a solid state drive. 5 pieces of WVGA (752×480 pixel) grayscale global shutter cameras are selected as the image sources. The angle between the neighbouring cameras is 47.5°. The joint image covers ~$265^0\times45^0$ field of view with 6 ° of overlap. The entire resolution is ~3000×480. [12]

The advantage of this multiple camera system is that, the distortion of the image is negligible compared to a fish-eye objective. Moreover, a high resolution camera with a high resolution, ultra large view angle precision lens would cost tens of thousands USD, and would weight kilos while this system is cheaper and lighter as well. To be able to hold the cameras in the required position, and avoid cross camera vibration, we have designed and manufactured a solid aluminium camera holder shown in Figure 7.4

An off-the-shelf FPGA card was selected (SPARTAN-6T), which had an appropriate compact design, and could handle solid-state-disk-drive (SSD). To be able to interface the ribbon cables of the cameras to the FPGA card, we have designed and manufactured an interface board.

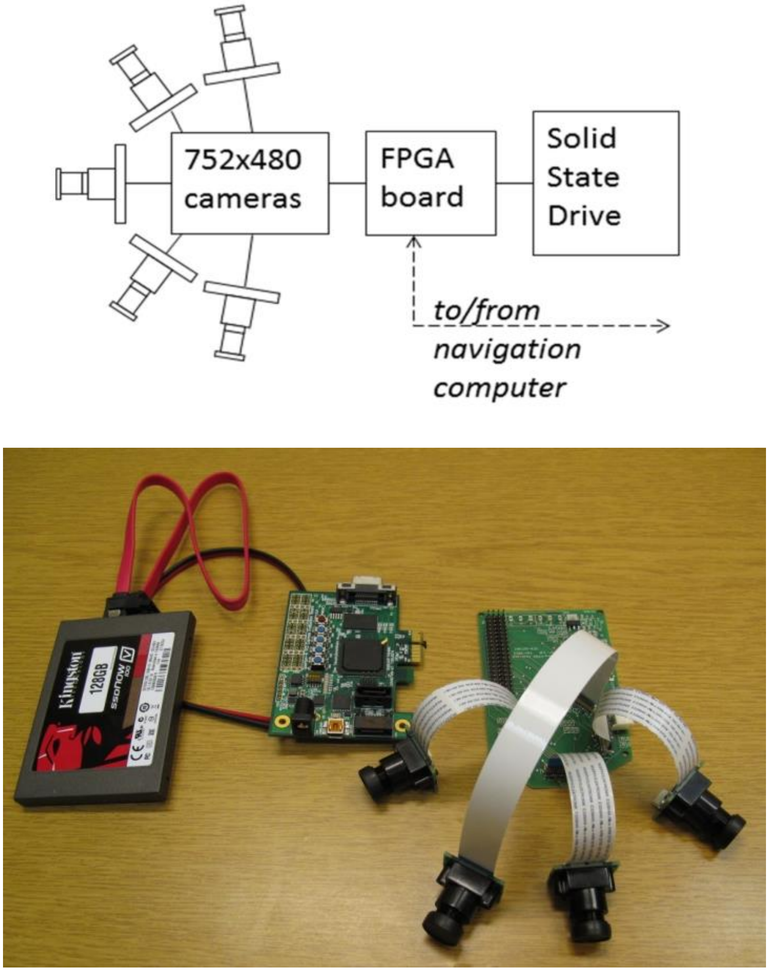


Figure 7.3 The image capturing, processing, and storing system; Diagram of the components (upper). The physical hardware components (lower) [12]

DOI:10.15774/PPKE.ITK.2014.008

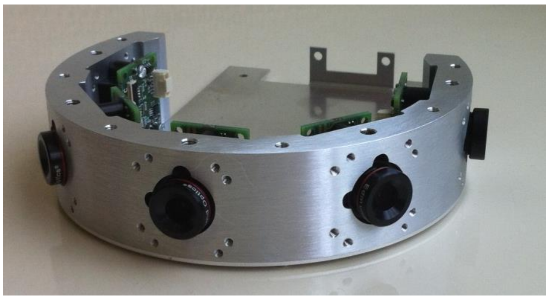


Figure 7.4 Solid aluminium camera holder for avoiding cross vibrations

The SSD is a key element of the system, which enables to record this high resolution, low distortion test image sequences from different approaching situations.

The FPGA based on-board image processing system executes several parallel tasks. Each task of the algorithm has a dedicated execution unit designed for the specific functionality of the task. Operation of the different units is synchronized by a Xilinx Microblaze soft processor core [90], [91]. The system can handle five micro cameras which are connected to the FPGA directly through parallel cables. Block diagram of the on-board image processing and motion estimation system is shown in Figure 7.5

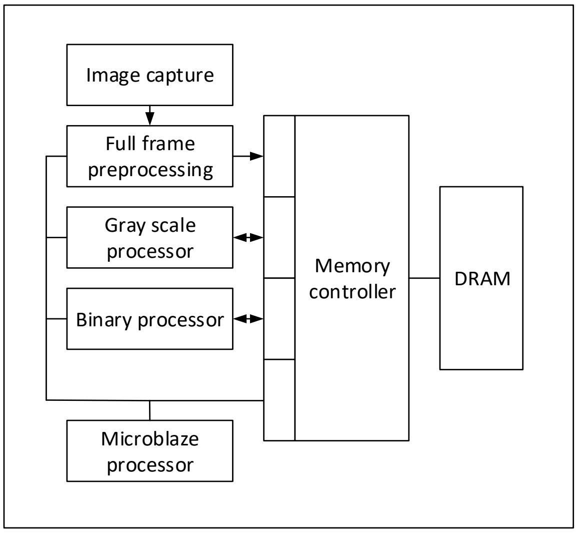


Figure 7.5 Block diagram of the image processing architecture; after the full frame preprocessing of each image the grayscale and binary operations are done in smaller regions. The operations are controlled by the Microblaze processor.

7.2 HIL simulation and measurement results

In this section the coupled GPS/IMU/Camera attitude estimator system is introduced. As a measurement example some of the datasets from the HIL simulation tests are run in this system.

The HIL simulation includes the aircraft dynamical model in MATLAB Simulink completed with the RC transmitter, and on-board microcontroller. The control inputs from the transmitter and microcontroller are sent into the PC through an RS-232 or CAN interface. The sensory system of the aircraft is emulated in Simulink, the sensor data is sent to the microcontroller again through an RS-232 or CAN interface. This way the real electronics controls the aircraft simulation.

7.3 Coupled GPS/IMU/Camera attitude estimator implementation

In this section the coupling of a GPS/IMU-based aircraft attitude estimation algorithm (from [89]) with the camera-based rotation matrix increment estimate (T_Δ) is introduced.

The original estimator is an EKF which uses the angular rate and acceleration measurements to propagate the attitude, velocity and (latitude, longitude, altitude = LLA) position dynamics of the aircraft. The Euler angles, earth relative velocity and position are predicted using system dynamic equations. In the correction step of the EKF GPS position and velocity measurements are used to calculate the prediction error and update the attitude, velocity and position accordingly. The rate gyro and accelerometer biases are also estimated.

The camera based rotation increment can be included into the measurement step as an information about the change of the direction cosine matrix (DCM). This is explained in the forthcoming part. The algorithm was implemented in MATLAB, and tested on the same data used in the previous section.

This data was generated in HIL excluding sensor noise and wind disturbance. The goal is to test the sensor fusion on exact data and so compare the performance of the different image processing algorithms in an ideal situation. From HIL, the real Euler angles are known. The attitude considers the error in the DCM (here $\overline{\overline{\mathbf{BE}}}$) instead of the error of Euler angles. The aircraft orientation in the second camera frame can now be represented in two different ways:

$$\overline{\overline{\mathbf{BE}}}' = \bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}^{\text{camera}} \text{ from the camera} \tag{7.1}$$

$$\overline{\overline{\mathbf{BE}}}' = \left(\bar{\bar{\mathbf{I}}} + \left[\overline{\boldsymbol{\delta}\mathbf{E}}\right]_\times\right)\overline{\overline{\mathbf{BE}}}^{\text{GPS/IMU}} \text{ from the GPS/IMU.} \tag{7.2}$$

Here $\overline{\overline{\mathbf{BE}}}^{\text{camera}}$ is the rotation matrix related to the first camera frame. $\overline{\overline{\mathbf{BE}}}^{\text{GPS/IMU}}$ is the rotation matrix predicted from actual IMU data. $\overline{\boldsymbol{\delta}\mathbf{E}}$ is the vector representing rotation errors and $[\overline{\boldsymbol{\delta}\mathbf{E}}]_\times$ is the skew-symmetric matrix created from it. Comparing the two equations $[\overline{\boldsymbol{\delta}\mathbf{E}}]_\times$ can be expressed:

DOI:10.15774/PPKE.ITK.2014.008

$$\bar{\bar{\Delta}}\,\overline{\overline{\mathbf{BE}}}^{\text{camera}}\left(\overline{\overline{\mathbf{BE}}}^{\text{GPS/IMU}}\right)^{\mathrm{T}}-\bar{\bar{\mathbf{I}}}=\left[\,\overline{\boldsymbol{\delta}\mathbf{E}}\,\right]_{\times}' \tag{7.3}$$

Of course, because of measurement and numerical errors $[\overline{\boldsymbol{\delta}\mathbf{E}}]_{\times}'$ will not be skew-symmetric (this is denoted by $()'$). But it can be made skew-symmetric with the following transformation:

$$\left[\,\overline{\boldsymbol{\delta}\mathbf{E}}\,\right]_{\times}=\frac{\left[\,\overline{\boldsymbol{\delta}\mathbf{E}}\,\right]_{\times}'-\left(\left[\,\overline{\boldsymbol{\delta}\mathbf{E}}\,\right]_{\times}'\right)^{\mathrm{T}}}{2} \tag{7.4}$$

From $[\overline{\boldsymbol{\delta}\mathbf{E}}]_{\times}$ the rotation error terms can be directly incorporated into the attitude estimator algorithm as measurements.

With the inclusion of camera data three working modes should be defined in the attitude estimator considering 5Hz GPS and 50Hz camera data:

1. Only GPS data, correction with GPS measurement
2. Only camera data, correction with camera measurement
3. Both GPS and camera data, correction with both of them

This means that the measurement equations of the attitude estimator EKF are changing according to the available data. In this application only the first and third modes are used, because the second mode needs some reformulation or tuning according to the simulation results.

7.4 Coupled GPS/IMU/Camera attitude estimator

Two examples are shown here. First the GPS/IMU solution and the error against the ground truth is plotted (Figure 7.6. and Figure 7.7.), and then the results of the homography and five point algorithm run with the random noise case are shown (Figure 7.8 and Figure 7.9.). In both the homography and the five point cases the sample time is minimum, that is 20ms, the CPAR is 0.093, and the sinusoidal path is used. For the five point algorithm only the errors are plotted (Figure 7.9.), because the angle comparison is very similar to the homography.

The comparison of the GPS/IMU results with the GPS/IMU/Camera solution shows that the latter has a better precision as with the inclusion of the Camera data the bias of the pitch estimation is removed.

The comparison of the homography and the five point algorithm shows that the homography is indeed less affected by the noise as it was stated in 5.3.2. The yaw angle error is less for the homography and the other two angles are at the same level. (Figure 7.10.)

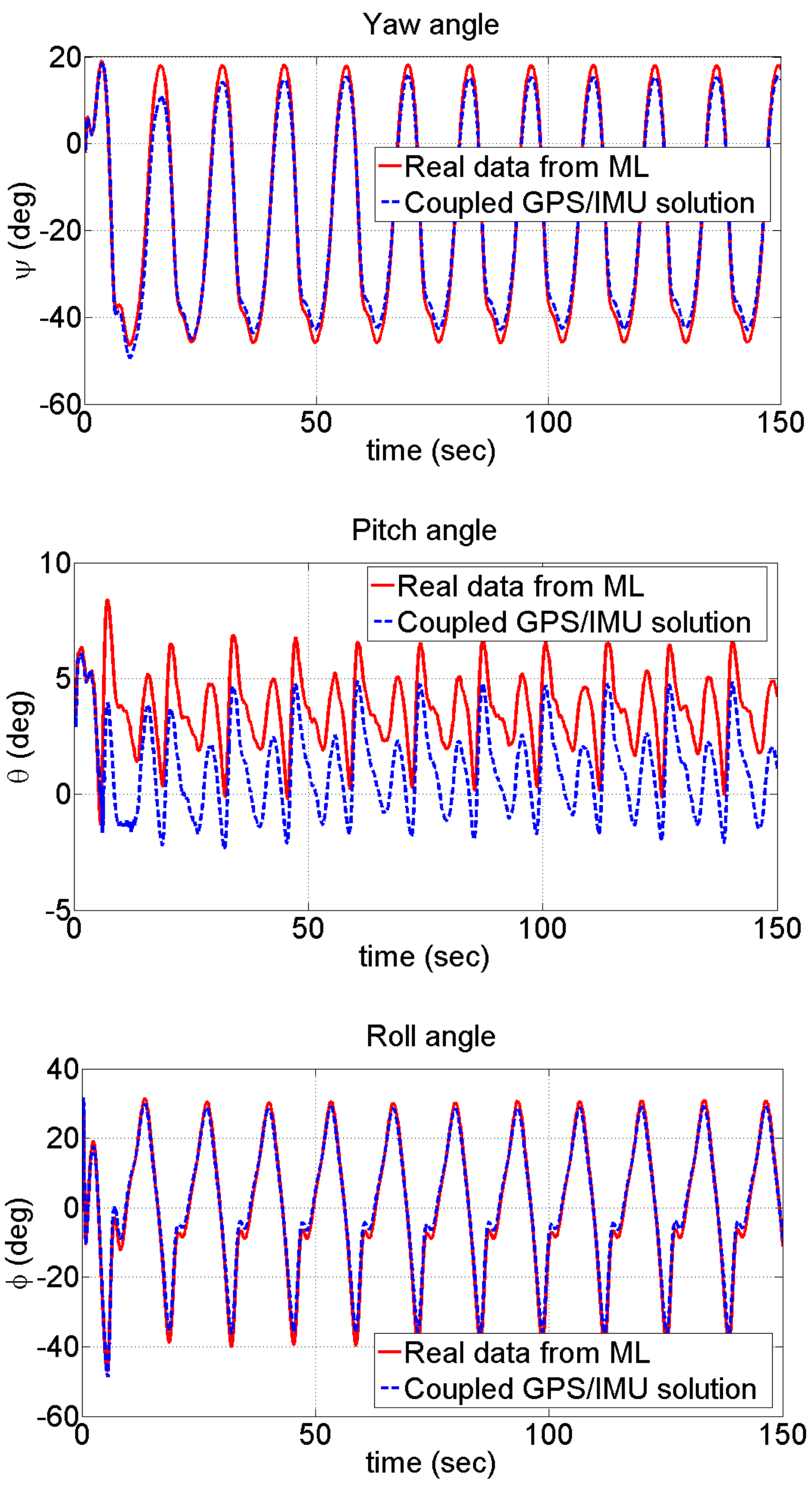


Figure 7.6 The result of the GPS/IMU fusion with respect to the ground truth; with red solid line the ground truth and with blue dashed line the result of the EKF; The bias in the pitch value can be seen in the middle figure

DOI:10.15774/PPKE.ITK.2014.008

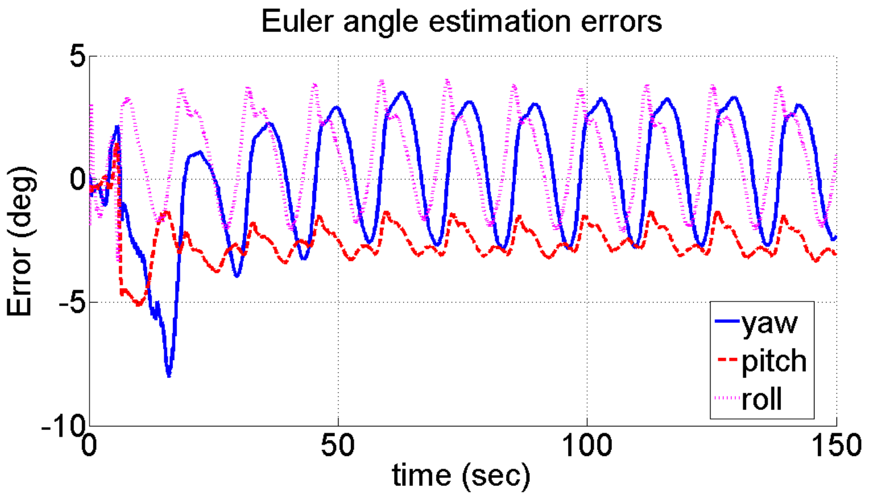


Figure 7.7 The error of the GPS/IMU fusion with respect to the ground truth

DOI:10.15774/PPKE.ITK.2014.008

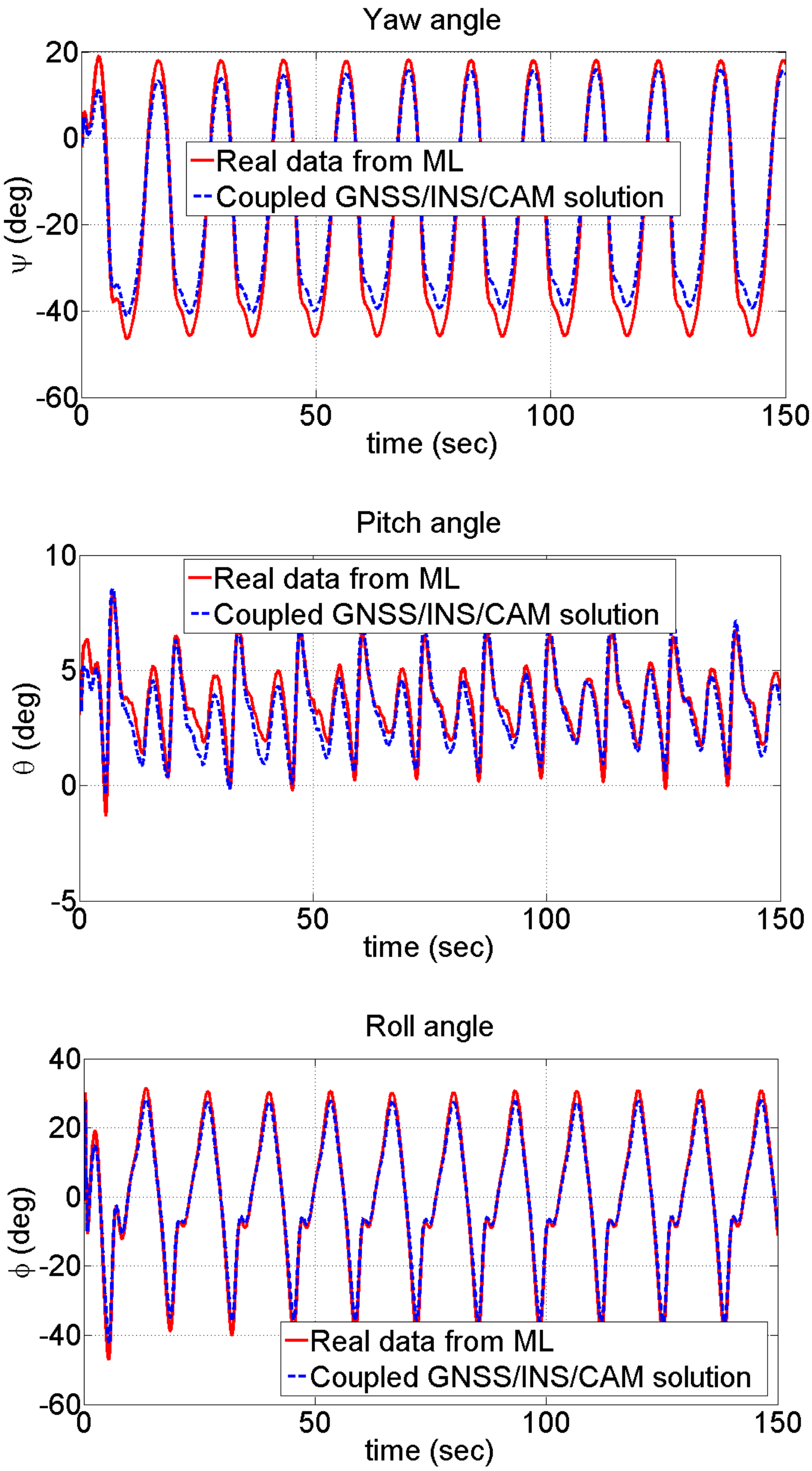


Figure 7.8 The result of the GPS/IMU/Camera fusion with the homography with respect to the ground truth; with red solid line the ground truth and with blue dashed line the result of the EKF; The pitch bias is eliminated

DOI:10.15774/PPKE.ITK.2014.008

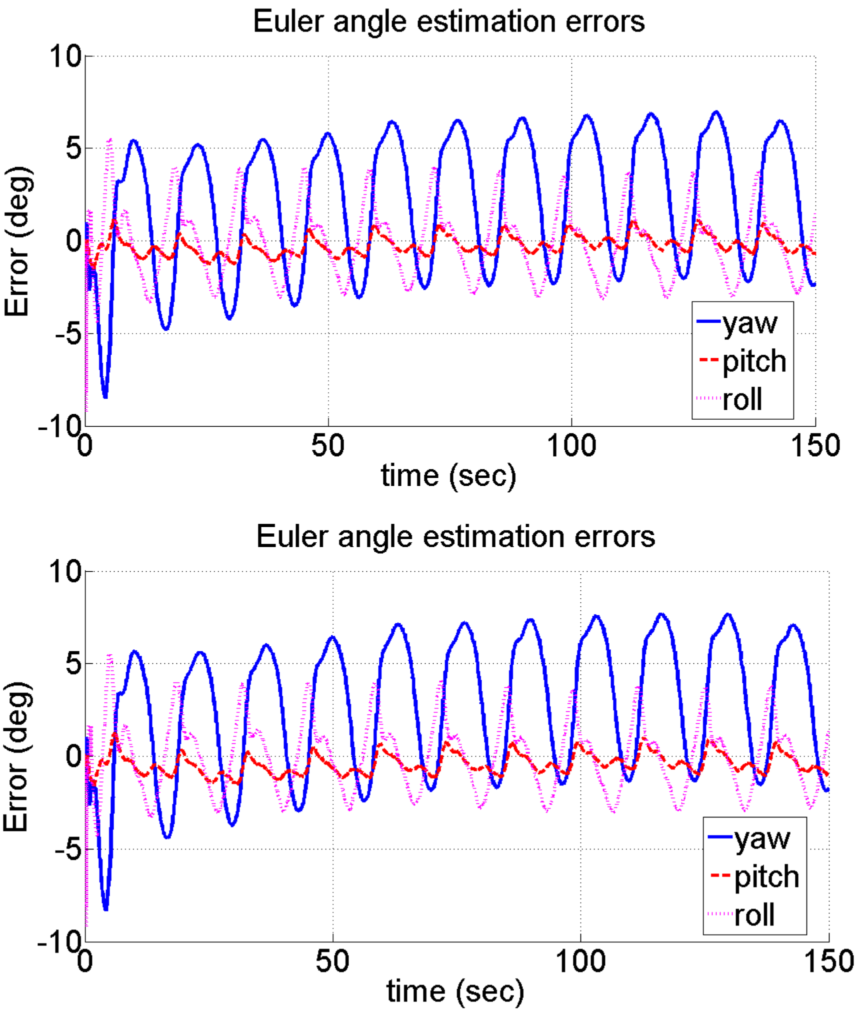


Figure 7.9 The Euler angle error of the GPS/IMU/Camera fusion with respect to the ground truth; top the results of the homography, bottom the results of the five point algorithm; the trends are similar

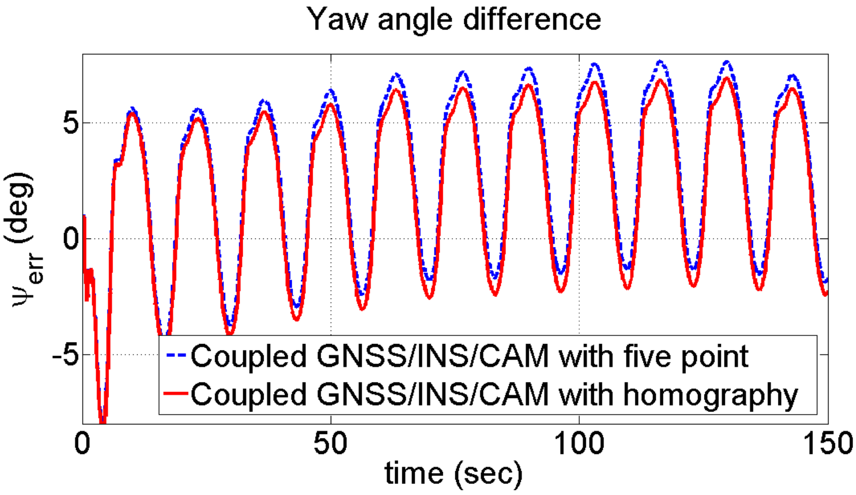


Figure 7.10 The yaw error of the GPS/IMU/Camera fusion with respect to the ground truth

DOI:10.15774/PPKE.ITK.2014.008

References

The author’s journal publications

[1] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Estimation of Relative Direction Angle of Distant, Approaching Airplane in Sense-and-Avoid,” *J. Intell. Robot. Syst.*, vol. 69, no. 1–4, pp. 407–415, Jan. 2013.

[2] A. Zarandy, M. Nemeth, Z. Nagy, A. Kiss, L. Santha, and T. Zsedrovits, “A real-time multi-camera vision system for UAV collision warning and navigation,” *J. Real-Time Image Process.*, Sep. 2014.

The author’s international conference publications

[3] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Collision avoidance for UAV using visual detection,” in *Proc. of 2011 IEEE Int. Sym. of Circuits and Systems (ISCAS)*, 2011, pp. 2173–2176.

[4] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Visual Detection and Implementation Aspects of a UAV See and Avoid System,” in *Proc. of 2011 20th European Conference on Circuit Theory and Design (ECCTD)*, 2011, pp. 472–475.

[5] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Estimation of Relative Direction Angle of Distant, Colliding Airplane in Sense-and-avoid and Tracking,” Presented at the *International Conference on Unmanned Aircraft Systems*, Philadelphia, Pennsylvania, 2012.

[6] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Azimuth estimation of distant, approaching airplane in See-and-avoid Systems,” in *Proc. of 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, Turin, Italy, 2012, pp. 1–6.

[7] T. Zsedrovits, P. Bauer, A. Zarandy, B. Vanek, J. Bokor, and T. Roska, “Towards Real-Time Visual and IMU Data Fusion,” Presented at the *AIAA Guidance, Navigation, and Control Conference*, Reston, Virginia, 2014.

DOI:10.15774/PPKE.ITK.2014.008

[8] T. Zsedrovits, P. Bauer, A. Zarandy, B. Vanek, J. Bokor, and T. Roska, “Error Analysis of Algorithms for Camera Rotation Calculation in GPS/IMU/Camera Fusion for UAV Sense and Avoid Systems,” Presented at the *International Conference on Unmanned Aircraft Systems*, Orlando, Florida, 2014.

The author’s other journal and conference publicaitons

[9] B. Vanek, T. Peni, J. Bokor, T. Zsedrovits, A. Zarandy, and T. Roska, “Performance analysis of a vision only Sense and Avoid system for small UAVs,” Presented at the *AIAA Guidance, Navigation, and Control Conference*, Reston, Virigina, 2011.

[10] B. Vanek, T. Péni, Á. Zarándy, J. Bokor, T. Zsedrovits, and T. Roska, “Performance Characteristics of a Complete Vision Only Sense and Avoid System,” Presented at the *AIAA Guidance, Navigation, and Control Conference*, 2012.

[11] Z. Nagy, A. Kiss, A. Zarandy, T. Zsedrovits, B. Vanek, T. Peni, J. Bokor, and T. Roska, “Volume and power optimized high-performance system for UAV collision avoidance,” in *Proc. of the 2012 IEEE Int. Symp. on Circuits and Systems*, 2012, pp. 189–192.

[12] A. Zarandy, T. Zsedrovits, Z. Nagy, A. Kiss, and T. Roska, “Visual sense-and-avoid system for UAVs,” in *Proc. of 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, 2012, pp. 1–5.

[13] A. Zarandy, T. Zsedrovits, Z. Nagy, A. Kiss, P. Szolgay, and T. Roska, “Cellular processor array based UAV safety system,” in *Proc. of 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, 2012, pp. 1–2.

A. Zarandy, Z. Nagy, B. Vanek, and T. Zsedrovits, “A five-camera vision system for UAV visual attitude calculation and collision warning,” *Comput. Vis. Syst. Lect. Notes Comput. Sci.*, vol. 7963, pp. 11–20, 2013.Other publications cited in the dissertation

[14] B. Vanek, P. Bauer, I. Gozse, M. Lukatsi, I. Reti, and J. Bokor, “Safety Critical Platform for Mini UAS Insertion into the Common Airspace,” Presented at the *AIAA Guidance, Navigation, and Control Conference*, 2014.

[15] T. Cox et al., “Civil UAV capability assessment,” Tech. Rep., Dec.2004.

[16] A. V. Koldaev, “Non-Military UAV Applications,” 2007.

DOI:10.15774/PPKE.ITK.2014.008

[17] P. Campoy et al., “Computer Vision Onboard UAVs for Civilian Tasks,” *J. Intell. Robot. Syst.*, vol. 54, pp. 105–135, 2008.

[18] P. Ross, “When will software have the right stuff?,” *IEEE Spectrum Magazine*, Dec-2011.

[19] S. Alderton, “UAS Help New Zealand Farmer Count Sheep and Assess Grass Quality,” *Farmers Weekly*, 2014. [Online]. Available: http://www.uasvision.com/2014/02/26/uas-help-new-zealand-farmer-count-sheep-and-assess-grass-quality/. [Accessed: 08-Apr-2014].

[20] D. Werner, “Making way for unmanned aircraft,” *Aerospace America January 2014,* no. January, pp. 28–32, 2014.

[21] B. Morton, “UAS to Help with Search & Rescue in Metro Vancouver.” [Online]. Available: http://www.uasvision.com/2014/02/11/uas-to-help-with-search-rescue-in-metro-vancouver/. [Accessed: 08-Apr-2014].

[22] S. Gorton, “UAS Firefighting Technology in New South Wales, Australia,” *Naaroma News,* 2014. [Online]. Available: http://www.uasvision.com/2014/02/11/uas-firefighting-technology-in-new-south-wales-australia/. [Accessed: 08-Apr-2014].

[23] P. Kannan, “UAE Plans to Use Delivery UAS Within a Year,” *The Vancouver Sun*, 2014. [Online]. Available: http://www.uasvision.com/2014/02/19/uae-plans-to-use-delivery-uas-within-a-year/. [Accessed: 08-Apr-2014].

[24] P. Magdirila, “Philippines TV Use UAS for News Reporting and Rescue Operations,” *Yahoo! News,* 2014. [Online]. Available: http://www.uasvision.com/2014/02/13/philippines-tv-use-uas-for-news-reporting-and-rescue-operations/. [Accessed: 23-Apr-2014].

[25] Renault, “Renault.com - Kwid concept-car,” 2013. [Online]. Available: http://www.renault.com/en/innovation/l-univers-du-design/pages/kwid-concept-car.aspx. [Accessed: 08-Apr-2014].

[26] M. T. Degarmo, “Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace,” Rep. MP 04W0000323, 2004.

[27] K. Dalamagkidis et al., “On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the National Airspace System,” *Prog. Aerosp. Sci.*, vol. 44, no. 7–8, pp. 503–519, Oct. 2008.

[28] M. D. Ebdon and J. Regan, “Sense-and-avoid requirement for remotely operated aircraft (ROA),” HQ ACC/DR-UAV SMO, White Paper, Jun. 25, 2004.

DOI:10.15774/PPKE.ITK.2014.008

[29] W.-Z. Chen et al., “Autonomous Sense and Avoid (SAA) for Unmanned Air Systems (UAS),” presented at the NATO RTO Symp. Intelligent Uninhabited Vehicle Guidance Systems, Brussels, Belgium, 2009, vol. 31.

[30] A. Ganguli and S. Avadhanam, “On the Limits of Collision Detection Performance of a Sense- and-Avoid System for Non-Cooperative Air Traffic,” in *Proc. of AIAA Infotech@Aerospace 2010*, Atlanta, 2010, no. April, pp. 1–9.

[31] G. L. Dillingham, “Unmanned Aircraft Systems: Measuring Progress and Addressing Potential Privacy Concerns Would Facilitate Integration into the National Airspace System,” U.S. Government Accountability Office, Washington, DC, Rep. GAO-12-981, Sep. 18, 2012.

[32] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton: Princeton University Press, 2012.

[33] U. S. Department of Transportation Federal Aviation Administration, “Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap,” Washington, DC, Rep. 2012-AJG-502, 2013.

[34] European RPAS Steering Group, “Roadmap for the integration of civil Remotely - Piloted Aircraft Systems into the European Aviation System,” 2013.

[35] Nemzeti Közlekedési Hatóság Légügyi Hivatala, “Konferencia a pilóta nélküli légi járművekről,” 2013. [Online]. Available: http://www.nkh.hu/Repules/UAS_konferencia/Lapok/default.aspx. [Accessed: 08-May-2014].

[36] A. Halászné dr. Tóth et al., “Esettanulmány a pilóta nélküli légijárművek jövőbeni alkalmazása tükrében,” *Repüléstudományi közlemények: on-line tudományos folyóirat*, vol. XXIV., no. 2, pp. 607–615, 2012.

[37] G. J. Harrison, “Unmanned Aircraft Systems (UAS): Manufacturing Trends,” Rep. R42938, 2013.

[38] J. Chesebro, “Unmanned Aircraft Systems (UAS),” 2011. [Online]. Available: http://www.trade.gov/static/aero_rpt_flight_plan_2011_uas.pdf. [Accessed: 08-May-2014].

[39] Teal Group, “World UAV Systems 2012: Market Profile and Forecast,” 2012.

[40] International Civil Aviation Organization, “Procedures for Air Navigation Services: Air Traffic Management,” Doc 4444 ATM/501, Nov. 22, 2007.

DOI:10.15774/PPKE.ITK.2014.008

[41] International Civil Aviation Organization, “Airborne Collision Avoidance System (ACAS) Manual,” Doc 9863 AN/461, 2012.

[42] T. Hutchings, “Architecting UAV sense & avoid systems,” in *Proc. of 2007 Institution of Engineering and Technology Conference on Autonomous Systems*, London, UK, 2007, pp. 1–8.

[43] B. Karhoff et al., “Eyes in the Domestic Sky: An Assessment of Sense and Avoid Technology for the Army’s ‘Warrior’ Unmanned Aerial Vehicle,” in *Proc. of 2006 IEEE Systems and Information Engineering Design Symposium*, 2006, pp. 36–42.

[44] Y. K. Kwag and J. W. Kang, “Obstacle awareness and collision avoidance radar sensor system for low-altitude flying smart UAV,” in *Proc. of The 23rd Digital Avionics Systems Conf. (IEEE Cat. No.04CH37576)*, 2004, pp. 12.D.2–121–10.

[45] Y. K. Kwag and C. H. Chung, “UAV based collision avoidance radar sensor,” *2007 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 639–642, 2007.

[46] A. Moses et al., “Radar-based detection and identification for miniature air vehicles,” in *Proc. of 2011 IEEE Int. Conf. on Control Applications (CCA)*, 2011, pp. 933–940.

[47] A. A. Moses et al., “UAV-borne X-band Radar for MAV Collision Avoidance,” in *Proc. of SPIE Unmanned Systems Technology XIII*, 2011, vol. 8045, p. 80450U.

[48] G. Fasano et al., “Airborne Multisensor Tracking for Autonomous Collision Avoidance,” in *Proc. of 2006 9th Int. Conf. on Information Fusion*, 2006, pp. 1–7.

[49] G. Fasano et al., “Multi-Sensor-Based Fully Autonomous Non-Cooperative Collision Avoidance System for Unmanned Air Vehicles,” *J. Aerosp. Comput. Information, Commun.*, vol. 5, no. 10, pp. 338–360, Oct. 2008.

[50] G. Fasano, “Multi-Sensor based Fully Autonomous Non-Cooperative Collision Avoidance System for UAVs,” Thesis, Università degli Studi di Napoli Federico II, 2008.

[51] D. Accardo et al., “Flight Test of a Radar-Based Tracking System for UAS Sense and Avoid,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1139–1160, Apr. 2013.

[52] G. Barrows et al., “Biomimetic visual sensing and flight control,” in *Proc. of Bristol UAV Conf.*, 2002, pp. 159–168.

[53] S. Bermudez i Badia et al., “A Biologically Based Flight Control System for a Blimp-based UAV,” in *Proc. of 2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3053–3059.

DOI:10.15774/PPKE.ITK.2014.008

[54] W. Green and P. Y. Oh, “The Integration of a Multimodal MAV and Biomimetic Sensing for Autonomous Flights in Near-Earth Environments,” in *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*, K. P. Valavanis, Ed. Dordrecht: Springer, 2007, pp. 407–430.

[55] J. Chahl and A. Mizutani, “Biomimetic Attitude and Orientation Sensors,” *IEEE Sens. J.*, vol. 12, no. 2, pp. 289–297, Feb. 2012.

[56] D. J. Lee et al., “See and avoidance behaviors for autonomous navigation,” *SPIE Opics East, Robot. Technol. Archit.*, vol. 5609, pp. 1–12, 2004.

[57] R. J. Carnie et al., “Computer Vision Based Collision Avoidance for UAVs,” presented at the Proceedings 11th Australian International Aerospace Congress, 2005.

[58] R. J. Carnie et al., “Image processing algorithms for UAV ‘sense and avoid,’” *Proc. 2006 IEEE Int. Conf. Robot. Autom. 2006. ICRA 2006.*, no. May, pp. 2848–2853, 2006.

[59] L. Mejias et al., “Towards the implementation of vision-based {UAS} sense-and-avoid,” in *Proc. of Proceedings of the 27th International Congress of the Aeronautical Sciences*, 2010, pp. 19–24.

[60] J. Lai et al., “Airborne vision-based collision-detection system,” *J. F. Robot.*, vol. 28, no. 2, pp. 137–157, Mar. 2011.

[61] L. Mejias et al., “Demonstration of closed-loop airborne sense-and-avoid using machine vision,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 27, no. 4, pp. 4–7, Apr. 2012.

[62] J. Lai et al., “Vision-Based Estimation of Airborne Target Pseudobearing Rate using Hidden Markov Model Filters,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2129–2145, Oct. 2013.

[63] J. Lai et al., “Characterization of Sky-region Morphological-temporal Airborne Collision Detection,” *J. F. Robot.*, vol. 30, no. 2, pp. 171–193, 2013.

[64] T. G. McGee et al., “Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation,” in *Proc. of 2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 4679–4684.

[65] J. Utt et al., “Development of a sense and avoid system,” *AIAA Infotech Aerosp.*, no. September 2005, pp. 12.D.2–121–10, 2005.

[66] J. McCalmont et al., “Sense and avoid technology for unmanned aircraft systems,” in *Proc. of Automatic Target Recognition XVII.*, 2007, vol. 6566, p. 65660P–65660P–11.

DOI:10.15774/PPKE.ITK.2014.008

[67] R. V. Iyer and P. R. Chandler, "On the computation of the ego-motion and distance to obstacles for a micro air vehicle," in *Proc. of 2006 American Control Conf.*, 2006, pp. 2554–2559.

[68] R. V. Iyer and P. R. Chandler, "Vision-Based UAV Flight Control and Obstacle Avoidance," *2006 Am. Control Conf.*, pp. 2166–2170, 2006.

[69] J. Byrne and C. J. Taylor, "Expansion segmentation for visual collision detection and estimation," in *Proc. of 2009 IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 875–882.

[70] B. Cohen and J. Byrne, "Inertial aided SIFT for time to collision estimation," in *Proc. of 2009 IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 1613–1614.

[71] J. Byrne and R. K. Mehra, "Method and System for Visual Collision Detection and Estimation," WO Patent, WO/2010/129,907, 11-Nov-2010.

[72] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[73] Y. C. Paw, "Synthesis and validation of flight control for UAV," PhD Dissertation, University of Minnesota, 2009.

[74] W. K. Pratt, *Digital Image Processing*. New York, USA: John Wiley & Sons, Inc., 2001.

[75] T. Roska et al., "CNN Software Library (Templates and Algorithms)," *User's Guid. Version*, 1999.

[76] R. F. Stengel, *Flight Dynamics*. Princeton University Press, 2004.

[77] Eutecus Inc., "Multitarget Tracking Library." [Online]. Available: http://www.eutecus.com/export/sites/eutecus/ProdServ/InstantVision/mtt_flyer.pdf. [Accessed: 19-May-2014].

[78] M. Hernandez, "Optimal sensor trajectories in bearings-only tracking," in *Proc. of The 7th Int. Conf. on Information Fusion*, Stockholm, Sweden, 2004, pp. 1–8.

[79] C.-C. Chu et al., "Performance comparison of tight and loose INS-Camera integration," in Proc. of Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), 2001, p. 3516.

[80] T. Chu et al., "Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environment," *Sensors (Basel).*, vol. 12, no. 3, pp. 3162–85, Jan. 2012.

[81] H. Stewénius et al., "Recent developments on direct relative orientation," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 4, pp. 284–294, Jun. 2006.

DOI:10.15774/PPKE.ITK.2014.008

[82] P. H. S. Torr and A. Zisserman, “MLESAC: A New Robust Estimator with Application to Estimating Image Geometry,” *Comput. Vis. Image Underst.*, vol. 78, no. 1, pp. 138–156, Apr. 2000.

[83] S. Choi et al., “Performance Evaluation of RANSAC Family,” in *Proc. of the British Machine Vision Conference 2009*, 2009, pp. 81.1–81.12.

[84] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–77, Jun. 2004.

[85] R. I. Hartley, “Chirality,” *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 41–61, 1998.

[86] G. L. Mariottini and D. Prattichizzo, “EGT for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras,” *Robot. Autom. Mag. IEEE*, vol. 12, no. 4, pp. 26–39, 2005.

[87] J.-Y. Bouguet, “Complete camera calibration toolbox for matlab,” 2004. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html. [Accessed: 19-May-2014].

[88] H. Stewénius, “Calibrated Fivepoint Solver,” 2010. [Online]. Available: http://www.vis.uky.edu/~stewe/FIVEPOINT/. [Accessed: 19-May-2014].

[89] S. Gleason and D. Gebre-Egziabher, *GNSS applications and methods*. Artech House, 2009.

[90] Z. Vörösházi et al., “Implementation of embedded emulated-digital CNN-UM global analogic programming unit on FPGA and its application,” *Int. J. Circuit Theory Appl.*, vol. 36, no. 5–6, pp. 589–603, Jul. 2008.

[91] “MicroBlaze Soft Processor.” [Online]. Available: http://www.xilinx.com/tools/microblaze.htm. [Accessed: 20-May-2014].