Universidad Politécnica de Madrid
Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica

Vision-Based Tracking, Odometry and Control for UAV Autonomy

A thesis submitted for the degree of
Doctor of Philosophy in Robotics and Automation

Changhong Fu
MEng Automation Engineer

2015

Escuela Técnica Superior De Ingenieros Industriales
Doctorado en Automática y Robótica
Universidad Politécnica de Madrid

Vision-Based Tracking, Odometry and Control for UAV Autonomy

A thesis submitted for the degree of
Doctor of Philosophy in Robotics and Automation

Author:
Changhong Fu
MEng Automation Engineer

Supervisor:
Dr. Pascual Campoy Cervera
Ph.D. Full Professor Industrial Engineer

2015

Titulo:

Vision-Based Tracking, Odometry and Control for UAV Autonomy

Autor:
Changhong Fu
MEng Automation Engineer

Tribunal nombrado por el Mgfco. y Excmo. Sr Rector de la Universidad Politécnica de Madrid el día .... de .... de 2015

Tribunal:

Presidente	:	....
Vocal	:	....
Vocal	:	....
Vocal	:	....
Secretario	:	....
Suplente	:	....
Suplente	:	....

Realizado el acto de lectura y defensa de la tesis el día .... de .... de 2015.

Calificación de la Tesis ....

El Presidente: Los Vocales:

El Secretario:

To my parents

To my wife: Wenyuan Wang

To my son: Yijia Fu

Changhong Fu

Acknowledgements

I would like to express my sincere appreciations and thanks to all the persons who helped me in these important four years of my life, without whom this dissertation would not have been possible.

First and foremost, I would like to thank my supervisor: Professor Dr. Pascual Campoy Cervera. It has been a great honor to be his first Chinese Ph.D. student in his Computer Vision Group (CVG). He has taught me, both consciously and unconsciously, how to correctly carry on the research works and projects related to computer vision and UAVs. I extremely appreciate all his contributions of time, energy, idea and funding to make my Ph.D. experience stimulating and productive. I firmly believe that his manner and enthusiasm will motivate and inspire me in my rest of life.

I would like to thank associate vice-rector for international relations at my university: Professor. Dr. Angel Alvarez-Rodriguez. He has kindly provided hundreds of assistances for me to solve my various problems from daily life and understand spanish culture in the past four years. To be frank, his patience and interpersonal communication are worth learning forever.

I would like to thank Professor Dr. Sergio Dominguez. His daily sport concepts have motivated me to keep a healthy and strong body for research. And his humorous has prompted me to release research pressures.

I would like to thank Professor Dr. Srikanth Saripalli and Professor Dr. Erdal Kayacan. They have provided great opportunies for me to carry out my vision-based UAV researches in their labs, i.e. the Autonomous System Technologies Research & Integration Laboratory (ASTRIL) in Arizona State University (ASU-USA) and the Flight Mechanics & Control (FMC) Laboratory in Nanyang Technological University (NTU-Singapore). The persistence they have for their scientific researches are contagious for me.

I would like to thank Dr. Miguel Angel Olivares-Mendez. He has enthusiastically taught me how to properly face and solve the UAV problems from the researches, thereby saving more time to focus on the reseach topics I am interested in. And he offered me a lot of helps in my UAV projects.

I would like to thank Professor Dr. Ivan Fernando Mondragon Bernal and Professor Dr. Carol Viviana Martinez Luna. Their successful examples in the CVG have encouraged me to be a good researcher.

I would like to thank my colleagues in the CVG. I thank Adrian Carrio and Hriday Bavle, who have discussed and shared many UAV autolanding ideas with me, especially in the MeSOANTEN and TAISAP-UAV projects. I appreciate Dr. Aneesh Chauhan, who has provided me the useful solutions in processing point cloud data in the EuRoC project. I thank Ramon Suarez-Fernandez, who has supplied the great helps in the OMNIWORKS project. I thank Jesus Pestana-Puerta and Ignacio Mellado-Bataller for their helps in the IMAV2012 competition. I also would like to thank my other colleagues: Jean François Collumeau, Masoud Shakeri, Jose Luis Sanchez-Lopez and Carlos Sampedro.

I would like to thank all members in the ASTRIL and FMC. Yucong Lin has provided me plenty of assistances and suggestions in the daily life of USA. Anand Biradar has often accompanied with me to carry out the UAV flight tests. Nursultan Imanberdiyev has helped me a lot during the UAV research and daily life of Singapore.

I would like to thank all the professors, secretaries, engineers in the Centre for Automation and Robotics (CAR), all my current or previous students from different top-level Chinese univerisities and Italian university, the friends from international summer school or conferences, everyone in international relation office/secretary office at UPM and ASU, staff in student services centre at NTU, teachers or leaders in the Chinese Embassy, and the financial supports from the China Scholarship Council (CSC), the Spanish National R&D programs, the Spanish Ministry of Science MICYT project, ECHORD Project in the European FP7 and IRSES project within the Marie Curie Program FP7.

Finally, I am most grateful for my wife, my son and my parents. Without a doubt, all their love, patiences and supports have encouraged me to pursuit my Ph.D. degree during these past four years. I love all of you faithfully!

Changhong Fu
CVG-UPM, *Madrid*
September, 2015

Resumen

El principal objetivo de este trabajo es proporcionar una solución en tiempo real basada en visión estéreo o monocular precisa y robusta para que un vehículo aéreo no tripulado (UAV) sea autónomo en varios tipos de aplicaciones UAV, especialmente en entornos abarrotados sin señal GPS.

Este trabajo principalmente consiste en tres temas de investigación de UAV basados en técnicas de visión por computador: (I) visual tracking, proporciona soluciones efectivas para localizar visualmente objetos de interés estáticos o en movimiento durante el tiempo que dura el vuelo del UAV mediante una aproximación adaptativa online y una estrategia de múltiple resolución, de este modo superamos los problemas generados por las diferentes situaciones desafiantes, tales como cambios significativos de aspecto, iluminación del entorno variante, fondo del tracking embarullado, oclusión parcial o total de objetos, variaciones rápidas de posición y vibraciones mecánicas a bordo. La solución ha sido utilizada en aterrizajes autónomos, inspección de plataformas mar adentro o tracking de aviones en pleno vuelo para su detección y evasión; (II) odometría visual: proporciona una solución eficiente al UAV para estimar la posición con 6 grados de libertad (6D) usando únicamente la entrada de una cámara estéreo a bordo del UAV. Un método Semi-Global Blocking Matching (SGBM) eficiente basado en una estrategia grueso-a-fino ha sido implementada para una rápida y profunda estimación del plano. Además, la solución toma provecho eficazmente de la información 2D y 3D para estimar la posición 6D, resolviendo de esta manera la limitación de un punto de referencia fijo en la cámara estéreo. Una robusta aproximación volumétrica de mapping basada en el framework Octomap ha sido utilizada para reconstruir entornos cerrados y al aire libre bastante abarrotados en 3D con memoria y errores correlacionados espacialmente o temporalmente; (III) visual control, ofrece soluciones de

control prácticas para la navegación de un UAV usando Fuzzy Logic Controller (FLC) con la estimación visual. Y el framework de Cross-Entropy Optimization (CEO) ha sido usado para optimizar el factor de escala y la función de pertenencia en FLC.

Todas las soluciones basadas en visión en este trabajo han sido probadas en test reales. Y los conjuntos de datos de imágenes reales grabados en estos test o disponibles para la comunidad pública han sido utilizados para evaluar el rendimiento de estas soluciones basadas en visión con ground truth. Además, las soluciones de visión presentadas han sido comparadas con algoritmos de visión del estado del arte. Los test reales y los resultados de evaluación muestran que las soluciones basadas en visión proporcionadas han obtenido rendimientos en tiempo real precisos y robustos, o han alcanzado un mejor rendimiento que aquellos algoritmos del estado del arte. La estimación basada en visión ha ganado un rol muy importante en controlar un UAV típico para alcanzar autonomía en aplicaciones UAV.

Abstract

The main objective of this dissertation is providing real-time accurate robust **monocular** or **stereo vision-based solution** for Unmanned Aerial Vehicle (UAV) to achieve the **autonomy** in various types of UAV applications, especially in **GPS-denied dynamic cluttered** environments.

This dissertation mainly consists of three UAV research topics based on computer vision technique: (I) **visual tracking**, it supplys effective solutions to visually locate interesting static or moving object over time during UAV flight with on-line adaptivity approach and multiple-resolution strategy, thereby overcoming the problems generated by the different challenging situations, such as significant appearance change, variant surrounding illumination, cluttered tracking background, partial or full object occlusion, rapid pose variation and onboard mechanical vibration. The solutions have been utilized in autonomous landing, offshore floating platform inspection and midair aircraft tracking for sense-and-avoid; (II) **visual odometry**: it provides the efficient solution for UAV to estimate the 6 Degree-of-freedom (6D) pose using only the input of stereo camera onboard UAV. An efficient Semi-Global Blocking Matching (SGBM) method based on a coarse-to-fine strategy has been implemented for fast depth map estimation. In addition, the solution effectively takes advantage of both 2D and 3D information to estimate the 6D pose, thereby solving the limitation of a fixed small baseline in the stereo camera. A robust volumetric occupancy mapping approach based on the Octomap framework has been utilized to reconstruct indoor and outdoor large-scale cluttered environments in 3D with less temporally or spatially correlated measurement errors and memory; (III) **visual control**, it offers practical control solutions to navigate UAV using Fuzzy Logic Controller (FLC) with the visual estimation. And the Cross-Entropy Optimization (CEO) framework has been used to optimize the scaling factor and the membership function in FLC.

All the vision-based solutions in this dissertation have been tested in real tests. And the real image datasets recorded from these tests or available from public community have been utilized to evaluate the performance of these vision-based solutions with ground truth. Additionally, the presented vision solutions have compared with the state-of-art visual algorithms. Real tests and evaluation results show that the provided vision-based solutions have obtained real-time accurate robust performances, or gained better performance than those state-of-art visual algorithms. The vision-based estimation has played a critically important role for controlling a typical UAV to achieve autonomy in the UAV application.

Nomenclature

Unless otherwise stated, the conventions utilized in this dissertation are defined as follows:

- Matrix will be represented by capital bold letters, e.g. $\mathbf{R}$
- Vectors will be represented by a lower case bold letters, e.g. $\mathbf{t}$
- Scalar variables are normal italic letters, e.g. i or k
- Images are normally represented by capital bold letters, e.g. $\mathbf{I}$ or $\mathbf{T}$

Acronyms

The most frequently-used acronyms in this dissertation are listed as follows:

UAV: Unmanned Aerial Vehicle

VTOL: Vertical Take-Off and Landing

SLAM: Simultaneous Localization And Mapping

FAST: Features from Accelerated Segment Test

BRIEF: Binary Robust Independent Elementary Features

SIFT: Scale-Invariant Feature Transform

SURF: Speeded Up Robust Features

BA: Bundle Adjustment

PnP: Perspective-n-Point(s)

RANSAC: RANdom SAmple Consensus

LKT: Lucas-Kanade Tracker

DVT: Discriminative Visual Tracker

CPU: Central Processing Unit

FPS: Frames Per Second

DOF: Degrees Of Freedom

FOV: Field Of View

GPS: Global Positioning System

LRF: Laser Range Finder

IMU: Inertial Measurement Unit

EKF: Extend Kalman Filter

PID: Proportional-Integral-Derivative

FLC: Fuzzy Logic Controller

CEO: Cross Entropy Optimization

RMSE: Root Mean Square Error

GCS: Ground Control Station

OFP: Offshore Floating Platform

SAA: Sense-And-Avoid

CTF: Coarse-To-Fine

MSF: Multiple-Sensor Fusion

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation and Overview

An Unmanned Aerial Vehicle (UAV), also known as Unmanned Aerial System (UAS) or Drone, is an aircraft without human pilot onboard. It is controlled by the remote operation of a pilot in the Ground Control Station (GCS) or autonomously with onboard sensors. Although it has been initially designed and deployed for military applications, but it has been widely utilized in a variety of civilian and commercial applications to save time, money and lives currently, e.g. disaster rescue, field reconnaissance, target tracking, orchard monitoring, forest management, animal protection, delivery service, media advertisement, building inspection, urban planning, tourist guide, environment assessment and 3D terrain reconstruction.

For a typical UAV, e.g. Asctec Pelican or Firefly [1], 3DR IRIS or X84

[1]http://www.asctec.de

or Aero-M [2], senseFly eXom or eBee [3], DJI F450 or F550 or Matrice 100 [4], UASTech LinkQuad [5] and AR.Drone Parrot I or II [6], it has limited size, payload, computation capability, power supply and expanded mounting space for other onboard sensors. Therefore, selecting appropriate onboard sensor and exploiting its maximum potentiality play the critically important roles to achieve the full autonomy of a typical UAV in those huge amount of civilian and commercial applications.

In commercial market, a large number of available sensors can be selected and utilized for a typical UAV. Global Positioning System (GPS) device is well researched for outdoor tasks to navigate UAV (Yun et al., 2007) (Abdelkrim et al., 2008), however, GPS signal is unreliable in urban canyons or dense forest, and it is completely lost in indoor environments. Laser Range Finder (LRF), e.g. Hokuyo URG-04LX-UG01 or UTM-30LX [7], is applied as an alternative sensor to provide both localization and environment information (Bachrach et al., 2009), but it often requires more power consumption, computing capability and payload from UAV, has restricted perception distance and generates a 2D reconstruction map because of the limited Field-Of-View (FOV), i.e. typically only in a plane.

Considering the size, weight, cost, power consumption, mounting flexibility and the capability to extract useful information from complex surrounding environments of available sensors, **camera** is the most competitive tool for a typical UAV. The rich visual information from camera can be utilized to provide real-time accurate robust vision-based estimation for UAV using computer vision technique in indoor and outdoor, small- and large-scale dynamic cluttered environments.

Although a RGB-D camera, e.g. Microsoft Kinect [8] and Asus Xtion Pro Live [9], is capable of offering vision-based localization and estimation of full-size surrounding environment for UAV (Bachrach et al., 2012)(Scherer and Zell, 2013), but the depth information is estimated by its Infrared (IR) camera and projector, which is not suitable for outdoor applications, and the hardware configurations, e.g. camera and baseline, in RGB-D camera are fixed.

[2]http://3drobotics.com/
[3]https://www.sensefly.com/
[4]http://www.dji.com/
[5]http://www.uastech.com/
[6]http://ardrone2.parrot.com/
[7]http://www.hokuyo-aut.jp/
[8]http://www.xbox.com/en-US/xbox-360
[9]http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

Therefore, this dissertation mainly aims to supply real-time accurate robust **monocular** or **stereo vision-based solution** for typical UAVs to achieve the **autonomy** in various types of UAV indoor and outdoor applications, especially in **GPS-denied dynamic cluttered large-scale** environments. It has included and discussed three main UAV research areas as follows:

- **Visual Tracking**: it provides the solutions how to visually locate interesting static or moving object over time during UAV flight with different challenging situations.
- **Visual Odometry** (**VO**): it supplies the solution how to estimate the six Degree-of-Freedom (DoF) pose of UAV using only the input of stereo camera. In this dissertation, the 3D mapping result also has been presented.
- **Visual Control**: it offers the solution how to utilize the real-time accurate robust vision-based estimation for navigating the UAV with fuzzy logic control approach.

1.2. Problem Statement

Nowadays, the computer vision technique of monocular or stereo camera has attracted extensive investigations in the UAV community. However, developing a visual algorithm for UAV to estimate its state, i.e. position and orientation, and reconstruct flight environments is not an easy task under the challenging situations. As a question has been mentioned and asked frequently: how to choose a suitable camera, i.e. hardware, and develop a real-time accurate robust visual algorithm, i.e. software, to close the control loop for navigating UAV in a specific application?

Monocular Camera has been utilized as a minimum vision configuration for a typical UAV to carry on indoor and outdoor vision-based applications. For hardware problem, different existing commercial cameras or lenses have various size, weight, cost, power consumption, mounting flexibility and specifications in sensor size, frame rate, resolution, shutter, pixel size, optical size, focal length and Field-Of-View (FOV) angle. And with the prosperous development of integrated circuit and 3D printing, researchers or end-users even should design and configure their own new camera sensors to accomplish their specific UAV applications. For software problem, monocular camera cannot sufficiently estimate the real absolute scale, i.e. depth, to its observed surrounding environments. If the hardware and software problems have been successfully solved, a specific UAV application can maximumly profit from planned monocular camera-based solution.

Stereo Camera has been applied as minimum number of camera configuration for solving scale ambiguity problem, i.e. it is able to effectively estimate depth information determined by the baseline between the left and the right cameras. And this configuration has significantly increased the visual information available to vision-based algorithm. However, stereo camera still has two bottlenecks: (I) when the distance between the UAV and the observed environment is much larger than the baseline, the depth estimation becomes inaccurate or simply invalid; (II) the features detected by only one side camera (e.g. occlusion) cannot be associated with the depths, but those 2D features can provide useful information to strengthen the visual estimation. Therefore, both 2D and 3D visual information should be effectively utilized to achieve more accurate robust stereo vision-based solution for UAVs.

Visual Tracking from UAV is defined as the problem of estimating the state, e.g. position, of a 2D or 3D object in the current image with given state in previous image. In literature, the appearance of tracking object is only pre-defined or selected on the first image frame without online appearance learning approach during the whole tracking process, i.e. the object appearance or lighting condition in ambient environment is assumed to be unchanged as time progresses. And many works have applied an off-line machine learning algorithm to recognize a specified object, however, a big amount of image training data recorded from UAV tracking application are required to be trained off-line with rich experience, time and energy. To build a visual tracking algorithm, three main requirements should be taken into account: (I) **adaptivity**: it requires a reliable and sustained online adaptation mechanism to learn the real appearance of a 2D or 3D object; (II) **robustness**: it means that the tracking algorithm should be capable of following the 2D or 3D object accurately even under challenging conditions, such as significant appearance change, variant illumination intensity, cluttered tracking background, partial or full object occlusion, rapid pose variation and onboard mechanical vibration; (III) **real-time**: it demands the tracking algorithm to process live image frames at high speed and with an acceptable tracking performance, generating consecutive and fast vision estimations as the input for closing the control loop.

Visual Odometry refers to the problem of using input images of camera(s), as the only source of external information, to incrementally estimate the 6 Degree-of-freedom (6D) pose of UAV. The monocular and stereo Visual Odometry (VO) systems has often been applied for UAV to carry on the vision-based applications. Most of those VO systems basically contain the following four steps: (I) environment feature detection, i.e. the selected feature should be salient, easily observable and fast computed, the relative pose to the UAV is able to be estimated at the real-time frame rates; (II) common feature matching, i.e. the common detected features with differnt viewpoints in the consecutive image frames should be matched, and an approach related to the outlier rejection should be utilized; (III) motion update, i.e. the 6 Degree-of-Freedom parameters (roll, pitch, yaw, x-. y- and z-translation) should be updated from the frame-to-frame estimation; (IV) pose refinement, i.e. the bundle adjustment has been applied for refining or optimizing the pose estimated by the initially frame-to-frame pose estimatiom. In this step, some critical image frames, i.e. the keyframes, are often selected based on the certain mechanism. In contrast to the monocular VO, stereo VO has better performance to solve the scale ambiguity problem.

Visual Control is the problem of using processed computer vision output, e.g. the vision-based estimation from visual tracking or VO approch, to control the motion of a UAV. The Proportional-Integral-Derivative (PID) controller has been widely utilized in navigating UAV based on a certain, accurate and completeness UAV model. However, the uncertainty, inaccuracy, approximation and incompleteness problems often exist in real UAV model. Fuzzy Logic Controller (FLC), also known as model-free control, has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying UAV system to solve those above problems. It mainly consists of three different types of parameters: (I) Scaling Factor (SF), which is defined as the gains for inputs and outputs. Its adjustment causes macroscopic effects to the behavior of the FLC, i.e. affecting the whole rule tables; (II) Membership Function (MF), typically, it is the triangle-shaped function, and its modification leads to medium-size changes, i.e. changing one row/column of the rule tables; (III) Rule Weight (RW), it is also known as the certainty grade of each rule, its regulation brings microscopic modifications for the FLC, i.e. modifying one unit of the rule tables. The FLC can be manually tuned or on-line optimized from macroscopic to microscopic effects, i.e. SF adjustment, MF modification and RW regulation. However, tuning the parameters of FLC with manual method not only requires the rich expert knowledge (exprience), a huge amount of UAV tests and time, but also increases the risks in operating UAV.

1.3. Dissertation Outline and Contributions

This dissertation has mainly focused on providing real-time arrurate robust vision-based estimations for typical UAVs to achieve the autonomy in various types of applications with monocular or stereo camera, especially in GPS-denied dynamic cluttered environments. And the codes of all visual and control algorithms in this dissertation have been developed in the Robot Operating System (ROS)[10] framework.

Nonetheless, the outline and specific contributions of this dissertation are listed below:

Chapter 2: State-Of-The-Art

- A detailed review of the state-of-art algorithms related to vision-based tracking, visual odometry and visual control for typical UAVs is introduced in this chapter.

Chapter 3: Visual Tracking

- An **online adaptive generative** visual tracking algorithm has been developed by learning the appearance of tracking object, i.e. **positive** sample [11], to land UAV on an arbitrary object, e.g. helipad or 3D object, even under challenging conditions, such as significant appearance change, different camera viewpoint, variant illumination intensity, cluttered tracking background, partial object occlusion, rapid pose variation and onboard mechanical vibration. The details of this visual tracker have been presented and discussed in Chapter 3, it mainly consists of three key parts:
  - **low-dimensional subspace representation method** (Belhumeur and Kriegman, 1996): it uses an **eigenspace** to represent the appearance of the tracking object instead of directly treating the tracking object as a set of pixels.
  - **online incremental learning approach** (Ross et al., 2008): it correctly updates both the sample mean and the eigenbasis using the information on previous consecutive image frames.

[10]http://www.ros.org/

[11]We denote tracking object as positive sample, and background information as negative sample.

- **hierarchical tracking strategy**: it adopts Multi-Resolution (MR) approach for each image frame to cope with the problems of strong motions (e.g. onboard mechanical vibration) or large displacements over time. In addition, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAV and Ground Control Station (GCS).

- An online adaptive **discriminative** visual algorithm (also called visual **tracking-by-detection** or **model-free tracking** approach) has been utilized to track objects, e.g. sensors, bolts, nuts and steel parts, on moving Offshore Floating Platform (OFP) for UAVs, i.e. the tracking object is separated from its dynamic surrounding background by an **adaptive binary classifier**, which is **on-line updated** with both **positive** and **negative** image samples. Using the hierarchical tracking strategy, especially in the Multi-Classifier (MC) voting mechanism, the importances of test samples have been used to reject samples, i.e. the lower resolution features are initially applied in rejecting the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, thereby ensuring the real-time performance and higher accuracy. The details of this Discriminative Visual Tracker (DVT) have been proposed and discussed in Chapter 3.

- An **online Multiple-Instance Learning (MIL)** (Dietterich et al., 1997) method has been integrated into discriminative algorithm to handle the **ambiguity problem** [12], which utilized the **positive and negative bags** [13] to update the adaptive binary classifier, and then trains a classifier in an online manner using bag likelihood function. This method has demonstrated good performance to handle drift, and can even solve significant appearance changes in the cluttered background. The details of this MIL-based DVT have been presented and discussed in Chapter 3 to track the midair intruder aircrafts from UAVs, which plays an important role in the UAV See-And-Avoid (SAA) application.

[12] The *exact* location of tracking object is unknown during cropping the positive samples

[13] The labels are provided for the bags rather than individual instances. A bag is positive if it contains at least one positive instance; otherwise, it is a negative bag.

Chapter 4: Visual Odometry

- A **stereo visual odometry** and **mapping** framework has been designed onboard a typical UAV to estimate the 6D pose and reconstruct the full-size flight environments. The details of this stereo visual odometry and mapping result have been presented in Chapter 4, which mainly contains:

  - A new light small-scale low-cost ARM-based stereo vision preprocessing system for typical UAV has been designed, which has advantages in terms of size, weight, cost and computational performance.
  - For the purpose of achieving real-time performance, depth map has been only estimated using one of two stereo image pairs instead of processing every consecutive image pairs, and the stereo **Semi-Global Block Matching** (SGBM) method (Hirschmuller, 2005) with a **Coarse-To-Fine** (CTF) strategy (Hermann and Klette, 2012) has been adopted to estimate the depth map.
  - The features have been extracted and tracked on the consecutive reference image parallelly using the **bucketing** method (Kitt et al., 2010) , i.e. each bucket are smoothed with a Gaussian kernel to reduce noise firstly, then the FAST detector (Rosten and Drummond, 2006) is used to extract the keypoints, and a modified version of the BRIEF descriptor (Calonder et al., 2010) has been employed to track FAST features.
  - A stereo visual odometry has been implemented for estimating the 6D pose of UAV, it effectively takes advantage of both **2D** (without depth) and **3D** (with depth) information to estimate the 6D pose **between each two consecutive image pairs**.
  - A **spherical coordinate system** has been applied for representing map point, i.e. a map point is represented by its radial distance, polar angle and azimuthal angle, which is similar to the sensing technology, i.e. the denser (sparser) point distribution that is closer (farther) to the UAV.
  - A **robust volumetric occupancy mapping** approach based on the original Octomap (Hornung et al., 2013) framework has been utilized for UAV to reconstruct arbitary indoor and outdoor large-scale cluttered environments in 3D with less temporally

or spatially correlated measurement errors and memory; This octree-based occupancy grid map models the occupied space (obstacles) and free areas clearly, and supports with coarse-to-fine resolutions.

Chapter 5: Visual Control

- A **Fuzzy Logic Controller** (FLC) has been designed and optimized as a model-free controller for UAV to carry on autonomous collision avoidance application. The details related to the FLC-based application have been proposed and discussed in Chapter 5.

  - The FLC has three inputs and one output. The Scaling Factors (SFs), triangle-shaped membership functions (MFs) and Rule Weights (RWs) have been set to the FLC.
  - A **Cross Entropy Optimization** (CEO) framework has been utilized as a **lazy** method to obtain the optimal SFs and MFs for FLC.
  - The monocular keyframe-based V-SLAM (SLAM) system has been utilized to estimate the 6D pose of UAV, and applied for real-time autonomous collision avoidance application.
  - A **Multiple-Sensor Fusion** (MSF) module based on the **Extended Kalman Filter** (EKF) has been applied for **fusing** the vision-based estimation and the measurement from **Inertial Measurement Unit** (IMU) .
  - Two different types of optimized FLCs have been compared and evaluated based on their control performances. One type is the FLC with optimized SFs, the other is the FLC with optimized SFs and MFs.

Chapter 6: Conclusions and Future Works

- The developed vision-based solutions related to visual tracking, odometry and control have been discussed and summarized in the conclusions.

- The directions of future works have been presented and discussed.

Chapter 2

State-Of-The-Art

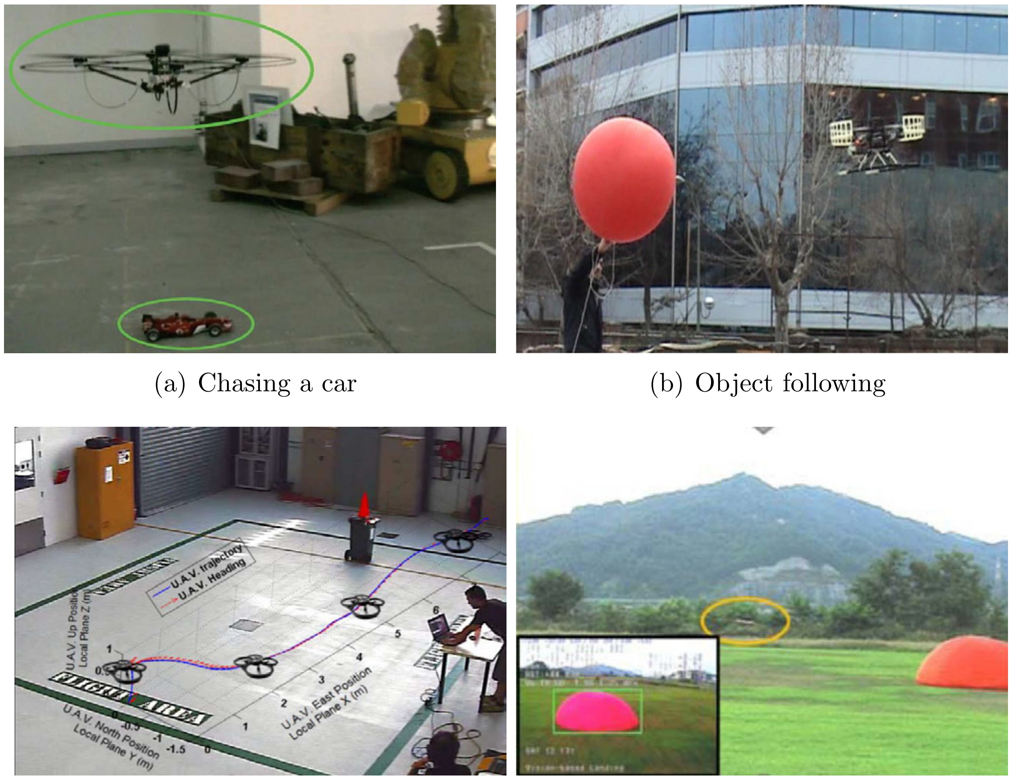
In literature, computer vision technique has been fruitfully researched and developed in the UAV community for different types of vision-based applications to estimate the state of UAV and even understand the surrounding flight environment. This chapter has introduced a detailed review of the state-of-art visual algorithms related to vision-based tracking, odometry and control of UAVs, and indicated the differences between the contributions of this dissertation and those state-of-art approaches.

2.1. Visual Tracking

In recent years, different visual object tracking methods have been applied for UAVs to robustly estimate the motion state, e.g. position, orientation and scale, of a 2D or 3D object. The typical visual tracking system or framework consists of three components ((Babenko et al., 2011)): (I) the appearance model, which can evaluate the likelihood that the target is at some particular locations; (II) the motion model, which relates the locations of the target over time; (III) the search strategy, which is applied for finding the most likely location in the current frame. And as the conclusion in a survey of visual tracking (Yilmaz et al., 2006), various state-of-art approaches are mainly differ from each other based on how to solve the following questions: (I) which object representation is suitable for tracking? (II) which image features should be used? (III) how should the motion, appearance and shape of the object be modeled?

The **color** information in image frame has played a critically important role in the visual tracking from UAVs. (Azrad et al., 2010) has proposed a color-based visual algorithm used to track a fixed target and autonomously stabilize a UAV. (Teuliere et al., 2011) has presented a robust color-based tracker for UAV to autonomously track and chase a moving red car, as shown in Fig. 2.1(a). An adaptive tracking method based on the color information has been adopted for a quadrotor UAV to follow a red 3D flying object and detect a red obstacle for see-and-avoid task in (Olivares-Mendez et al., 2011) and (Olivares-Mendez et al., 2012), as shown in Fig. 2.1(b) and 2.1(c). (Huh and Shim, 2010) has adopted a color- and moment-based target detection method to track a red monotone hemispherical airbag for UAV autonomous landing, as shown in Fig. 2.1(d). In addition, (Fu et al., 2012) has applied the color information to autnomously segment and recognize the number or character on the wall or floor for UAV to finish the indoor exploration task. Although the color-based object tracking is very efficient and different types of color spaces have been utilized in many visual tracking works, this kind of visual information is sensitive to image noise and illumination changes.

A **rectangle** shape has usually been applied to represent a static or moving object tracked by UAV. Therefore, the translation, affine or homography transformation (Hartley and Zisserman, 2004) have often been utilized to model the motion of object. The most common method with this representation is **template matching** (Brunelli, 2009), which searches a region in the current image frame similar to the object template defined in the previous image frame. For example, (Martínez et al., 2014) has adopted the **direct method**, i.e. directly represent the object using the intensity information of all pixels, to track the insteresting object, as the helipad shown in Fig. 2.2(a). (Mejias et al., 2006a), (Campoy et al., 2009), (Mondragón et al., 2010) and (Yang et al., 2014) have applied the **feature-based** approaches, e.g. Harris Corner (Harris and Stephens, 1988), SIFT (Lowe, 2004), SURF (Bay et al., 2008) or ORB (Rublee et al., 2011) features, for visual object tracking from UAVs, some examples are shown in Fig. 2.2(b), 2.2(c) and 2.2(d). However, template matching approach is suitable



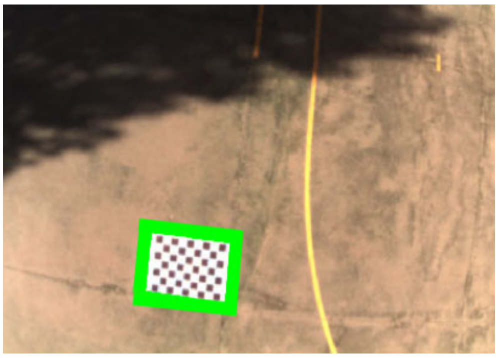
(a) Chasing a car (b) Object following

(c) See-and-avoid (d) Autonomous Landing

Figure 2.1: Visual object tracking from UAVs using color information.

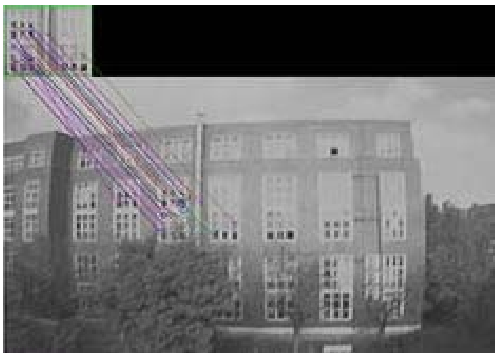
for tracking planar objects. And since the object template has been defined and fixed in the previous image frame, those visual trackers cannot learn the object appearance during the whole UAV tracking process with the challenging situations, such as significant appearance change, variant surrounding illumination, cluttered tracking background and full or partial object occlusion.

Machine learning approaches have been widely utilized in the UAV vision-based tracking applications. They have been divided into two categories based on the learning schemes: **off-line** and **on-line** learning methods. (Sanchez-Lopez et al., 2013) has applied an off-line learning algorithm to recognize the specified 2D planar object for UAV. And a supervised learning approach for solving the tower detection and classification problem has been presented in (Sampedro et al., 2014). However, all these works with off-line learning methods are requiring a large amount of image training data, which should maximumly include images captured from all the chal-

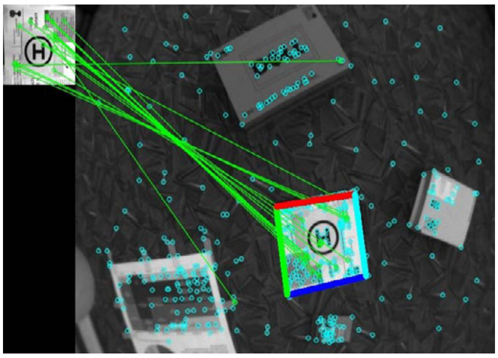


(a) 2D Helipad detection

(b) 2D Object following



(c) Window tracking



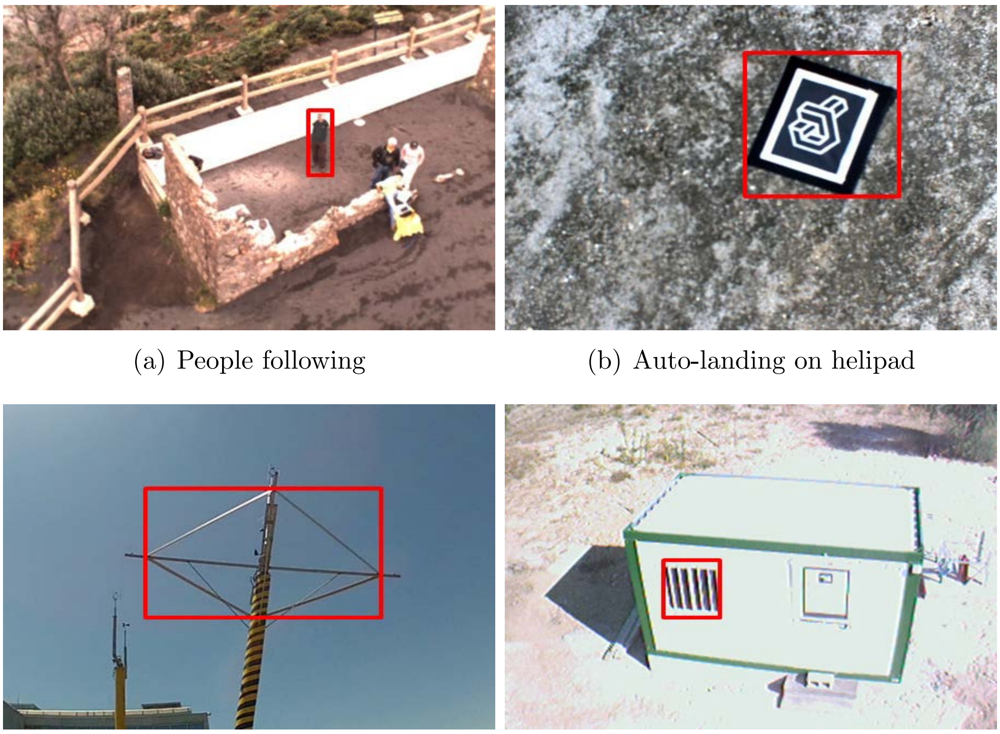
(d) Helipad tracking

Figure 2.2: Visual object tracking with direct and feature-based methods.

lenging conditions and labelled by human operator with plenty of time and energy, to be trained off-line.

On-line learning-based visual trackers, also known as **model-free** tracking or **tracking-by-detection** approaches, have been applied as the most popular tracking methods to handle the appearance variations of a 2D or 3D object currently. In general, on-line learning algorithms can be also divided into two categories: **generative** methods and **discriminative** methods. Generative methods (Jepson et al., 2003)(Zhou et al., 2004)(Lee and Kriegman, 2005)(Ross et al., 2008)(Kwon et al., 2009), which have been used to online learn only the appearance of a 2D or 3D tracking object itself. In literature, (Fu et al., 2015) has adopted a generalized visual tracking framework based on the hierarchical and incremental subspace learning method with a sample mean update to track a 2D or 3D object for a quadrotor UAV. All these works have obtained the promising tracking results, however, the background (i.e. negative) information as an useful visual cue has not been utilized to improve the accuracy of visual tracking from UAVs, especially when the background is cluttered and when multiple objects are appear.

Discriminative methods (Collins et al., 2005)(Wang et al., 2005)(Avidan, 2007)(Tian et al., 2007)(Saffari et al., 2009), which have applied an on-line trained and updated binary classifier to distinguish a 2D or 3D object from the background with positive (i.e. tracking object) and negative (i.e. background) information. (Fu et al., 2013) has presented a real-time adaptive Multi-Classifier Multi-Resolution (AMCMR) Discriminative Visual Tracking (DVT) framework for UAVs to track 2D or 3D objects. However, each update step of the visual tracker may introduced some noises, leading to tracking failure, i.e. drift problem. In literature, (Babenko et al., 2009) has proposed a novel tracking method based on the online **Multiple-Instance Learning** (MIL) method, which has successfully resolved the uncertainties of where to carry on positive updates during tracking process, i.e. ambiguity problem. Therefore, (Fu et al., 2014a) has integrated online MIL approach to AMCMR framework to track different objects from the fixed-wing UAVs. Some real-time tracking results with DVT have been shown in Fig. 2.3.



(a) People following

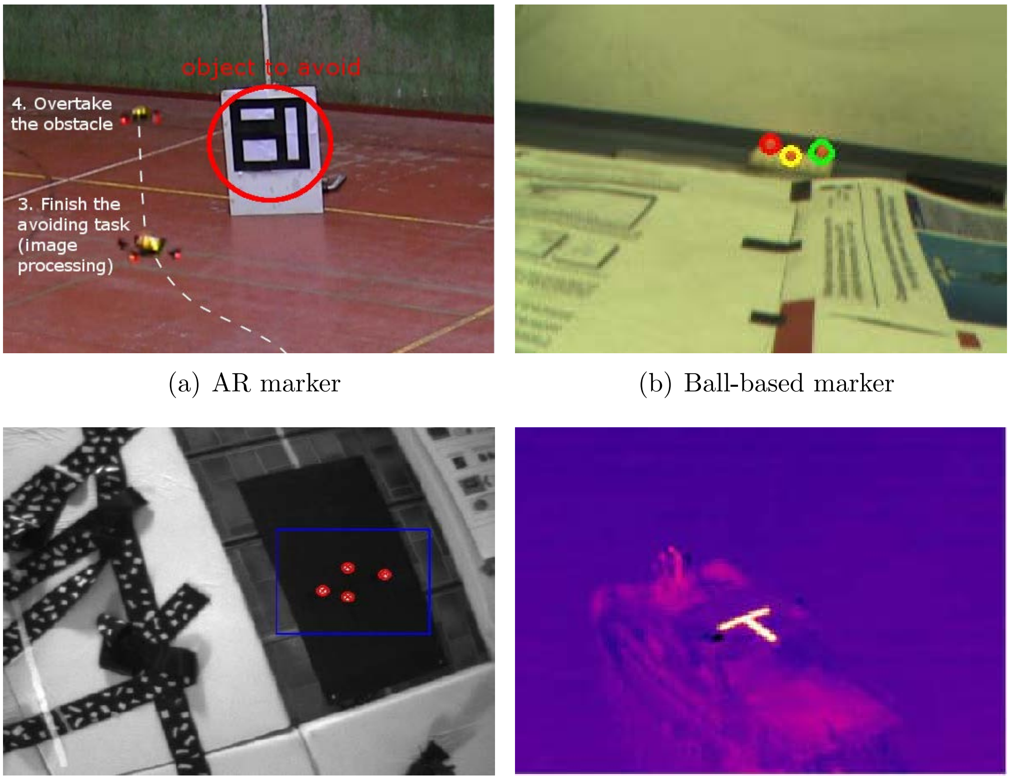
(b) Auto-landing on helipad

(c) Platform inspection

(d) Window tracking

Figure 2.3: Real-time visual object tracking from UAVs with DVT.

In addition, the **marker**-based visual tracking approaches has been utilized to estimate the 6D pose of UAV in different applications, this kind of method can estimate the **absolute** pose related to the tracking object. (Olivares-Mendez et al., 2014) has presented a case study of see-and-avoid application for UAV based on the detection of an Augmented Reality (AR) marker, i.e. ArUco [1] marker, as shown in Fig. 2.4(a). (Masselli et al., 2014) has developed a 6D pose estimation method for UAV using a pattern of four table tennis balls, as shown in Fig. 2.4(b). And (Breitenmoser et al., 2011) has utilized the Light-Emitting Diode (LED) to accurately estimate the 6D pose of UAV at real-time frame rates, as shown in Fig. 2.4(c). Additionally, (Xu et al., 2009) has adopted a thermal imager mounted on UAV to capture the infrared radiation images for autonomous landing on a ship, and a T shape marker has been applied for visual tracking, as shown in Fig. 2.4(d).



(a) AR marker

(b) Ball-based marker

(c) LED-based marker

(d) T-shape marker

Figure 2.4: Visual object tracking from UAVs with marker-based approaches.

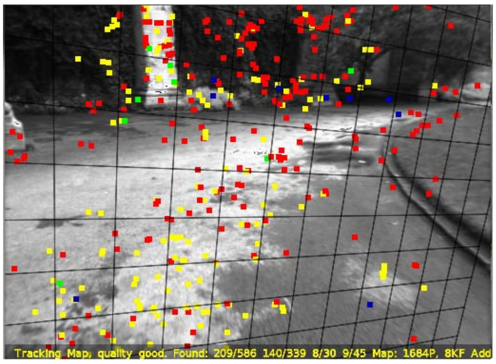
[1] http://www.uco.es/investiga/grupos/ava/node/26

2.2. Visual Odometry

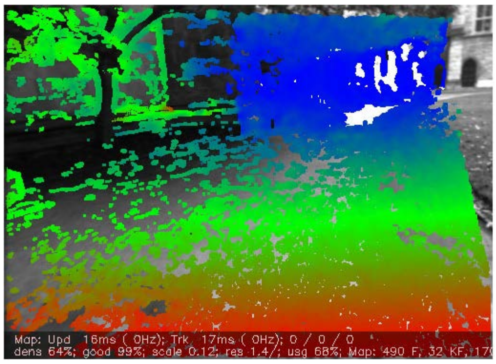
Recently, the Visual Odometry (VO) also has been utilized for various applications to estimate the 6D pose of UAVs. The VO approaches can be classified into two main categories based on the number of cameras adopted: monocular and stereo VO methods. In literature, (Klein and Murray, 2007) has proposed the most representative monocular keyframe-based tracking and mapping system, i.e. PTAM, for real time pose estimation applications, as shown in Fig. 2.5(a). (Brockers et al., 2014) has modified this PTAM system for UAV to estimate 6D pose on an onboard **embedded computer**, i.e. Odroid U2, at 30 Frames Per Second (FPS). (Forster et al., 2014) also presented a semi-direct monocular visual odometry algorithm, i.e. SVO, on the same embedded computer, which runs at 55 FPS and outputs a sparse 3D reconstructed environment model. (Pizzoli et al., 2014) proposed a real-time probabilistic monocular pose estimation method for 3D dense environment reconstruction, i.e. REMODE. (Faessler et al., 2015) has applied the SVO and REMODE algorithms for 6D pose estimation and dense 3D mapping task with Odroid U3 embedded computer. (Engel et al., 2014) described a direct monocular Simultaneous Localization and Mapping (SLAM) algorithm for building consistent semi-dense reconstructions of the environments, as shown in Fig. 2.5(b). And (Mur-Artal and Tardós, 2015) has presented a keyframe-based monocular SLAM system with ORB features to estimate the 6D pose and reconstruct a sparse environment model, as shown in Fig. 2.5(c).

However, monocular VO is not adequately estimate the real absolute scale (i.e. scale ambiguity), especially in large-scale environments, generating accumulated scale drifts. Although an IMU sensor is adopted in many works (Faessler et al., 2015)(Forster et al., 2015)(Fu et al., 2014b) with a Multi-Sensor Fusion (MSF) module, or a lidar device (i.e. a motor actuated rotated Hokuyo UTM-30LX) is applied (Zhang et al., 2014) to solve this problem, but the performance of finally fused or enhanced pose estimation in these works mainly lie on the measurement accuracy of these extra sensors, and the higher performance or quality of these extra sensors will result in more expensive sensor system. And some of extra sensors are still too heavy to be carried onboard a typical UAV, and they require more computational capability and power consumption from UAV.

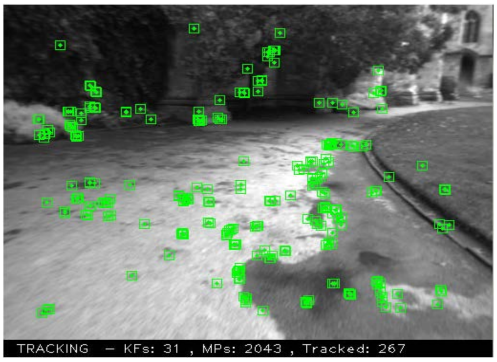
A stereo pair is applied as minimum number configuration of cameras for solving scale ambiguity problem to carry on the stere visual odometry (Herath et al., 2006)(Paz et al., 2008) (Mei et al., 2009)(Brand et al., 2014).



(a) PTAM



(b) LSD-SLAM



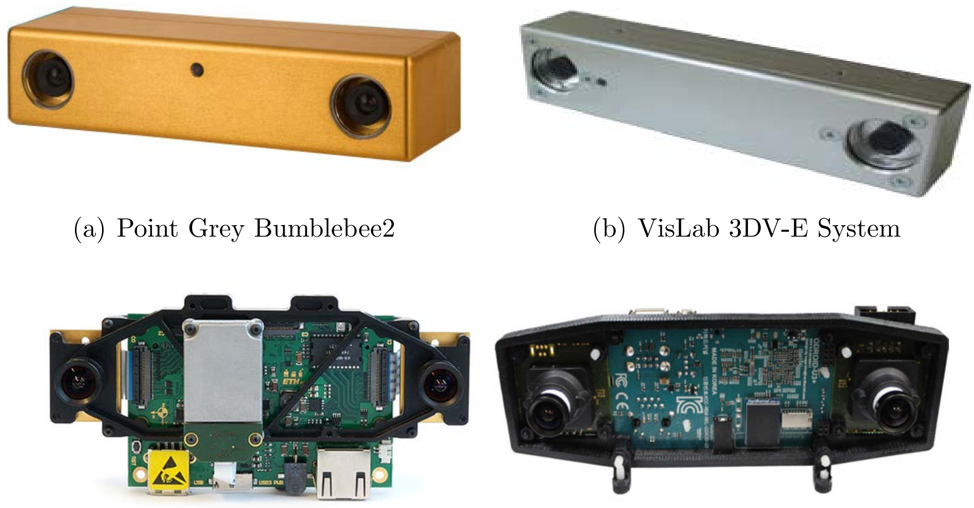
(c) ORB-SLAM

Figure 2.5: The well-known monocular VO systems. Images from (Mur-Artal and Tardós, 2015).

Especailly, (Strasdat et al., 2011) has implementated a double window optimization framework for constant-time visual stereo SLAM, i.e. ScaViSLAM[2]. As introduced in chapter 1, a typical UAV has limited size, payload, computation capability, power supply and expanded mounting space for other sensors. Although many stereo cameras are available to be sold on the commercial markets currently, e.g. Skybotix VI-sensor[3], Point Grey Bumblebee2[4] and VisLab 3DV-E[5], as shown in Fig. 2.8. However, the high cost (e.g. Skybotix VI-sensor and VisLab 3DV-E), big weight (e.g. Point Grey Bumblebee2 and VisLab 3DV-E) or incompatible communication interface (e.g. Point Grey Bumblebee2) reduce a number of potential university or

[2]https://github.com/strasdat/ScaViSLAM/
[3]http://www.skybotix.com/
[4]http://www.ptgrey.com/
[5]http://vislab.it/products/

company end-users to use these stereo cameras for a wide variety of UAV applications. Therefore, we designed a new light small-scale low-cost embedded stereo vision system for UAV to process rich visual information, the details of our embedded system are introduced in Section 4.2.1.



(a) Point Grey Bumblebee2

(b) VisLab 3DV-E System

(c) Skybotix VI Sensor

(d) Our Stereo System

Figure 2.6: Commercial stereo systems and our designed stereo device.

(Geiger et al., 2011) and (Kitt et al., 2010) have developed a well-known open-source algorithm, i.e. Libviso2 [6], for real-time autonomy exploration in dynamic environments with stereo camera, and this library has been widely applied in different types of robot applications. For depth estimation in stereo camera, the frequently-used matching algorithms consist of Block Matching (BM) (Konolige, 1997), Semi-Global Block Matching (SGBM) (Hirschmuller, 2008) and Libelas (Geiger et al., 2010). In literature, the SGBM is often applied as a memory efficient implementation for embedded stereo camera. (Gehrig et al., 2009) presesnted an efficient approach to speed up the processing of depth map estimation with standard SGBM for embedded system. Moreover, (Hermann and Klette, 2012) proposed a coarse-to-fine strategy based on standard SGBM, i.e. SGBM$_{\mathcal{F}}$, to estimate the depth map with (40%) faster processing and denser disparities, a comparison between SGBM$_{\mathcal{F}}$ and standard SGBM results has been shown in Fig. 2.7.

[6]http://www.cvlibs.net/software/libviso/

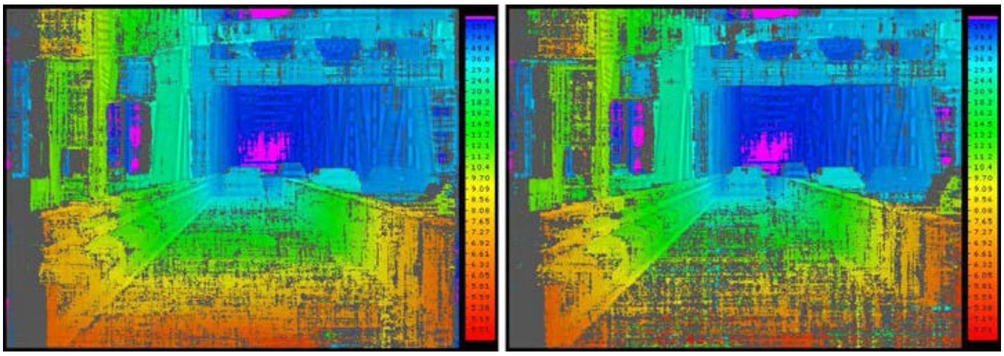


Figure 2.7: SGBM$_{\mathcal{F}}$ (Left) and standard SGBM (Right). Image from (Hermann and Klette, 2012).

To reconstruct the surrounding envrionment in 3D, (Hornung et al., 2013) implemented an efficient octree-based probabilistic 3D occupancy grid mapping approach, i.e. OctoMap, which is applied in numerous works, e.g. (Fossel et al., 2013), (Nieuwenhuisen et al., 2014) and EuRoC Challenge[7]. In practice, the original Octomap is prone to generate a number of falsely mapped artifact grids with stereo camera, (Schauwecker and Zell, 2014) presented a robust volumetric occupancy mapping method based on the original Octomap approach to solve this problem, it is more robust against high temporally or spatially correlated measurement errors, requires less memory and processes faster than original Octomap.

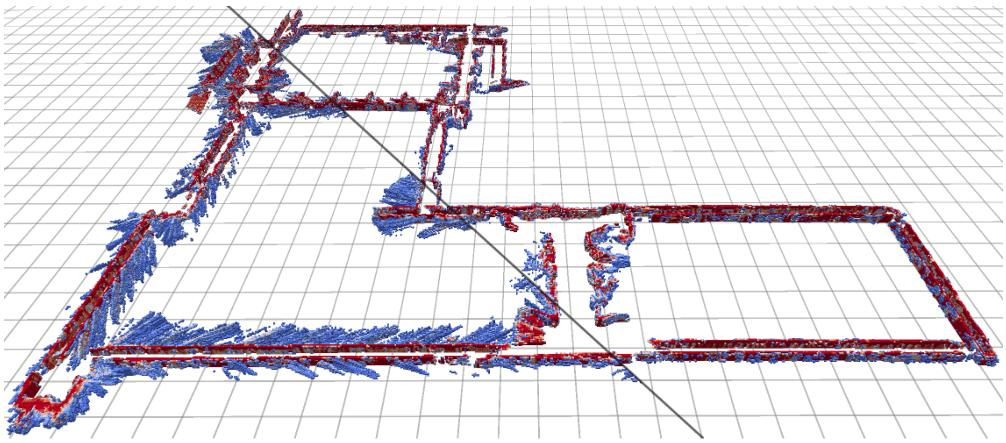


Figure 2.8: Original Octomap (Left-Bottom) and Schauwecker's method (Right-Up). Image from (Schauwecker and Zell, 2014).

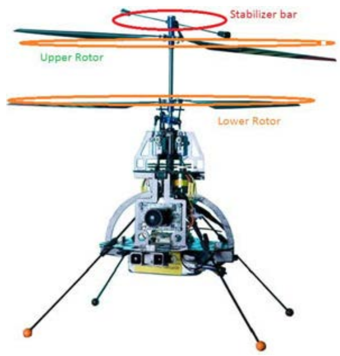
[7]http://www.euroc-project.eu/

2.3. Visual Control

Nowadays, different types of UAVs have been designed to carry on the various vision-based applications. To control the motions of these different UAVs using traditional PID control (Pestana et al., 2014), combination of Sliding mode and Backstepping (Bouabdallah and Siegwart, 2005), integration of PID and H_∞ (Wang et al., 2006) and Sliding mode control (Espinoza et al., 2014), the exact mathematics model of UAV should be identified in advanced. However, the uncertainty, inaccuracy, approximation and incompleteness problems widely exist in real controlling techniques. The Fuzzy Logic Controller (FLC) as one of the most active and fruitful soft computing methods can well deal with these issues. In addition, this *model-free* control approach often has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying UAV systems.

In literature, many applications have utilized FLC to control UAVs recently. The design process of a fuzzy logic based controller for a coaxial micro helicopter is presented in (Limnaios and Tsourveloudis, 2012), their UAV has been shown in Fig. 2.9(a). (Doitsidis et al., 2004) proposed a FLC for Unmanned Aerial Vehicle (UAV) to fly through specified waypoints in a 3D environment repeatedly, perform trajectory tracking, and duplicate or follow another vehicle's trajectory. (Coza and Macnab, 2006) provides a new method to design the adaptive-fuzzy controller to achieve stabilization of a quadrotor helicopter in the presence of sinusoidal wind disturbance. (Santos et al., 2010) also implemented a fuzzy controller to control the quadrotor. And (Kurnaz et al., 2009) proved its FLC can obtain the expected performances in the UAV control and navigation tasks. Moreover, (Olivares-Mendez et al., 2010) has presented a FLC for UAV in a landing application, the UAV used in this application has been shown in Fig. 2.9(c). (Olivares-Mendez et al., 2011) has proposed a FLC for a quadrotor UAV, as shown in Fig. 2.9(b), to follow a 3D object. And (Olivares-Mendez et al., 2009) proposed an implementation of two FLCs working parallelly for a pan-tilt camera platform on an UAV. Additionally, (Gomez and Jamshidi, 2011) has designed a fuzzy adaptive controller to stabllize and navigate a fixed-wing UAV, the UAV they applied is shown in Fig. 2.9(d). However, the parameters of FLC are manually adjusted in a huge amount of tests in these works, this kind of regulation method for FLC not only requires the rich expert knowledge (exprience) and time, but also increases the risks in operating UAV.

Therefore, autonomous tuning or *lazy* methods for FLC are more com-



(a) Coaxial UAV

(b) Multi-rotor UAV

(c) Helicopter UAV

(d) Fixed-wing UAV

Figure 2.9: Fuzzy Logic Controllers worked in different types of UAVs

petitive. A robustness comparison between model-based with self-tunable fuzzy inference system (STFIS) has been studied to control a drone in presence of disturbances in (K.M.Zemalache and H.Maaref, 2009). (Kadmiry and Driankov, 2004) designed an gain scheduler-based FLC for an unmanned helicopter to achieve stable and robust aggressive maneuverability. An adaptive neuro-fuzzy inference system (ANFIS) based controller for UAV was developed to adjust its altitude, the heading and the speed together in (Kurnaz et al., 2010). The classical and multi-objective genetic algorithm (GA) based fuzzy-genetic autopilot are also designed and used for UAV in (A.R.Babaei et al., 2011), which validated the time response characteristics, the robustness and the adaptation of fuzzy controller with respect to the large commands.

(E.Haber et al., 2010) have proved that the Cross-Entropy (CE) is the best optimization technique for FLC, they use CE to tune the Scaling Factor (SF) of a PD fuzzy controller for cutting force regulation in a drilling process. And a CE-based optimization for SF in a PID fuzzy controller to command the UAV for avoiding a small obstacle with special color has been presented in (Olivares-Mendez et al., 2013) and (Olivares-Mendez et al., 2012). Nonetheless, the CE in these works was limited to only optimizing the SF of FLC.

Chapter 3

Visual Tracking

This chapter[1] has presented different visual tracking algorithms for typical UAVs to work in various civilian applications. All these presented visual tracking algorithms have adopted on-line learning approches to adapt and update the appearance of a 2D or 3D object during real-time visual tracking

[1] publications related to this chapter:
- *"Towards an Autonomous Vision-Based Unmanned Aerial System Against Wildlife Poachers"*, Sensors, 2015
- *"SIGS: Synthetic Imagery Generating Software for the Development and Evaluation of Vision-based Sense-And-Avoid Systems"*, Journal of Intelligent & Robotic Systems, 2015
- *"Robust Real-Time Vision-Based Aircraft Tracking from Unmanned Aerial Vehicles"*, IEEE ICRA, 2014
- *"Online Learning-Based Robust Visual Tracking for Autonomous Landing of Unmanned Aerial Vehicles"*, IEEE ICUAS, 2014
- *"A Ground-Truth Video Dataset for the Development and Evaluation of Vision-Based Sense-and-Avoid Systems"*, IEEE ICUAS, 2014
- *"Real-Time Adaptive Multi-Classifier Multi-Resolution Visual Tracking Framework for Unmanned Aerial Vehicles"*, RED-UAS, 2013

processes.

3.1. Autonomous Landing

3.1.1. Introduction

In this section, a new on-line adaptive visual tracking algorithm has been developed for VTOL UAV to carry out autonomous landing application.

The new proposed visual algorithm applies a low-dimensional subspace represention scheme to model a 2D helipad or 3D object during the tracking process. Additionally, an online incremental learning approach to update the appearance of the helipad or 3D object is adopted. A particle filter is employed to estimate the motion model of helipad or 3D object. Moreover, we utilized a hierarchical tracking strategy, based on the multi-resolution of a frame, to cope with the problems of large displacements or strong motions over time. With this strategy, especially in the multiple particle filters voting mechanism, multiple motion models will be estimated at different resolution levels, i.e. lower resolution textures are initially applied to estimate a few motion parameters (e.g. location of helipad or 3D object) at a relatively low cost, leaving more motion paramters (e.g. scale, orientation and location of helipad or 3D object) to be estimated at higher resolutions. Besides this mechanism, a multiple-block-sizes adapting method has been utilized to update the helipad or 3D object with different frequencies, i.e. a smaller block size means more frequent updates, making it quicker to model appearance changes. All these approaches are integrated to ensure higher accuracy and real-time performance of the helipad or 3D object tracking from UAV. The details of this new visual algorithm have been introduced in section 3.1.4.

In addition, a novel light small-scale low-cost ARM [2] architecture-based efficient monocular vision system has also been designed for on-board UAV, thereby saving enough computing capability for the onboard primary computer to process the path planning, sensor fusion, flight control and other tasks, the details of this new system are introduced in section 3.1.3.

Nonetheless, the main contributions of this work are listed below:

(I) Designed a novel light small-scale low-cost ARM architecture-based on-board monocular vision system for the UAV.

(II) Developed a new online learning and tracking algorithm for the

[2] http://www.arm.com/

designed system using the Robot Operating System (ROS) and vectorized NEON [3] instructions.

(III) Applied this visual algorithm to solve the freewill selected helipad and 3D object tracking problems for real autonomous landing applications of VTOL UAVs.

(IV) Summarized the state-of-art works related to the standalone preprocessing systems and vision-based autolanding applications.

3.1.2. Related Works

In literature, monocular visual SLAM-based approaches, e.g. (Fu et al., 2014b), have obtained promissing performances for UAV applications, however, for the autonomous landing flights of UAVs, this kind of methods require an accurate relative pose between the UAV take-off place and the landing field or object, and the accuracy of the pose estimation depends on the measurements of extra sensors, e.g. IMU. Therefore, visual tracking-based algorithms are more popular to be applied in the UAV autolanding task.

A vision-based real-time landing algorithm for an autonomous helicopter was implemented by (Saripalli et al., 2003). They used moment descriptors to determine the orientation and location of the landing pad, however, it is difficult to apply this visual tracking algorithm in variant outdoor environments, because intensity values in the image vary significantly depending on the sunlight, vibration of the camera, helicopter heading and so on. Moreover, it does not have the adpative characteristic necessary for tackling with the appearance changes of the landing pad, and a differential GPS is used in their work to provide the altitude of the helicopter instead of having a vision-based altitude estimation.

A visual tracking algorithm, based on the Lucas-Kanade optical flow, was presented by (Mondragón et al., 2010) for a UAV to land on a helipad, where, the 3D position of the UAV is estimated using a pre-defined reference helipad selected on the first image frame, therefore, this tracker also cannot learn the helipad appearance during tracking, and the RANSAC (Fischler and Bolles, 1981) requires a big number of iterations (heavy time consumption) to reach optimal estimation. Similarily, the SIFT (Lowe, 2004), SURF (Bay et al., 2008) features have been used in visual tracking algorithms for autolanding of UAV. All these methods are known as *feature-based* visual tracking approaches.

[3]http://www.arm.com/products/processors/technologies/neon.php

The *direct tracking method* (i.e. directly represent the helipad using the intensity information of all pixels in the image) was utilized by (Martinez et al., 2013) to track helipad from a UAV. They have demonstrated that the direct method-based tracker performs better than those well-known *feature-based* algorithms, obtaining superior results, but they also employed a fixed helipad template for the whole UAV tracking process. Although this tracker has been improved in (Martínez et al., 2013) by manually adding many other templates, but still it does not provide online self-taught learning. And gradient descent method often falls into a local minimum value and is relatively slow to be close to the global minimum.

(Sanchez-Lopez et al., 2013) applied an off-line learning algorithm to recognize the specified 2D helipad in UAV autolanding application, i.e. a big amount of image training data is trained off-line using a Multi-Layer Perceptron Artificial Neural Network (MLP-ANN). However, the target recognition for landing, an **H** character as mentioned in their work, is fixed or predefined instead of freewill 2D or 3D objects selected online, and the collection of those image training data is difficult to cover all the challenging conditions from the real UAV flights. Moreover, it is time-consuming or empirical to obtain the optimal parameters for this kind of off-line learning methods.

In this work, to handle the problems of drift, rapid pose variation and variant surrounding illumination, motivated by (Black and Jepson, 1998), (Murase and Nayar, 1995), (Belhumeur and Kriegman, 1996), (Ke and Sukthankar, 2004), the low-dimensional subspace represention scheme is applied as the practicable method to represent or model the helipad or 3D object. And the online incremental learning approach is utilized as the effective technique for learning or updating the appearance of helipad or 3D object. Moreover, the Particle Filter (PF) (Arulampalam et al., 2002) and hierarchical tracking strategy are also employed to estimate the motion model of the helipad or 3D object for UAV autolanding applications.

3.1.3. Monocular Vision System

Our new monocular vision system is shown in Fig. 3.1, and its details have been introduced as follows:

(I) *computer*: it is the modification of hardkernel ODROID U3 [4] ($69), which has one 1.7 GHz *Quad*-Core processor (i.e. Samsung Exynos 4412 Prime Cortex-A9), 2 GByte RAM, 64 GByte eMMC-based storage memory

[4] http://www.hardkernel.com/

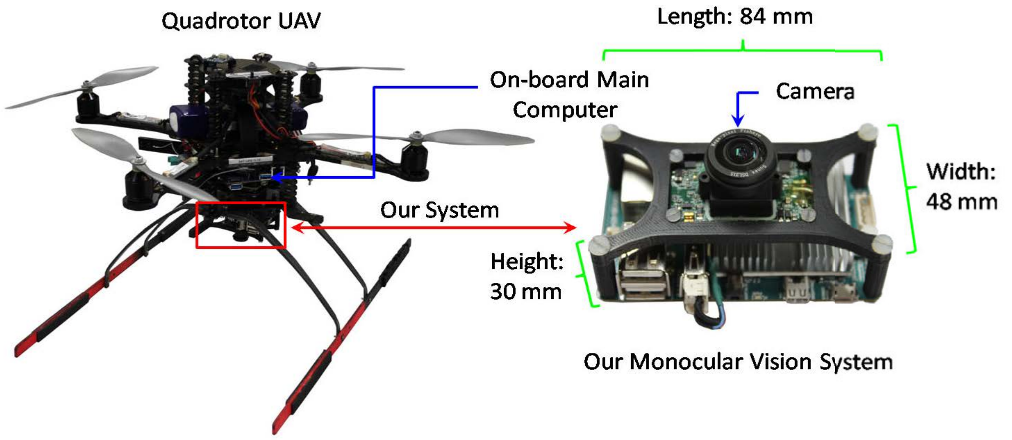


Figure 3.1: The prototype of our light on-board small-scale low-cost ARM architecture-based monocular vision system. It is mounted on the bottom of a quadrotor UAV (i.e. Asctec Pelican) for an autonomous landing application.

($79), 10/100 Mbps Ethernet with RJ-45 LAN Jack, 3 High speed USB2.0 Host ports, 1 micro HDMI, 1 micro USB, and GPIO/UART/I2C ports. Its size is 83mm×48mm×16mm, the weight is 48g (including heat sink). In our current monocular vision system, the tested operating systems are Ubuntu 13.04, 13.10 and 14.04, supporting with the Hydro and Indigo versions of the Robot Operating System (ROS), and the OpenCV [5] library is used to manage the image processing. In addition, it also supports a wireless communication module.

(II) *camera*: the system is equipped with one Point Grey Firefly MV camera [6] (type: FMVU-03MTC-CS) based on CMOS type sensor (model: Aptina MT9V022) with USB 2.0. The readout method of this camera is global shutter. The frame rate reaches up to 60 FPS. In our monocular vision system, the maximum image resolution is 752×480 pixels. The focal length of the lenses (i.e. Lensagon[7] BM2820) is 2.8mm, the horizontal and vertical fields of view are 98^o and 73^o, respectively. The camera size is 40mm×25mm×20mm, and the weight is 18 grams. In addition, the camera is fixed on a light multi-function mechanical part, which is also used to flexibly mount on the robots.

The total weight of whole system is 75 grams, which is lighter than other frequently-used sensors for UAVs, e.g. RGB-D sensor (Asus Xtion

[5]http://opencv.org/
[6]http://eu.ptgrey.com/
[7]http://www.lensation.de/

Pro Live): ∼200 grams, 2D Laser (Hokuyo UTM30-LX): ∼270 grams. The dimension is 83mm×48mm×35mm. Additionally, the cost of our monocular vision system is only 360 Euros. To the authors's best knowledge, this is the first work to present such a light low-cost ARM architecture-based monocular vision pre-processing system.

3.1.4. Adaptive Visual Tracking

In this section, the details of the proposed visual tracking algorithm for UAV autolanding has been introduced, as shown in Fig. 3.2. In the Fig. 3.2, the whole 3D car has been selected as a landing object, and a visual Augmented Reality (AR) marker has been *only* applied for obtaining the ground truths of 3D position and heading angle, i.e. yaw, of UAV to evaluate the performance of visual tracking estimation.

We assume that the camera is modeled as a pinhole camera (Hartley and Zisserman, 2004), and the intrinsic parameters of camera, e.g. optical center (c_x, c_y), focal length (f_x, f_y), are estimated using the ROS camera calibration tool [8].

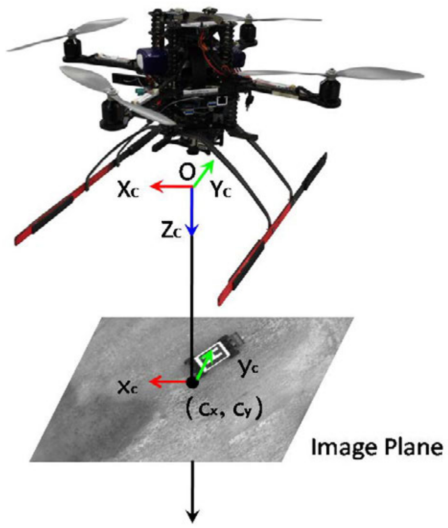


Figure 3.2: Vision-based autolanding for Asctec Pelican quadrotor UAV.

Recently, online incremental subspace learning methods, e.g. (Li et al., 2008), (Wang et al., 2007), (Wang et al., 2010), (Hu et al., 2011), have obtained promising tracking performances. Especially, (Ross et al., 2008) has presented an online incremental learning approach for effectively modelling

[8] http://wiki.ros.org/camera_calibration

and updating the tracking object with a low dimensional PCA (i.e. Principal Component Analysis) subspace representation method, which demonstrated that PCA subspace representation with online incremental update is robust to the appearance changes caused by rapid pose variation, variant surrounding illumination and partial target occlusion, as explained by Eq. 3.1 and shown in Fig. 3.3. In addition, PCA has also been demonstrated in (Ke and Sukthankar, 2004) (Juan and Gwon, 2009) to have those above advantages in tracking applications. (Levey and Lindenbaum, 2000) and (Hall et al., 2002) have done similar works to (Ross et al., 2008), however, (Levey and Lindenbaum, 2000) did not consider the changing of subspace mean when the new data arrive, and the forgetting factor is not integrated in (Hall et al., 2002), which generates a higher computational cost during the tracking process.

$$\mathbf{O} = \mathbf{U}\mathbf{c} + \mathbf{e} \tag{3.1}$$

where, $\mathbf{O}$ represents an observation vector, $\mathbf{c}$ indicates the target coding coefficient vector, $\mathbf{U}$ denotes the matrix of column basis vectors, and $\mathbf{e}$ is the error term, which is the Gaussian distribution with small variances.

The main precedures of online incremental PCA subspace learning algorithm with subspace mean update are as follows: Given a set of training image $\mathcal{S}_a = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_n\} \in \mathbb{R}^{d\times n}$, the appearance model of the helipad or 3D object can be computed by the Singular Value Decomposition (SVD) of the centered data matrix $[(\mathbf{S}_1 - \bar{\mathbf{S}}_a)\cdots(\mathbf{S}_n - \bar{\mathbf{S}}_a)]$, denoted by $(\mathcal{S}_1 - \bar{\mathbf{S}}_a)$, i.e. $(\mathcal{S}_a - \bar{\mathbf{S}}_a) = U\Sigma V^\top$, where, $\bar{\mathbf{S}}_a = \frac{1}{n}\sum_{i=1}^{n}\mathbf{S}_i$ is the sample mean of the training images.

If a new set of images $\mathcal{S}_b = \{\mathbf{S}_{n+1}, \mathbf{S}_{n+2}, ..., \mathbf{S}_{n+m}\} \in \mathbb{R}^{d\times m}$ arrives, then the mean vectors of $\mathcal{S}_b$ and $\mathcal{S}_c = [\mathcal{S}_a \ \mathcal{S}_b]$ are computed, i.e. $\bar{\mathbf{S}}_b = \frac{1}{m}\sum_{i=n+1}^{n+m}\mathbf{S}_i$, $\bar{\mathbf{S}}_c = \frac{n}{n+m}\bar{\mathbf{S}}_a + \frac{m}{n+m}\bar{\mathbf{S}}_b$. Because the SVD of $(\mathcal{S}_c - \bar{\mathbf{S}}_c)$ is equal to the SVD of concatenation of $(\mathcal{S}_a - \bar{\mathbf{S}}_a)$, $(\mathcal{S}_b - \bar{\mathbf{S}}_b)$ and $\sqrt{\frac{nm}{n+m}}(\bar{\mathbf{S}}_a - \bar{\mathbf{S}}_b)$, which is denoted as $(\mathcal{S}_c - \bar{\mathbf{S}}_c) = U'\Sigma'V'^\top$, this can be done efficiently by the SVD algorithm, i.e.:

$$U' = [U\,\tilde{E}]\tilde{U}, \qquad \Sigma' = \tilde{\Sigma} \tag{3.2}$$

where, $\tilde{U}$ and $\tilde{\Sigma}$ are calculated from the SVD of R: $\begin{bmatrix} \Sigma & U^\top E \\ 0 & \tilde{E}(E - UU^\top E) \end{bmatrix}$, E is the concatenation of $(\mathcal{S}_b - \bar{\mathbf{S}}_b)$ and $\sqrt{\frac{nm}{n+m}}(\bar{\mathbf{S}}_a - \bar{\mathbf{S}}_b)$, $\tilde{E}$ represents the orthogonalization of $E - UU^\top E$, U and Σ are the SVD of $(\mathcal{S}_a - \bar{\mathbf{S}}_a)$.

Taking the forgetting factor, i.e. $\eta \in (0, 1]$, into account for balancing between previous and current observations to reduce the storage and computation requirements, the R and $\bar{\mathbf{S}}_c$ are modified as below:

$$R = \begin{bmatrix} \eta\Sigma & U^\top E \\ 0 & \tilde{E}(E - UU^\top E) \end{bmatrix} \tag{3.3}$$

$$\bar{\mathbf{S}}_c = \frac{\eta n}{\eta n + m}\bar{\mathbf{S}}_a + \frac{m}{\eta n + m}\bar{\mathbf{S}}_b \tag{3.4}$$

where, $\eta = 1$ means that all previous data are included to adapt the changing appearance of the helipad or 3D object.

During the PCA subspace-based object tracking, as shown in Fig. 3.3, each object image is re-sized to the 32×32 pixels, and the reconstructed object image is constructed using the eigenbasis. Moreover, the eigenbasis images are sorted based on their according eigenvalues.

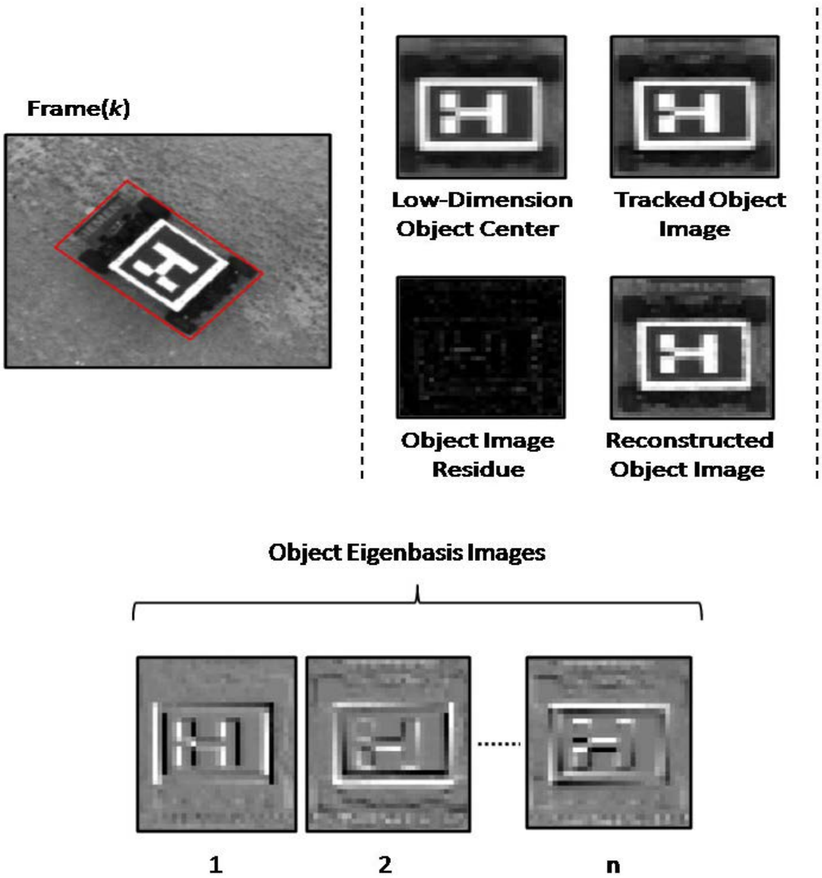


Figure 3.3: The PCA subspace-based tracking of a 3D object.

For the UAV autonomous autolanding task of the UAV, the visual helipad or 3D object tracking can be formulated as an inference problem with a Markov model and hidden state variables. Given a set of observed images $\mathcal{O}_k = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_k\}$ at the kth frame, the hidden state variable $\mathbf{X}_k$ can be estimated as below:

$$p(\mathbf{X}_k|\mathcal{O}_k) \propto p(\mathbf{O}_k|\mathbf{X}_k) \cdot \int p(\mathbf{X}_k|\mathbf{X}_{k-1})p(\mathbf{X}_{k-1}|\mathcal{O}_{k-1})\,d\mathbf{X}_{k-1} \tag{3.5}$$

where, $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ is the dynamic (motion) model between two consecutive states, as shown in Fig. 3.4 and Equ. 3.7. $p(\mathbf{O}_k|\mathbf{X}_k)$ represents the observation model that estimates the likelihood of observing $\mathbf{O}_k$ at the state $\mathbf{X}_k$. The optimal state of the tracking helipad or 3D object given all the observations up to kth frame is obtained by the maximum a posteriori estimation over N samples at time k by

$$\widehat{\mathbf{X}}_k = \arg\max_{\mathbf{X}_k^i} p(\mathbf{O}_k^i|\mathbf{X}_k^i)p(\mathbf{X}_k^i|\mathbf{X}_{k-1}), i = 1, 2, ..., N \tag{3.6}$$

where, $\mathbf{X}_k^i$ is the ith sample of the state $\mathbf{X}_k$, and $\mathbf{O}_k^i$ denotes the image patch predicted by $\mathbf{X}_k^i$.

Dynamic Model

In this application, we aim to utilize four parameters for constructing motion model X_k of the helipad or 3D object to close the vision control loop: (I) location x and y; (II) scale factor s; (III) rotation angle θ of the target in the image plane, which can be modelled between two consecutive frames, i.e. $\mathbf{X} = (x, y, s, \theta)$, it is called *Similarity* Transformation in (Hartley and Zisserman, 2004).

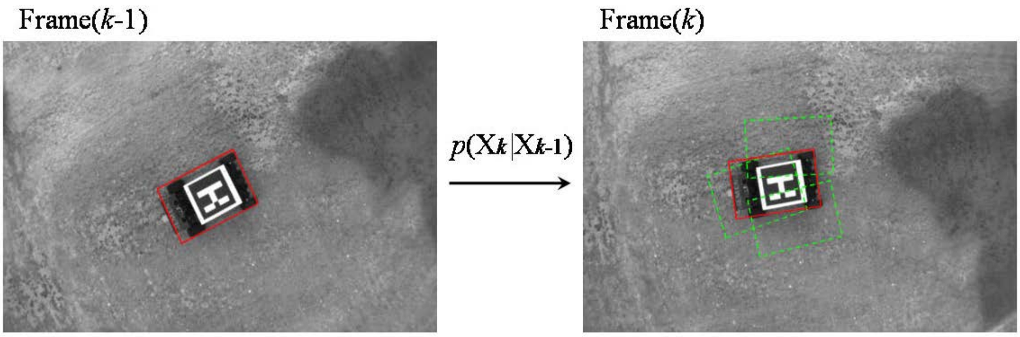


Figure 3.4: Dynamic motion model of a 3D object tracking.

The state transition is formulated by random walk:

$$p(\mathbf{X}_k|\mathbf{X}_{k-1}) = \mathcal{N}(\mathbf{X}_k; \mathbf{X}_{k-1}, \mathbf{\Psi}) \tag{3.7}$$

where, Ψ is the diagonal convariance matrix, i.e. $\mathbf{\Psi} = (\sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_\theta^2)$. Fig. 3.4 shows the dynamic motion model of a 3D object tracking between the (k-1)th frame and the kth frame, where, the Green bounding box represents the test sample generated from particle filter, while the Red one is the tracking result with maximum posteriori estimation. However, the efficiency (i.e. how many particles should be generated) and effectiveness (i.e. how well particle filter should approximate the posteriori distribution, which depends on the values in $\mathbf{\Psi}$) of the PF should be a trade off. Larger values in $\mathbf{\Psi}$ and more particles will obtain the higher accuracy, but at the cost of more storage and computation expenses. We solved this problem in following subsections.

Observation Model

In this work, we apply the low-dimensional PCA subspace representation to describe the tracked target, thus, a probabilistic interpretation of PCA should be modelled for the image observations. The probability is inversely proportional to the distance from the sample to the reference point (i.e. center) of the subspace, which includes two types of distances: (i) the distance-to-subspace: d_{to}; (ii) the distance-within-subspace: d_{within}.

The probability of d_{to} is defined as:

$$p_{d_{to}}(\mathbf{O}_k|\mathbf{X}_k) = \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, UU^\top + \varepsilon I) \tag{3.8}$$

where, $\boldsymbol{\mu}$ is the center of the subspace, I represents the identity matrix, and εI denotes the Gaussian noise.

$$p_{d_{within}}(\mathbf{O}_k|\mathbf{X}_k) = \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, U\Sigma^{-2} U^\top) \tag{3.9}$$

where, Σ represents the matrix of singular values corresponding to the columns of U.

Hence, the probability of the obervation model is as follows:

$$\begin{aligned} p(\mathbf{O}_k|\mathbf{X}_k) &= p_{d_t}(\mathbf{O}_k|\mathbf{X}_k) p_{d_w}(\mathbf{O}_k|\mathbf{X}_k) \\ &= \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, U\Sigma^{-2} U^\top)\mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, U\Sigma^{-2} U^\top) \end{aligned} \tag{3.10}$$

Moreover, the robust error norm, i.e. $\rho(x, y) = \frac{x^2}{x^2+y^2}$, rather than quadratic error norm has been applied to reduce the noise effects.

Hierarchy Tracking Strategy

In the autolanding application of UAVs, an Incremental PCA Subspace Learning-based (IPSL) visual tracker is sensitive to the large displacements or strong motions. Although the value in $\mathbf{\Psi}$ (in Eq. 3.7) can be set to be larger, and more particles can be generated to get more tolerance for these problems, however, more noises will be incorporated from those particles, and the requirements of storage and computation cost will be higher, which influences the real-time and accuracy performances. Therefore, the hierarchical tracking strategy, based on the multi-resolution structure, has been proposed to deal with these problems, as shown in Fig. 3.34, the kth frame is downsampled to create the multi-resolution structure (middle). In the motion model propagation, lower resolution textures are initially used to reject the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions. Finally, the tracking results are sent as the inputs to the on-board host computer. The IPSL^p represents the IPSL tracker in the pth level of pyramid.

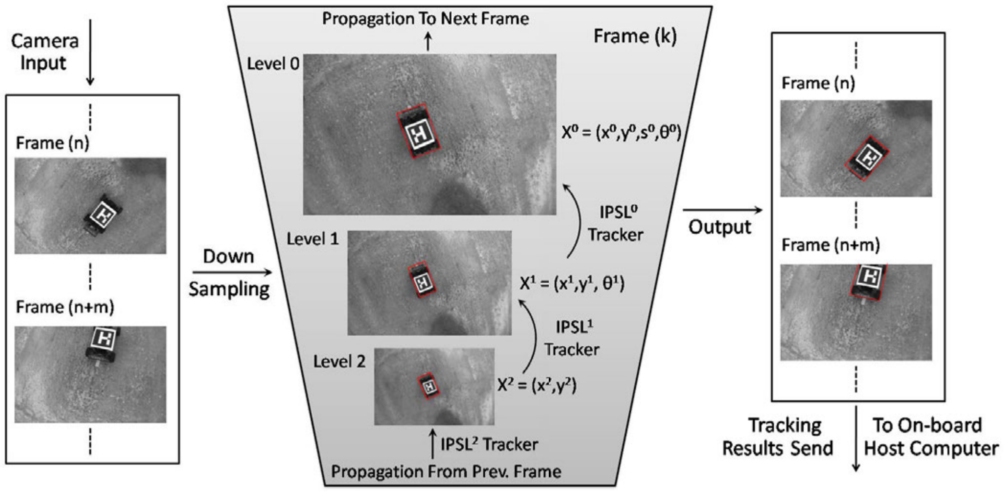


Figure 3.5: Our online learning visual tracker.

Nevertheless, there must be a compromise between the number of levels required to overcome the large inter-frame motion and the amount of visual information required to update the appearance of the target for estimating the motions. The main configurations for hierarchical visual target tracking are as follows:

Considering the image frames are downsampled by a ratio factor 2, the Number of Pyramid Levels (N_{PL}) of the multi-resolution structure are defined as a function below:

$$N_{PL} = \lfloor log_2 \frac{min\{\mathbf{T}_W, \mathbf{T}_H\}}{minSizes} \rfloor \tag{3.11}$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $\mathbf{T}_W$, $\mathbf{T}_H$ represent the width and height of target $\mathbf{T}$ in the highest resolution image (i.e. the highest-level of pyramid: 0 level), respectively. And $minSizes$ is the minimum size of the target in the lowest resolution image (i.e. the lowest-level of pyramid: p_{min} level, $p_{min} = N_{PL}$-1), in order to have enough information to estimate the motion model in that level. Thus, if the $minSizes$ is set in advance, the N_{PL} directly depends on the width/height of tracking target $\mathbf{T}$. In this application, the number of pyramid levels is $N_{PL} = 3$, then p is initialized as $p = \{2, 1, 0\}$.

Particle Filter Setup

Since the multi-resolution structure provides the computational advantage to analyze textures and update appearance model in low resolution images, and the lowest resolution image is good for estimating the location of tracking target, with the increase of resolution, more details from visual information can be used to estimate more parameters in the motion model. In this work, the motion models estimated in different resolution frames are defined as follows based on (Hartley and Zisserman, 2004):

Level 2:
$\mathbf{X}_k^2 = (x_k^2, y_k^2)$, i.e. *translation*
Level 1:
$\mathbf{X}_k^1 = (x_k^1, y_k^1, \theta_k^1)$, i.e. *translation + rotation*
Level 0:
$\mathbf{X}_k^0 = (x_k^0, y_k^0, s_k^0, \theta_k^0)$, i.e. *similarity*

Motion Model Propagation

Taking into account that the motion model estimated in each level is used as the initial estimation of motion for the next higher resolution image, the motion model propagation is defined as follows:

$$x_k^{p-1} = 2x_k^p, y_k^{p-1} = 2y_k^p$$
$$\theta_k^{p-1} = \theta_k^p \tag{3.12}$$

$$s_k^{p-1} = s_k^p$$

where, p represents the pth level of the pyramid, $p = \{p_{min}, p_{min}-1, ..., 0\} = \{N_{PL} - 1, N_{PL} - 2, ..., 0\}$, and k is the kth frame.

After finding the motion model in the kth frame, this motion model is sent as the initial estimation to the highest pyramid level of $(k+1)$th frame, as shown in Fig. 3.34:

$$\begin{aligned} x_{k+1}^{p_{min}} = \frac{x_k^0}{2^{p_{min}}}, y_{k+1}^{p_{min}} &= \frac{y_k^0}{2^{p_{min}}} \\ \theta_{k+1}^{p_{min}} &= \theta_k^0 \\ s_{k+1}^{p_{min}} &= s_k^0 \end{aligned} \tag{3.13}$$

Besides the propagation of motion models, the majority of particles will be rejected based on their particle weights in the lower resolution image, in other words, it is not necessary to generate larger number of samples to estimate the same parameters in the higher resolution image, leaving a higher tracking speed, better accuracy than a single full resolution-based voting process, the reject particle number is defined as:

$$N_R^p = \alpha^p N_P^p \tag{3.14}$$

where, α^p is the reject ratio ($0 < \alpha^p < 1$) in the pth level in the pyramid, and N_P^p is the number of particles.

In the rejected particles, the particle with maximum weight is called *Critical* particle (C_k^p). Taking the x position for example, the distance between x of C_k^p and x_k^p is denoted as *Heuristic* distance (H_k^p). Therefore, for the searching range propagation, i.e. σ_x, it is defined as:

$$\sigma_{(k,x)}^{p-1} = 2H_k^p \tag{3.15}$$

where, $\sigma_{(k,x)}^{p-1}$ is the variance of x translation in the $(p-1)$th level of the pyramid of the kth frame. And the other motion model parameters have similar propagations during the tracking process.

Block Size Recurison

The multi-block size adapting method has been utilized to update the helipad/3D object with different frequencies, i.e. a smaller block size means more frequent updates, making it faster for modelling appearance changes.

Because the image in the lowest level of the pyramid has fewer texture information, thus, the recurison of block size (N_B) is given as below:

$$N_B^{p-1} = \lfloor \frac{N_B^p}{log_2(1+p)} \rfloor \tag{3.16}$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, p represents the pth level in the pyramid, and k is the kth frame.

All the approaches introduced in this section are integrated to ensure higher accuracy and real-time performance of the helipad or 3D object tracking from VTOL UAVs. The flight tests are discussed in the section below.

3.1.5. Visual Tracking Evaluation

In this section, we have compared our visual tracker with ground truth databases in two different UAV autolanding flight tasks. The Robot Operating System (ROS) framework has been used to manage and process image data.

Ground Truth Collections

Ground truth databases have been applied to analyze the performance of our visual tracker. Figure 3.6 shows the reference points, i.e. red cross, of ground truth, which have been zoomed in and clicked by the mouse to obtain the location of each point. The center location, rotation and area of these two different helipads can be calculated frame-to-frame based on these lablled reference point locations.

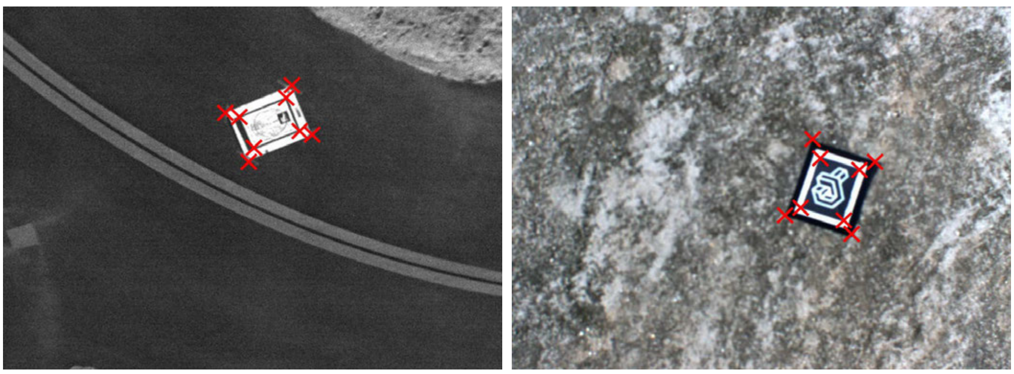
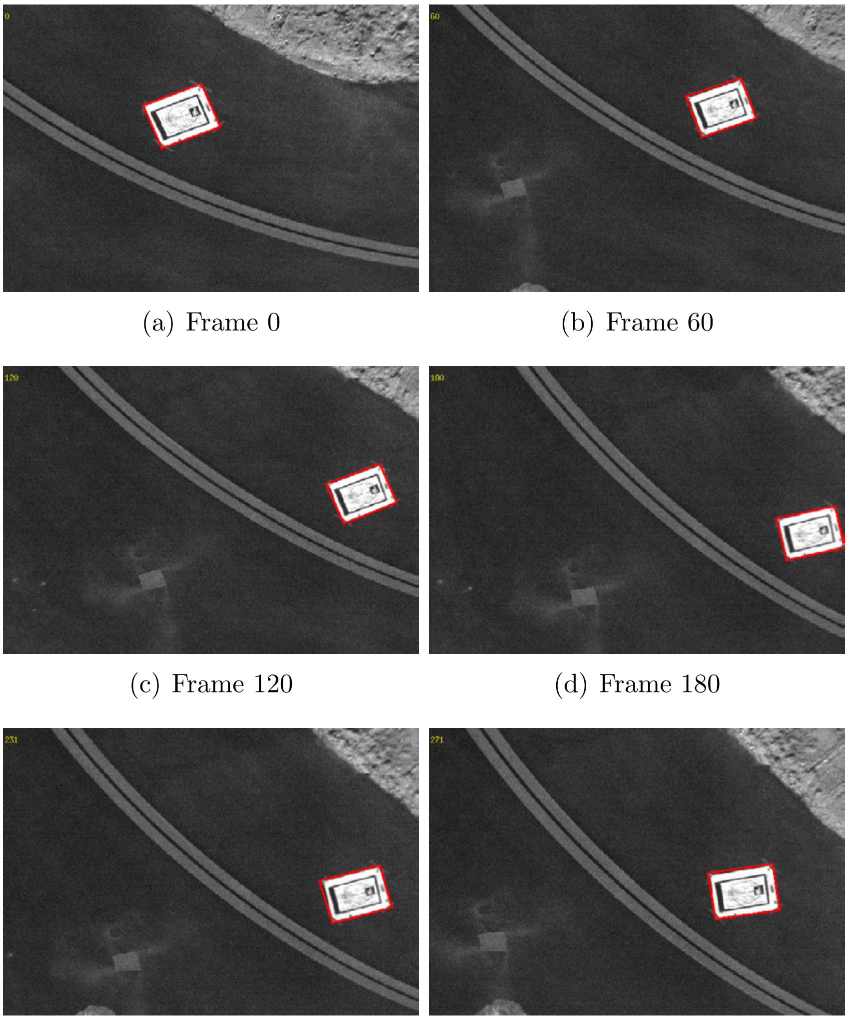


Figure 3.6: The reference points of ground truth.

Comparisons with Ground Truth

Test 1: In this test, the vision estimation contains three main challenging factors: (I) strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) rapid pose variation; (III) illumination variation. Some tracking results in Test 1 have been shown in Figure 3.12. And the comparisons between the vision estimation with related ground truth have been shown in Figure 3.8 to 3.11, where, the average RMSE errors of location in X and Y, rotation angle and area are 2 pixels, 3 pixels, 2 degrees and 133 pixel^2, respectively.

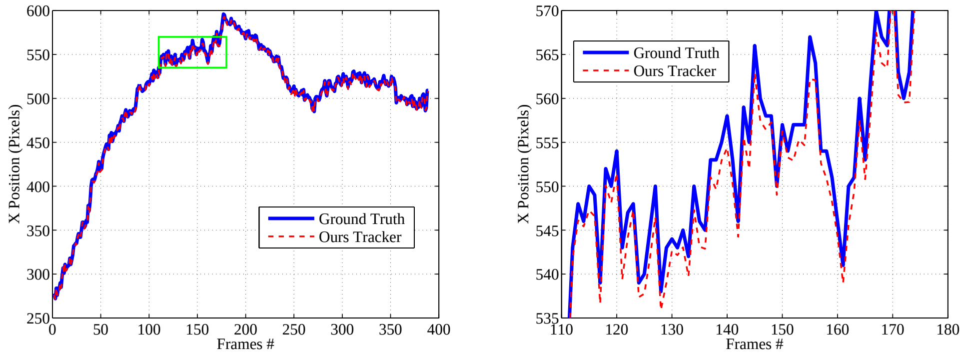


(a) Frame 0 (b) Frame 60

(c) Frame 120 (d) Frame 180

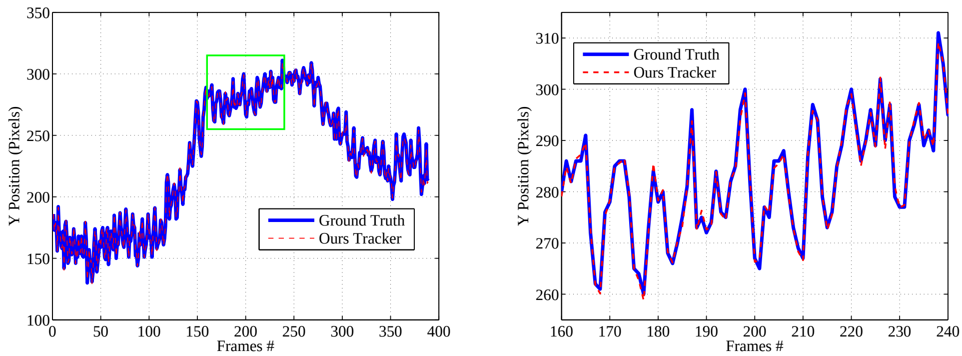
(e) Frame 231 (f) Frame 271

Figure 3.7: Some tracking results with our visual tracker in Test 1.



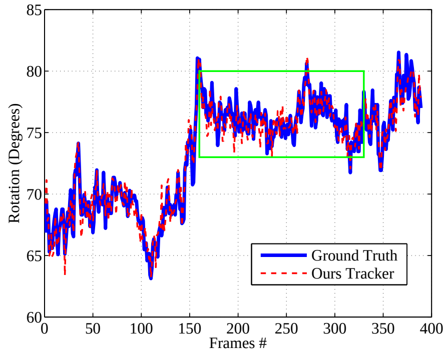
(a) X position comparison (b) Enlarged result from green rectangle

Figure 3.8: Comparsion of estimated X position with its ground truth in Test 1.

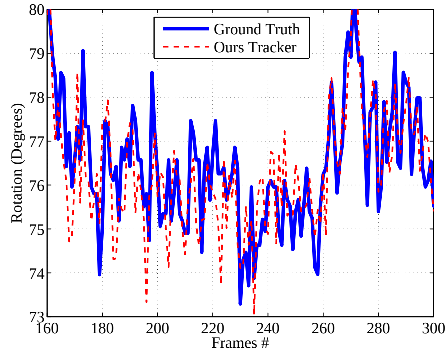


(a) Y position comparison (b) Enlarged Y position comparison

Figure 3.9: Comparsion of estimated Y position with its ground truth in Test 1.

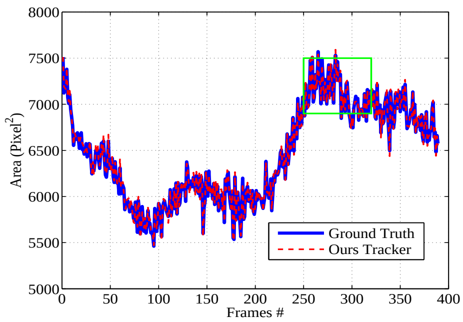


(a) Rotation angle comparison

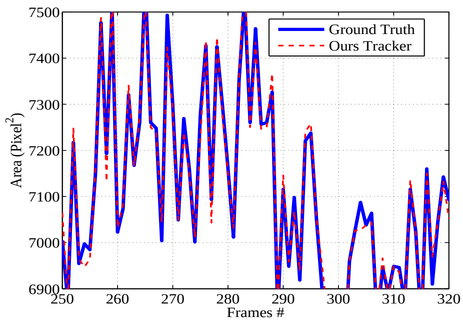


(b) Enlarged rotation angle comparison

Figure 3.10: Comparsion of estimated rotation with its ground truth in Test 1.



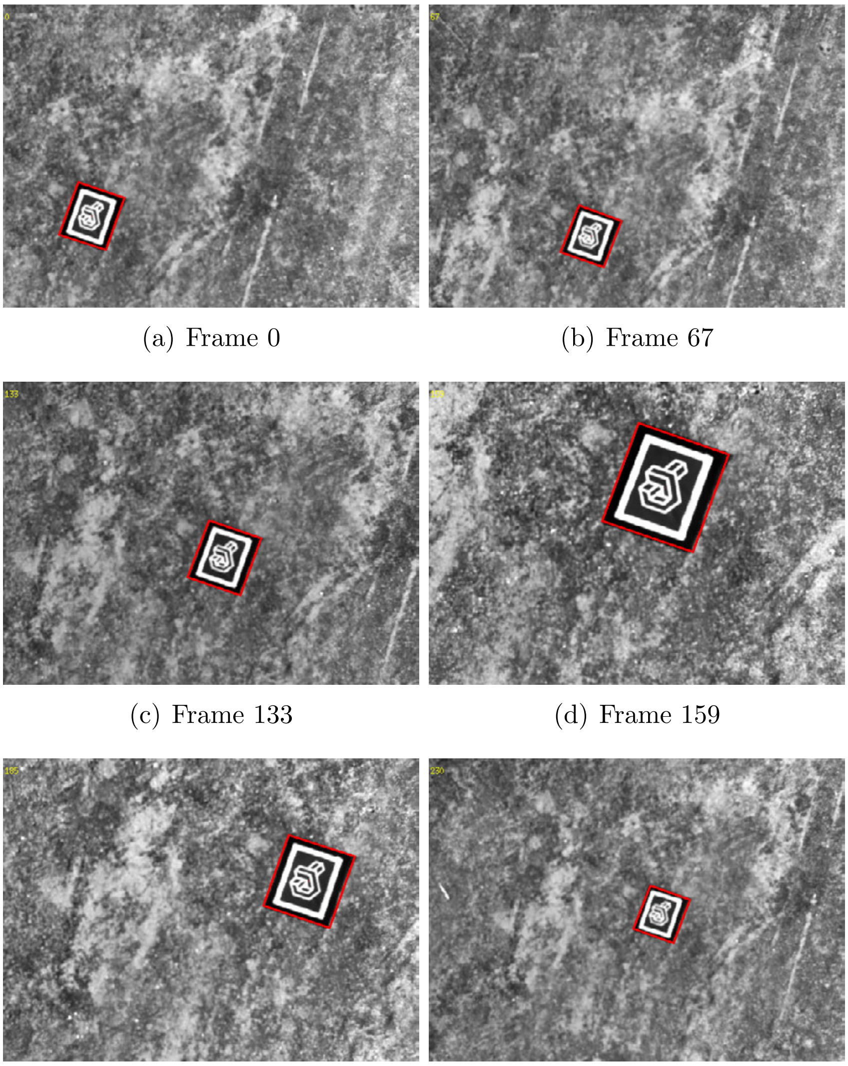
(a) Area comparison



(b) Enlarged area comparison

Figure 3.11: Comparsion of estimated area with its ground truth in Test 1.

Test 2: In this test, the tracking consists of four main challenging factors: (I) strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) scale change; (III) illumination variation; (IV) rapid pose variation. Some tracking results in the Test 2 have been shown in Figure 3.12. The comparisons between tracking result with ground truth have been shown in Figure 3.13 to 3.16, where, the average RMSE errors of location in X and Y, rotation angle and area are 3 pixels, 4 pixels, 3 degrees and 235 pixel^2, respectively.

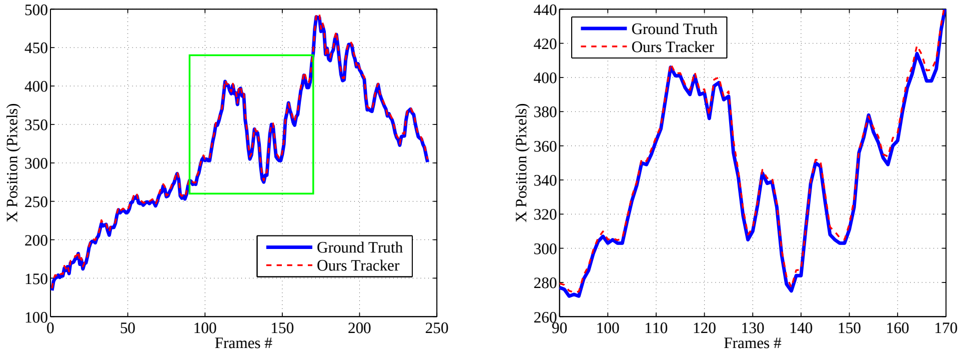


(a) Frame 0 (b) Frame 67

(c) Frame 133 (d) Frame 159

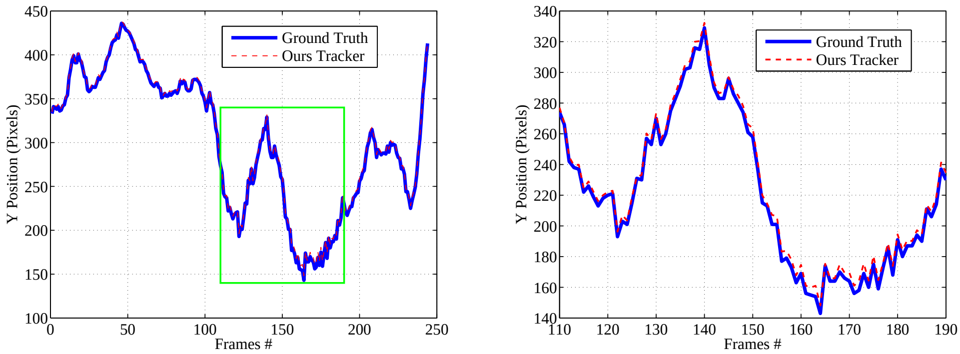
(e) Frame 185 (f) Frame 230

Figure 3.12: Some tracking results using our visual tracker in Test 2.



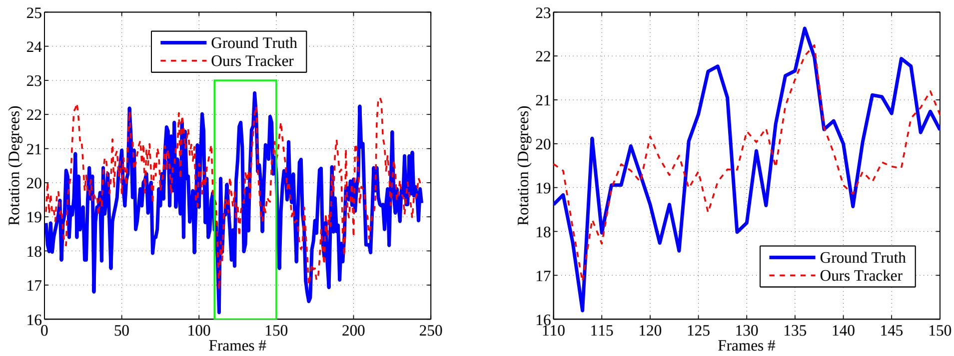
(a) X position comparison (b) Enlarged result from green rectangle

Figure 3.13: Comparsion of estimated X position with ground truth in Test 2.



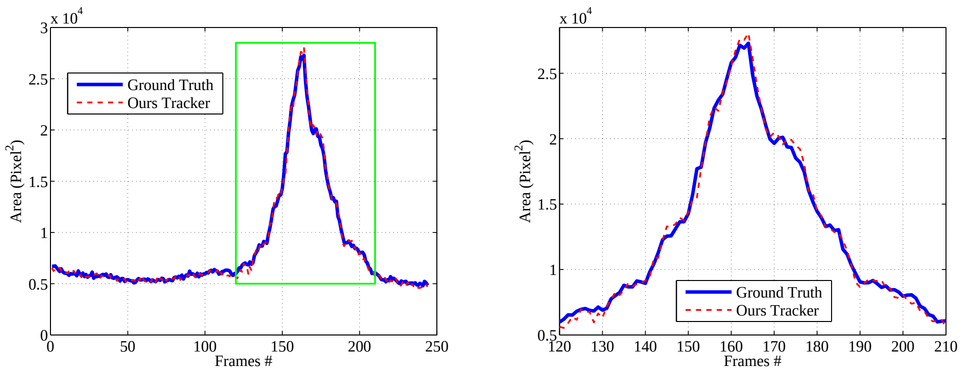
(a) Y position comparison (b) Enlarged Y position comparison

Figure 3.14: Comparsion of estimated Y position with ground truth in Test 2.



(a) Rotation angle comparison (b) Enlarged rotation angle comparison

Figure 3.15: Comparsion of estimated rotation with ground truth in Test 2.



(a) Area comparison (b) Enlarged area comparison

Figure 3.16: Comparsion of estimated area with ground truth in Test 2.

3.1.6. Real Flight Tests and Discussions

In this section, the real indoor and outdoor UAV flights have been presented and discussed. The captured input image resolution is set to 376×240 pixels, which significantly increased the image processing speed (average rate calculated by **rostopic hz** [9]: > 20) and still maintaining sufficient resolution for autonomous landing application. And the tracking performance of our proposed algorithm has been evaluated with the ground truth datasets collected from those real UAV flight tests.

Ground Truth Collections

Considering UAV flights in both indoor and outdoor environments, and taking into account that an on-board IMU sensor can not provide the stable pose estimation because of its measurement drifts. As we have proved in (Olivares-Mendez et al., 2014), an Augmented Reality (AR) library, i.e. ArUco (Garrido-Jurado et al., 2014), has been selected and applied to obtain the ground truth data of 3D position and heading angle of UAV. Note: the ArUco markers used in these tests are randomly selected, and tracking area size of helipad or 3D object is different from the ArUco size, as shown in the figure below.

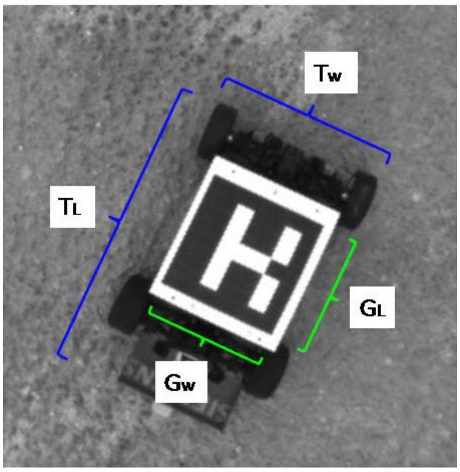


Figure 3.17: The differences between tracking target size (T_L, T_W) and the AR marker scale (G_L, G_W).

[9]The **rostopic** is a command-line tool for displaying debug information related to ROS topics. The **rostopic hz** command outputs the publishing rate of a specified topic.

Real Indoor and Outdoor UAV Flights

Test 1. Helipad-based Indoor Test: As shown in Fig. 3.18, the quadrotor Asctec Pelican is hovering above the helipad to test the tracking performances using our new designed monocular system in an indoor environment, i.e. gym (GPS coordinates: 40.439648, -3.688431).

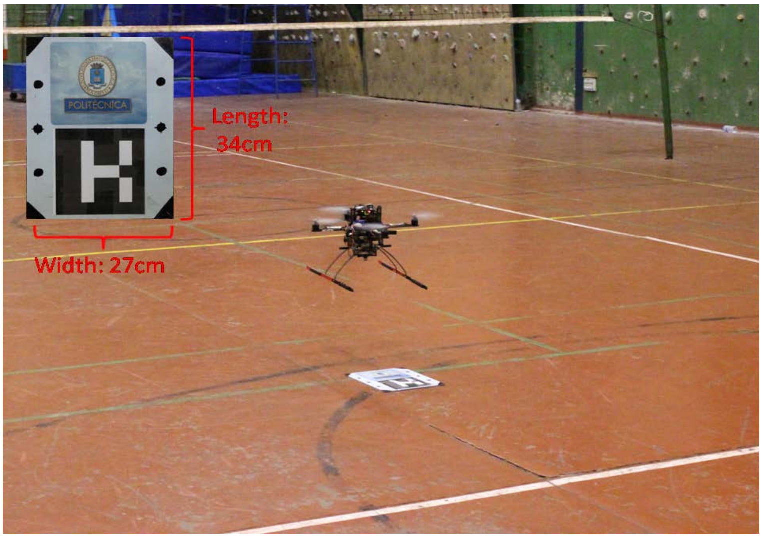


Figure 3.18: Helipad-based *indoor* UAV test.

And Fig. 3.19(a) shows the 3D trajectory of the VTOL UAV indoor flight, which is reconstructed with Ground Truth data, the green and red points represent the start and end tracking positions. The size of helipad is 34cm$\times$27cm, as the left-top corner in Fig. 3.18.

In addition, the comparisons between onboard visual tracking estimations (red curve) and ground truths (blue curve) have been shown in Fig. 3.19(b), 3.19(c), 3.19(d) and 3.19(e), where, the vision estimations are well-matched with the ground truths, without many obvious outliers. The indoor average RMSE errors of x position, y position, altitude (z), UAV heading angle are 2.73cm, 2.98cm, 2.26cm and 3.12 degrees, respectively.

From the above average RMSE errors, we can find that our new visual algorithm can accurately track the target during the UAV indoor autolanding flights even with the motion blur caused by mechanical vibrations.

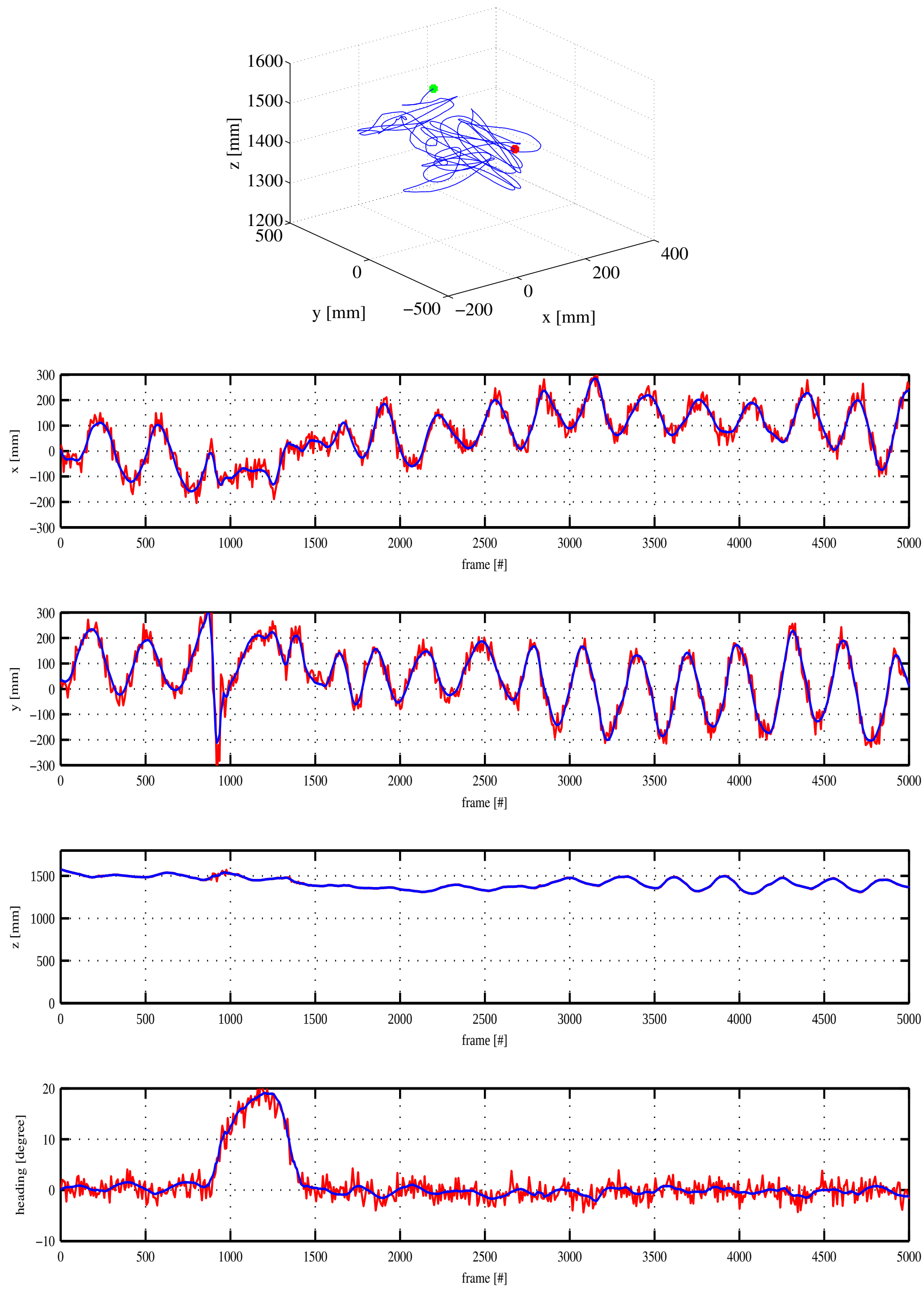


Figure 3.19: 3D trajectory and comparisons between visual helipad tracking estimations (red curve) and ground truths (blue curve) in indoor UAV flight test.

Test 2. Helipad-based Outdoor Test: In this test, a new helipad with the same size as the one used in the test 1 is utilized in outdoor environment, i.e. tennis court (GPS coordinates: 40.439478, -3.688894), as shown in Fig. 3.20.

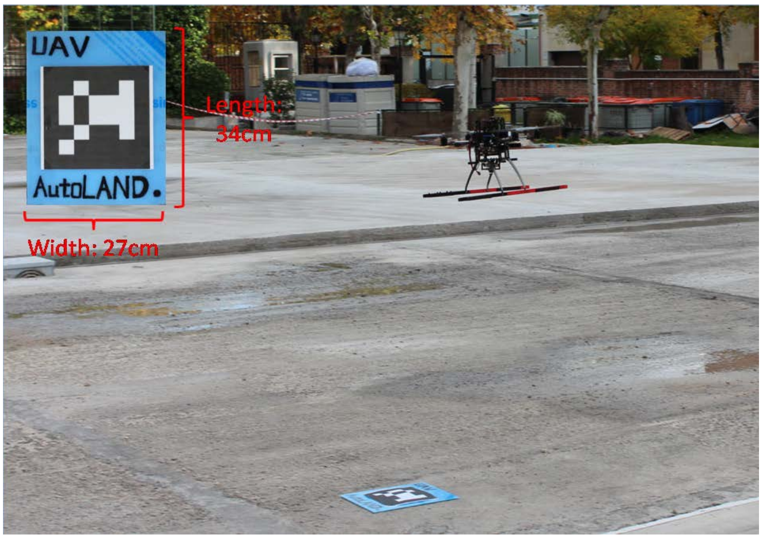


Figure 3.20: Helipad-based *outdoor* UAV test.

Fig. 3.21(a) provides the 3D trajectory during the UAV outdoor flight. The details of the UAV flight performances have been shown in Fig. 3.21(b), 3.21(c), 3.21(d) and 3.21(e), where, the visual tracking results are basically coincident with the ground truth data, however, the performances are worse compared to the indoor flight, mainly because of the wind disturbances et al.

In addition, from the Fig. 3.21(d), we can see that the UAV quickly reduce its altitude between Frame 3940 and Frame 4000, but our vision algorithm can track the helipad robustly and adaptively. The average RMSE errors of x position, y position, altitude, UAV heading angle are 3.25cm, 3.31cm, 3.06cm and 4.43 degrees during the outdoor tracking helipad tests, respectively.

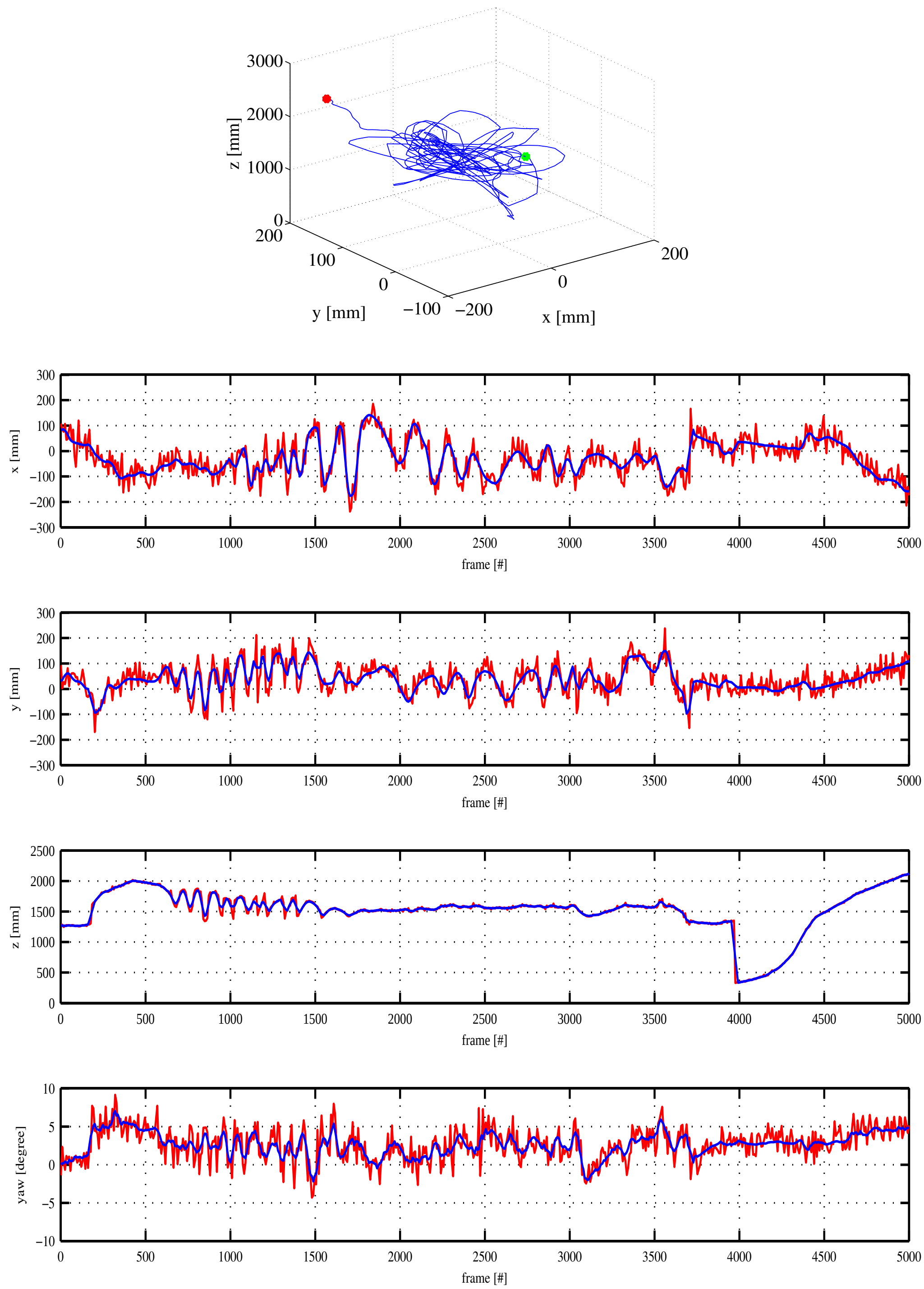


Figure 3.21: 3D trajectory and comparisons between visual helipad tracking estimations (red) and ground truths (blue) in outdoor UAV flight test 1.

Test 3. 3D Object-based Outdoor Test: In the real world, most objects are showing three dimensions (3D), for a 3D object, its appearance also varies from the different camera angle of views, therefore, a 3D target, i.e. a car (size: 40cm×25cm×12cm), is used to demonstrate the robustness of online learning method using our *model-free* tracker in outdoor environment, as shown in Fig. 3.22.

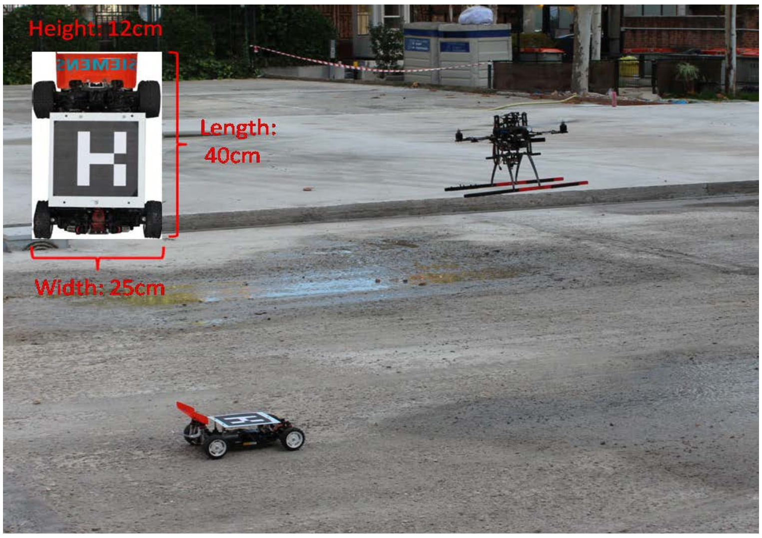


Figure 3.22: 3D object-based *outdoor* UAV test.

Fig. 3.23(a) presents the 3D trajectory of the UAV flight in the process of tracking a car. And the tracking results, i.e. x position, y position, altitude (z) and heading angle have been shown in Fig. 3.23(b), 3.23(c), 3.23(d) and 3.23(e). Compared to the indoor test, UAV had to fly against the wind influences, and therefore its performance is also worse. However, the visual algorithm achieved similar performances as the ones in the outdoor tracking helipad. In outdoor 3D car tracking flights, the average RMSE errors of x position, y position, altitude, UAV heading angle are 3.65cm, 3.72cm, 3.39cm and 4.99 degrees, respectively.

From these two outdoor flight tests, we can also conclude that the novel vision algorithm shows robustness and good adaptation for outdoor 2D or 3D target tracking from VTOL UAV.

And all UAV landing flights (total number of tests: 27) demonstrated that this newly proposed visual algorithm has obtained the accurate tracking estimations, which can be stably sent to the host computer on-board UAV to close the control loop.

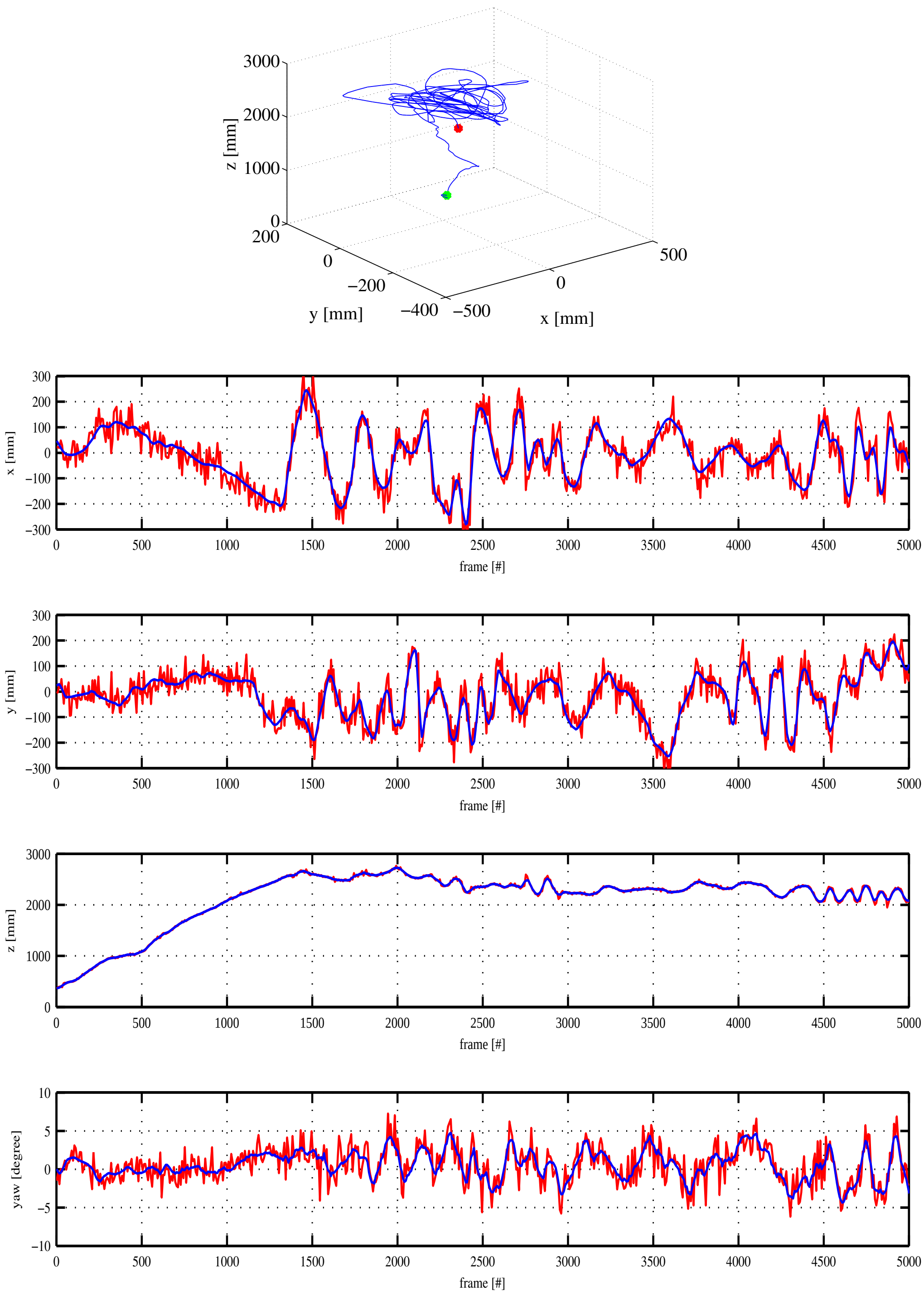
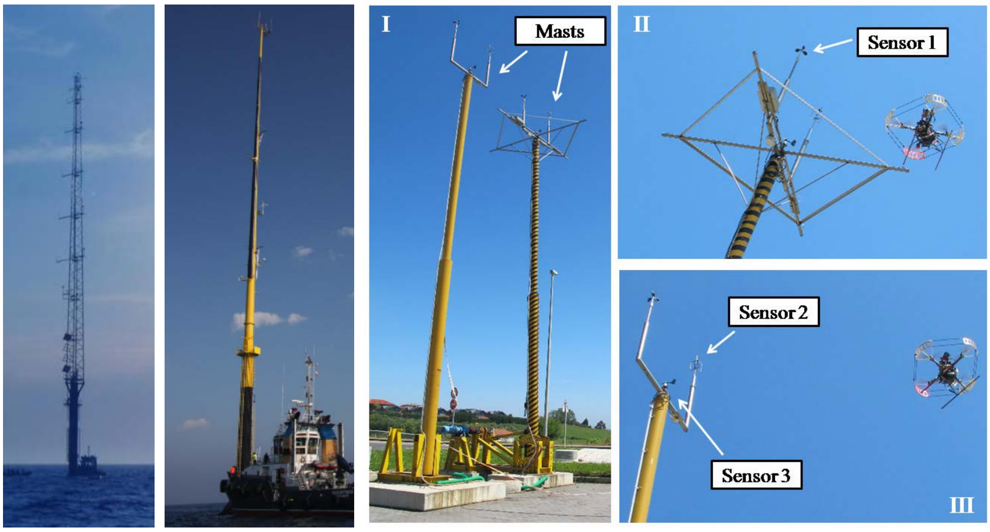


Figure 3.23: 3D trajectory and comparisons between visual 3D object tracking estimations (red) and ground truths (blue) in the outdoor UAV flight test 2.

3.2. Offshore Floating Platform Inspection

In this section, a civilian UAV application to inspect the Offshore Floating Platform (OFP) has been presented. The developed visual tracking algorithm should have the capability to detect and track different sensors, e.g. anemometer, instrumentation and signaling (antenas) systems installed in the OFP. This application aims to replace the human engineer, who should climb on the OFP, to remotely check those sensors, thereby reducin the risks, time, equipments and inspection costs.



(a) OFP 1 (b) OFP 2 (c) Test platform and UAV inspection test

Figure 3.24: Different offshore floating platforms and UAV inspection test.

In literature, many visual trackers have obtained the promising tracking performances for specific or arbitrary objects. (Martinez et al., 2013) utilized the direct method (i.e. directly represent the target using all the pixels in the selected and fixed image template) to track target. (Mejias et al., 2006b) selected the target model on the first frame using Lucas-Kanade tracker. (Mei and Ling, 2011) modeled the target as a sparse linear combination of target and trival templates, i.e. l_1- tracker. But these feature and direct method-based algorithms are unstable to track the targets with obvious appearance changes and other challenging factors, and they did not use the valuable background information to improve the tracking performances (Wang et al., 2011).

While discriminative algorithms (also called visual *tracking-by-detection* methods (Avidan, 2004)) employ an adaptive binary classifier to separate the target from background during the frame-to-frame tracking ((Zhang et al., 2013), (Collins et al., 2005), (Grabner and Bischof, 2006), (Kalal et al., 2012)). The important stage for discriminative visual tracking is *classifier update* using online selected features, many works update their classifiers using only one positive sample and some surrounding negative samples, but this method often causes the tracking drift (failure) problem becasue of noisy and misaligned samples. Recently, both multiple positive samples and negative samples are used to update classifier, the location of sample with maximum classifier score (i.e. most *correct or important* sample) is the new target location at current frame, this method can even solve significant appearance changes in cluttered background. However, these discriminative algorithms did not take into account the information about the importance of the sample, i.e. classifier score for each sample, to improve the tracking stability and accuracy.

Although many discriminative approaches often achieve superior tracking results, and tolerate the motions in the range of search radius, but in the tracking on-board UAV for control and navigation, we have observed that discriminative visual tracking algorithms are sensitive to strong motions (e.g. onboard mechanical vibration) or large displacements over time. Therefore, we adopt Multi-Resolution (MR) strategy to cope with these problems. Additionlly, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAV with Ground Control Station (GCS). Thus, this section proposes to address the UAV tracking problem using discriminative visual tracker with compressive sensing technique, i.e. Compressive Tracking (CT) (Zhang et al., 2012), and extending it in a hierarchy-based framework with different image resolutions and adaptive classifiers applied to estimate the motion models in these resolutions: the Adaptive Multi-Classifier Multi-Resolution framework (AMCMR). Using this strategy, especially in the Adaptive Multi-Classifier structure, the importance of the sample has been used to reject samples, i.e. the lower resolution features are initially applied in rejecting the majority of samples (called Rejected Samples (RS)) at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, thereby ensuring the real-time performance and higher accuracy.

To the author's best knowledge, this framework has not been presented for solving the online learning and tracking freewill 2D or 3D target problems in the UAV. The proposed AMCMR-CT framework runs at real-time and performs favorably on challenging public and aerial image sequences in terms of efficiency, accuracy and robustness. For this reason, the intention of this section is also to expand this discriminative method-based framework in more real-time UAV control and navigation applications.

3.2.1. Discriminative Visual Tracking

Discriminative Visual Tracking (DVT) takes the tracking problem as a binary classification task to separate target from its surrounding background, as shown in Figure 3.25, the visual tracker is tracking a sensor installed on the OFP, i.e. anemometer, at real-time frame rates on-board UAV, the image patch with green rectangle represents the positive sample, while the one with blue rectangle is the negative sample.

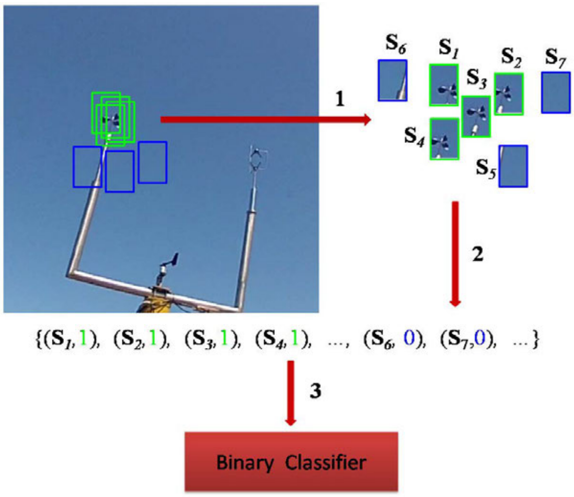


Figure 3.25: Discriminative visual tracking.

It trains a classifier in an online method using positive and negative samples extracted from the current frame. When the next frame is coming, the samples around the old target location are extracted at this frame, and then the afore-trained classifier is applied to these samples. The location of the sample with the maximum classifier score is the new target location at this frame. A generic process of the DVT is presented in the Algorithm 1.

In the Algorithm 1, the parameter α is called search radius, which is used to extract the test samples in the k-th frame, the parameter β is the radius applied for extracting the positive samples, while the parameter γ and δ are the inner and outer radii, which are used to extract the negative samples.

Algorithm 1 Discriminative Visual Tracking.

Input: the k-th frame

1. Extract a set of image samples:
$\mathcal{S}^{\alpha}$=$\{\mathbf{S} | \|\mathbf{l}(\mathbf{S}) - \mathbf{l}_{k-1}\| < \alpha\}$, where, $\mathbf{l}_{k-1}$ is the target location at (k-1)th frame, and online select feature vectors.
2. Use classifier trained in the (k-1)th frame to these feature vectors and find the target location $\mathbf{l}_k$ with the maximum classifier score.
3. Extract two sets of image samples:
$\mathcal{S}^{\beta}$=$\{\mathbf{S} | \|\mathbf{l}(\mathbf{S}) - \mathbf{l}_{k}\| < \beta\}$ and $\mathcal{S}^{\gamma,\delta}$=$\{\mathbf{S} | \gamma < \|\mathbf{l}(\mathbf{S}) - \mathbf{l}_{k}\| < \delta\}$, where, $\beta < \gamma < \delta$.
4. Online select the feature using these two sets of samples, and update the classifier.

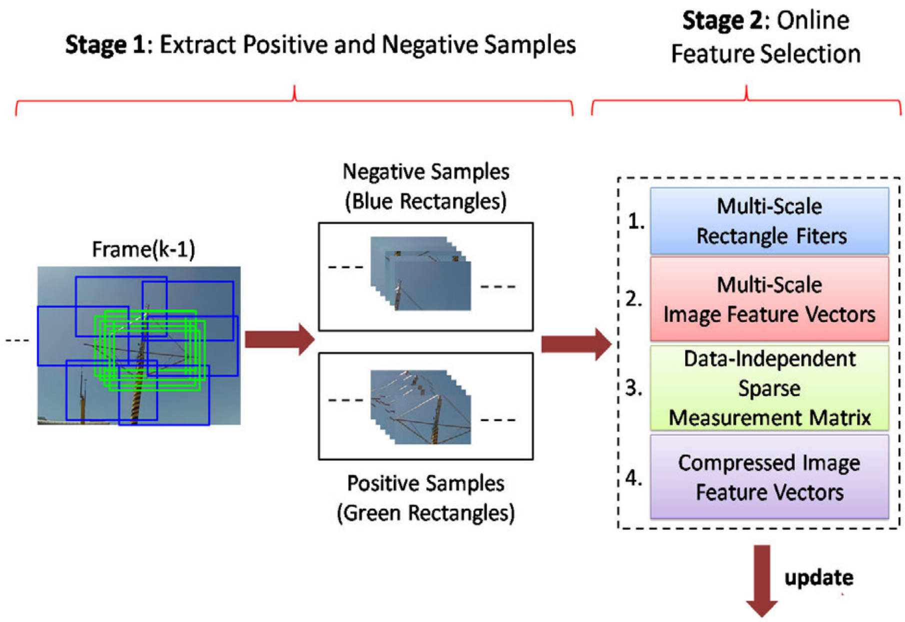
Output: (1) Target location $\mathbf{l}_k$
(2) Classifier trained in the kth frame

Compressive Tracking

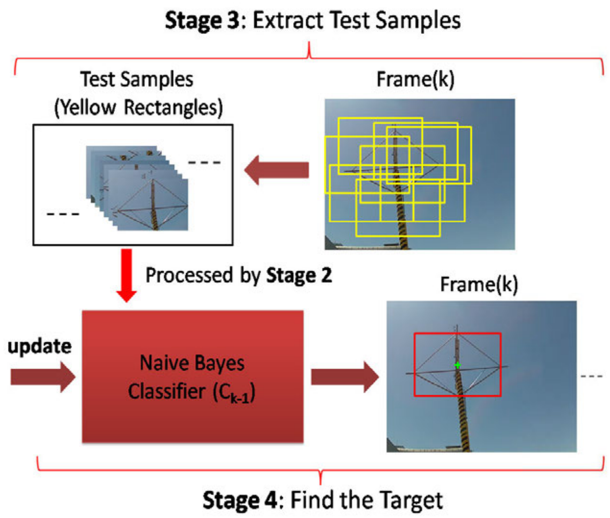
As introduced in above sections, updating classifier depends on online selecting features, (Collins et al., 2005) demonstrated that the most discriminative features can be learned online. In this section, we have adopted an effective and efficient DVT-based Compressive Tracking (CT) algorithm proposed by (Zhang et al., 2012), which selects the features in the compressed domain, as shown in the Figure 3.26, the adaptive classifier is updated with compressed image feature vectors in the (k-1)th frame, and then applied to estimate the target location in the kth frame. The CT runs at real-time frame rates and outperforms the existing state-of-art visual trackers. In addition, the positive and negative samples are compressed with the same data-independent sparse measurement matrix discriminated by a simple naive Bayes classifier.

The CT mainly consists of three stages, as shown in Figure 3.26:

- *Selecting*: it means that selecting image feature vector from each positive or negative sample croped from input image frame;
- *Compressing*: compress these feature vectors to the low-dimensional feature vectors with data-independent sparse measurement matrix;
- *Updating*: on-line update the naive Bayes classifier with these low-dimensional feature vectors.



(a) Stage 1 and 2.



(b) Stage 3.

Figure 3.26: Real-time Compressive Tracking (CT) algorithm.

1. Selecting

As the Part 1 and 2 in the dash line rectangle, for each sample $\mathbf{S} \in \mathbb{R}^{\underline{w}\times\underline{h}}$, it is processed with a set of rectangle filters at multiple scales, $\{h_{1,1}, ..., h_{\underline{w},\underline{h}}\}$ defines as

$$h_{i,j}(x, y) = \begin{cases} 1, & 1 \leqslant x \leqslant i, 1 \leqslant y \leqslant j \\ 0, & otherwise \end{cases} \tag{3.17}$$

where i and j are the width and height of a rectangle filter, respectively. Then, each filtered image is represented as a colum vector in $\mathbb{R}^{\underline{w}\times\underline{h}}$, and then these vectors are concatenated as a very high-dimensional multi-scale image feature vector $\mathbf{x} = (x_1, ..., x_m)^T \in \mathbb{R}^m$, where $m = (\underline{w} \times \underline{h})^2$. The dimensionality m is typically in the order of 10^6 to 10^{10}.

2. Compressing

After obtained the high-dimensional multi-scale image feature vector $\mathbf{x} \in \mathbb{R}^m$, as shown in Part 3 and 4, the random matrix $R \in \mathbb{R}^{n\times m}$ is used to compress it to a lower-dimensional vector $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{x} \xrightarrow{\times R} \mathbf{v}, \quad i.e. \mathbf{v} = R\mathbf{x} \tag{3.18}$$

where $n \ll m$. In the Equation 3.18, a very sparse random Gaussian matrix $R \in \mathbb{R}^{n\times m}$, where $r_{ij} \sim N(0, 1)$, with entries is defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 & with\, possibility \ \frac{1}{2s} \\ 0 & with\, possibility \ 1 - \frac{1}{s} \\ -1 & with\, possibility \ \frac{1}{2s} \end{cases} \tag{3.19}$$

where, when $s = 3$, it is very sparse where two thirds of the computation can be avoided.

From Equation 3.19, it is only necessary to store the nonzero entries in R and the positions of rectangle filters in an input image corresponding to the nonzero entries in each row of R. Then, $\mathbf{v}$ can be efficiently computed by using R to sparsely measure the rectangle features which can be efficiently computed using the integral image method, where, the compressive features compute the relative intensity difference in a way similar to the generalized Haar-like features. The compressive sensing theories ensure that the extracted features of CT algorithm preserve almost all the information of the original image.

3. Updating

For each sample $\mathbf{S} \in \mathbb{R}^{\underline{w} \times \underline{h}}$, its low-dimensional representation is $\mathbf{v} = \{v_1, ..., v_n\} \in \mathbb{R}^n$ with $m \gg n$. All elements in $\mathbf{v}$ are assumed to be independently distributed and modeled with a naive Bayes classifier (Ng and Jordan, 2002),

$$\begin{aligned} H(\mathbf{v}) &= \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}\right) \\ &= \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \end{aligned} \tag{3.20}$$

where, uniform prior are assumed: $p(y=1) = p(y=0)$, and $y \in \{0,1\}$ is a binary variable which represents the sample label.

The conditional distributions $p(v_i|y=1)$ and $p(v_i|y=0)$ in the classifier $H(\mathbf{v})$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ where

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1), \quad p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0) \tag{3.21}$$

The scalar parameters in Equation (3.21) are incrementally updated

$$\begin{aligned} \mu_i^1 &\leftarrow \eta\mu_i^1 + (1-\eta)\mu^1 \\ \sigma_i^1 &\leftarrow \sqrt{\eta(\sigma_i^1)^2 + (1-\eta)(\sigma^1)^2 + \eta(1-\eta)(\mu_i^1 - \mu^1)^2} \end{aligned} \tag{3.22}$$

where, $0 < \eta < 1$ is a learning parameter,
$\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and
$\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k)$.
The update schemes for μ_i^0 and σ_i^0 have similar formations.

3.2.2. Hierarchy-based Tracking Strategy

In the UAV tracking applications, compressive tracking is sensitive to the strong motions or large displacements. Although the search radius for extracting test samples can be set to be larger, as shown in Algorithm 1, to get more tolerance for these problems, however, more test samples (noises) will be generated (introduced), which influence the real-time and accuracy performances. Therefore, Multi-Resolution (MR) approach was proposed to deal with these problems, as shown in Figure 3.34, the kth frame is downsampled to create the MR structure. In Adaptive Multi-Classifier structure, lower resolution features are initially used to reject the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions. The C^p_{k-1} represents the adaptive classifier updated in the pth level of pyramid of (k-1)-th frame. . Nevertheless, there must be a compromise between the number of levels required to overcome the large inter-frame motion and the amount of visual information required to update the adaptive multiple classifiers for estimating the motions.

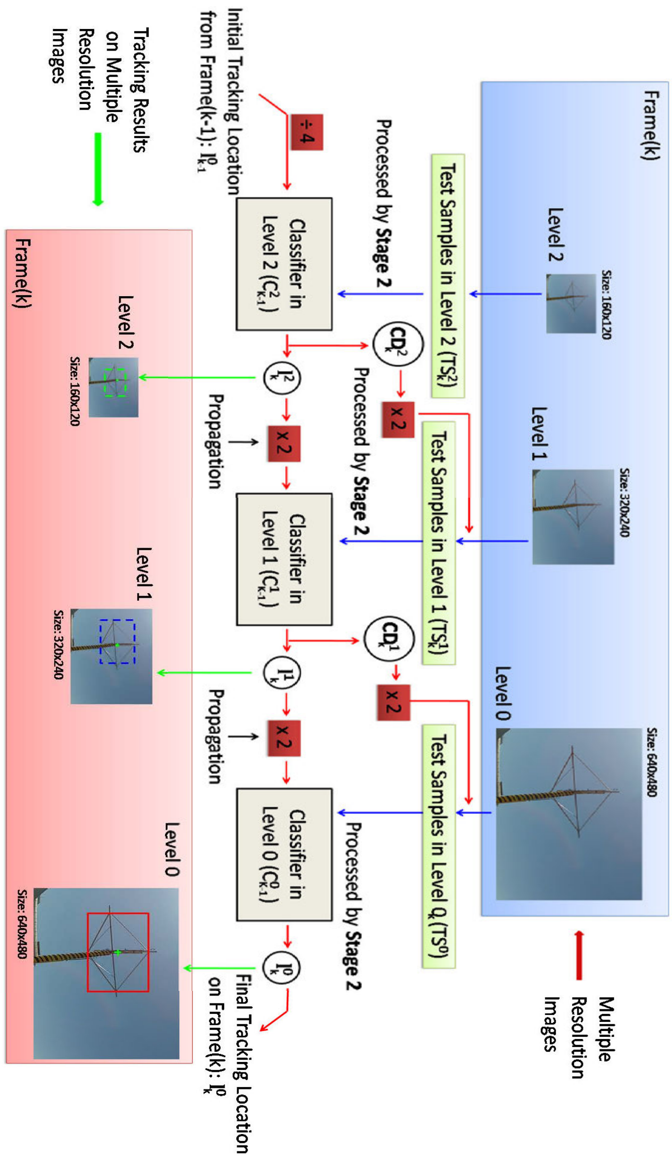


Figure 3.27: AMCMR-CT visual tracking framework.

Configurations

1. Number of Pyramid Levels (N_{PL})

Considering the images are downsampled by a ratio factor 2, the Pyramid Levels of the MR structure are defined as a function below:

$$N_{PL} = \lfloor log_2 \frac{min\{\mathbf{T}_W, \mathbf{T}_H\}}{minSizes} \rfloor \tag{3.23}$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $\mathbf{T}_W$, $\mathbf{T}_H$ represent the width and height of target $\mathbf{T}$ in the highest resolution image (i.e. the lowest-level of pyramid: 0 level), respectively. And $minSizes$ is the minimum size of target in the lowest resolution image (i.e. the highest-level of pyramid: p_{max} level, $p_{max} = N_{PL}$-1), in order to have enough information to estimate the motion model in that level. Thus, if the $minSizes$ is set in advanced, the N_{PL} directly depends on the width/height of tracking target $\mathbf{T}$.

2. Motion Model (l) Propagation

Taking into account that the motion model estimated by CT in each level is used as the initial estimation of motion for the next higher resolution image, therefore, the motion model propagation is defined as follows:

$$\mathbf{l}_k^{p-1} = 2\mathbf{l}_k^p \tag{3.24}$$

where, p represents the pth level of the pyramid, $p = \{p_{max}, p_{max}-1, ..., 0\} = \{N_{PL}-1, N_{PL}-2, ..., 0\}$, and k is the kth frame.

3. Number of Rejected Sample (N_{RS})

In the Adaptive Multi-Classifier structure, since the MR approach provides the computational advantage to analyze features and update classifiers in low resolution images, the majority of samples will be rejected based on their classifier scores (i.e. sample importances) in the lower resolution image, leaving a fewer number of samples to be processed in the higher resolution image. Thus, the AMC structure obtains higher tracking speed, better accuracy than a single full resolution-based adaptive classifier, the rejected sample number is defined as:

$$N_{RS}^p = \xi^p N_S^p \tag{3.25}$$

where, p represents the pth level in the pyramid, ξ^p is the reject ratio ($0 < \xi^p < 1$), and N_S^p is the number of test samples. Especially, the sample with maximum score in the rejected samples is the Critical Sample (CS_k^p).

4. Search Radius Propagation

The euclidean distance between the location of CS_k^p and $\mathbf{l}_k^p$ is the Critical Distance (CD_k^p), which will be propagated to next higher resolution image as the search radius:

$$\alpha_k^{p-1} = 2CD_k^p \tag{3.26}$$

where, p represents the pth level in the pyramid, and k is the kth frame.

3.2.3. Experiment Evaluation

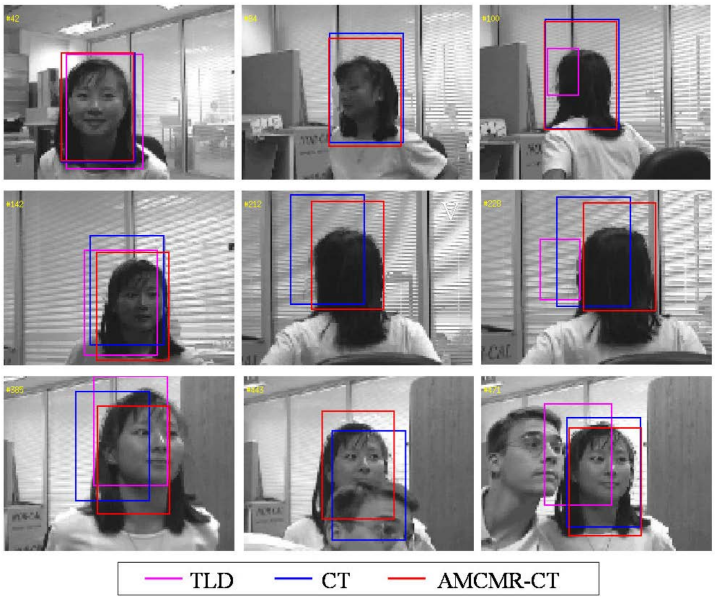
In this subsection, we compared our AMCMR-CT tracker with 2 latest state-of-art trackers (TLD((Kalal et al., 2012)) and CT) on two different types of challenging image data: (I) Public Image Datasets, which are used to test and compare visual algorithms in computer vision community; (II) Aerial Image Databases, which are captured from our former vision-based UAV inspection projects. The evaluation measure ((Bai and Li, 2012)) is the Center Location Error (CLE), which is defined as the Euclidean distance from the detected target center to the (manually labeled) ground truth center at each frame.

Test 1: Comparison with Public Image Datasets

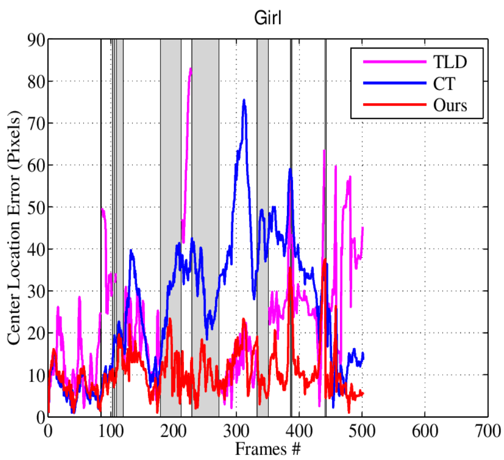
The most challenging public image datasets have been applied. The total number of evaluated frames is more than 10^4. Here, the tracking performance of *Girl* image dataset released by Birchfield[10] is shown below. This sequence contains one main challenging factor that is full 360-degree out-of-plane rotation.

During the tracking process, as shown in Figure 3.28, although TLD tracker is able to relocate on the target, it is easy to lose the target completely for many frames. For the CT, it can track its target, but the tracking drift generates easily in some frames when the target is out of plane, changing scale , jumping suddenly and occluded by other people. However, the AMCMR-CT outperforms these two trackers.

[10]http://www.ces.clemson.edu/∼stb/



(a) Tracking with different algorithms.

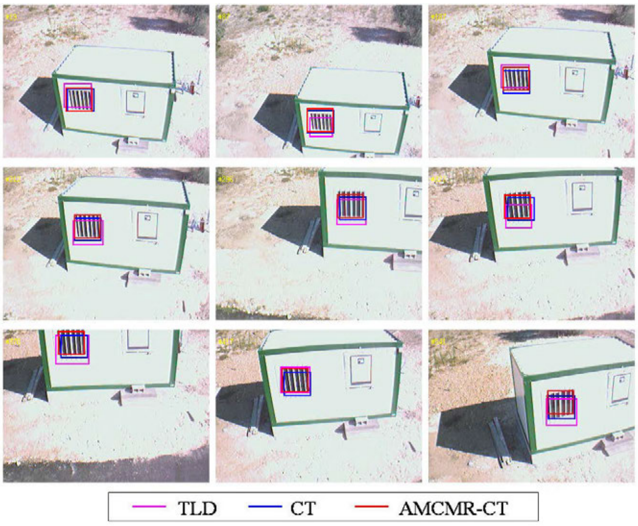


(b) The Performances of Visual Tracking.

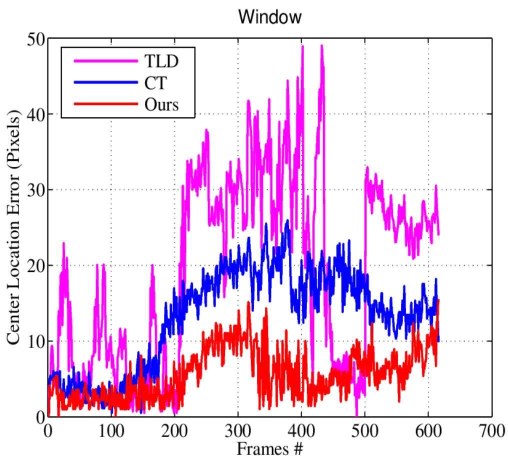
Figure 3.28: Tracking *Gril* with different algorithms and evaluations.

Test 2: Comparison with Real Flight-based Aerial Image Databases

The real flight-based aerial images captured by different UAVs are processed. The evaluated frame number is more than 8000. Here, one aerial image database for tracking *window* recorded by CVG-UPM[11] is tested. This database includes one obvious challenging factor that is the strong motion or large displacements over time.



(a) Tracking with different algorithms.



(b) The Performances of Visual Tracking.

Figure 3.29: Tracking *Window* with different algorithms and evaluations.

[11]http://www.vision4uav.com

For the tracking performances, as shown in Figure 3.29, TLD can finish its tracking task in this sequence, and sometimes relocate its target when the appearance of target is similar to the initialized target appearance, but it often misaligns its target, and the misaligned error is larger than those generated by the CT and AMCMR-CT. Although the CT can tolerate the motions to some extent based on the range of searching radius, its tracking performance is not superior to the AMCMR-CT's performance.

3.2.4. Visual Inspection Applications and Discussions

The different evaluations described in the Section 3.2.3 have shown that the proposed AMCMR-CT framework is able to track arbitrary 2D or 3D targets under different challenging conditions and obtains the better performances. In this subsection, this new algorithm is used in a real application, i.e. OMNIWORKS Project[12], to control and navigate UAV.

The input image resolution captured from onboard uEye USB3 camera (type: UI-3240CP-C-HQ) has been set to 640×480 pixels. The onboard processing computer is the AscTec Atomboard [13], the average image processing rate calculated by **rostopic hz** is 25.32. And the tracking performance of our proposed algorithm has been evaluated with the manually-generated ground truth datasets.

Tracking Performance Analysis

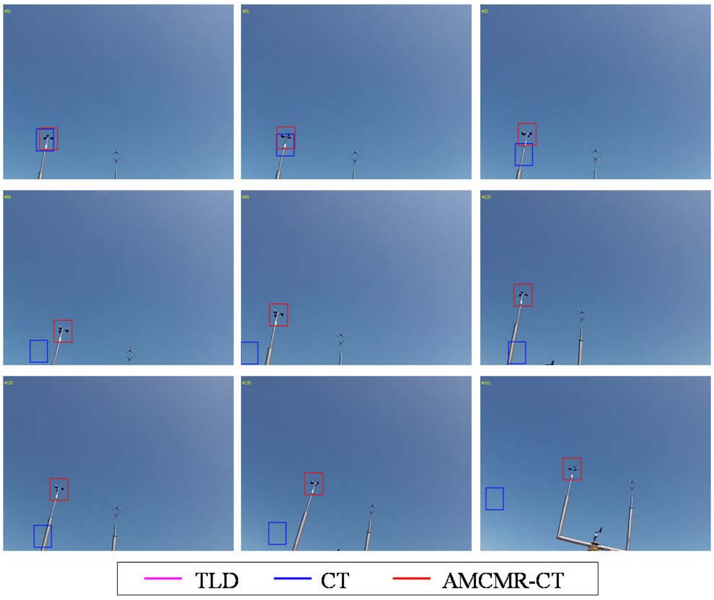
In this subsubsection, the TLD and CT trackers are applied to compare with our AMCMR-CT tracker in two different inspection tasks for OFP, the tracking targets inlcude: (I) Anemometer; (II) Moving mast.

1. Anemometer Tracking

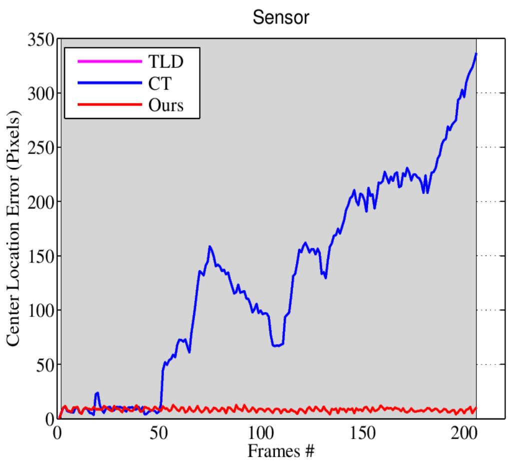
The anemometer is used to measure the speed of wind. The main challenging factors for anemometer tracking in UAV are obvious appearance change and strong motions.

In the anemometer tracking, the TLD completely lose its target from the 2nd frame. Although the CT tracks its target well at the beginning, the drift problem occurs from the 51th frame, and then lose its target, as shown in Fig. 3.30. While the AMCMR-CT obtains the best performance.

[12]http://www.echord.info/wikis/website/omniworks
[13]http://www.asctec.de/asctec-atomboard/



(a) Tracking with different algorithms.



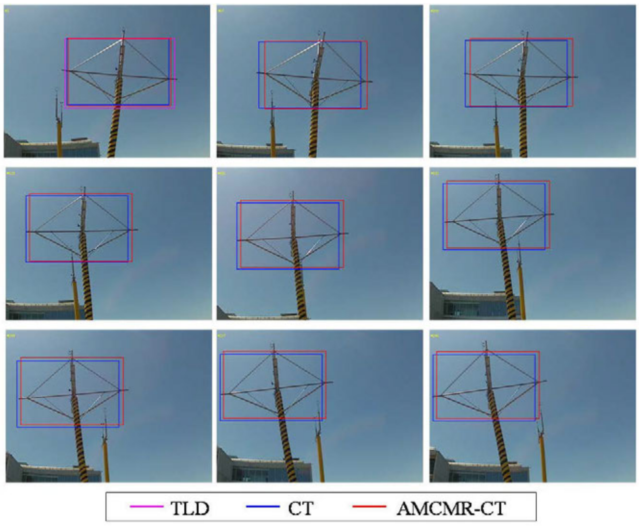
(b) The Performances of Visual Tracking.

Figure 3.30: Tracking *Anemometer* with different algorithms and evaluations.

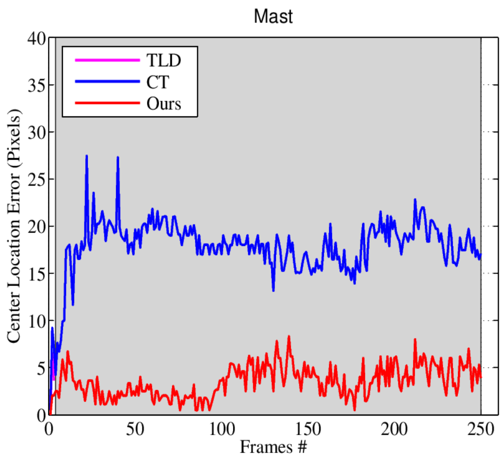
2. Moving Mast Tracking

The mast is applied to fix the different sensors. This aerial image dataset includes one main challenging factor that is strong motions or large displacements.

For the moving mast tracking, as shown in Fig. 3.31, from the 4th frame, the TLD completely lose its target again. The CT and AMCMR-CT both can finish their mast tracking, however, the CT still has the drift problem, its tracking accuracy is worse than the one AMCMR-CT obtained.



(a) Tracking with different algorithms.



(b) The Performances of Visual Tracking.

Figure 3.31: Tracking *Moving Mast* with different algorithms and evaluations.

The Center Location Error (CLE) (Unit: Pixels) for each image sequence is shown in the below Table:

Sequences-Trackers	TLD	CT	AMCMR-CT
Girl	NaN	25	**11**
Window	21	13	**6**
Sensor	NaN	127	**8**
Mast	NaN	18	**5**

3.3. Midair Aircraft Tracking

In this section, an arbitrary aircraft, i.e. intruder for UAV, robustly tracked by the vision-based approach has been researched, as shown in Fig. 3.32, the monocular camera sensor is fixed on the tail of a fixed-wing UAV [14]. The visual aircraft tracking has played an important role in the sense-and-avoid, i.e. See-And-Avoid (SAA), application of UAV in the midair. In addition, a unique evaluation system for aircraft tracking from UAV has been develpoed for evaluating the performances of presented visual algorithm.

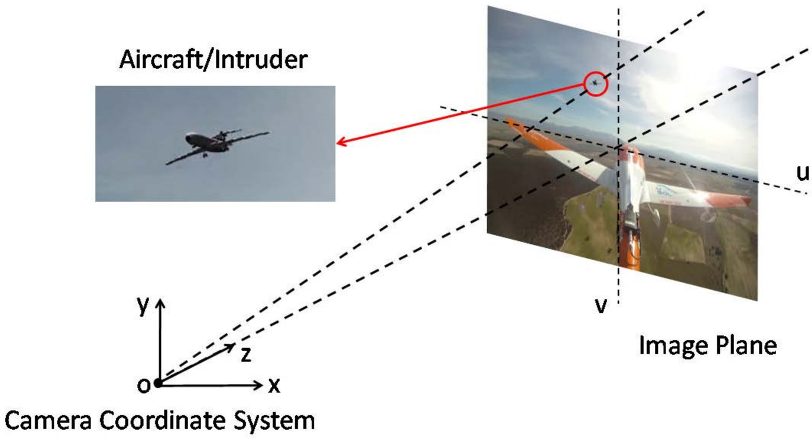


Figure 3.32: Vision-based aircraft inspection from UAV.

Visual aircraft tracking has been researched and developed fruitfully in the robot community recently. However, real-time robust visual tracking for arbitrary aircraft (also referred to visual *aircraft model-free* tracking), especially in Unmanned Aerial Vehicle (UAV), remains a challenging task due to significant appearance change, variant surrounding illumination, partial aircraft occlusion, blur motion, rapid pose variation, and onboard mechanical vibration, low computation capacity and delayed information communication between UAVs and Ground Control Station (GCS).

In literature, many visual trackers have obtained the promising tracking performances for arbitrary aircrafts, where, the morphological filtering technology as the most popular method has been applied in many vision-based Sense-and-Avoid (i.e. See-and-Avoid) systems, e.g. (Wainwright and Ford, 2012), (Gandhi et al., 2000), (Carnie et al., 2006) and (Lai et al., 2012). However, a big number of false positives will be generated by this

[14]The fixed-wing UAVs shown in this section are manufactured by the Unmanned Solutions (USol) company: http://www.usol.es/

approach, and it requires the reliable morphological operators to adaptively detect the aircraft under different backgrounds. Although (Dey et al., 2010) have utilized shape descriptor and SVM-based classifier to reduce false positives, however, it should be trained offline with hand-labeled samples in large amounts of image data with plenty of time and energy.

(Mccandless, 1999) presented an optical flow method for aircraft detection, and (Mian, 2008) proposes a modified KLT tracking algorithm to track aircrafts, which uses a feature clustering criterion to track aircraft based on its multiple local features, and this local features are continuously updated to make the tracker robust to appearance changing of the aircraft. However, all these methods can be generally categorized as the generative-based method, and they did not use the valuable background information to improve the tracking performances (Wang et al., 2011).

In this section, we apply the discriminative-based algorithm (also called visual *tracking-by-detection* method) to track aircraft or intruder in the midair using UAVs, this kind of visual algorithm employs an adaptive binary classifier to separate the aircraft from background during frame-to-frame tracking, and online Multiple-Instance Learning (MIL) method (Babenko et al., 2011) has been used to handle the ambiguity problem, which put the positive samples and negative ones into positive and negative bags, respectively, and then trains a classifier in an online manner using bag likelihood function. This method has demonstrated good performance to handle drift, and can even solve significant appearance changes in the cluttered background.

Moreover, we adopt Multi-Resolution (MR) strategy to cope with the problems of strong motions (e.g. onboard mechanical vibration) or large displacements over time. Additionally, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAVs and Ground Control Station (GCS). Using this strategy, especially in the Multi-Classifier voting mechanism, the importances of test samples have been used to reject samples, i.e. the lower resolution features are initially applied in rejecting the majority of samples (called Rejected Samples (RS)) at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, thereby ensuring the real-time performance and higher accuracy.

3.3.1. Visual Aircraft Tracking

As introduced in section 3.2, Discriminative Visual Tracking (DVT) has obtained a promising tracking result in the UAV application. However, the ambiguity problem can confuse the classifier. (Viola et al., 2006) used a Multiple-Instance Learning (MIL) (Dietterich et al., 1997) approach to solve this ambiguity problem in face detection task successfully.

Tracking with Online Multiple-Instance Learning

Recently, (Babenko et al., 2011) also presented an online Multiple-Instance Learning (MIL) algorithm, i.e. MIL tracker, to track arbitrary targets robustly. In this section, we adopted this method for visual aircraft tracking, as shown in Figure 3.33, the adaptive MIL classifier is updated with online boosting features in the (k-1)th frame, and then applied to estimate the aircraft location in the kth frame.

And the Algorithm 2 shows the pseudo code of this tracker.

Algorithm 2 MIL.

Input: Dataset $\{\mathcal{S}_i, y_i\}_{i=0}^{1}$, where $\mathcal{S}_i = \{\mathbf{S}_{i1}, \mathbf{S}_{i2}, ...\}$ is the ith bag, and $y_i \in \{0, 1\}$ is a binary label of sample $\mathbf{S}_{ij}$

Update weak classifier pool Φ=$\{h_1, h_2, ..., h_M\}$ with data $\{\mathbf{S}_{ij}, y_i\}$
Initialize $H_{ij} = 0$ for all i, j
for k=1 to K **do**
    Set $\mathcal{L}_m$ =0, m=1, ..., M
        for m=1 to M **do**
            for i=0 to 1 **do**
                for j=0 to N+L-1 **do**
                    p_{ij}^m=$\sigma(H_{ij} + h_m(\mathbf{S}_{ij}))$
                end for
                    p_i^m=1-$\prod_j (1 - p_{ij}^m)$
              $\mathcal{L}_m \leftarrow \mathcal{L}_m + y_i log(p_i^m) + (1 - y_i) log(1 - p_i^m)$
            end for
        end for
    m^*=$argmax_m(\mathcal{L}_m)$
    $h_k(\mathbf{S}_{ij}) \leftarrow h_{m^*}(\mathbf{S}_{ij})$
    $H_{ij} = H_{ij} + h_k(\mathbf{S}_{ij})$
end for

Output: Classifer $H_K(\mathbf{S}_{ij}) = \sum_k h_k(\mathbf{S}_{ij})$, and $p(y = 1|\mathbf{S}_{ij})$=$\sigma(H_K(\mathbf{S}_{ij}))$

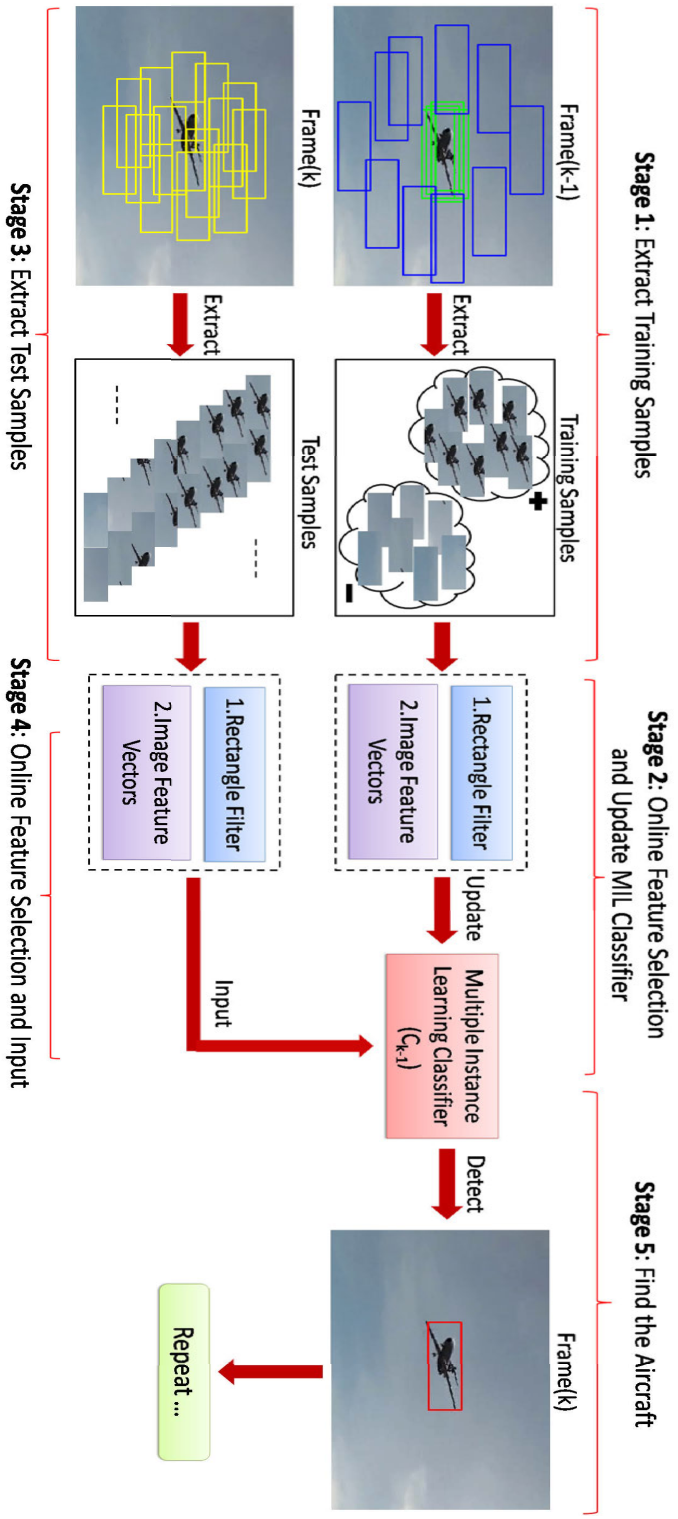


Figure 3.33: Visual aircraft tracking via Multiple-Instance Learning (MIL).

In the Algorithm 2, the posterior probability of sample $\mathbf{S}_{ij}$ to be positive, i.e. $p(y=1|\mathbf{S}_{ij})$, is computed by the Bayesian theorem, $\sigma(z) = 1/(1+e^{-z})$ is a sigmoid function, the strong classifier H_K is constructed by selected K weak classifiers, i.e. $H_K{=}\sum_{k=1}^{K} h_k$. And $\mathcal{L}$ is the bag log-likehood function: $\mathcal{L} = \sum_i (y_i logp_i + (1-y_i)log(1-p_i))$.

For each image sample, it is represented as a vector of Haar-like features (Viola and Jones, 2001), which is denoted by function $\mathbf{f}(\mathbf{S}_{ij})$, i.e. $(f_1(\mathbf{S}_{ij}), f_2(\mathbf{S}_{ij}), ..., f_K(\mathbf{S}_{ij}))^T$. Each feature consists of 2 to 4 rectangles, and each rectangle has a real valued weight. The feature value is then a weighted sum of the pixels in all the rectangles.

We assume that Haar-like features in $\mathbf{f}(\mathbf{S}_{ij})$ are independently distributed and assume uniform prior $p(y=0) = p(y=1)$. Then, the classifier $H_K(\mathbf{S}_{ij})$ is described with the Haar-like feature $\mathbf{f}(\mathbf{S}_{ij})$ as

$$H_K(\mathbf{S}_{ij}) = ln\left(\frac{p(\mathbf{f}(\mathbf{S}_{ij})|y=1)p(y=1)}{p(\mathbf{f}(\mathbf{S}_{ij})|y=0)p(y=0)}\right) = \sum_{k=1}^{K} h_k(\mathbf{S}_{ij}) \tag{3.27}$$

where,

$$h_k(\mathbf{S}_{ij}) = ln\left(\frac{p(f_k(\mathbf{S}_{ij})|y=1)}{p(f_k(\mathbf{S}_{ij})|y=0)}\right) \tag{3.28}$$

and

$$\begin{aligned} p(f_k(\mathbf{S}_{ij})|y_i=1) &\sim N(\mu^1, \sigma^1), \\ p(f_k(\mathbf{S}_{ij})|y_i=0) &\sim N(\mu^0, \sigma^0) \end{aligned} \tag{3.29}$$

The update schemes for the parameters μ_1 and σ_1 are:

$$\begin{aligned} \mu^1 &\leftarrow \eta\mu^1 + (1-\eta)\frac{1}{N}\sum_{j|y_i=1} f_k(\mathbf{S}_{ij}) \\ \sigma^1 &\leftarrow \eta\sigma^1 + (1-\eta)\sqrt{\frac{1}{N}\sum_{j|y_i=1}(f_k(\mathbf{S}_{ij}) - \mu^1)^2} \end{aligned} \tag{3.30}$$

where, N is the number of positive samples and η is a learning rate parameter. The update schemes for μ_0 and σ_0 have similar formulas.

Hierarchy-based Tracking

Although the discriminative-based approaches often achieve superior tracking results, and tolerate the motions in the range of search radius,

Table 3.1: Relationship between Search Radius (α) and Number of Extracted Test Samples (N_S)

Radius α	Sample N_S	Radius α	Sample N_S
30	2809	17	889
29	2617	16	793
28	2449	15	697
27	2285	14	609
26	2109	13	517
25	1941	12	437
24	1789	11	373
23	1649	10	305
22	1513	9	249
21	1369	8	193
20	1245	7	145
19	1125	6	109
18	1005	5	69

but for the tracking on-board UAV, we have observed that discriminative visual tracking algorithms are sensitive to the strong motions or large displacements. The search radius for extracting test samples can be set to be larger, as shown in Algorithm 1, to get more tolerance for these problems, however, more test samples (including noises) will be generated, which influence the real-time and accuracy performances, as shown in TABLE 3.1. Therefore, Multiple Resolution (MR) approach was proposed to deal with these problems, as shown in Figure 3.34, the kth frame is downsampled to create MR structure, the lower resolution features are initially used to reject the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, and the C^p_{k-1} represents the adaptive classifier updated in the pth level of pyramid of (k-1)th frame. This approach also can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAVs and Ground Control Station (GCS).

Configurations

The configurations of Multiple Resolution (MR) approach is the similar to the one introduced in section 3.1.4. Figure 3.35 and TABLE 3.1 show the details of our presented tracker, which are constructed by the importances of test samples (Blue Circle) in the kth frame, where, the Green Circle represents the Ground Truth (GT), the White Circle is the track-

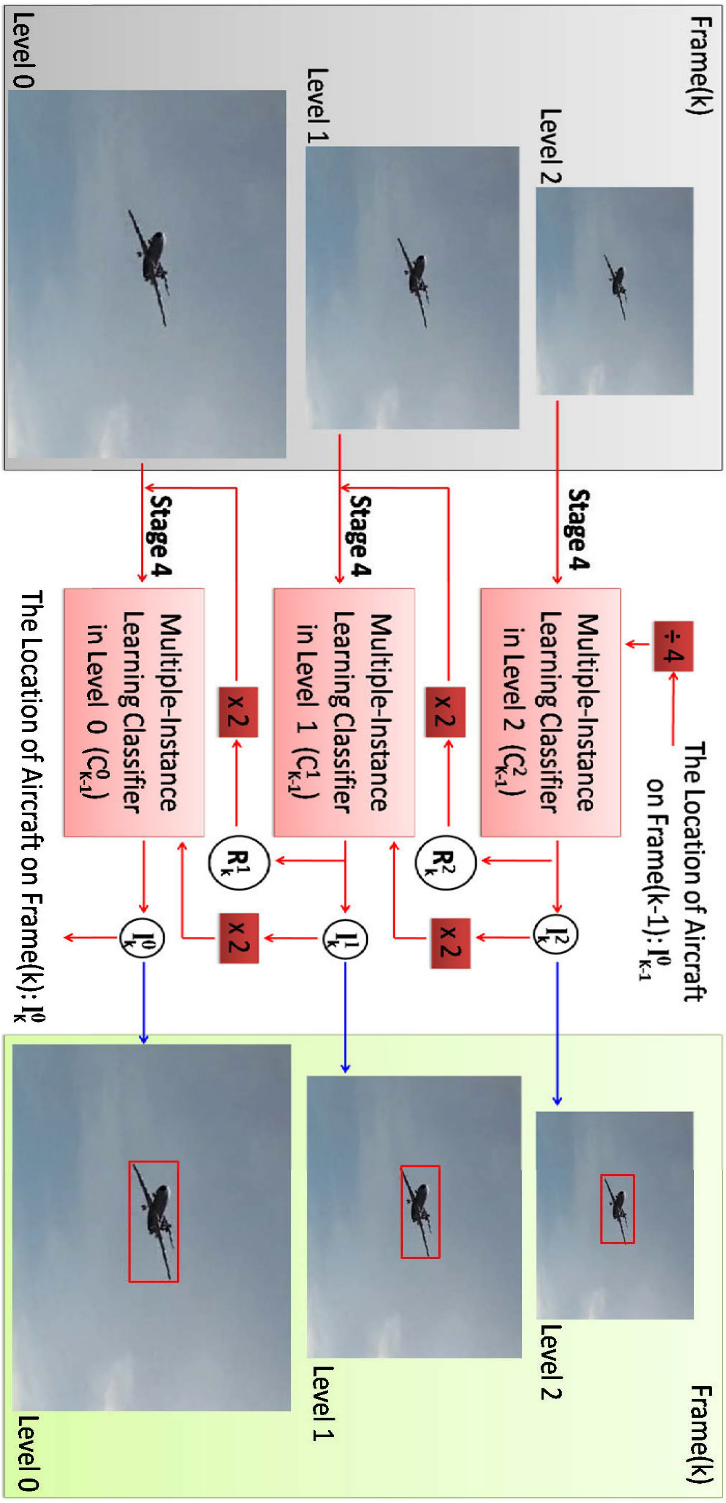
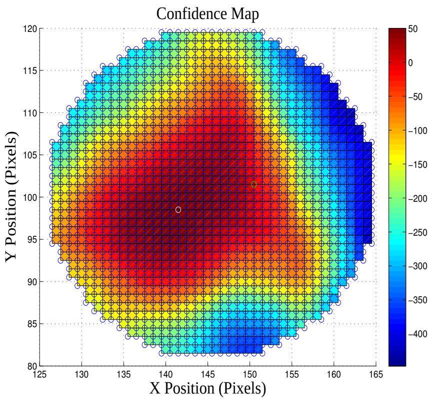
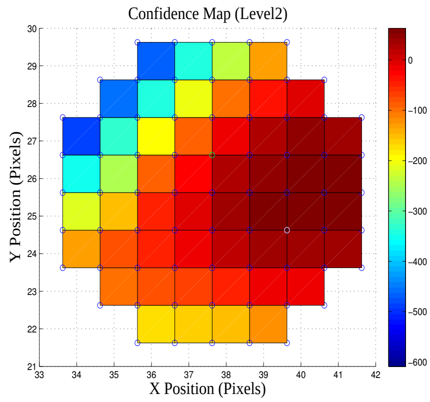


Figure 3.34: AM^3 visual tracker.

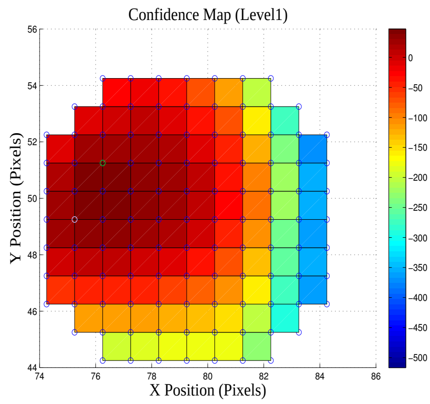
ing result. And Fig. 3.35(a) is the non-hierarchical tracking result, Fig. 3.35(b)3.35(c)3.35(d) are the hierarchical tracking results in different resolution frames, i.e. low, middle and high resolutions. We assume that the tracker requires radius 20 (Unit: Pixels) to search the aircraft in the full (high) resolution frame, then 1245 samples will be extracted to test with classifier, however, our tracker just need a small number of samples (371 in total) within different resolution frames, and obtains higher accuracy, as shown in Fig. 3.36, the Red and Blue Bars represent the hierarchical and non-hierarchical tracking results, respectively.



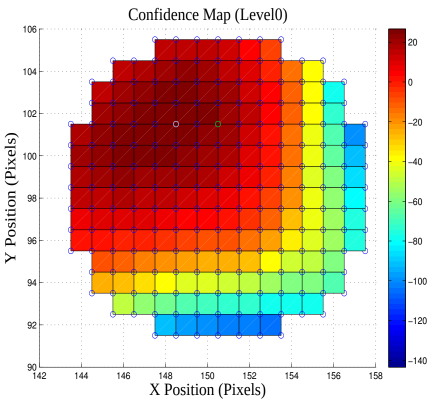
(a) $\alpha = 20$, $N = 1245$



(b) $\alpha = 5$, $N = 69$



(c) $\alpha = 6$, $N = 109$



(d) $\alpha = 8$, $N = 193$

Figure 3.35: Confidence maps from visual tracking on the kth frame.

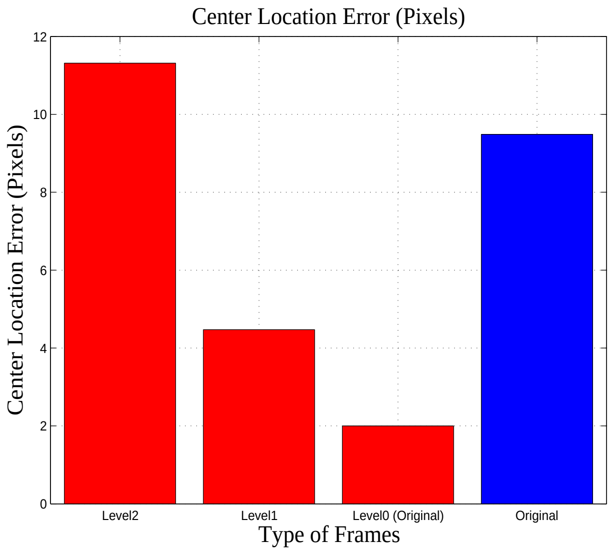


Figure 3.36: Comparison of Center Location Errors in the kth frame.

3.3.2. Evaluation System

In this section, the Collision Visual Warning and Avoidance Evaluation System (CVWAES) has been introduced.

Vision-based aircraft detection and avoidance algorithms demand real scenario images to be tested. These images sometimes are difficult or dangerous to obtain, especially for detecting collision course. For this reason, a new midair collision visual warning and avoidance evaluation system has been developed, this system allows the user to define any flight trajectories and backgrounds using different aircrafts/intruders, where real world images took from some UAVs are fused with virtual images containing 3D aircraft model. These virtual images are obtained taking into account scene illumination, camera vibrations and lens distortions, thereby producing the very realistic video stream.

The 3D pose and attitude of aircraft are pre-defined frame-by-frame, therefore, the performances of different tracking algorithms can be evaluated and compared. The main part of system in software is accomplished with three steps. Firstly, image vibration information is collected from the real world images. Secondly, the virtual image of an aircraft/intruder 3D model is constructed. Finally, both real frames and vitrual images are fused.

Real Image Vibration Information Collection

Due to the existence of vibrations in the real world images, this image vibration effects should be reproducted in the virtual images in order to obtain the most realistic results. The virtual image is transformed according to the homography transformation, which is a (3×3) matrix that links coordinates between two views of the same scene, i.e.:

$$x_i' = Hx_i \tag{3.31}$$

The homography matrices that map the relationship between the first and the other consecutive frames are obtained with below processes:

- Corner feature extraction from the first frame
- Optical flow calculation on the new frame
- Homography matrix collection using RANSAC

Virtual Image Construction

In order to obtain a virtual image displaying an aircraft, a 3D virtual scenario is generated using OpenGL. A virtual camera system and a virtual 3D aircraft are placed and orientated, where, the virtual camera system is configured with the same angle of view in the on-board real camera system, and the virtual 3D aircraft is constructed using a 3D geometry model of the aircraft and a texture, which allows the 3D model to have a realistic appearance.

Additionally, the 3D secene is rendered with a green background, which allows to easily distinguish the aircraft pixels from the background pixels, i.e. chroma key technique[15].

Real and Virtual Image Fusion

The original background image is undistorted and backwarped so that the subsequent warping and distortion applied to both the aircraft and the background will help to generate an unaltered background. Performing the fusion with this way, the interpolation during the warping and distortion processes will produce a more realistic result. The fusion results are shown in the Figure 3.32, 3.33, 3.34, 3.39(a) and 3.40(a) with a common commercial plane: Boeing 727. In addition, more fusion results with different types of intruders have been shown in Fig. 3.37 and 3.38.

[15] http://en.wikipedia.org/wiki/Chroma_key

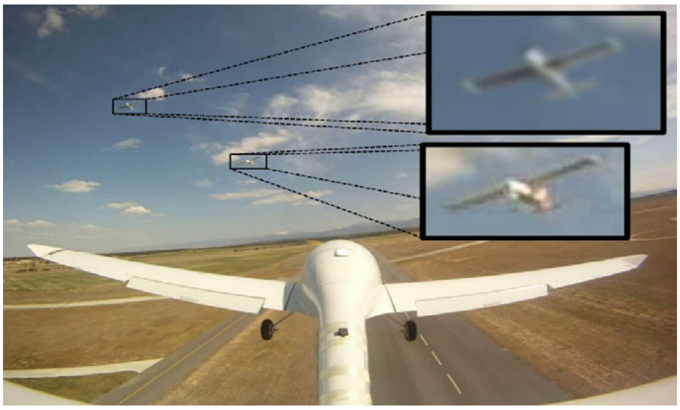


Figure 3.37: Real (above) and simulated (below) intruders.

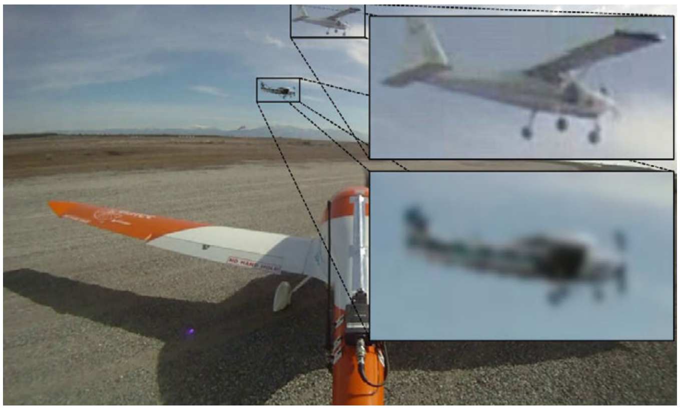


Figure 3.38: Real (above) and simulated (below) intruders.

3.3.3. Comparisons and Discussions

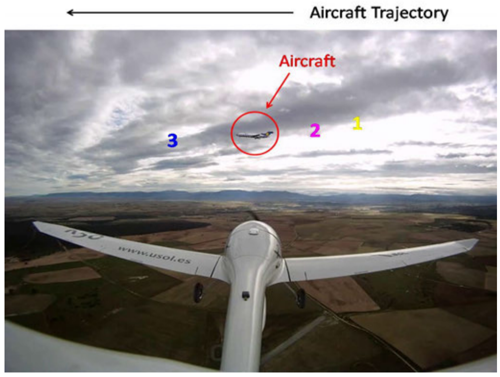
In this section, we compared our AM^3 tracker with 3 latest state-of-art trackers (Frag (Adam et al., 2006), TLD (Kalal et al., 2012) and MIL (Babenko et al., 2011)) on two different types of challenging situations: (I) *Cloudy*; (II) *Strong light*.

The Frames Per Second (FPS) of input evaluation videos is 30. The computation is performed on a 2.4 GHz Intel i5 (4 cores), the average image processing rate, i.e. output of **rostopic hz**, is more than 25.

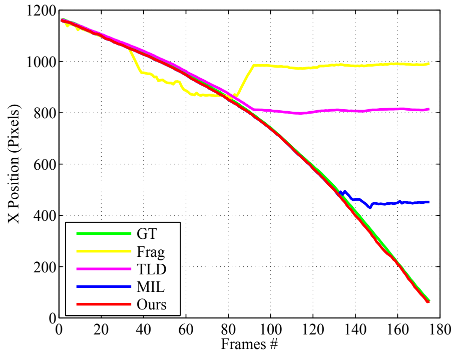
The performances of these trackers were evaluated with the Ground Truth (GT), as shown in the Figure 3.39(b), 3.39(c), 3.40(b) and 3.40(c). And the Center Location Error (CLE) is used to be the evaluation measurement (Bai and Li, 2012), which is defined as the Euclidean distance from the detected aircraft center to its ground truth center at each frame, as shown in the Figure 3.39(d) and 3.40(d).

Test 1: Comparison under the cloudy background

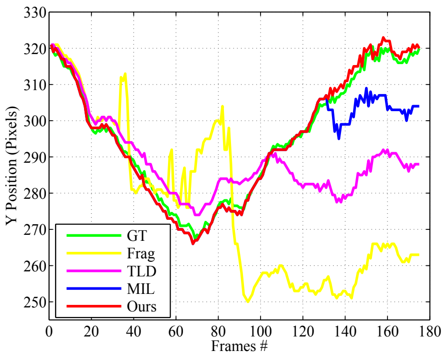
This situation contains four main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Scale change; (III) Illumination Variation; (IV) Background Clutters.



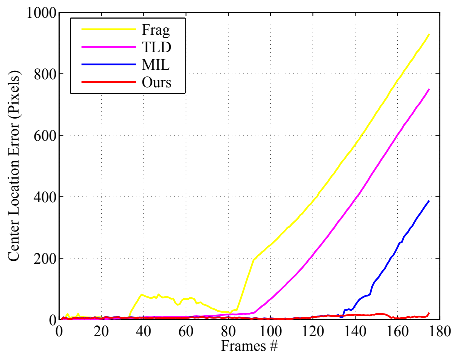
(a) Trajectory and Lost Locations.



(b) X Position Errors.



(c) Y Position Errors.



(d) Center Location Errors.

Figure 3.39: Visual aircraft or intruder tracking on-board UAV under *Cloudy* background.

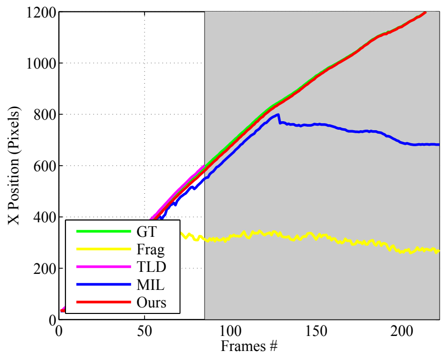
In Fig. 3.39, No.1 (Yellow) represents the lost location tracked by Frag tracker. For TLD and MIL trackers, their lost locations are marked with No.2 (Pink) and No.3 (Blue), respectively. Their tracking performances are evaluated with Ground Truth (Green).

For the tracking performances, as shown in Figure 3.39(b), 3.39(c) and 3.39(d), Frag tracker lost its target firstly when the aircraft was flying from the non-cloud area to the cloud area. While the TLD tracker also lost its target when the illumination of aircraft is similar to the edge of cloud. MIL can track its aircraft well at the beginning, however, it also lost the aircraft when the target was flying from cloud area to non-cloud area. Our new proposed AM^3 can locate the aircraft in all evaluation processes, and the performances of these four trackers have been shown in the TABLE 3.2.

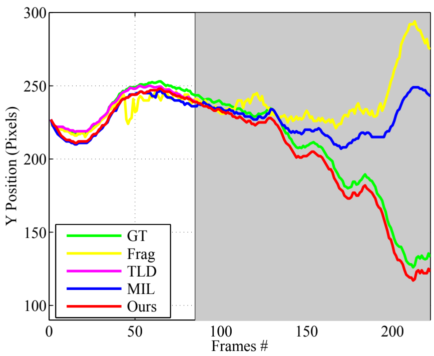
Test 2: Comparison under the strong light background

This situation also contains three main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Scale change; (III) Illumination Variation.

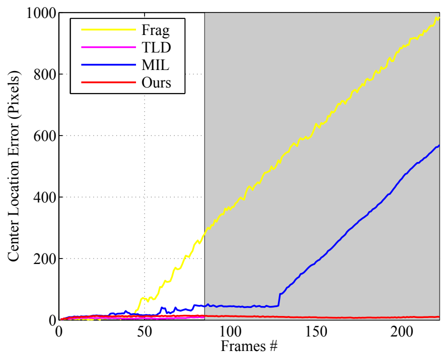
(a) Trajectory and Lost Locations.



(b) X Position Errors.



(c) Y Position Errors.



(d) Center Location Errors.

Figure 3.40: Visual aircraft/intruder tracking on-board UAV under the *Strong Light* background.

In Fig. 3.40, the Grey Shadows show that the TLD tracker lost the aircraft or intruder completely.

During the tracking process, as shown in Figure 3.40(b), 3.40(c) and 3.40(d), the Frag tracker lost its target when a small cloud confused it, as the yellow **1** shown in Figure 3.40(a). For TLD tracker, it is able to relocate on the target at the beginning, but it lost the aircraft completely from the 85th frame. For the MIL tracker, it prones to locate the tail of aircraft, but it also lost the aircraft when the aircraft was flying from the non-strong light area to the strong light area. Our new presented visual tracker AM^3 can track the aircraft all the time until the aircraft flow out of the FOV.

The Center Location Error (CLE) (in pixels) for these two evaluations in this section is shown in below Table:

Table 3.2: Center Location Error (in pixels)

Situations-Trackers	Frag	TLD	MIL	AM^3
Cloudy	275	172	48	**7**
Strong Light	425	NaN	154	**10**

Chapter 4

Visual Odometry

This chapter[1] has presented a robust visual odometry and mapping solution for UAV to estimate all 6 Degrees of Freedom (6DOF) egomotion and full-size flight environments using the fully calibrated stereo camera.

4.1. Introduction

Nowadays, Unmanned Aerial Vehicle (UAV) has been widely utilized in a variety of civilian applications, e.g. earthquake disaster rescue, 3D indoor environment reconstruction and greenhouse orchard monitoring. All these UAV applications mainly depend on the real-time stable accurate localiza-

[1] publications related to this chapter:
- *"Efficient Visual Odometry and Mapping for Unmanned Aerial Vehicle Using ARM-Based Stereo Vision Pre-Processing System"*, IEEE ICUAS, 2015
- *"Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers"*, Journal of Intelligent and Robotic Systems, 2014

tion and full-scale flight environment inputs estimated from the sensor onboard UAV. Global Positioning System (GPS)-based localization approach is well researched for UAV outdoor applications, however, GPS signal is completely lost in indoor environments. Laser Range Finder (LRF) is applied as an alternative sensor to provide both localization and environment information, but it often requires more power consumption, computing capability and payload from UAV, has restricted perception distance and generates a 2D reconstruction map because of the limited Field-Of-View (FOV), i.e. typically only in plane. Considering size, weight, cost, computation performance, mounting flexibility and the capability to process information from complex surrounding environments, camera is the most competitive device for UAV. In literature, monocular and stereo cameras are frequently applied as two main tools for vision-based UAV applications. However, a monocular camera cannot sufficiently estimate the real absolute scale to its observed surrounding environments, leading to generate a lot of accumulated scale drifts for visual localization, especially for UAV autonomous navigation in large-scale environments. While stereo camera is able to effectively estimate depth information determined by the baseline between the left and the right cameras, thereby supplying preferable 6 Degrees-Of-Freedom (DOF) pose estimation and real-size environment information for UAV to autonomously explore in various types of GPS-denied indoor and outdoor environments. In addition, there is no special initilization module to be required in stereo localization in contrast to monocular one. Therefore, stereo camera is selected as a promissing sensor for our UAV applications in this work.

For a typical UAV, it has limited size, payload, computation capability, power supply and expanded mounting space for other sensors, e.g. Asctec Pelican/Firefly[2], DJI F450/F550[3], LinkQuad[4] and 3DR IRIS/X8[5] (diagonal wheelbase: ∼50cm, without propellers, the maximum payload: 300-650 grams). Although many stereo cameras are available to be sold on the commercial markets currently, e.g. Skybotix VI-sensor[6], Point Grey Bumblebee2[7] and VisLab 3DV-E[8]. However, the high cost, big weight or incompatible communication interface reduce a number of potential univer-

[2]http://www.asctec.de
[3]http://www.dji.com/
[4]http://www.uastech.com/
[5]http://3drobotics.com/
[6]http://www.skybotix.com/
[7]http://www.ptgrey.com/
[8]http://vislab.it/products/

sity/company end-users to use these stereo cameras for a wide variety of UAV applications. Therefore, we designed a new light small-scale low-cost embedded stereo vision system for UAV, the details of our embedded system are introduced in Section 4.2.1. However, stereo camera still has two bottlenecks: (I) when the distance between the UAV and the observed environment is much larger than the baseline, the depth estimation becomes inaccurate or simply invalid. (II) the features detected by only one side camera (e.g. occlusion) cannot be associated with the depth via real-time stereo matching, but those 2D features can provide useful information to strengthen visual pose estimation. In addition, a software plugin is developed, it effectively takes advantage of those 2D and 3D information together to estimate the 6D pose between each two consecutive image pairs.

Efficient 3D reconstruction of unknown cluttered GPS-denied environment during UAV flight plays an important role in online obstacle avoidance, online path planning and offline post-processing (e.g. environment analysis and 3D printing). It is classified into two categories based on the density of reconstructed environment representation: sparse and dense. However, building a dense reconstruction model is more practical for UAV than sparse one, as the requirements in mentioned online and offline applications above. In this work, we reconstructed the UAV flight environments to dense 3D models with a robust and efficient 3D volumetric occupancy mapping approach[9] (Schauwecker and Zell, 2014), which is developed based on the OctoMap[10] (Hornung et al., 2013) framework, models the occupied space (i.e. obstacles) or free areas clearly, generates less erroneous artifacts, requires less memory, processes dense depth measurement data faster and supports coarse-to-fine resolution strategy. Moreover, we utilized sperical coordinate system to efficiently represent the depth map and fast access the map data. More details of efficient 3D reconstruction are introduced in Section 4.2.2.

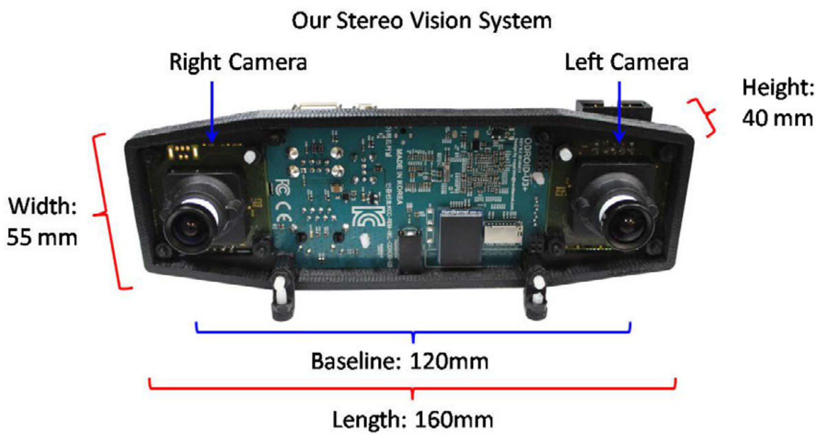
The outline of this chapter is organized as follows: Section 4.2.1 describes the details of our designed light on-board small-scale low-cost stereo vision system. The software plugin developed for visual odometry and 3D reconstruction approach are introduced in Section 4.2.2. Section 4.2.3 evaluated the performance of our presented stereo visual odometry algorithm and a 3D reconstruction model of flight environment. In Section 4.2.4, two field test results from indoor and outdoor typical environments are presented, in addition, the comparison to the well-known stereo visual odometry algorithm have been proposed and discussed.

[9]http://www.ra.cs.uni-tuebingen.de/software/occmapping
[10]http://octomap.github.io/

4.2. Stereo Visual Odometry and Mapping

4.2.1. Stereo Vision System

Fig. 4.1 has shown our light on-board small-scale low-cost stereo vision system, which has been mounted on the Asctec Pelican quadrotor UAV platform, and the on-board host computer of the UAV is Asctec Mastermind. The Gigabit Ethernet cable onboard UAV can be utilized to communicate between onboard host computer and our stereo vision system. The details of our stereo vision system are introduced as follows:



(a) Our stereo vision system.

(b) Quadrotor UAV field test. 1) stereo vision system. 2) onboard host computer.

Figure 4.1: Our stereo vision system and UAV field test.

- *computer*: it is the modification of hardkernel ODROID U3 computer ($69), which has been introduced in the section 3.1.3.

- *cameras*: the system is equipped with two IDS uEye industry cameras[11] (type: UI-1221LE-C-HQ) based on CMOS type sensors (model: MT9V032C12STC) with USB 2.0. The camera supports High Dynamic Range (HDR) mode and global shutter. The frame rate reaches up to 87.2 FPS with freerun mode. In our stereo vision system, two uEye cameras are parallelly fixed on the two sides of a light multi-function mechanical part (which is also used to flexibly mount on the robots), and the stereo image pairs are synchronized, their maximum image resolutions are 752×480 pixels. The focal length of the lenses (i.e. Lensagon[12] BM2820) is 2.8mm, the horizontal and vertical fields of view are 98^o and 73^o, respectively. Each camera size is 36.0mm×36.0mm×25.2mm, and the weight is 20 grams.

The total weight of whole system is 100 grams, which is also lighter than other frequently-used sensors, e.g. RGB-D sensor (Asus Xtion Pro Live): ∼200 grams, 2D Laser (Hokuyo UTM30-LX): ∼270 grams. And it also has less weight than other embedded system-based stereo vision devices, as mentioned in the Section 4.1. The dimension is 160mm×55mm×40mm, and its baseline is 12 centimeters. Additionally, the cost of our stereo camera system is only 800 Euros. To authors's best knowledge, this is the first work to present such a new light low-cost ARM-based stereo vision system.

4.2.2. Visual Odometry and 3D Mapping

This section mainly describes the stereo visual odometry and 3D mapping algorithms.

Coordinate System

The disparity or depth map is generated in the optical frame of left camera, i.e. the coordinate system $\{O\}$ is located on its optical center, as shown in Fig. 4.2, X-, Y- and Z-axis point at Right, Down and Forward, repectively.

[11] http://en.ids-imaging.com/
[12] http://www.lensation.de/

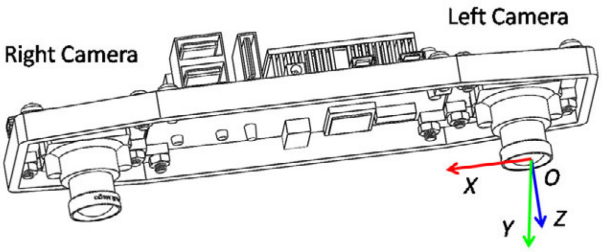


Figure 4.2: Coordinate system $\{O\}$ of our stereo vision system.

We assume that the disparity/depth map is estimated by the kth stereo image pair, the coordinate of the ith feature with depth in $\{O^k\}$, i.e. $\mathbf{X}_i^k \in \mathbb{R}^3$, is defined as:

$$\begin{aligned}\mathbf{X}_i^k &= g(u_r^k, v_r^k, u_l^k, v_l^k) = [x_i^k, y_i^k, z_i^k]^T \\ &= \left[\frac{(u_l^k - c_x)b}{d^k}, \frac{(v_l^k - c_y)b}{d^k}, \frac{fb}{d^k}\right]^T\end{aligned} \tag{4.1}$$

and

$$x_i^k = \frac{z_i^k(u_l^k - c_x)}{f}, \quad y_i^k = \frac{z_i^k(v_l^k - c_y)}{f}$$

where, (u_l^k, v_l^k) and (u_r^k, v_r^k) are the pixels on the kth pair of left and right images, f represents the focal length, c_x and c_y are the coordinates of optical center, $d^k = u_r^k - u_l^k$ is the horizontal disparity, and b is the baseline of the stereo vision system.

Calibration

The Camera Calibration Toolbox[13] has been utilized to calibrate the left and right cameras in our stereo vision system. Fig. 4.3 shows the result of stereo calibration.

And the extrinsic parameters, i.e. pose of right camera wrt left camera, are given with rotation vector $\mathbf{r}_{r-l}$ and translation vector $\mathbf{t}_{r-l}$ below:

$$\mathbf{r}_{r-l} = \begin{bmatrix} 0.01458 \\ -0.00019 \\ 0.00255 \end{bmatrix}^T, \quad \mathbf{t}_{r-l} = \begin{bmatrix} -119.82036 \\ -0.84963 \\ -1.10543 \end{bmatrix}^T \tag{4.2}$$

[13]http://www.vision.caltech.edu/bouguetj/calib_doc/

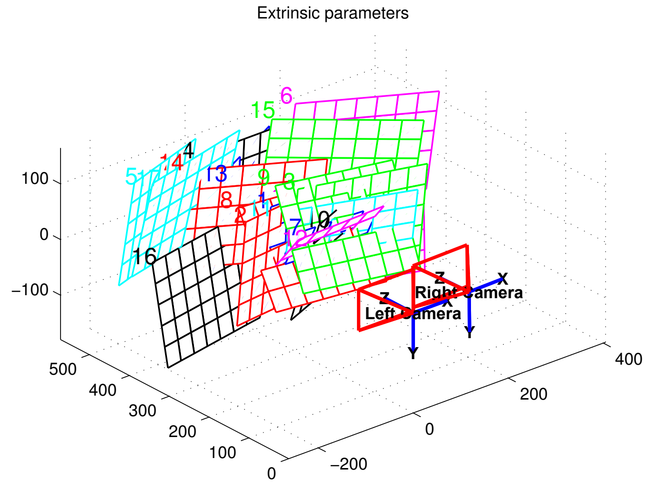


Figure 4.3: Result of stereo calibration (Unit: mm).

where, the rotation vector $\mathbf{r}_{r-l}$ can be converted to a rotation matrix $\mathbf{R}_{r-l}$ with Rodrigues transform (Murray et al., 1994).

After introduced the coordinate system and calibration of our stereo vision system, the whole algorithm flowchart is presented in Fig. 4.4 below:

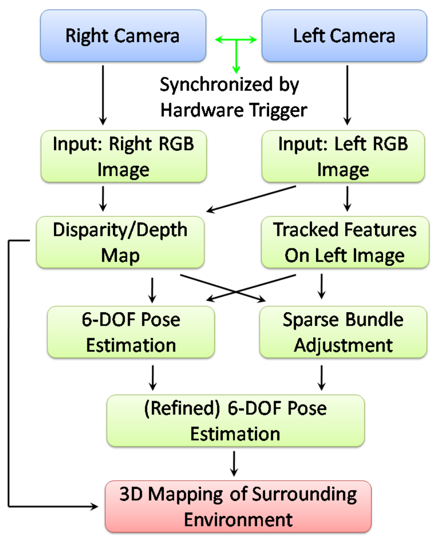


Figure 4.4: Algorithm flowchart.

Disparity/Depth Map Estimation

We synchronized both cameras to publish the image pairs at 15 Hz with hardware trigger, and all the image resolutions are at 376×240. We only estimate the disparity/depth map on the kth stereo image pair, i.e. *one* of two stereo image pairs is utilized instead of processing every consecutive image pairs for disparity/depth estimation, we have $\mathbf{X}_i^{k+1} = z_i^{k+1}\hat{\mathbf{X}}_i^{k+1}$ and $\hat{\mathbf{X}}_i^{k+1} = [\hat{x}_i^{k+1}, \hat{y}_i^{k+1}, 1]^T$. And the disparity/depth map is estimated by a fast stereo Semi-Global Block Matching (SGBM) (Hirschmuller, 2008) approach. To speed up the processing time of SGBM, a Coarse-To-Fine (CTF) strategy (Hermann and Klette, 2012) has been applied, i.e. SGBM$_{\mathcal{F}}$, a prior disparity/depth map is generated from half-resolution stereo image pair firstly, then the prior map is applied for restricting the disparity or depth search space on full-resolution stereo image pair.

We denote the processing time of standard SGBM and SGBM$_{\mathcal{F}}$ as t_S and t_C, the saved processing time performance γ (unit: %) of SGBM$_{\mathcal{F}}$ compared to standard SGBM is:

$$\gamma = 1 - \frac{t_C}{t_S} = \eta\frac{D-9}{D} - \frac{1}{8} \tag{4.3}$$

where, D is the maximum possible disparity, η is the density of disparity/depth map, i.e.

$$\eta = \frac{Number\ of\ Valid\ Disparities}{Size\ of\ Disparity\ Map} \tag{4.4}$$

In practical tests, $\eta < 1$. And some depth estimation results are shown in Fig. 4.8 and 4.11.

Visual Feature Detection and Tracking

We extract and track the features on the consecutive left image frame I using the *bucketing* (Kitt et al., 2010) method, i.e. the input left image I is divided into non-overlapped rectangle regions ($N = N_x \times N_y$), the size of the ith region (bucket) (B^i, $i = 1, 2, 3, \ldots, N$) is defined as:

$$B_W^i = \frac{I_W - 2\times O_W}{N_x}, \qquad B_H^i = \frac{I_H - 2\times O_H}{N_y} \tag{4.5}$$

where, B_W^i and B_H^i represent the width and height of the ith bucket, I_W and I_H are the width and height of image frame, $I_W - 2\times O_W$ and $I_H - 2\times O_H$

represent the width and height of interest area located on the center of image frame.

And each bucket are smoothed with a Gaussian kernel to reduce noise firstly, then the FAST (Rosten and Drummond, 2006) detector is used to extract the keypoints, and a modified version of the BRIEF descriptor (Calonder et al., 2010) is adopted to match the keypoints, the modified BRIEF descriptor of a keypoint $\mathbf{p}$, i.e. $D(\mathbf{p})$, is defined as:

$$D_j(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} + \mathbf{x}_j < \mathbf{p} + \mathbf{y}_j \\ 0 & \text{otherwise} \end{cases}, \forall j \in [1, ..., N_b] \tag{4.6}$$

where, $D_j(\mathbf{p})$ is the jth bit of the binary vector, $(\mathbf{x}_j, \mathbf{y}_j)$ is sampled in a $S_r \times S_r$ local neighbour region based on the locations of keypoints in the previous left image frame, however, the $\mathbf{x}_j$ and $\mathbf{y}_j$ are defined as:

$$\begin{aligned} \mathbf{x}_j &= \mathcal{N}(0, (\frac{1}{5}S_r)^2) \\ \mathbf{y}_j &= \mathcal{N}(\mathbf{x}_j, (\frac{2}{25}S_r)^2) \end{aligned} \tag{4.7}$$

The parameter N_b is the number of bits in the binary vector. The distance of two vectors is calculated by counting the number of different bits between them, i.e. Hamming distance, which is faster than computing the Euclidean distance done in SURF or SIFT. The FAST feature tracking results captured from EuRoC [14] stereo dataset and our real indoor/outdoor field tests have been shown in Fig. 4.5.

6-DOF Motion Estimation of Stereo Camera

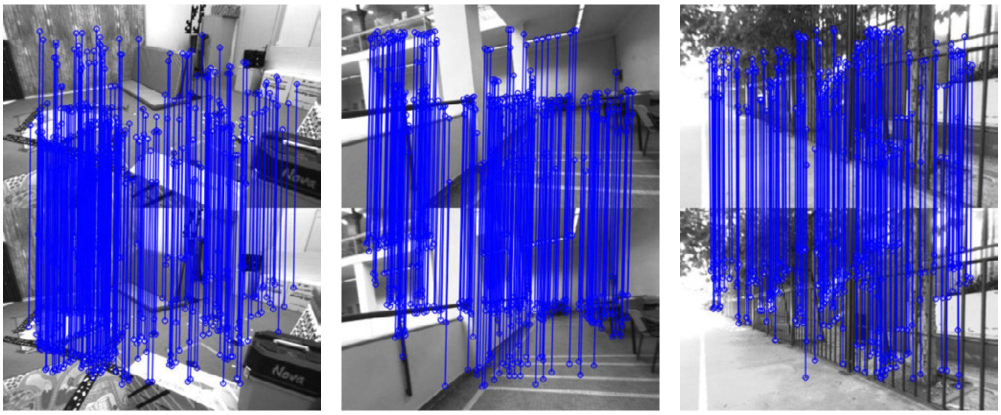
In general, we define $\mathbf{R} \in SO(3)$ as the rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ as the translation vector between two consecutive image frames, then the camera motion can be defined as below:

$$\mathbf{X}_i^{k+1} = \mathbf{R} \cdot \mathbf{X}_i^k + \mathbf{t} \tag{4.8}$$

where, the $\mathbf{X}_i^k$ and $\mathbf{X}_i^{k+1}$ are the coordinates of the ith FAST features tracked by BRIEF descriptor in $\{O^k\}$ and $\{O^{k+1}\}$, respectively.

For parameterizing the translation vector of the camera, its 3 entries are utilized as the variables. For parameterizing the rotation with minimal parameterization to reflect its 3-DOF, since the Lie algebra $\mathfrak{so}_3$ can be

[14] www.euroc-project.eu/



(a) Evaluation: EuRoC Data. (b) Indoor: Library. (c) Outdoor: Football Field.

Figure 4.5: FAST feature tracking results in Indoor and Outdoor environments.

considered as the tangent space of SO_3 at the identity, we used the exponential mapping to project the Lie algebra to the Lie group, then the rotation matrix $\mathbf{R}$ can be expressed with Rodrigues formula (Murray et al., 1994) as:

$$\mathbf{R} = \exp(\mathbf{\Omega}(\omega)) \tag{4.9}$$

$$= \mathbf{I}_3 + \mathbf{\Omega}(\omega) \cdot \frac{\sin(\|\omega\|)}{\|\omega\|} + \mathbf{\Omega}^2(\omega) \cdot \frac{1 - \cos(\|\omega\|)}{\|\omega\|^2} \tag{4.10}$$

where, the isomorphism $\mathbf{\Omega}$ is:

$$\mathbf{\Omega}(\omega) : \mathbb{R}^3 \longrightarrow \mathfrak{so}_3, \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{4.11}$$

Therefore, the motion parameters of the camera is denoted as $\mathbf{M} = [\omega^T \ \mathbf{t}^T]^T \in \mathbb{R}^6$.

As introduced in Section 4.2.2, we only estimated the disparity/depth map on the kth stereo image pair, then the Eq. 4.8 is converted as:

$$z_i^{k+1} \cdot \hat{\mathbf{X}}_i^{k+1} = \mathbf{R} \cdot \mathbf{X}_i^k + \mathbf{t} \tag{4.12}$$

In addtion, a certain number of tracked FAST features only have 2D information because of the two bottlenecks of stereo camera introduced in section B.4, the Eq. 4.8 is converted to:

$$z_i^{k+1} \cdot \hat{\mathbf{X}}_i^{k+1} = z_i^k \cdot \mathbf{R} \cdot \hat{\mathbf{X}}_i^k + \mathbf{t} \tag{4.13}$$

To estimate the camera motion parameters $\mathbf{M} = [\omega^T \ \ \mathbf{t}^T]^T$ of our embedded *stereo* vision pre-processing system with real-time performance, we applied the motion estimation approach done by (Zhang et al., 2014), which has demonstrated and utilized on the *RGB-D* sensor and *Monocular Camera-Lidar* device, taking advantage of 2D and 3D Harris corner (Harris and Stephens, 1988) features tracked by the KLT (Lucas and Kanade, 1981) tracker. In brief, the non-linear cost function c over $\mathbf{M} \in \mathbb{R}^6$ is solved by Levenberg-Marquardt (LM) algorithm (Levenberg, 1944) after eliminated the z_i^k or z_i^{k+1}. Here, we have simply introduced their motion estimation approach.

For Eq. 4.12, after eliminating the z_i^{k+1}, then the following equations can been obtained:

$$(\mathbf{R}_1 - \widehat{x}_i^{k+1} \cdot \mathbf{R}_3) \cdot \mathbf{X}_i^k + T_1 - \widehat{x}_i^{k+1} \cdot T_3 = 0 \tag{4.14}$$

$$(\mathbf{R}_2 - \widehat{y}_i^{k+1} \cdot \mathbf{R}_3) \cdot \mathbf{X}_i^k + T_2 - \widehat{y}_i^{k+1} \cdot T_3 = 0 \tag{4.15}$$

For Eq. 4.13, after eliminating the z_i^k and z_i^{k+1}, then we get:

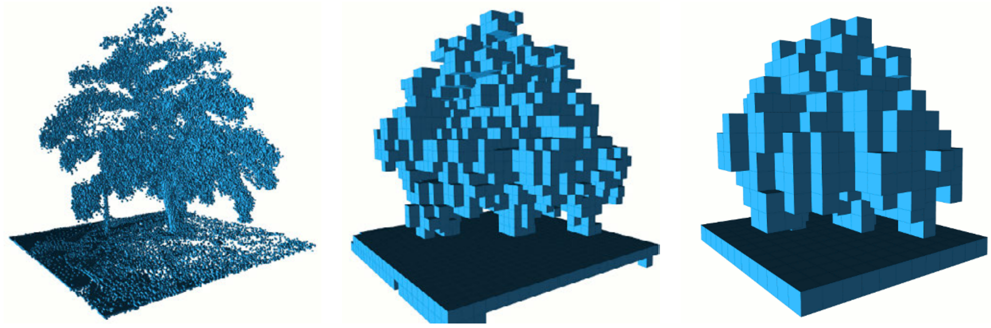
$$\begin{bmatrix} T_2 - \widehat{y}_i^{k+1} \cdot T_3 \\ \widehat{x}_i^{k+1} \cdot T_3 - T_1 \\ -\widehat{x}_i^k \cdot T_3 + \widehat{y}_i^k \cdot T_1 \end{bmatrix}^T \cdot \mathbf{R} \cdot \widehat{\mathbf{X}}_i^k = 0 \tag{4.16}$$

After have estimated the initial motion, a Bundle Adjustment (BA) (Hartley and Zisserman, 2004) using keyframes is adopted to refine the camera motion parallelly, in our work, keyframe is added when the tracked FAST feature number is less than a threshold or camera motion is larger than a threshold.

3D Mapping of UAV Flight Environment

Robust dynamic 3D mapping plays a key role for navigation or exploration and path planning during safely autonomous flights of UAV. In our work, the OctoMap[15] (Hornung et al., 2013) has been utilized as an efficient probabilistic 3D mapping framework for reconstructing arbitary UAV flight environments. This octree-based occupancy grid map models the occupied space (obstacles) and free areas clearly, and supports with coarse-to-fine resolutions, as shown in Fig. 4.6.

[15] http://octomap.github.io/



(a) Resolution: 0.08m (b) Resolution: 0.64m (c) Resolution: 1.28m

Figure 4.6: Different resolutions of Octomap. Images from (Hornung et al., 2013).

However, the original Octomap is prone to generate a number of falsely mapped artifact grids with stereo camera in practical tests, therefore, we have utilized a robust volumetric occupancy mapping method presented by (Schauwecker and Zell, 2014), which is based on the original Octomap approach to solve the above problem, and it is more robust against high temporally or spatially correlated measurement errors, requires less memory and processes faster than original Octomap.

It is worthing to notice that the main parameters applied for final mapping result:

- *Resolution*: the size for representing the obstacles in 3D map (Unit: meter), as shown in the Fig. 4.6.
- *sensor_model/max_range*: the maximum inspection range for inserting point cloud data when dynamically building the 3D map (Unit: meter)
- *sensor_model/hit* or *sensor_model/miss*: the probability for hiting or missing of stereo camera model when dynamically building the 3D map
- *sensor_model/min* or *sensor_model/max*: the minimum or maximum probability for clamping when dynamically building the 3D map
- *pointcloud_min_z* or *pointcloud_max_z*: the minimum or maximum height of point cloud utilized to build the 3D map

Additionally, we applied a spherical coordinate system for representing map point, i.e. a map point $[x_i, y_i, z_i]^T$ is represented by its radial distance r_i, polar angle φ_i and azimuthal angle θ_i, as shown in Fig. 4.7.

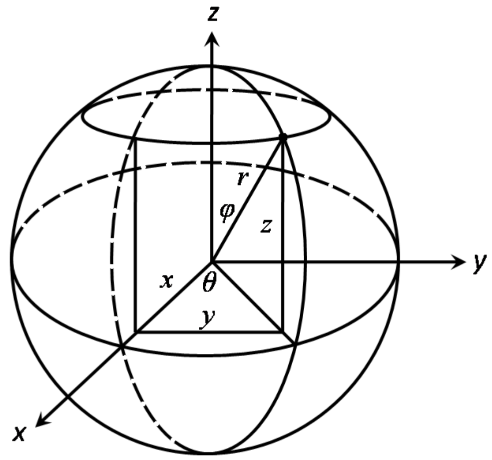


Figure 4.7: A map point is represented in the spherical coordinate system, i.e. $[r_i, \varphi_i, \theta_i]^T$.

The conversion to spherical coordinate system is:

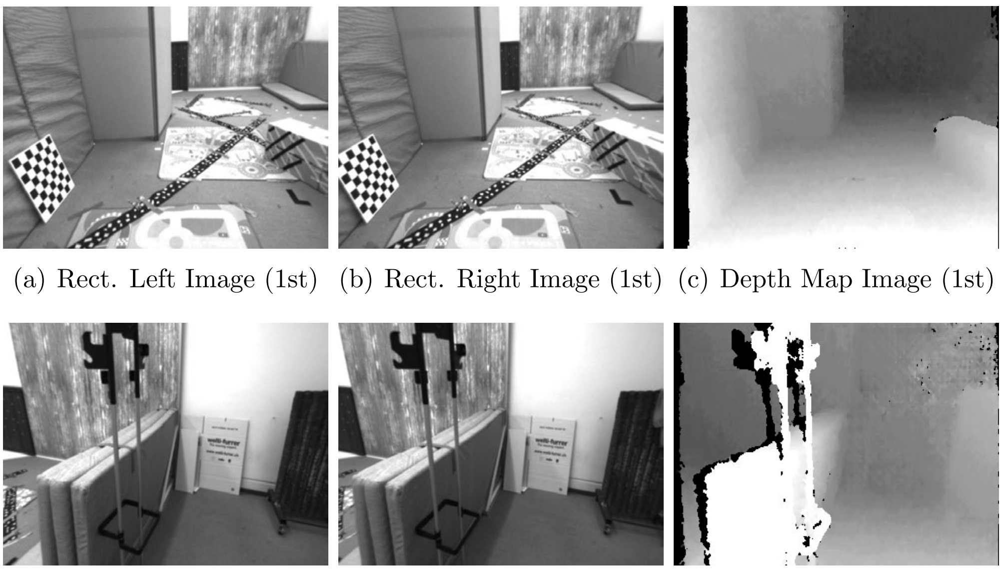
$$\begin{aligned} r_i &= \sqrt{x_i^2 + y_i^2 + z_i^2} \\ \theta_i &= \tan^{-1}(\frac{y_i}{x_i}) \\ \varphi_i &= \cos^{-1}(\frac{z_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}) \end{aligned} \tag{4.17}$$

The spherical depth representation method is similar to the sensing technology, i.e. the denser (sparser) distribution that is closer (farther) to the UAV. And the depth map in the local environment of UAV flight can be stored based on the polar angle, azimuthal angle and resolution of the map.

4.2.3. Performance Evaluation

In this section, the performance of the presented visual algorithm is evaluated using the well-known stereo video datasets from EuRoC[16] Challenge III project, which adopts the Asctec Firefly Hexcopter UAV to record stereo image pairs with Skybotix VI sensor, provides the Ground Truth (GT) of flying UAV pose and includes different dynamic environments with variant surrounding illuminations, blur motions and other challenging conditions.

For each input image frame, we divided it into 6×4 non-overlapped rectangle regions (i.e. buckets), and the maximum number of FAST features in each bucket was set to 20, generating maximumly 480 FAST features to track in total. This method guarantees FAST features to evenly locate in each frame, enhancing the performance of pose estimation. Fig. 4.8, 4.9 and 4.10 show captured images, evaluation performance and 3D mapping result from the first challenging stereo video. In Fig. 4.8, the localizations of captured images in row are shown in Fig. 4.10 with No.1 and No.2. In addition, both voxel size and resolution are 0.1 meter in Fig. 4.10. Red grids represent high altitudes, and purple grids indicate low altitudes, as the color bar shown in Fig. 4.10.



(a) Rect. Left Image (1st) (b) Rect. Right Image (1st) (c) Depth Map Image (1st)

(d) Rect. Left Image (2nd) (e) Rect. Right Image (2nd) (f) Depth Map Image (2nd)

Figure 4.8: Captured images from Asctec Firefly Hexcopter UAV flight.

[16]www.euroc-project.eu/

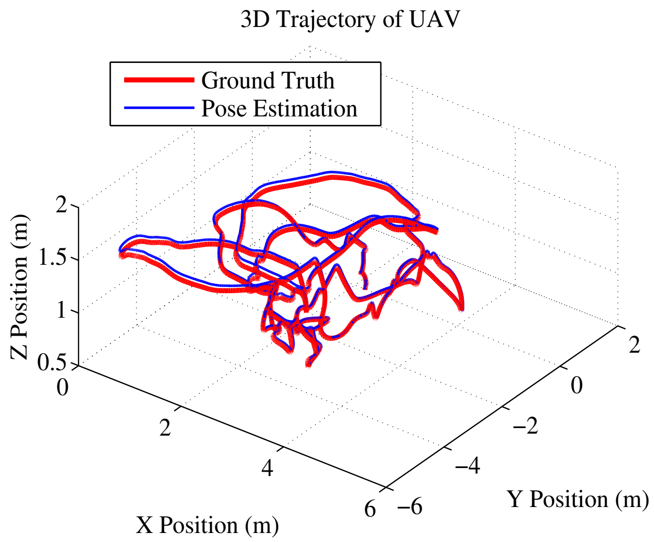


Figure 4.9: Comparison of UAV 3D position estimation..

Table 4.1: The evaluation result. (Unit: Position error in mm, Orientation error in $degree$)

Parameter	X	Y	Z	Roll	Pitch	Yaw
RMSE	18.6	28.4	26.7	4.03	4.36	2.76

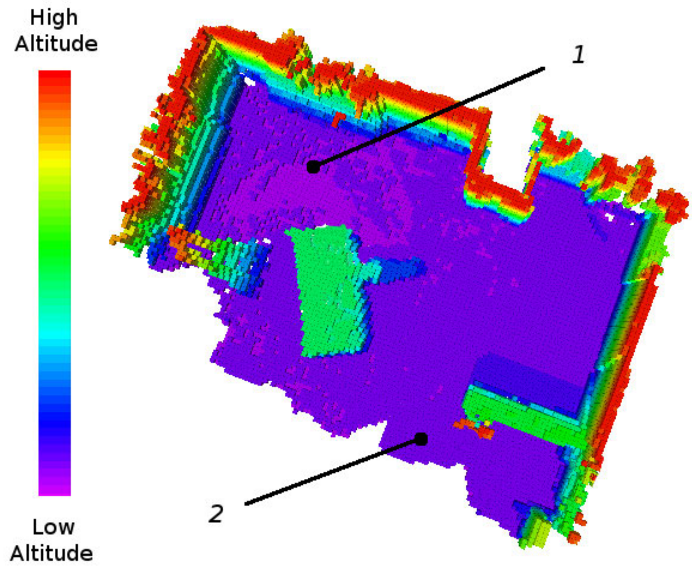


Figure 4.10: 3D mapping result of UAV flight environment.

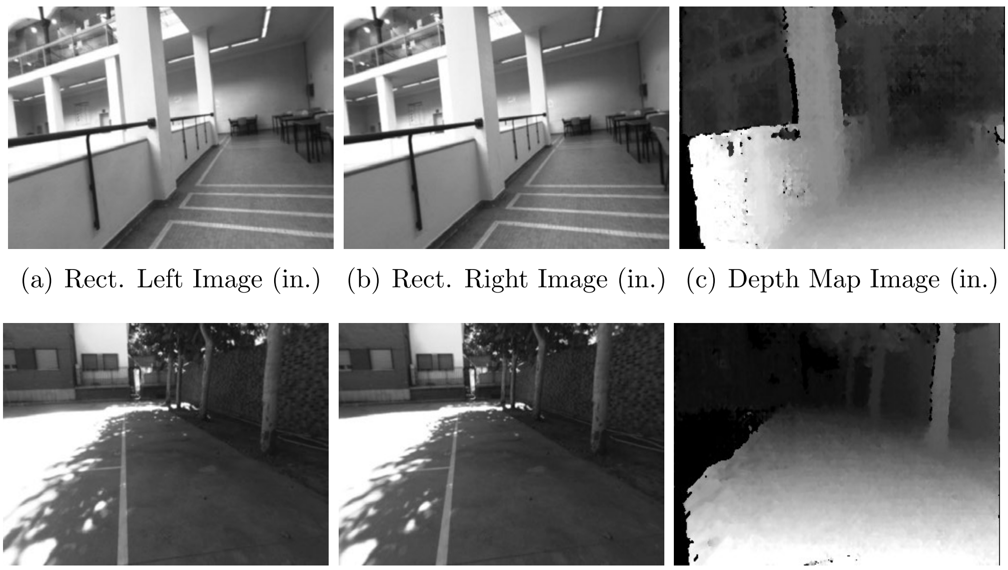
4.2.4. Real Indoor/Outdoor Tests and Comparisons

After having evaluated our stereo visual algorithm, we have tested the algorithm with real stereo vision image datasets, which are recorded in our university campus to demonstrate the performance of visual algorithm with various types of indoor and outdoor dynamic GPS-denied environments. In these indoor and outdoor stereo vision image datasets, we have **re-visited** the starting places. In addition, the variant illuminations (e.g. sunshine, shadow, floor or wall reflections) and blur motions as the main challenging situations have been included in the real indoor and outdoor tests.

In this section, two test results have been presented and discussed:

- Library: the main obstacles include chair, desk, door, wall and pole. Its approximate location in GPS coordinates is (40.440560, -3.689581).
- Football field: the main obstacles consist of tree, iron fence, football door and wall. Its approximate location in GPS coordinates is (40.439449, -3.688328).

Fig. 4.11, 4.12 and 4.14 have shown captured images and 3D mapping results from real indoor and outdoor tests. In Fig. 4.11, the localizations of captured images in row are shown with Black Point in Fig. 4.12 and 4.14 .



(a) Rect. Left Image (in.) (b) Rect. Right Image (in.) (c) Depth Map Image (in.)

(d) Rect. Left Image (out.) (e) Rect. Right Image (out) (f) Depth Map Image (out.)

Figure 4.11: Captured images from real Indoor (Up) and Outdoor (Bottom) tests.

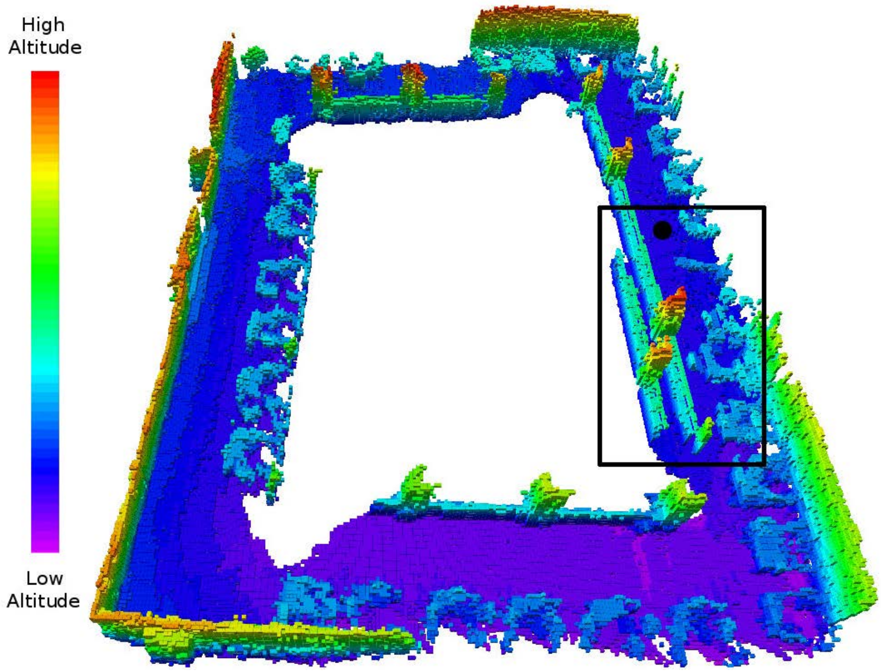


Figure 4.12: 3D mapping result of real indoor test.

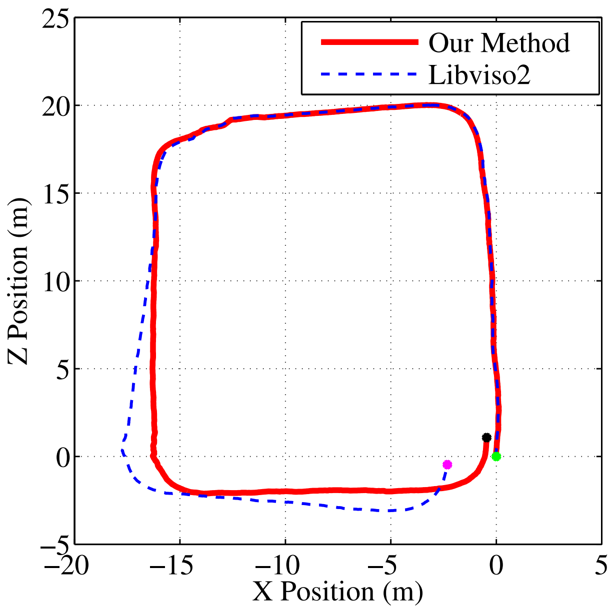


Figure 4.13: Indoor trajectory comparison between our method and Libviso2.

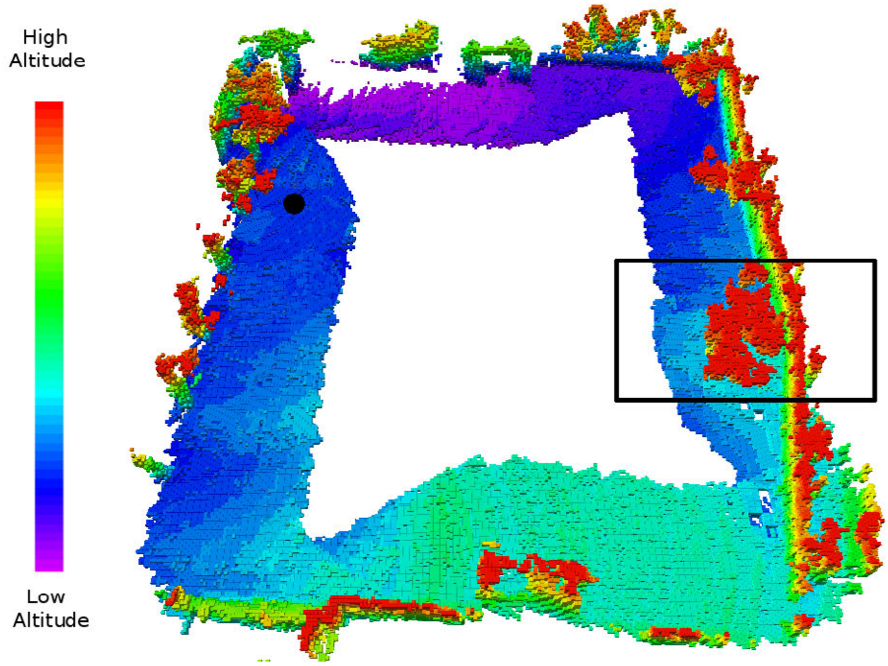


Figure 4.14: 3D mapping result of real outdoor test.

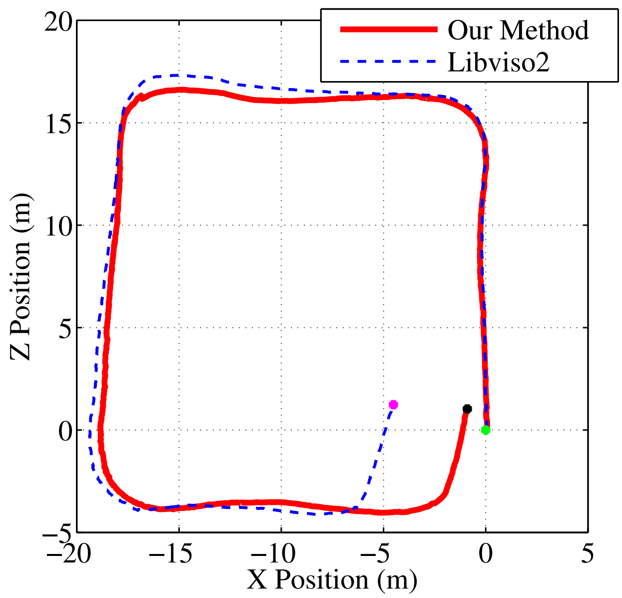


Figure 4.15: Outdoor trajectory comparison between our method and Libviso2.

The indoor and outdoor performances of stereo visual odometry algorithm have also been compared with well-known stereo visual algorithm, i.e. Libviso2[17]. Fig. 4.13 and 4.15 have shown the indoor and outdoor trajectory comparison results, where, the green point is the test stating place, while the black and pink points are the test ending places estimated by those two stereo visual odometry algorithms.

In practice, the stereo algorithm performance in the indoor environments outperforms the ones in the outdoor environments. Especially, the depth map estimation in the indoor environments are more stable than the ones estimation in the outdoor environments, and less noises have been included in the depth map estimated in the indoor environments. The depth map is difficult to be fully estimated when the stereo camera is moving from the places with shadow to the places with strong sunshine, as shown in the Fig. 4.11(f), the depth map information is NULL or invalid in the area with strong sunshine. Therefore, it can be find that the pose estimation, e.g. the altitude estimation, will be unstable, as the large amount of green grids in the bottom of the Fig. 4.14.

In general, the average pre-processed pose estimation rate calculated by the **rostopic hz** is 15 Hz. And the Trajectory Drift (TD), as the Black Rectangle shown in the Fig. 4.12 and 4.14, in the both indoor and outdoor real tests have been calculated as follows:

$$S_{TD} = \sqrt{{X_F}^2 + {Y_F}^2 + {Z_F}^2} \tag{4.18}$$

where, the $*_F$ represents the estimated ending coordinate along with $x-$, $y-$ or $z-$ axis.

Table 4.2: The average Trajectory Drift (TD). (Unit: meter)

Stereo VO	Indoor Environment	Outdoor Environment
Our method	1.3865	2.1664
Libviso2	2.5636	4.6820

[17]http://www.cvlibs.net/software/libviso/

Chapter 5

Visual Control

This chapter [1] has presented a vision-based application for UAV using Fuzzy Logic Controller (FLC). As introduced in the previous chapters, real-

[1] publications related to this chapter:
- *"Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers"*, Journal of Intelligent and Robotic Systems, 2014
- *"A General Purpose Configurable Controller for Indoors and Outdoors GPS-Denied Navigation for Multirotor Unmanned Aerial Vehicles"*, Journal of Intelligent and Robotic Systems, 2014
- *"Using the Cross-Entropy Method for Control Optimization: A Case Study of See-And-Avoid on Unmanned Aerial Vehicles"*, IEEE MED, 2014
- *"Floor Optical Flow Based Navigation Controller for Multirotor Aerial Vehicles"*, ROBOT, Advances in Intelligent Systems and Computing, 2013
- *"UAS See-And-Avoid Strategy using a Fuzzy Logic Controller Optimized by Cross-Entropy in Scaling Factors and Membership Functions"*, IEEE ICUAS, 2013
- *"A General Purpose Configurable Navigation Controller for Micro Aerial Multirotor Vehicles"*, IEEE ICUAS, 2013
- *"AR Drone Identification and Navigation Control at CVG-UPM"*, XXXIII Jornadas Nacionales de Automatica

time robust vision-based solutions have been implemented and presented for different UAV applications. These vision estimations can be utilized as the input of UAV controller to navigate the motion of UAV. However, the uncertainty, inaccuracy, approximation and incompleteness problems widely exist in real controlling techniques. The Fuzzy Logic Controller (FLC) as one of the most active and fruitful soft computing methods can well deal with these above issues. In addition, this *model-free* control approach often has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying robot systems, e.g. UAV.

This chapter has designed and proposed an optimization framework for tuning fuzzy logic controller with Cross-Entropy method, which is motivated by an adaptive approach to estimate probabilities of rare events in complex stochastic networks. The different parts of PID-type fuzzy logic controller have been optimized with this optmization approach. The optimized fuzzy logic controller have been applied for a quadrotor UAV platform to carry on the See-And-Avoid application. Fig. 5.1 shows that a UAV is sensing its surrounding environment using monocular visual-inertial SLAM-based method. This real-time and accurate localization approach allows to expand the UAV's collision avoidance capabilities in the event of failures.

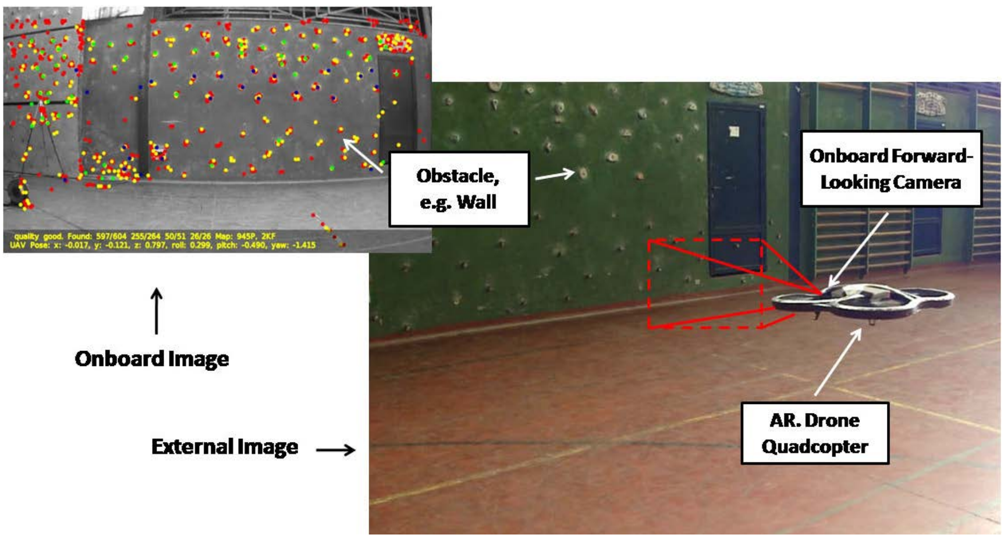


Figure 5.1: UAV see-and-avoid application with monocular visual-Inertial SLAM approach.

5.1. Fuzzy Logic Controller

5.1.1. Introduction

The FLC is based on the fuzzy logic (Zadeh, 1973) that imitates human thinking and decision making with natural language. Its essential part is a set of linguistic control rules, as shown in Table 5.1 to 5.5, related by the dual concepts of fuzzy implication and the compositional rule of inference. In other words, FLC provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control method. It mainly consists of three different type of parameters: (I) Scaling Factor (SF), which is defined as the gains for inputs and outputs. Its adjustment causes *macroscopic* effects to the behavior of the FLC, i.e. affecting the whole rule tables; (II) Membership Function (MF), typically, it is the triangle-shaped function, as shown in Figure 5.6 to 5.9, and its modification leads to *medium-size* changes, i.e. changing one row/column of the rule tables; (III) Rule Weight (RW), it is also known as the certainty grade of each rule, its regulation brings *microscopic* modifications for the FLC, i.e. modifying one unit of the rule tables. (Zheng, 1992) presents a practical guide to tune FLC, which points out that the FLC can be manually tuned from *macroscopic* to *microscopic* effects, i.e. SF adjustment, MF modification and RW regulation.

The Cross-Entropy (CE) method derives its name from the Cross-Entropy (or Kullback-Leibler) distance, which is a fundamental concept of modern information theory. The method was motivated by an adaptive algorithm for estimating probabilities of rare events in complex stochastic networks, which involves variance minimization. In a nutshell, the CE method involves an iterative procedure where each iteration can be broken down into two phases. In the first stage, a random data sample (e.g. SF or MF of FLC) is generated according to a specified mechanism. Then, the parameters of the random mechanism are updated based on the data in order to produce a *better* sample in the next iteration. The CE method provides a unifying approach to simulation and optimization (R.Y.Rubinstein and D.P.Kroese, 2004).

The localization techniques for UAV have obtained many promising performances, which use Global Positioning System (GPS), Motion Capture System (MCS), laser, camera, kinect (RGB-D Sensor) et al. However, considering the cost, size, power consumption, weight and surrounding information different sensors can obtained steadily, camera is the best onboard option. It can achieve the Visual Odometry (VO) (Scaramuzza and Fraundorfer, 2011) to estimate the 6D pose of UAV. Many related works are presented using the advanced monocular Simultaneous Localization and Mapping (SLAM) (Durrant-Whyte and Bailey, 2006) algorithms, they have overcome the drawback of monocular SLAM to estimate the real absolute scale to environments by fusing other sensors, e.g. Inertial Measurement Unit (IMU), in order to navigate the UAV accurately.

Collision avoidance (also referred to sense-and-avoid) problem has been identified as one of the most significant challenges facing the integration of aircraft into the airspace. Here, the term *sense* relates to the use of sensor information to automatically detect possible aircraft conflicts, whilst the term *avoid* relates to the automated control actions used to avoid any detected/predicted collisions (Angelov, 2012). The onboard single or multiple sensors can provide the sense-and-avoid capability for flying aircraft. However, as what has been mentioned above, the camera sensor is the best onboard candidate for UAV, which can be used in collision avoidance applications. Especially, Fail-Safe UAV requires this collision avoidance ability in the event of failures, e.g. GPS has dropped out, INS generated the drift, pilot sent the wrong control commands, the software or hardware of UAV has the faults suddenly et al.

Nonetheless, the main contribution of this chapter are:

(I) Developing the Robot Operating System (ROS)-based FLC, which is a node providing three inputs and one output.

(II) Presenting the FLC training framework integrating with CE in Matlab Simulink, which can be used as the *lazy* method to obtain the optimal SF and MF parameters of FLC.

(III) Applying this framework to solve a challenging task: collision avoidance for Fail-Safe UAV.

(IV) Designing a FLC-based Fail-Safe UAV with a monocular SLAM system.

(V) Optimizing two different kind of FLCs: (1) only SF is optimized in FLC (called SF-FLC)(Olivares-Mendez et al., 2013)(Olivares-Mendez et al., 2012); (2) both SF and MF are optimized in FLC (named SFMF-FLC), and comparing their control performances with Fail-Safe UAV.

The outline of the chapter is organized as follows: The monocular visual-inertial SLAM-based collision avoidance strategy is described in section 5.1.2. Section 5.1.3 designed the FLC with its initial SFs, MFs and rule base. Then, the Cross-Entropy theory and its optimization method for FLC are introduced in section 5.1.4. In section 5.1.5, the UAV training framework and the optimized results are shown. In section 5.1.6, the real flight results have been given and discussed.

5.1.2. Monocular Visual-Inertial SLAM-based Collision Avoidance Strategy

Collison Avoidance Stategy

Many typical civil tasks, such as forest fire monitoring (Merino et al., 2012) in tree lines, fault inspection for buildings or bridges (Murphy et al., 2011) in cluttered urban, are carrying out by UAV currently, and a field study after Hurricane Katrina (Pratt et al., 2009) concluded one of the most important recommendations for autonomy UAV is that the minimum emergent standoff distance from inspected structures is 2-5m. This section aims to discuss and research how to prevent crashes by UAV itself when UAV fly into this recommended distance based on the former working distance, i.e. Fail-Safe UAV avoids the collision in its (local) surrounding envrionment.

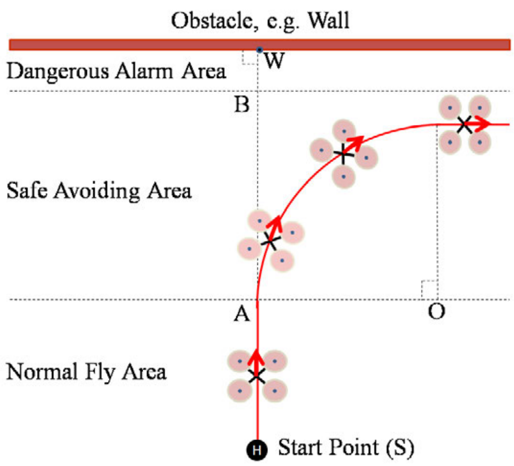


Figure 5.2: 2D Description for Collision Avoidance Task.

Considering a flying Fail-Safe UAV, e.g. AR. Drone Parrot, moving forwardly with a constant flight speed to an obstacle, e.g. wall, where, the heading of quadrotor helicopter is parallel to the normal vector of the obstacle. The control goal is to command it to avoid the obstacle, at least making it flying parallelly to the obstacle with a safe distance. Figure 5.2 shows the collision avoidance strategy, we divided the whole area into three parts:

(I) Dangerous Alarm Area (DAA): it is set based on our quadcopter size (52.5×51.5cm) and its inertance, as shown in the Figure 5.3, the scale of each big grid in white is equal to 1 meter in reality;

(II) Safe Avoiding Area (SAA): it is designed for avoidance, which is based on the recommended distance from 1 meter to 4 meter in length;

(III) Normal Fly Area (NFA): it is the safe working area, the Start Point (S) can be set on any place in the NFA.

Therefore, for Fail-Safe UAV, once it fly into the SAA, the emergency control is activated to prevent crash. And different constant flight speeds and sizes of SAA will be tested and evaluated in the simulations and real flights.

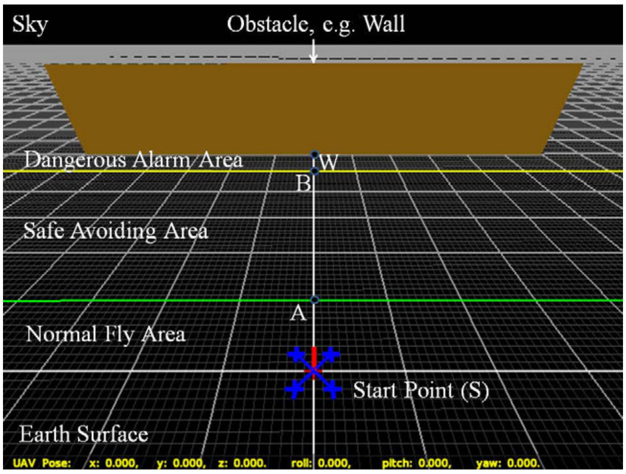


Figure 5.3: Real-time 3D Synchronization Map.

Monocular Visual-Inertial SLAM-based Control

The real-time and accurate 6D pose estimations can provide the comprehensive and reliable information for Fail-Safe UAV during collision avoiance tasks, in the literatures, one single monocular camera and IMU sensor can be used to obtain this localization information. However, fusion of vision and IMU can be classified into 3 different categories (Nutzi et al., 2011). The first section is named *Correction*, where it use the results from one kind of sensor to correct or verify the data from another sensor. The second category is *Colligation*, where one uses some variables resulting from the inertial data together with variables from the visual data. The third category is called *Fusion* and is by far the most popular method to efficiently combine inertial and visual data to improve pose estimation.

Thus, in this section, the *fusion* method is applied in monocular visual-inertial SLAM-based control for Fail-Safe UAV. For monocular vision, several visual odometry and visual SLAM frameworks have been launched in recent years. But the keyframe-based Parallel Tracking and Mapping (PTAM) (Klein and Murray, 2007) is more robust than filter-based SLAM, and it has the parallel processing threads and fast response performance. Besides, it is more suitable for common scenarios. It can provide 6 Degrees of Freedom (DOF) estimation. For Inertial Measurement Unit (IMU), it is a 3D acceleration and rotation estimator.

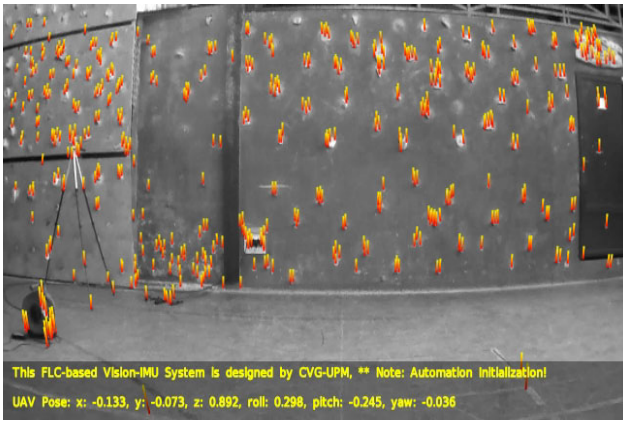


Figure 5.4: FLC-based Fail-Safe UAV during Automation Initilization Stage, where, the orange-yellow line stands for the tracked keypoint (FAST corners) movement from the first keyframe to current frame.

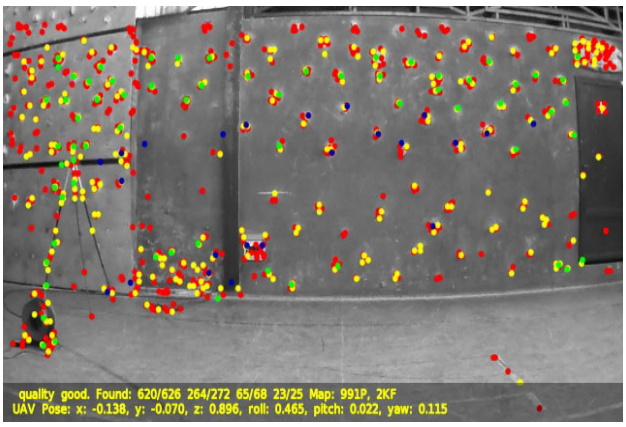


Figure 5.5: FLC-based Fail-Safe UAV during Visual Fuzzy Servoing Stage, where, the dot represents the localization of keypoint. And the colors correspond to which pyramid-level the keypoint is measured in.

Stephan Weiss et al (Weiss et al., 2011) presented that the Inertial Measurement Unit (IMU) as proprioceptive measurement can be used in monocular SLAM (exteroceptive information) to estimate the absolute scale, thereby navigating UAV precisely[2]. Jakob Engel et al (Engel et al., 2012) has presented an Visual-Inertial SLAM system to enable UAV to precisely fly with differnt trajectories[3]. Both of them has published their works as the open sources in the Robot Operating System (ROS). Considering the UAV in our real flight tests, and the precise 6D pose estimation in the latter work, we developed a Fuzzy Logic Controller (FLC)-based Monocular Visual-Inertial SLAM system using his parts of codes, and all the parameters in this new system has been optimized. Figure 5.4 and 5.5 show the automation initialization of PTAM and real-time image processing, which are fusing visual pose estimation with IMU measurement, the average pose estimation rate calculated by the **rostopic hz** is 30 Hz.

5.1.3. Fuzzy Logic Controller

Fuzzy Logic Controller (FLC) has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying Fail-Safe UAV, thus, a FLC is designed to control its orientation. As previously developed FLCs in (Olivares-Mendez et al., 2013)(Olivares-Mendez et al., 2012), this FLC is developed in ROS based on the MOFS (Miguel Olivares' Fuzzy Software).

This FLC is a PID-like controller, which provides three inputs and one output. The first input is the angle error estimation in degrees between the angle reference (e.g. 90^o) and the heading of Fail-Safe UAV. Other two inputs are the derivate and the integral value of this estimated angle error. The output is the command in degrees per seconds to change Fail-Safe UAV's heading. The initial Scaling Factors without CE optimization have the default value, which are equal to one. Since the collision avoiance task is identical for right or left side avoiding, and the Fail-Safe UAV has a symmetric design with the same behavior for left and right heading movements, this FLC has the symmetric definitions in the inputs, output and rule base.

Figure 5.6 to 5.9 show the initial definition for the Membership Functions in the inputs and output before Cross-Entropy (CE) Optimization. Each input has 5 sets, and the output has 9 sets. The symmetry of the FLC implies that any modification of the left side of each variable (input and output) can be applied to the right side.

[2]http://www.ros.org/wiki/asctec_mav_framework
[3]http://www.ros.org/wiki/tum_ardrone

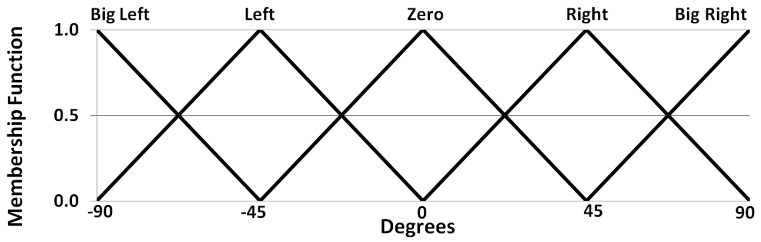


Figure 5.6: Membership Functions for the *First input* (Yaw Error), *without* CE optimization.

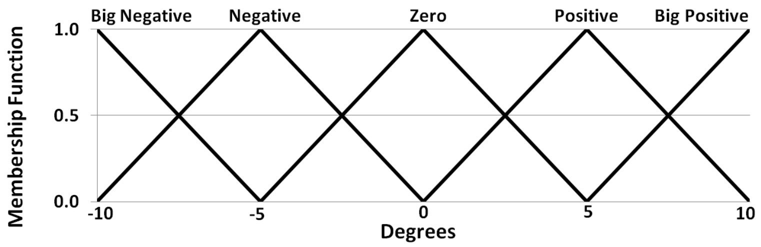


Figure 5.7: Membership Functions for the *Second input* (Derivative of Yaw Error), *without* CE optimization.

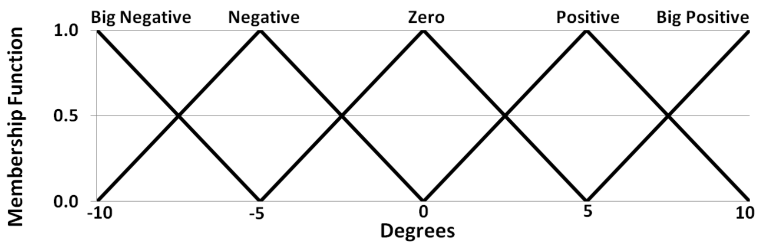


Figure 5.8: Membership Functions for the *Third input* (Integral of Yaw Error), *without* CE optimization.

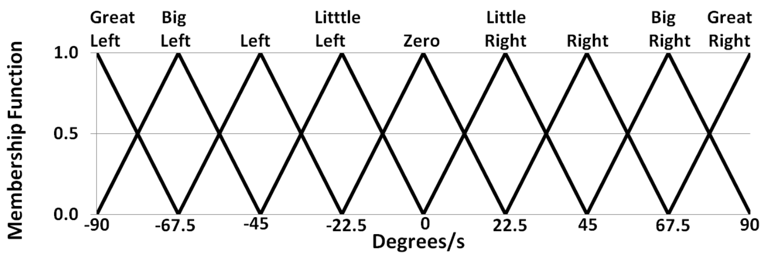


Figure 5.9: Membership Functions for the *Output* (Yaw Command), *without* CE optimization.

The rule base was designed using the heuristic information based on expert knowledge. Each rule without CE optimization has a default weight, which is also equal to one. In other words, each rule has the same importance and effect to the FLC behavior. As shown in the definition of MFs, the three inputs in this FLC imply that the rule base has a cube construction, which is $5 \times 5 \times 5$ dimension. In this section, 5 tables in 5×5 dimension are presented in order to show the rule base clearly. Each table is related to one of the 5 linguistic values of the third variable (i.e. the integral of error). Table 5.1 shows the rule base slide for the *zero* value. Table 5.2 shows the slide for the *negative* value. Table 5.3 shows the slide for the *big negative* value. Table 5.4 shows the slide for the *positive* value. Finally, table 5.5 shows the slide for the *big positive* value.

Table 5.1: Rules based on the *Zero* in the third input (integral of the error), *before* CE Optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Big Left	Left	Little Left	Zero
Negative	Big Left	Left	Little Left	Zero	Little Right
Zero	Left	Little Left	Zero	Little Right	Right
Positive	Little Left	Zero	Little Right	Right	Big Right
Big Positive	Zero	Little Right	Right	Big Right	Great Right

Table 5.2: Rules based on the *Negative* in the third input (integral of the error), *before* CE Optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Big Left	Left	Little Left	Zero	Little Right
Negative	Left	Little Left	Zero	Little Right	Right
Zero	Little Left	Zero	Little Right	Right	Big Right
Positive	Zero	Little Right	Right	Big Right	Great Right
Big Positive	Little Right	Right	Big Right	Great Right	Great Right

Table 5.3: Rules based on the *Big Negative* in the third input (integral of the error), *before* CE Optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Left	Little Left	Zero	Little Right	Right
Negative	Little Left	Zero	Little Right	Right	Big Right
Zero	Zero	Little Right	Right	Big Right	Great Right
Positive	Little Right	Right	Big Right	Great Right	Great Right
Big Positive	Right	Big Right	Great Right	Great Right	Great Right

The product t-norm is used for rules conjunction, and the defuzzification method used in this section is a modification of the Height Weight method. It introduces the value of the weight assigned to each rule in the defuzzification process. Equation 5.1 shows the defuzzification method.

Table 5.4: Rules based on the *Positive* in the third input (integral of the error), *before* CE Optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Great Left	Big Left	Left	Little Left
Negative	Great Left	Big Left	Left	Little Left	Zero
Zero	Big Left	Left	Little Left	Zero	Little Right
Positive	Left	Little Left	Zero	Little Right	Right
Big Positive	Little Left	Zero	Little Right	Right	Big Right

Table 5.5: Rules based on the *Big Positive* in the third input (integral of the error), *before* CE Optimization

Dot error/error	Big Left	Left	Zero	Right	Big Right
Big Negative	Great Left	Great Left	Great Left	Big Left	Left
Negative	Great Left	Great Left	Big Left	Left	Little Left
Zero	Great Left	Big Left	Left	Little Left	Zero
Positive	Big Left	Left	Little Left	Zero	Little Right
Big Positive	Left	Little Left	Zero	Little Right	Right

$$y = \frac{\sum_{l=1}^{M} \overline{y}^l \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) w_i\right)}{\sum_{l=1}^{M} \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) w_i\right)} \tag{5.1}$$

where, N and M represent the number of inputs variables and total number of rules, respectively. $\mu_{x_i^l}$ denotes the merbership function of the lth rule for the ith input variable. $\overline{y}^l$ represents the output of the lth rule. w_i corresponds to the weight of the ith rule, which could takes values from 0 to 1.

5.1.4. Cross-Entropy Optimization

Optimization Principle

The CE method involves an iterative procedure where a random data sample $(x_1, ..., x_N)$ in the χ space is generated according to a specified random mechanism. A probability density function (pdf), such as the normal distribution, is used to update the data in order to produce a better sample in the next iteration. Let $g(-, v)$ be a family of probability density functions in χ, which is parameterized by a real value vector $v \in \Re$: $g(x, v)$. Let ϕ be a real function on χ, so the aim of the CE method is to find the minimum (as this section proposed) or maximum of ϕ over χ, and the corresponding states x^* satisfying this minimum/maximum: $\gamma^* = \phi(x^*) = min_{x \in \chi} \phi(x)$.

In each iteration, the CE method generates a sequence of $(x_1, ..., x_N)$ and $\gamma_1, ..., \gamma_N$ levels, such that γ converges to γ^* and x to x^*. Estimating the probability $l(\gamma)$ of an event $E_v = \{x \in \chi \mid \phi(x) \geq \gamma\}, \gamma \in \Re$ is concerned.

A collection of functions for $x \in \chi, \gamma \in \Re$ are defined:

$$I_v(x,\gamma) = I_{\{\chi(x_i)>\gamma\}} = \begin{cases} 1 & if\phi(x) \leq \gamma \\ 0 & if\phi(x) > \gamma \end{cases} \tag{5.2}$$

$$l(\gamma) = P_v(\chi(x) \geq \gamma) = E_v \cdot I_v(x,v) \tag{5.3}$$

where, E_v denotes the corresponding expectation operator.

In this manner, Equation 5.3 transforms the optimization problem into an stochastic problem with very small probability. The variance minimization technique of importance sampling is used, in which the random sample is generated based on a pdf h. The sample $x_1, ..., x_N$ from an importance sampling density h on ϕ is evaluated by:

$$\hat{l} = \frac{1}{N} \cdot \sum_{i=1}^{N} I_{\{\chi(x_i)>\gamma\}} \cdot W(x_i) \tag{5.4}$$

where, $\hat{l}$ is the importance sampling and $W(x) = \frac{g(x,v)}{l}$ is the likelihood ratio. The search for the sampling density $h^*(x)$ is not an easy task, because the estimation of $h^*(x)$ requires that l should be known $h^*(x) = I_{\{\chi(x_i)>\gamma\}} \cdot \frac{g(x,v)}{l}$. So the referenced parameter v^* must be selected in the situation that the distance between h^* and $g(x,v)$ is minimal, therefore, the problem is simplified to a scalar case. The method used to measure the distance between these two densities is the Kullback-Leibler, also known as Cross-Entropy:

$$D(g,h) = \int g(x) \cdot ln\ g(x)dx - \int g(x) \cdot ln\ h(x)dx \tag{5.5}$$

The minimization of $D(g(x,v),h^*)$ is equivalent to maximize $\int h^* ln[g(x,v)]dx$, which implies $max_v D(v) = max_v E_p \left(I_{\{\chi(x_i)>\gamma\}} \cdot ln\ g(x,v) \right)$, in terms of importance sampling, it can be rewritten as:

$$max_v \hat{D}(v) = max \frac{1}{N} \sum_{i=1}^{N} I_{\{\chi(x_i)>\gamma\}} \cdot \frac{p_x(x)}{h(x_i)} \cdot ln\ g(x_i,v) \tag{5.6}$$

where, h is still unknown, therefore, the CE algorithm will try to overcome this problem by constructing an adaptive sequence of the parameters $(\gamma_t \mid t \geq 1)$ and $(v_t \mid t \geq 1)$.

FLC Optimization Description

The Cross-Entropy method generates N FLCs: $x_i = (x_{i1}, x_{i2}, ..., x_{ih})$, where, $i = 1, 2, .., N$, h represents the number of optimization objects. The probability density functions: $g(x, v) = (g(x_1, v), g(x_2, v), ..., g(x_h, v))$ are also generated from CE method. Then, the objective function values are calculated for each FLC. The optimization objects: $x_{i1}, x_{i2}, ..., x_{ih}$ in the first optimization stage correspond to SFs, i.e. K_p, K_d and K_i, and then they represent the position and size of the membership function sets in the second stage. After all the generated controllers are tested in one iteration, the $g(x, v)$ is updated using a set of best FLCs (i.e. elite FLCs). The number of elite FLCs used to update the pdf is denoted as N^{elite}. Then, new N FLCs are generated, which are tested in the next iteration. When the maximum number of iteration is reached, the optimization process is over. A generic version of the optimization process for FLCs is presented in the Algorithm 3.

Algorithm 3 Cross-Entropy Algorithm for Fuzzy controller optimization

1. Initialize $t = 0$ and $v(t) = v(0)$
2 Generate N FLCs: $(x_i(t))_{1\leq i\leq N})$ from $g(x, v(t))$, being each $x_i = (x_{i1}, x_{i2}, ..., x_i h)$
3. Compute $\phi(x_i(t))$ and order $\phi_1, \phi_2, ..., \phi_N$ from smallest $(j = 1)$ to biggest $(j = N)$.
Get the N^{elite} first controllers $\gamma(t) = \chi_{[N^{elite}]}$.
4. Update $v(t)$ with $v(t+1) = arg_v min \frac{1}{N^{elite}}$
$\sum_{j=1}^{N^{elite}} I_{\{\chi(x_i(t))\geq\gamma(t)\}} \cdot ln\ g(x_j(t), v(t))$
5. Repeat from step 2 until convergence or ending criterion.
6. Assume that convergence is reached at $t = t^*$, an optimal value for ϕ can be obtained from $g(., v(t)^*)$.

In these two optimization processes, the gaussian distribution function was used. The mean μ and the variance σ of h parameters are calculated for each iteration as $\tilde{\mu}_{th} = \sum_{j=1}^{N^{elite}} \frac{x_{jh}}{N^{elite}}$, $\tilde{\sigma}_{th} = \sum_{j=1}^{N^{elite}} \frac{(x_{jh}-\mu_{jh})^2}{N^{elite}}$. The mean vector $\bar{\tilde{\mu}}$ should converge to γ^* and the standard deviation $\bar{\tilde{\sigma}}$ to zero, where, $N^{elite} = 5$.

In order to obtain the smooth updates for the mean and variance in iterations, a set of parameters, i.e. α, η, β in Equation 5.7, have been applied, where α is a constant value used for the mean, η is a variable value, which is applied to the variance to avert the occurrences of $0s$ and $1s$ in the parameter vectors, and β is a constant value, which changes the value of $\eta(t)$.

$$\begin{array}{l} \eta(t) = \beta - \beta \cdot (1 - \frac{1}{t})^q \\ \hat{\mu}(t) = \alpha \cdot \tilde{\mu}(t) + (1 - \alpha) \cdot \hat{\mu}(t-1) \\ \hat{\sigma}(t) = \eta(t) \cdot \tilde{\sigma}(t) + (1 - \eta(t)) \cdot \hat{\sigma}(t-1) \end{array} \tag{5.7}$$

where, $\hat{\mu}(t-1)$ and $\hat{\sigma}(t-1)$ are the previous values of $\hat{\mu}(t)$ and $\hat{\sigma}(t)$. The values of the smooth update parameters are set: $0.4 \leq \alpha \leq 0.9$, $0.6 \leq \beta \leq 0.9$ and $2 \leq q \leq 7$. In order to get an optimized controller, the objective function named Integral Time of Square Error (ITSE) is selected.

5.1.5. Training Framework and Optimized Results

Training Framework

The training framework developed in Matlab Simulink is presented in this section, which is used to optimize the FLC to change Fail-UAV's heading. In this framework, the optimization will be processed by itself, and each controller generated by the Cross-Entropy method is tested. The main simulink blocks include UAV model, virtual camera, obstacle and FLC.

UAV Model

This *UAV Model* block is designed for AR. Drone Parrot, Asctec Pelican and LinkQuad[4] in CVG-UPM[5] according to (Mellinger et al., 2012) and (Michael et al., 2010), as shown in the Figure 5.10. This block has four different type of input commands, however, the presented avoiding task only require to control the orientation of Fail-Safe UAV, and it will move forwardly with a constant flight speed to an obstacle, thus, the pitch and yaw commands are only to be controlled, i.e. roll and altitude commands were set to 0. For the UAV pitch commands, different constant speeds were sent in all the tests in order to improve the generalization performance of FLC. For the yaw commands, they are generated by the optimizing FLCs. The outputs of this block are the current 6D pose of the quadcopter.

[4]http://www.uastech.com/
[5]http://www.vision4uav.com

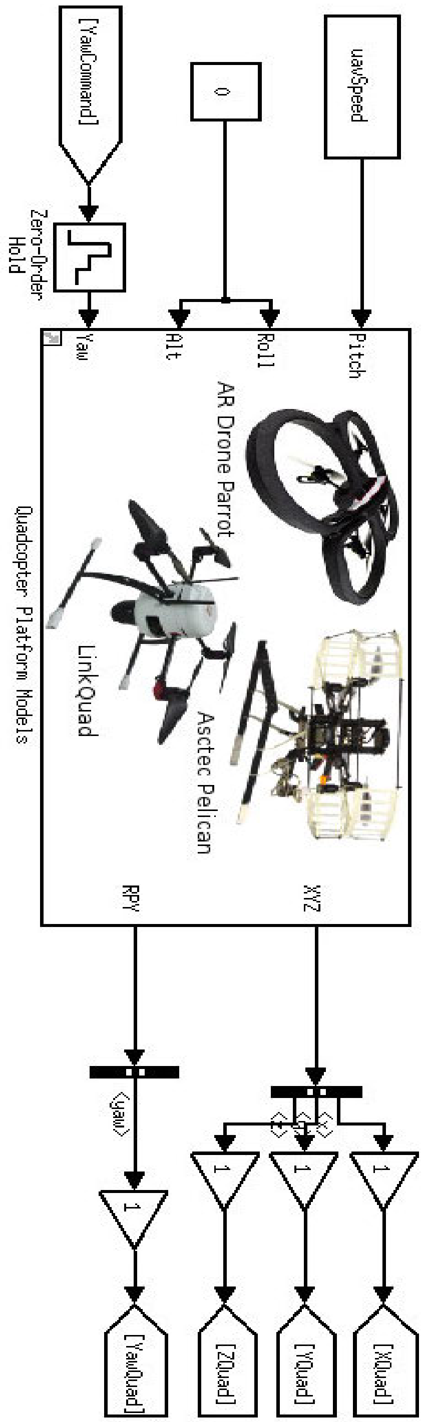


Figure 5.10: UAV Model Block in Matlab Simulink, where, it is suitable for different quadcopter platforms.

Obstacle

The *Obstacle* block is implemented for Fail-Safe UAV to avoid, as shown in the Figure 5.11. This block contains six inputs: the initial positions, i.e. x, y, z of obstacle, linear speed, angular speed and the orientation. In the collision avoiance task, the obstacle, e.g. wall, is static, thus, the speed commands are set to 0. In the other hand, the heading of UAV is parallel to the normal vector of the obstacle at the beginning of task, therefore, the orientation is also given to 0. The x position of obstacle is set according to the minimum emergent standoff distance mentioned in (Pratt et al., 2009), in our work, it is equal to 5.

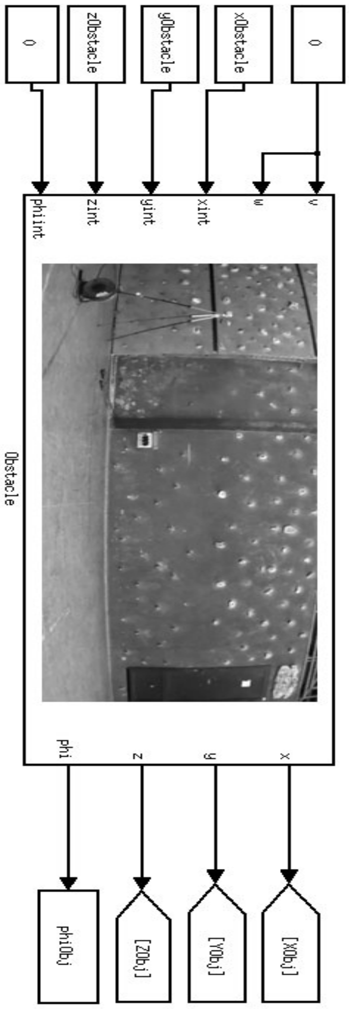


Figure 5.11: Obstacle Block in Matlab Simulink.

Virtual Camera

The *virtual camera* block is constructed to simulate the onboard forward camera in Fail-Safe UAV, which is used to detect the target (i.e. obstacle) and provide the angle reference information to the FLC. The Figure 5.12 shows the onboard virtual camera block implemented in Matlab Simulink. The inputs of this block includes the current positions of the target and the quadcopter. The output is the horizontal angle, which is equal to 90 degree according to the above parameters set for obstacle.

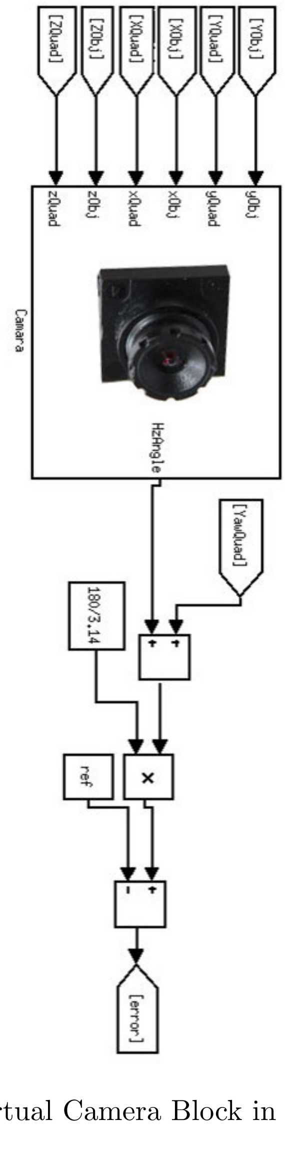


Figure 5.12: Virtual Camera Block in Matlab Simulink.

Fuzzy Logic Controller

Figure 5.13 is the Fuzzy Logic Controller (FLC) block for controlling the heading of Fail-Safe UAV. It has 3 inputs and 1 output. The three inputs are the yaw error, its derivate and integral value. And the initial membership functions and rule base are set to it based on the Table 5.1 to 5.5 and Figure 5.6 to 5.9. And the initial scaling factors are set to 1. This FLC generates the yaw command to the Fail-Safe UAV to avoid the collision.

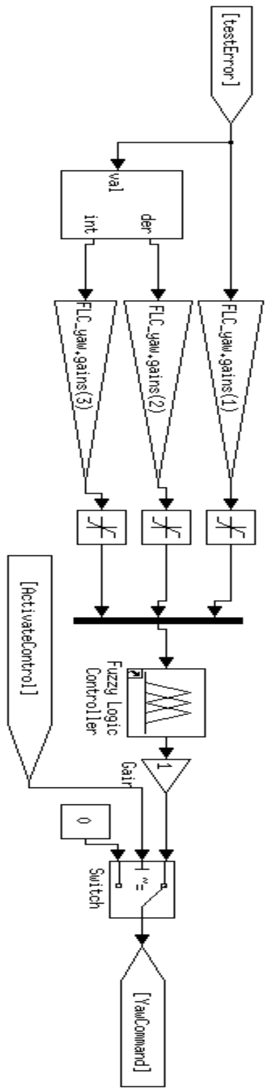


Figure 5.13: Fuzzy Logic Controller (FLC) Block in Matlab Simulink.

Optimized Results

In the literatures, although CE has been used to optimized many FLCs in different systems, their optimized parameters are only limited to Scaling Factors. In this section, an new application of this method was presented for optimizing both Scaling Factor (SF) and Membership Function (MF) in the FLC as mentioned above, i.e. a Macroscopic and Medium-size optimization are presented.

The initial parameters for Cross-Entropy method were set based on (E.Haber et al., 2010), (Olivares-Mendez et al., 2013) and (Z.Botev and D.P.Kroese, 2004). Figure 5.14 shows the whole training process.

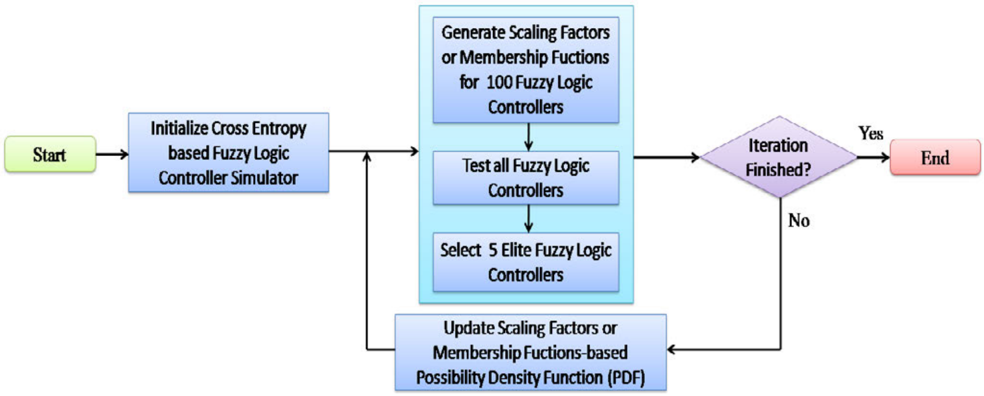


Figure 5.14: Flowchart of Cross-Entropy Optimization for Scaling Factors and Membership Functions in FLCs.

Scaling Factors Optimization Results

Figure 5.15 shows the control loop during the scaling factors, i.e. K_p, K_d, K_i, optimization stage. The evolution can be shown with mean and sigma values associated with each scaling factor. Both values can be used to represent the Probability Density Function (PDF) in each iteration.

Figure 5.16 shows the evolution of the PDF for the SF of first input in FLC, its optimized value is 4.6739. Similarily, the optimal scaling factors for second and third input, as shown in Figure 5.17 and 5.18, are 0.03 and -0.5003, respectively. In the SFs optimization stage, its winner iteration is the 85th iteration in 100 iterations.

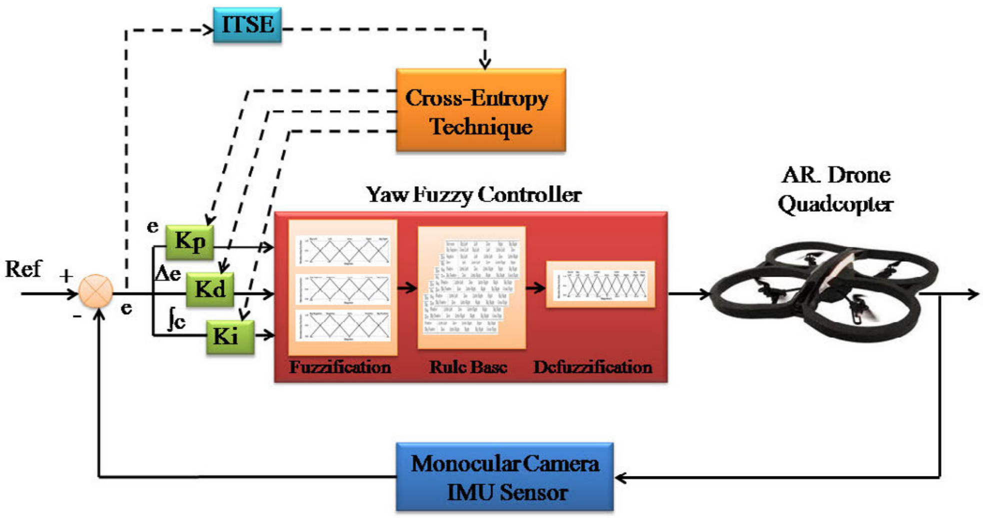


Figure 5.15: Cross-Entropy Optimization for Scaling Factors in FLC.

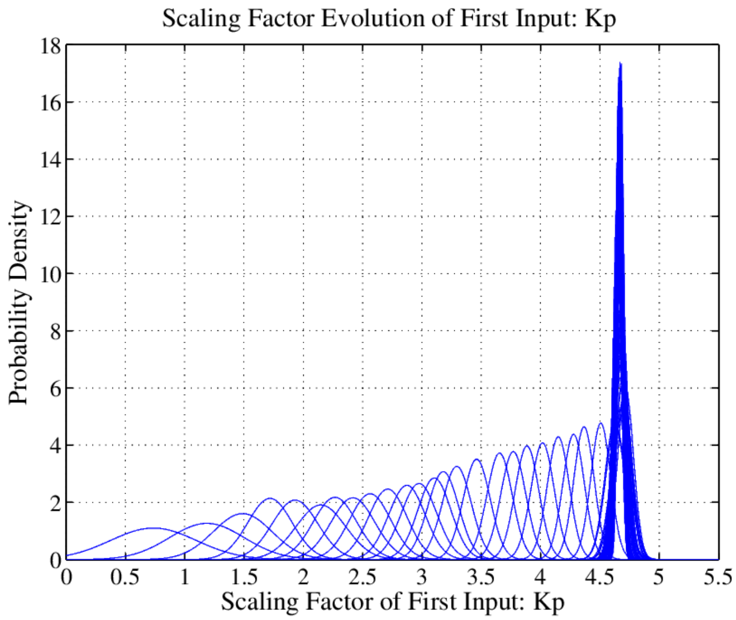


Figure 5.16: The evolution of the PDF for the Scaling Factor of first input (K_p) in FLC using CE method. The optimal Scaling Factor for first input is 4.6739.

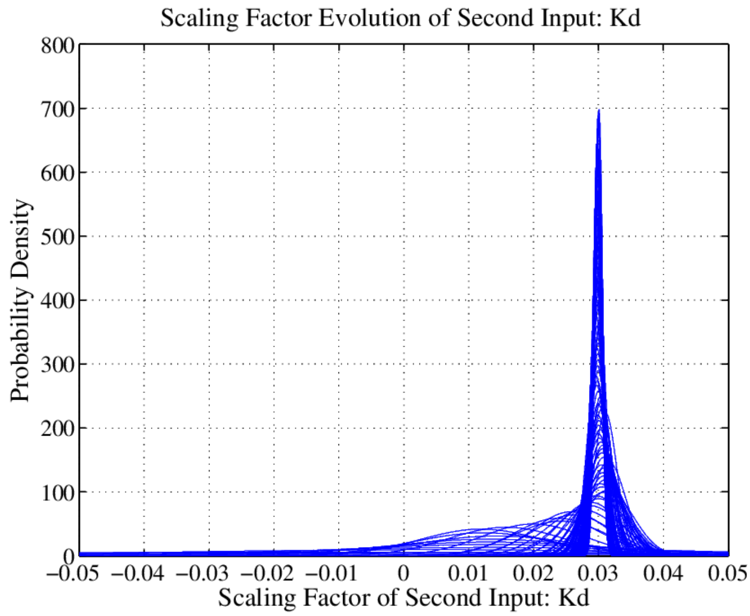


Figure 5.17: The evolution of the PDF for the Scaling Factor of second input (K_d) in FLC using CE method. The optimal Scaling Factor for second input is 0.03.

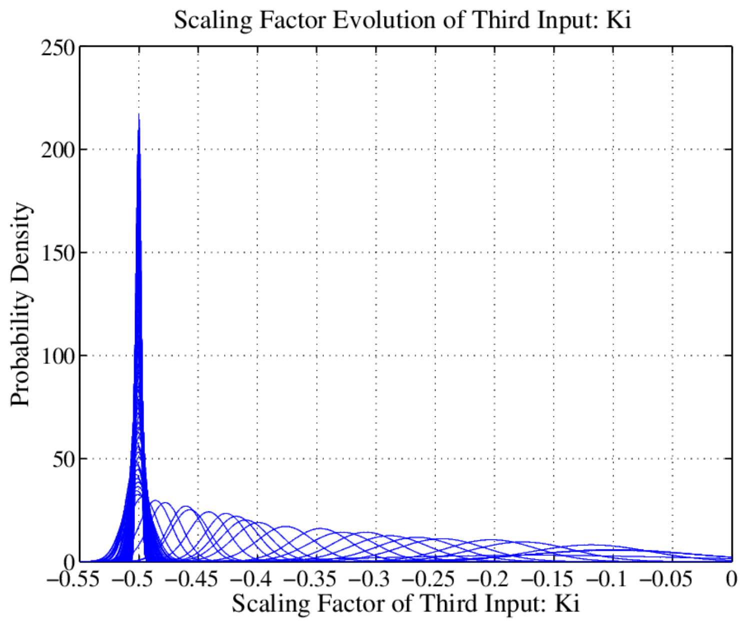


Figure 5.18: The evolution of the PDF for the Scaling Factor of third input (K_i) in FLC using CE method. The optimal Scaling Factor for third input is -0.5003.

Membership Functions Optimization Results

After obtained the optimal SFs (K_{po}, K_{do}, K_{io}) for FLC, the Membership Functions should be optimized. Figure 5.19 shows the control loop during the membership functions optimization stage. Considering the membership functions are symmetric, any position modification of the left side of each variable (input and output) can be applied to the right side.

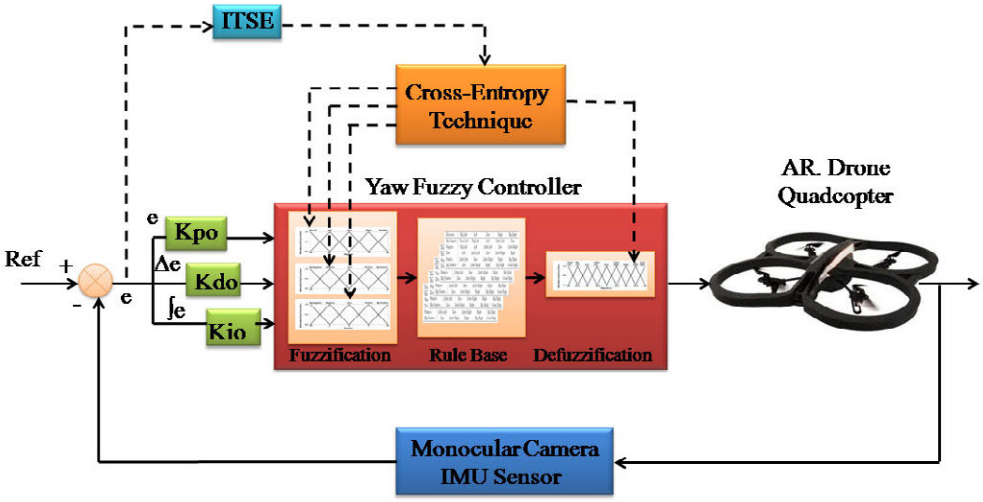


Figure 5.19: Cross-Entropy Optimization for Membership Functions based on the optimized Scaling Factors in FLC.

Figure 5.20 shows the evolution of the PDF for the membership functions of first input (*Left*) of FLC, its optimized value is -89.6960, thus, the optimal membership function for first input (*Right*) is 89.6960. Similarily, the optimal membership function for second input (*Negative*), third input (*Negative*) are -8.1166 and -9.9782, respectively. Thus, the optimal membership function for second input (*Positive*), third input (*Positive*) are 8.1166 and 9.9782, respectively. The evolution of the PDF for second and third input have been shown in Figure 5.21 and 5.22.

Figure 5.23 shows the evolution of the PDF for the membership functions of output (*Big Left*) of FLC, its optimized value is -88.974, thus, the optimal membership function for output (*Big Right*) is 88.974. Similarily, the optimal membership function for output *Left* and *Little Left* are -88.191 and -74.952, respectively. Hence, the optimal membership function for output *Right* and *Little Right* are 88.191 and 74.952, respectively. The evolution of the PDF for output *Left* and *Little Left* also have been shown in Figure 5.24 and 5.25. In the MFs optimization stage, its winner iteration is the 93th iteration in 100 iterations.

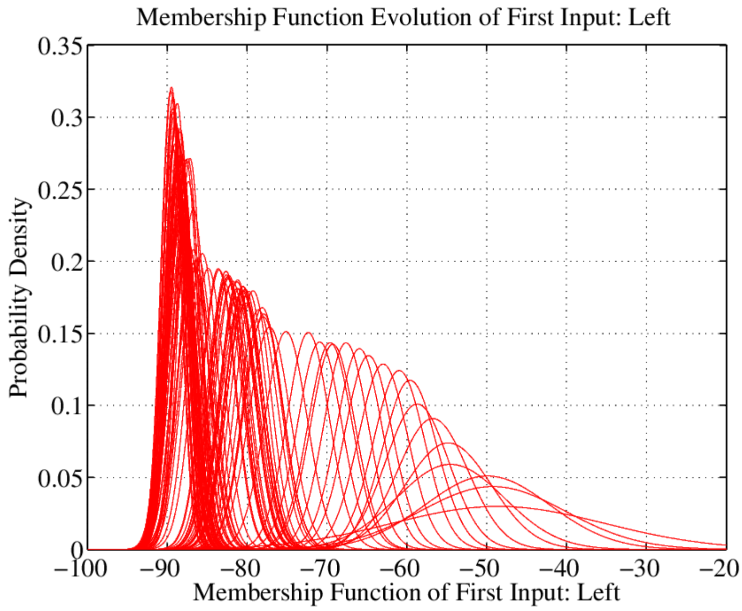


Figure 5.20: The evolution of the PDF for the Membership Function of first input (*Left*) in FLC using CE method. The optimal Membership Function for *Left* is -89.6960, then, the optimal *Right* is 89.6960.

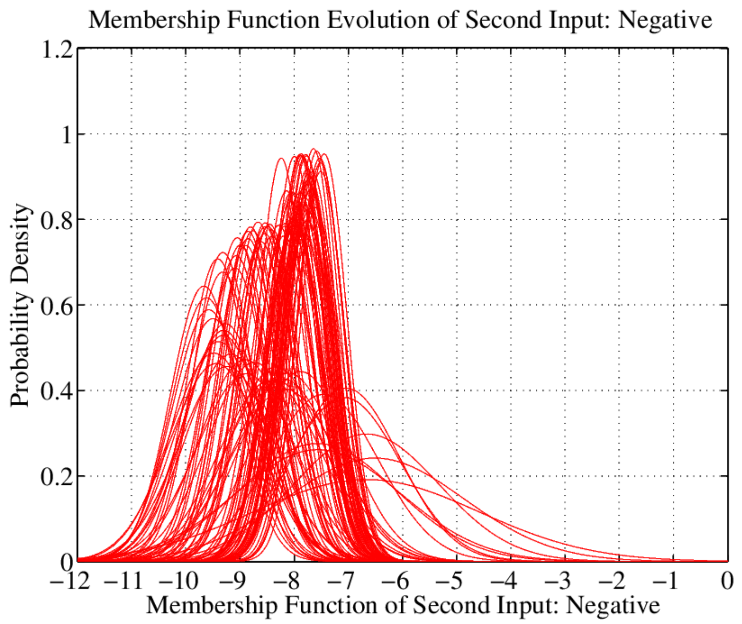


Figure 5.21: The evolution of the PDF for the Membership Function of second input (*Negative*) in FLC using CE method. The optimal Membership Function for *Negative* is -8.1166, then, the optimal *Positive* is 8.1166.

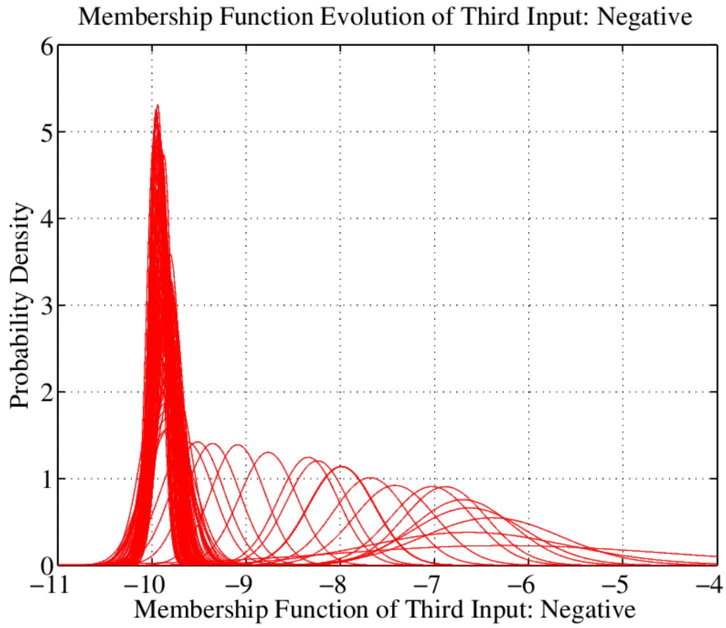


Figure 5.22: The evolution of the PDF for the Membership Function of third input (*Negative*) in FLC using CE method. The optimal Membership Function for *Negative* is -9.9782, then, the optimal *Positive* is 9.9782.

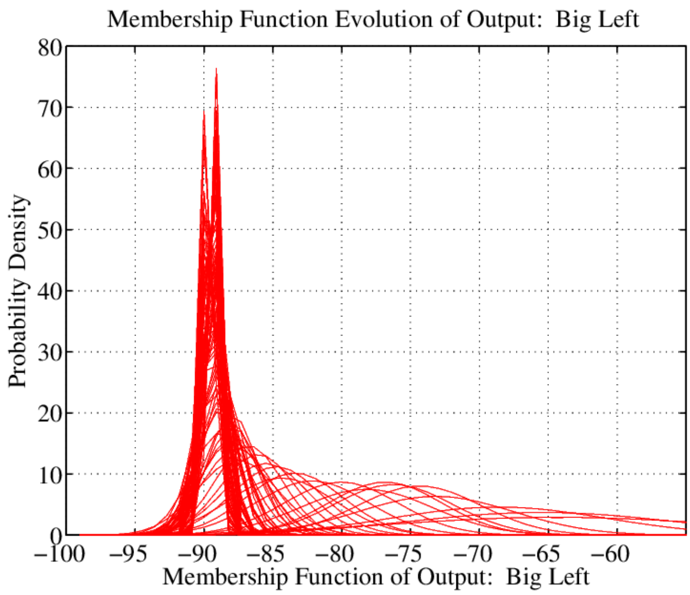


Figure 5.23: The evolution of the PDF for the Membership Function of output (*Big Left*) in FLC using CE method. The optimal Membership Function for *Big Left* is -88.974, then, the optimal *Big Right* is 88.974.

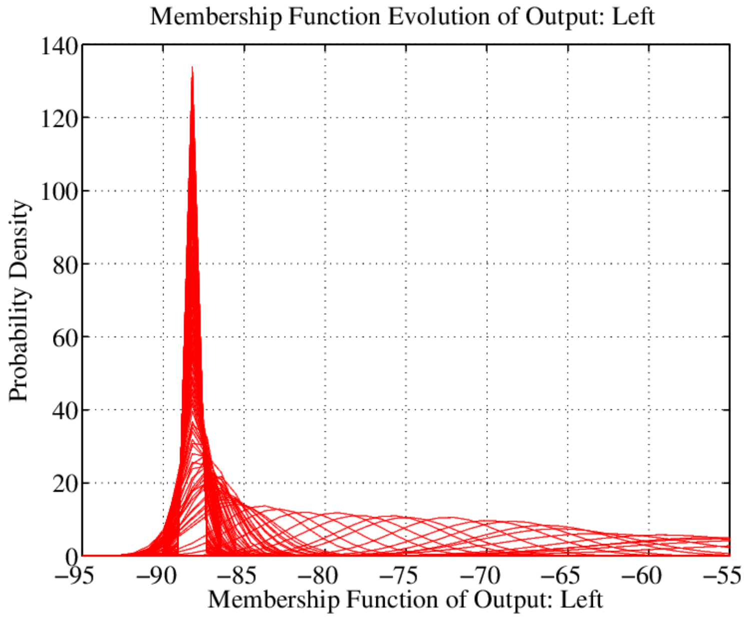


Figure 5.24: The evolution of the PDF for the Membership Function of output (*Left*) in FLC using CE method. The optimal Membership Function for *Left* is -88.191, then, the optimal *Right* is 88.191.

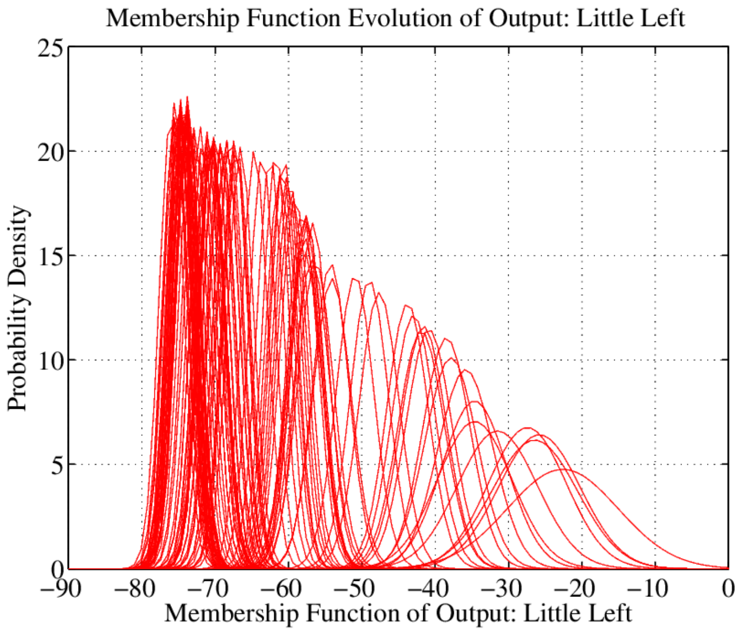


Figure 5.25: The evolution of the PDF for the Membership Function of output (*Little Left*) in FLC using CE method. The optimal Membership Function for *Little Left* is -74.952, then, the optimal *Little Right* is 74.952.

After CE-based optimization for MFs, the new MFs have been generated for FLC, as shown in the Figure 5.26 to 5.29, where, some sets of membership functions are nearly overlapped between each other, therefore, some sets can be deleted. Two sets of membership functions has been reduced in Figure 5.26 and 5.28, and four sets are deleted in Figure 5.29.

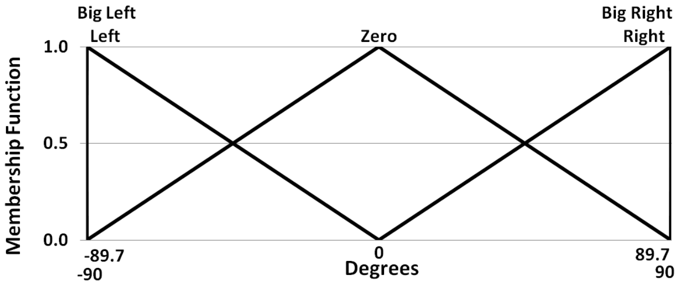


Figure 5.26: MFs for the *First input* (Yaw Error), *after* CE optimization, where, the *Left* (*Right*) has been optimized to -89.6960 (89.6960).

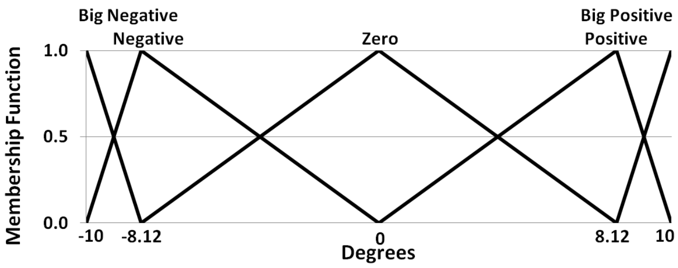


Figure 5.27: MFs for *Second input* (Derivative of Yaw Error), *after* CE optimization, where, the *Negative* (*Positive*) has been optimized to -8.1166 (8.1166).

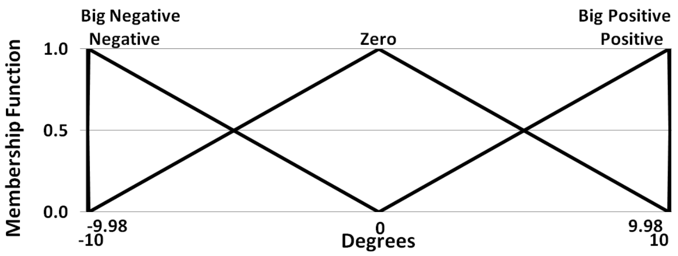


Figure 5.28: MFs for *Third input* (Integral of Yaw Error), *after* CE optimization, where, the *Negative* (*Positive*) has been optimized to -9.9782 (9.9782).

These reductions lead to the cancellation of rules directly. Table 5.6, 5.7 and 5.8 shows the final rule base, 64% of rules has been cancelled from 125 rules to 45 rules, where, Table 5.5 (25 rules) and 5.3 (25 rules) have been cancelled, 10 rules have been reduced in Table 5.1, 5.2 and 5.4, respectively.

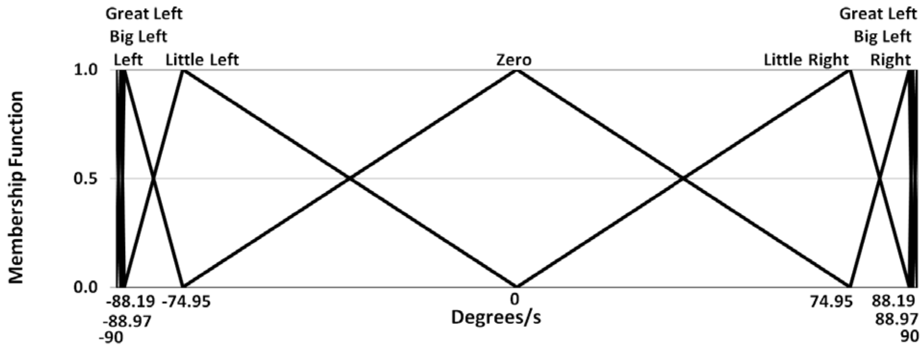


Figure 5.29: MFs for *Output* (Yaw Command), *after* CE optimization, where, the *Big Left, Left, Little Left* (*Big Right, Right, Little Right*) have been optimized to -88.974, -88.191, -74.952 (88.974, 88.191, 74.952).

Table 5.6: Rules based on the *Zero* in the third input (integral of the error)

Dot error/error	Left	Zero	Right
Big Negative	Left	Left	Little Left
Negative	Left	Little Left	Zero
Zero	Little Left	Zero	Little Right
Positive	Zero	Little Right	Right
Big Positive	Little Right	Right	Right

Table 5.7: Rules based on the *Negative* in the third input (integral of the error)

Dot error/error	Left	Zero	Right
Big Negative	Left	Little Left	Zero
Negative	Little Left	Zero	Little Right
Zero	Zero	Little Right	Right
Positive	Little Right	Right	Right
Big Positive	Right	Right	Right

Table 5.8: Rules based on the *Positive* in the third input (integral of the error)

Dot error/error	Left	Zero	Right
Big Negative	Left	Left	Left
Negative	Left	Left	Little Left
Zero	Left	Little Left	Zero
Positive	Little Left	Zero	Little Right
Big Positive	Zero	Little Right	Right

5.1.6. Real Flights and Discussions

In the Section 5.1.5, we have obtained the optimal SFs and MFs for FLCs, thus, a large number of real flight tests should be done. A quadcopter, i.e. AR.Drone Parrot platform, is used to test with the FLCs, it connects to the ground station via wireless LAN.

In this section, we use two different type of CE optimized FLCs: (I) only SF is optimized in FLC (called SF-FLC) (Olivares-Mendez et al., 2013)(Olivares-Mendez et al., 2012); (II) both SF and MF are optimized in FLC (named SFMF-FLC). And the flight speeds, i.e. $0.4m/s$, $0.6m/s$ and $0.8m/s$, are selected to compare their control performances, the size of Safe Avoiding Area (SAA) is 3 meter. Thus, the collision avoiance process is that the Fail-Safe UAV flies one meter toward the obstacle in Normal Fly Area (NFA), then the reference command (90^o) is sent in SAA, and visual fuzzy servoing is activated to avoid collision. Additionally, we set a threshold, e.g. 5 meters, to the X-axis movements in order to compare all the tests in the same conditions, as shown in the Figure 5.33, 5.36 and 5.39.

Figure 5.30 shows the external images and real-time processing images in the tests, where, the first column: forward flight (Yaw Estimation: 0.068^o), the second column: avoiding with little turning (Yaw Estimation: 32.585^o) and the last column: finish the see-and-avoid task (Yaw Estimation: 90.506^o).

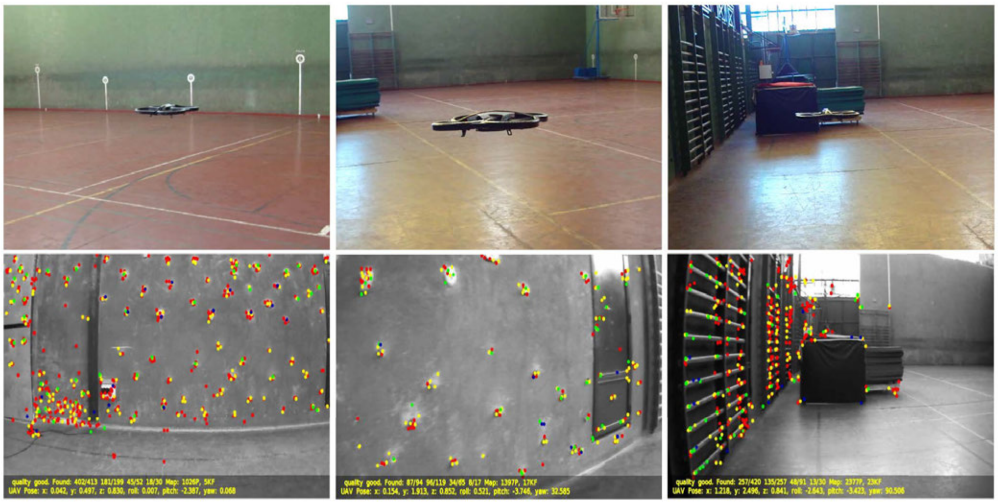


Figure 5.30: UAV in collision avoidance task.

Test 1: Comparison with Flight Speed: $0.4m/s$

Figure 5.31 shows the measurements of yaw angle in the whole collision avoiance task, and Figure 5.32 is the enlarged image to show the performance in the steady state. With this flight speed, SF-FLC and SFMF-FLC avoided the obstacle successfully and did not fly into the Dangerous Alarm Area (DAA). However, the average RMSE is 4.812 degree for SF-FLC, while the average RMSE is 2.583 degree for SFMF-FLC in all the tests. The SFMF-FLC's control performance is better than the one in SF-FLC.

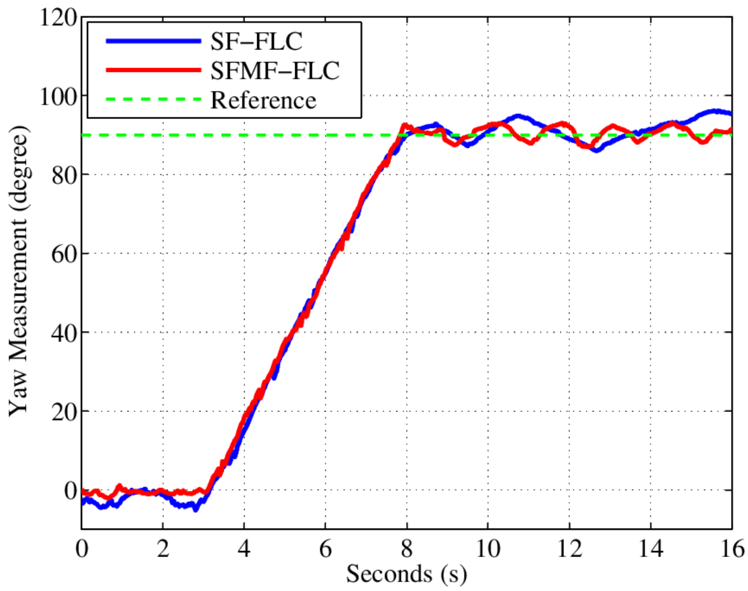


Figure 5.31: Measurements of Fail-Safe UAV' heading in the whole collision avoiance task, the flight speed is $0.4m/s$.

Figure 5.33 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic heading angles for Fail-Safe UAV using the SFMF-FLC.

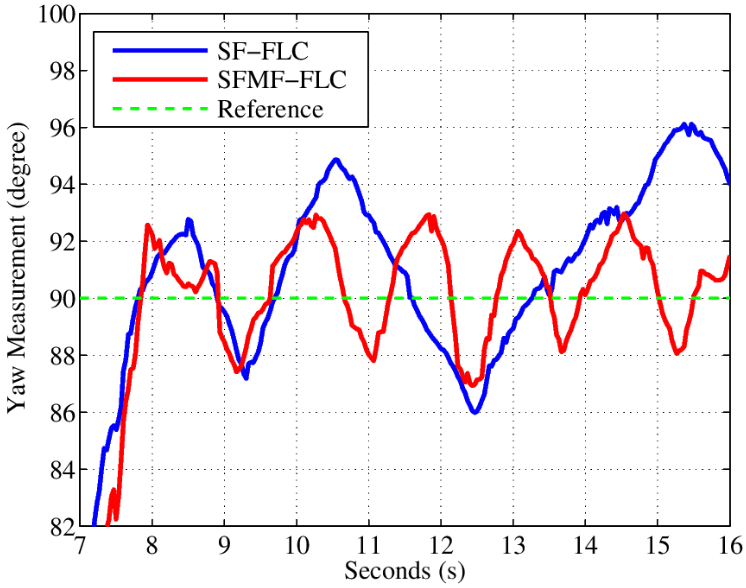


Figure 5.32: Enlarged image for steady state performances.

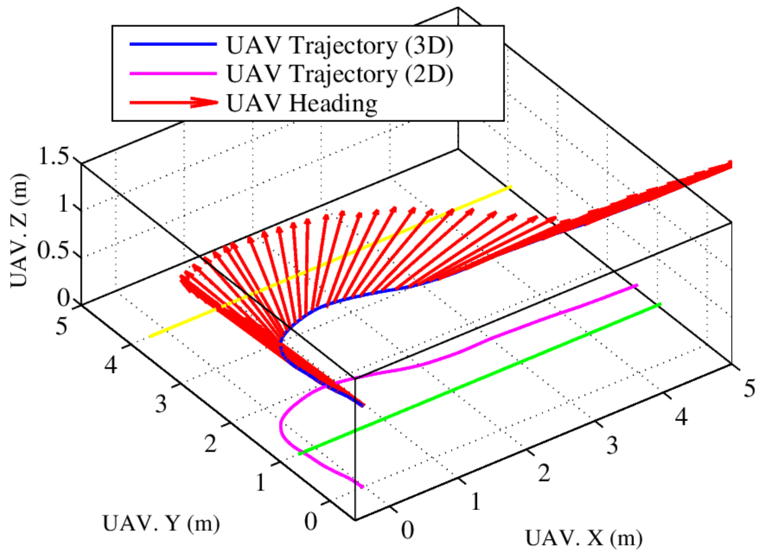


Figure 5.33: 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic change of heading angle, where, along with Y-axis, NFA: 0-1m . SAA: 1-4m. DAA: 4-5m. Obstacle: 5m.

Test 2: Comparison with Flight Speed: $0.6m/s$

Figure 5.34 shows the heading of Fail-Safe UAV in the whole see-and-avoid task, and Figure 5.35 is the enlarged image to show the performance in steady state. Similarily, with this flight, both controllers also avoided the obstacle successfully and did not fly into the DAA. However, the average RMSE is 6.060 degree for SF-FLC, while the average RMSE is 3.218 degree for SFMF-FLC in all the tests. For the control performances, the SFMF-FLC outperforms the SF-FLC.

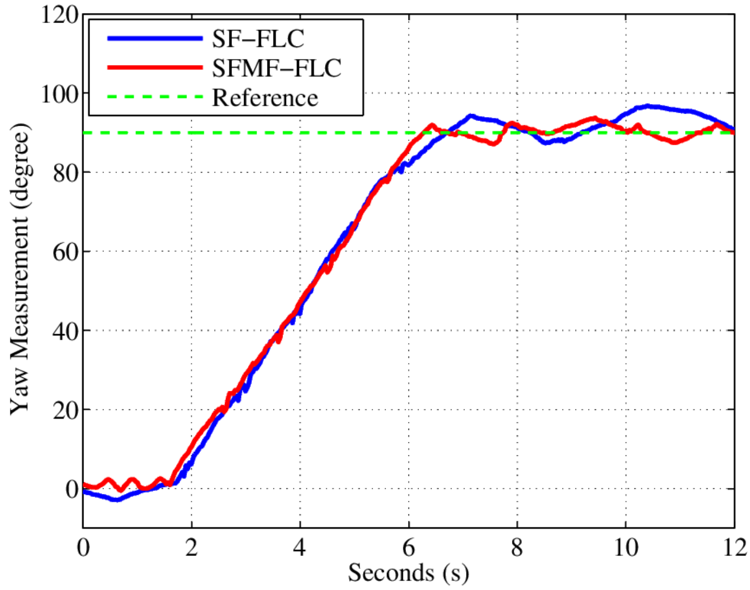


Figure 5.34: Measurements of Fail-Safe UAV' heading in the whole collision avoiance task, the flight speed is $0.6m/s$.

Figure 5.36 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic heading angles for Fail-Safe UAV using the SFMF-FLC.

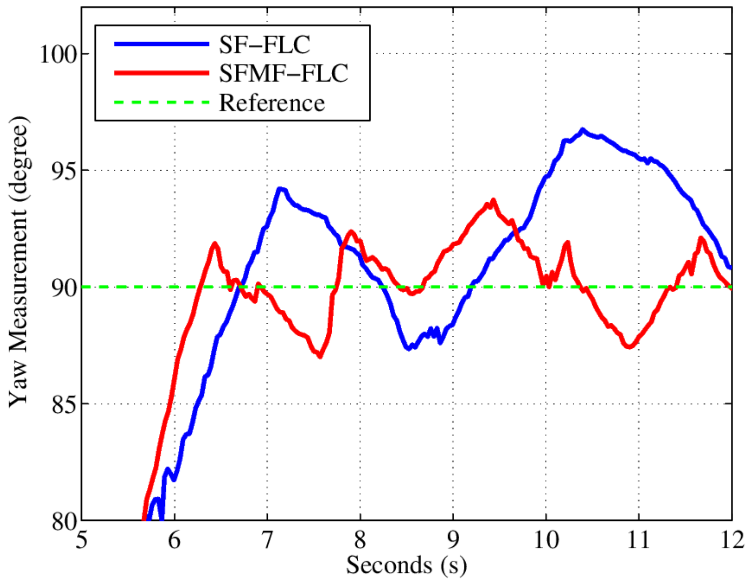


Figure 5.35: Enlarged image for steady state performances.

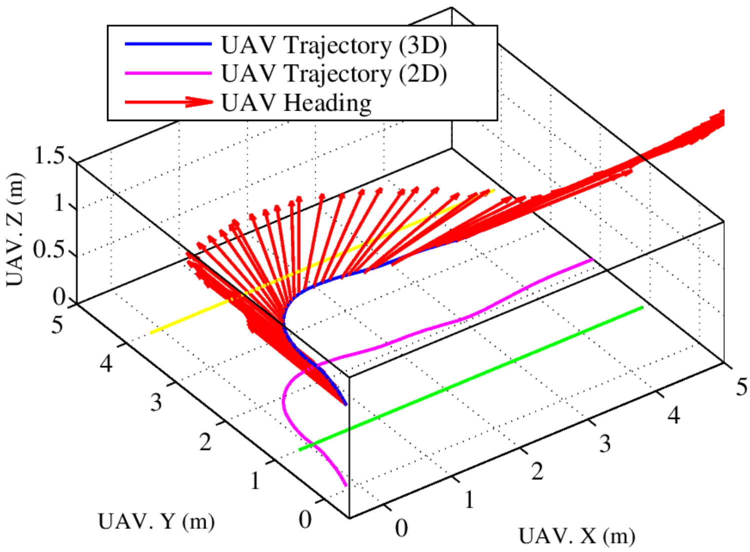


Figure 5.36: 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic change of heading angle.

Test 3: Comparison with Flight Speed: $0.8m/s$

Figure 5.37 shows the Fail-Safe UAV's yaw angle in the whole collision avoiance task, and Figure 5.38 is the enlarged image to show the performance in steady state. Similarily, both controllers avoided the obstacle successfully and did not fly into the dangerous alarm area. However, the average RMSE is 7.427 degree for SF-FLC, while the average RMSE is 4.069 degree for SFMF-FLC in all the tests. For the control performances, the SFMF-FLC is superior to the SF-FLC.

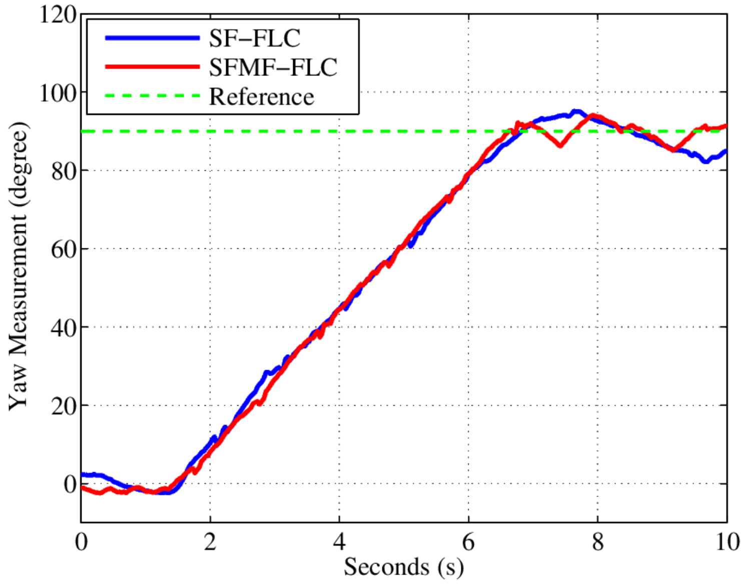


Figure 5.37: Measurements of Fail-Safe UAV' heading in the whole collision avoiance task, the flight speed is $0.8m/s$.

Figure 5.39 shows the 2D and 3D reconstructions of trajectories and the corresponding dynamic heading angles for Fail-Safe UAV using the SFMF-FLC.

In general terms, all the tests show that by adopting a SFMF-FLC, the Fail-Safe UAV can obtain the better control performances.

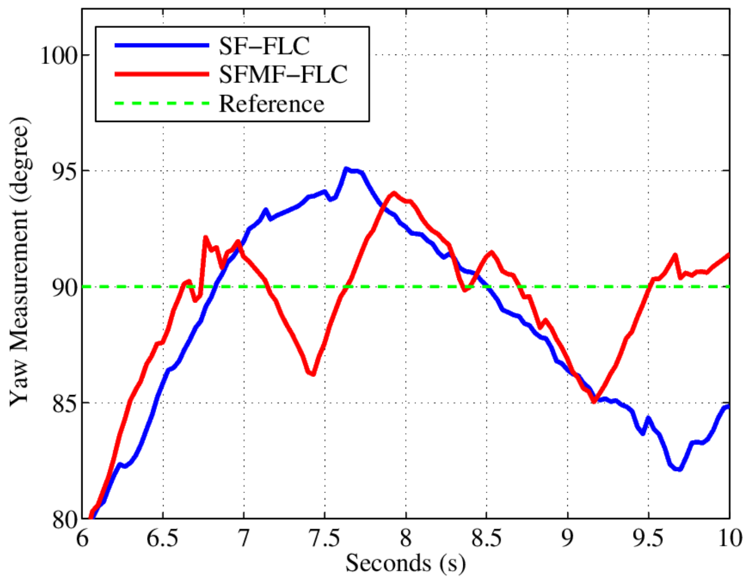


Figure 5.38: Enlarged image for steady state performances.

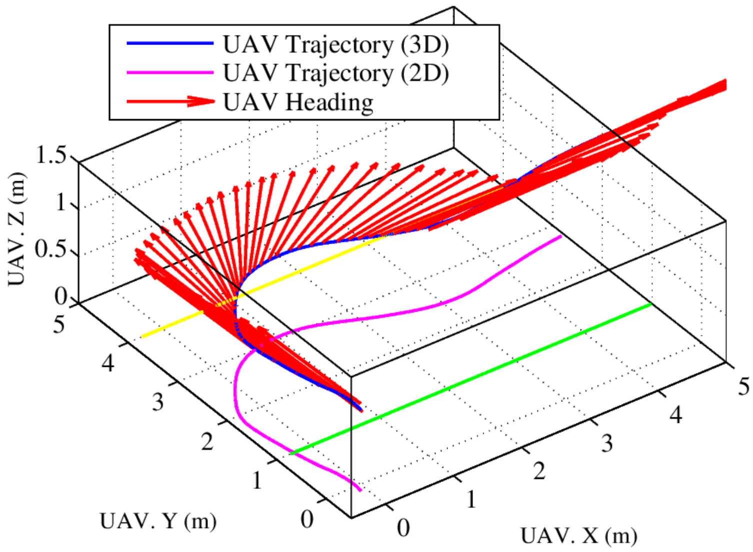


Figure 5.39: 2D and 3D recontructions for Fail-Safe UAV' trajectories and dynamic change of heading angle.

Chapter 6

Conclusions and Future Works

Monocular and stereo cameras have been utilized as the competitive and promissing sensors for Unmanned Aerial Vehicle (UAV) to carry on different vision-based applications. The rich visual information extracted from the input image frames can be applied in the real-time robust visual tracking, visual odometry and 3D reconstruction. Moreover, the vision-based estimation can be adopted as the input of controller to navigate the motion of UAV for achieving autonomous flight.

This dissertation has supplied novel accurate robust **monocular** or **stereo vision-based solutions** for typical UAVs, e.g. ARDrone Parrot, Asctec Pelican and fixed-wing UAV platform, at the real-time rates to achieve the **autonomy** in various types of UAV indoor and outdoor applications, especially in **GPS-denied dynamic cluttered large-scale** environments, e.g. autonomous landing, offshore floating platform inspection, midair aircraft or intruder tracking, dense 3D reconstruction, Fuzzy Logic Controller (FLC)-based see-and-avoid.

Developing the real-time robust visual tracking algorithms onboard UAV to track an arbitrary 2D or **3D** object is the one of main objectives in this dissertation. In the Chapter 2, the state-of-art algorithms related to visual tracking have been presented, it can be found that the most of visual tracking algorithms have took advantages of color information, the approaches based on the features or direct pixel intensity, off-line machine learning methods to track the object. In this dissertation, real-time **on-line learning**-based visual algorithms have been utilized to deal with the tracking problems under the challenging situations, e.g. ignificant appearance change, different camera viewpoint, variant illumination intensity, cluttered tracking background, partial object occlusion, rapid pose variation and onboard mechanical vibration. The online adaptive visual tracking often has obtained the promissing performances with more test samples or when the object moving distance is less than its searching radius. In this dissertation, we have provided different solutions, e.g. **multiple-resolution** strategy, **multiple-classifier** approach, **multiple-particle filter** method and **multiple-instance learning** algorithm, to solve these problems, and made the visual algorithms working faster and more accurate. In addition, the performances of our visual tracker outperforms the state-of-art visual algorithms. Here, we have concluded the advantages of these solutions we adopted in our visual tracking algorithms:

- Multiple-resolution strategy: it mainly solved the problems of the strong motions (e.g. onboard mechanical vibration) or large displacements over time on the image frame. Additionally, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAV and Ground Control Station (GCS).

- Multiple-classifier approach: its voting mechanism can help to reject test samples based on classifier scores, i.e. the lower resolution features are initially applied in rejecting the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, and select the searching radius in the next higher solution of image frame.

- Multple-instance learning algorithm: it can help to solve the ambiguity problem, i.e. the correct sample features will be selected to on-line update the binary classifier, thereby reducing the noises introduced from visual tracking.

- Multiple-particle filter method: besides the advantages mentioned in Multiple-classifer approach, it has been utlized to estimate the different motion models based on the useful visual information in different resolutions of image frame.
- Multiple-block size approach: it can help to select the frequence for updating the appearance of tracking object, i.e. a smaller block size means more frequent updates, making it quicker to model the appearance changes.

For future work in visual tracking algorithm, we will consider the improvements as follows:

- developing **multiple-object** tracking algorithm using on-line learning methods, as the autonomous landing application for UAV shown in Appendix A.3.
- incorporating the **scale adaptation** in discriminative-based visual tracking methods
- applying the **off-line machine learning** method for recovering the tracking location of object

Real-time estimating the 6D pose of UAV and mapping the surrounding flight environment are the other objectives in this dissertation. Although many works have utilized monocular camera to carry on the visual odometry, but the monocular visual odometry is not adequately estimate the real absolute scale (i.e. scale ambiguity) to the observed environments, especially in large-scale environments, generating accumulated trajectory drift. In the Chapter 4, the stereo camera has been applied as minimum number configuration of cameras for solving scale ambiguity problem, i.e. the depth information can be effectively estimated based on the baseline between the left and the right cameras, thereby supplying preferable 6 Degrees-Of-Freedom (DOF) pose estimation and real-size environment information for UAV. A novel stereo visual odometry algorithm is effectively takes advantage of both 2D (without depth) and 3D (with depth) information to estimate the 6D pose **between each two consecutive image pairs**. In this dissertation, a new light small-scale low-cost embedded stereo vision system for UAV has been designed because of the limitations of typical UAV and available commercial stereo cameras, i.e. a typical UAV has the limiation of size, payload, computation capability, power supply and expanded mounting space for other sensors, and the available commercial stereo cameras are high cost, big weight or incompatible communication interface. The new embedded stereo vision system also can be employed to pre-process the rich information, thereby saving more computing capability for the onboard host computer to execute other online tasks.

For future work in stereo visual odometry and mapping algorithm, the new modules will be added:

- adopting a **new visual cue** for visual tracking module, i.e. Canny edge information, instead of using FAST detector, because it can be find that the FAST features are prone to be **clustered** [1], if a large number of FAST feature are not set in the tracking module, the FAST features are difficult to be evenly distributed over the image frame, however, a large number of FAST features will result in expensive computation. It is worthing to notice that **all** canny edge information in the image will **not** be utilized for tracking, only **critical** canny edge information will be applied, i.e. **CEPIG feature**, the details of this CEPIG feature have been introduced in Appendix A.1.

- developing a **global** loop closure module for reducing the whole UAV trajectory drift, and achieving the full **stereo V-SLAM**. It is worthing to notice that the **re-visited place recognition** has been finished, the detaild of this palce recognition module based on Bag-of-Binary Words (**DBOW2**) has been introduced in Appendix A.2.

- adding an Inertial Measurement Unit (**IMU**) sensor to our newly designed stereo vision system.

Designing the vision-based controller to navigate the motion of UAV in different GPS-denied indoor and outdoor environments is the final objective in this dissertation, as presented in the Chapter 5. The developed visual algorithms in the former two chapters can be adopted for various typical UAVs to robustly estimate their states at the real-time rates. In literature, many works have utilized the traditional PID control, Sliding mode control, Backstepping control and robust H_∞ control, however, the mathematics model of each UAV should be exactly identified in advanced. In real UAV controlling problems, the uncertainty, inaccuracy, approximation and incompleteness problems are widely included in the UAV systems.

[1] The distance between each neighbour FAST feature is very small in a local image area, other point-like features also have the same property.

Since the Fuzzy Logic Controller (FLC) often has the good robustness and adaptability in the highly nonlinear, dynamic, complex and time varying UAV systems, this *model-free* control approach as one of the most active and fruitful soft computing methods have been applied for quadrotor UAV platform in our vision-based applications.

For a FLC, the Scaling Factor (SF), Membership Function (MF) and Rule Weight (RW) can be tuned from *macroscopic* to *microscopic* effects to improve the control performance of UAV, i.e. SF adjustment, MF modification and RW regulation. In many works, the FLCs have been optimized with different Artificial Intelligence (AI) algorithms, e.g. genetic algorithm, simulated annealing, ant colony optimization, particle swarm optimization, bees algorithm. In this dissertation, a Fuzzy Logic Controller (FLC) has been optimized by a new approach, i.e. the Cross-Entropy (CE) framework, for UAV, both scaling factors and membership functions have been optimized. In the UAV vision-based collision avoidance application, we have demonstrated that the FLC with SF and MF optimization is more accurate than only SF optimization-based FLC. And this novel CE optimization has not just improved the behavior of FLC, but also reduced 64% of the initial rule base.

For the future works, some works have been considered as follows:

- optimizing the **rule weights** of FLC, it will casue Microscopic effects to the behavior of FLC for further improving the control peformance of UAV.

- optimizing the scaling factors, membership functions and the weights of rule for other Degree-Of-Freedom (**DOF**) of UAV.

- considering the FLC utilized in this dissertation is a speed-based controller, the **position**-based FLC will be optmized from from *macroscopic* to *microscopic* effects, it is important for UAV to carry on the high-level vision-based applications.

Appendix A

Other Developed Algorithms

The other developed algorithms listed in this Appendix will be utilized to improve the presented monocular or stereo vision-based solutions.

A.1. CEPIG Feature Detection and Tracking

The **cluster** problem is widely exsited in the most of point-like features, as shown in Fig. A.1.

However, the edge information often represents environments better, therefore, a new edge-based sparse feature detection is introduced as following:

Firstly, the gray values of the color RGB image I are utilized, i.e.

$$I = \frac{I^R + I^G + I^B}{3} \tag{A.1}$$

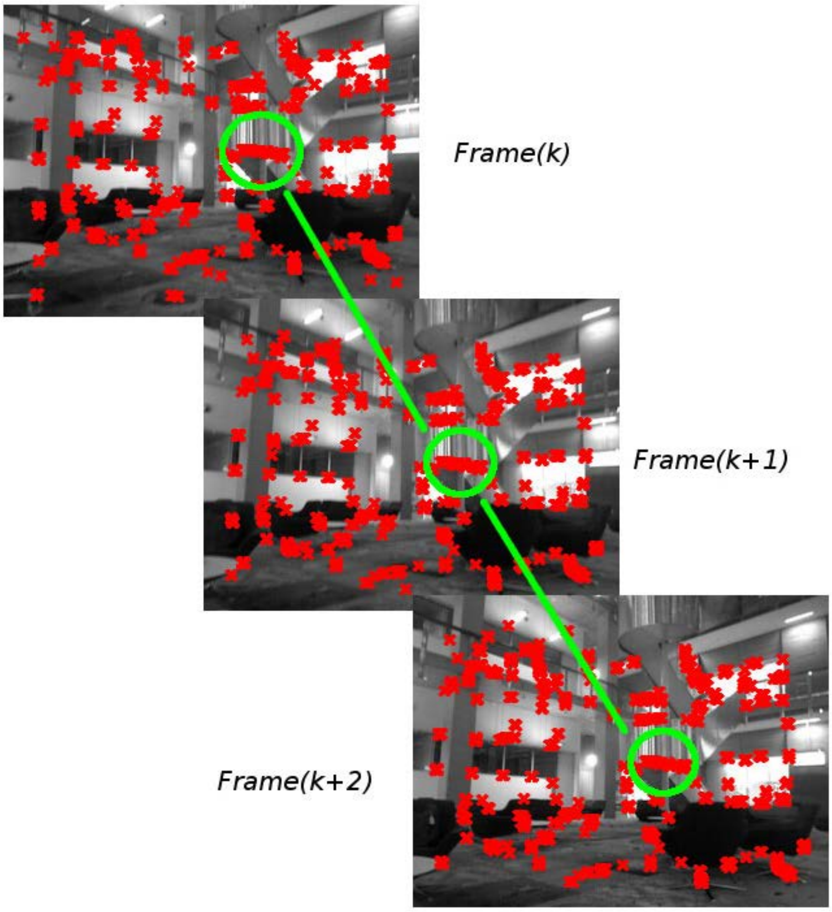


Figure A.1: Harris corner detection with **cluster** problem. The maximum tracking number is set to **800**.

and a Gaussian blur filter is applied for smoothing the gray image to reduce noises.

The maximum number of CEPIG feature to track in each smoothed gray image is N. Then, the smoothed gray image is partitioned with $N = N_x \times N_y$ non-overlapped grids, the size of the ith grid (G^i, $i = 1, 2, 3, \ldots, N$) is defined as:

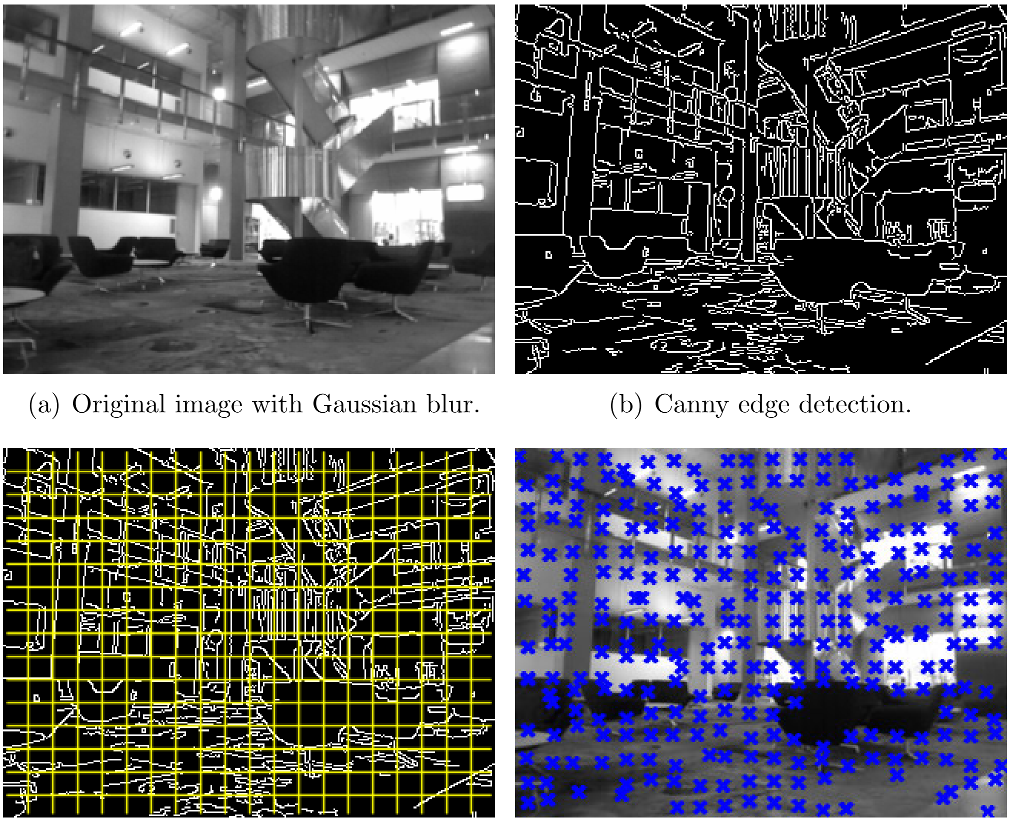
$$G_W^i = \frac{I_W - 2 \times O_W}{N_x}, \qquad G_H^i = \frac{I_H - 2 \times O_H}{N_y} \tag{A.2}$$

where, G_W^i and G_H^i represent the width and height of the ith grid, I_W and I_H are the width and height of gray image, $I_W - 2 \times O_W$ and $I_H - 2 \times O_H$ represent the width and height of interest area located on the center of gray image.

The Canny edge detector (Canny, 1986) is used to find edges in the ith grid, then, the Canny Edge Pixel of Interest in Grid, i.e. CEPIG feature, is defined as:

$$P_{ij}^{*} = \arg\min_{\mathbf{l}_{P_{ij}}} \{ \| \mathbf{l}_{P_{ij}} - \mathbf{l}_{G^i}^{C} \|^2 \leqslant \sigma \} \tag{A.3}$$

where, $\|*\|^2$ represents Euclidean distance, $\{P_{i1}, P_{i2}, P_{i3}, \ldots, P_{iM}\}$ is the set of detected canny edge pixels in the ith grid, $i = 1, 2, 3, \ldots, N$, $\mathbf{l}_{P_{ij}}$ is the location of the jth canny edge pixel in the ith grid, $j = 1, 2, 3, \ldots, M$, $\mathbf{l}_{G^i}^{C}$ represents the centre location of the ith grid, σ is a distance threshold.



(a) Original image with Gaussian blur. (b) Canny edge detection. (c) Grid configuration. (d) Final CEPIG features.

Figure A.2: The detail of CEPIG feature detection. The maximum tracking number is set to **320**.

The details of CEPIG feature detection has been shown in Fig. A.2 with pictures, it is worthing to notice that the O_W and O_W are set to 0.

After detected the CEPIG features, the BRIEF descriptor (Calonder et al., 2010) is adopted to track them, for a IPEG feature, its location is $\mathbf{l}$, the BRIEF descriptor of it, i.e. binary vector $\mathbf{D}(\mathbf{l})$, is defined as:

$$D_j(\mathbf{l}) = \begin{cases} 1 & \text{if } I_{\mathbf{l}+\mathbf{x}_j} < I_{\mathbf{l}+\mathbf{y}_j} \\ 0 & \text{otherwise} \end{cases}, \forall j \in [1, ..., N_b] \tag{A.4}$$

where, I_* is the intensity value of smoothed gray image, $\mathbf{x}_j$ and $\mathbf{y}_j$ are sampled in a local neighbour window $W_l \times W_l$ based on the location of the CEPIG feature, $\mathbf{x}_j = \mathcal{N}(0, (\frac{1}{5}W_l)^2)$, however, $\mathbf{y}_j = \mathcal{N}(\mathbf{x}_j, (\frac{2}{25}W_l)^2)$, which is different from the definition of original BRIEF descriptor (Calonder et al., 2010). $D_j(\mathbf{l})$ is the jth bit of the binary vector, and N_b is the number of bits in the binary vector, i.e. N_b comparisons. Since the distance of two binary vectors is calculated by counting the number of different bits between them, i.e. Hamming distance, it is faster than computing the Euclidean distance as done in SIFT and SURF features.

Fig.A.3 shows one example result of the CEPIG feature tracking with images captured in the library of our university, the red and blue circles represent the presented CEPIG features and BRIEF-tracked CEPIG features. In addition, the same BRIEF descriptor is also applied for real-time mapping and detecting loop in our stereo V-SLAM, as introduced in Section A.2.

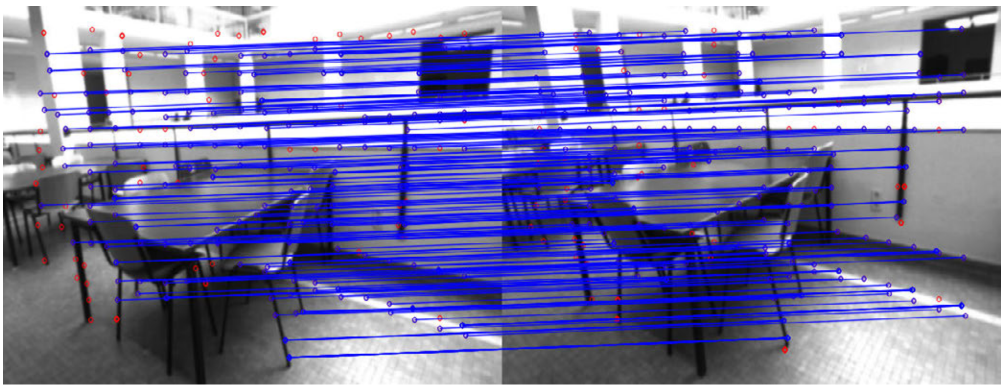


Figure A.3: One example result of the CEPIG feature tracking.

A.2. Re-visited Place Recognition

A re-visited place recognizer module plays a critically important role in one Monocular or Stereo SLAM system for detecting loop closure. (Williams et al., 2009) have compared and analyzed three various types of methods for re-visited place recognition in both indoor and outdoor environments, i.e. map-to-map, image-to-image and image-to-map. They have concluded that the image-to-image method, i.e. appearance-based approach, scales better than other two approaches in larger environments.

This section has presented some place recognition results with the challenging KITTI image datasets [1] using image-to-image matching method, which is based on DBoW2 technique (Gálvez-López and Tardós, 2012) [2] with CEPIG features.

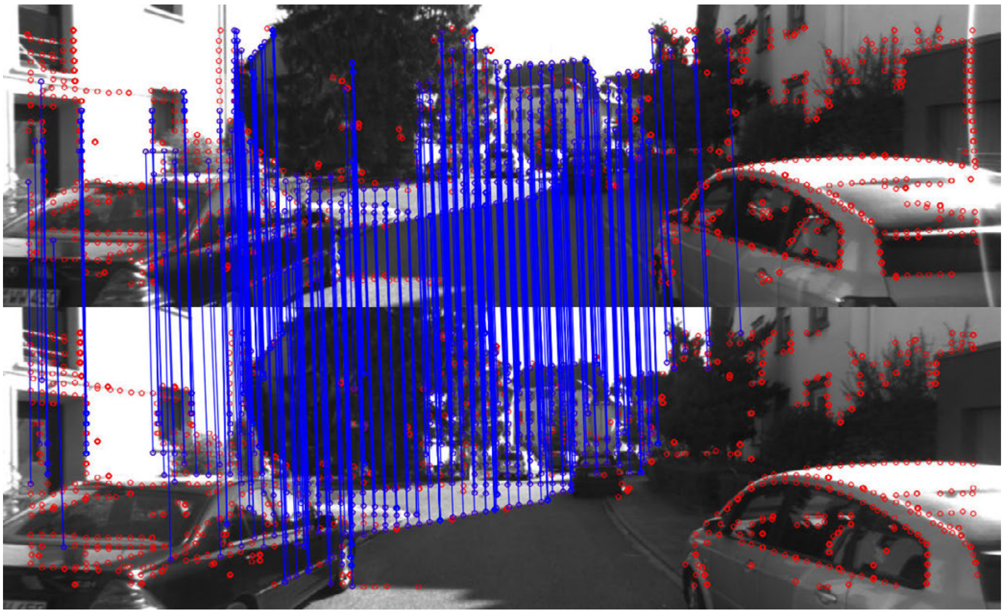


Figure A.4: One example result of the CEPIG feature tracking in KITTI-00.

[1] http://www.cvlibs.net/datasets/kitti/
[2] https://github.com/dorian3d/DBoW2

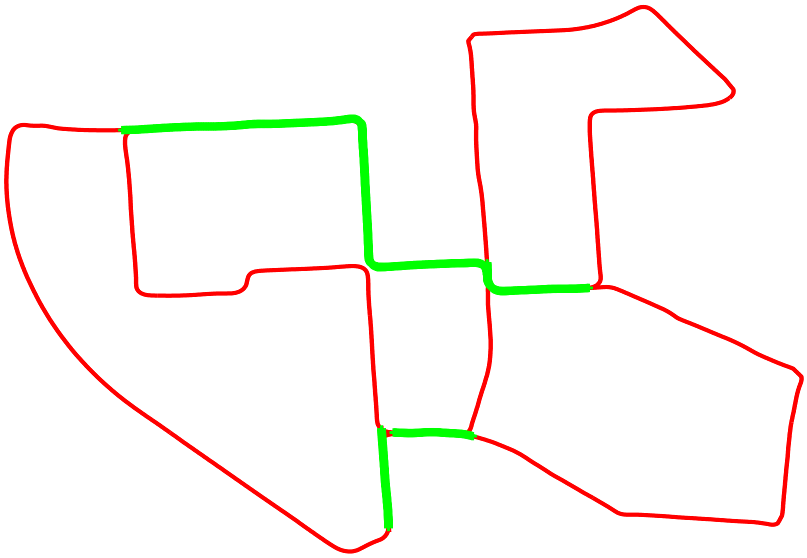


Figure A.5: Recognition of Re-visited Places in KITTI-00. The **Green** color represents the re-visited places.

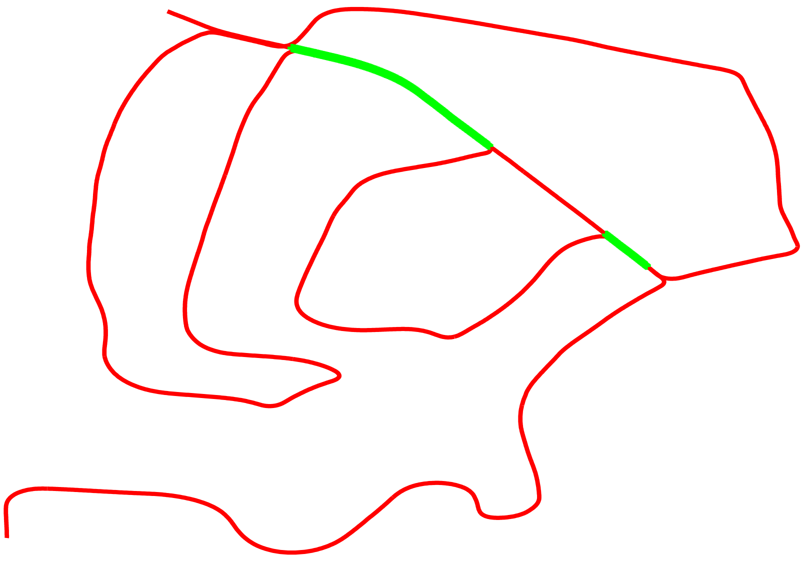


Figure A.6: Recognition of Re-visited Places in KITTI-02.

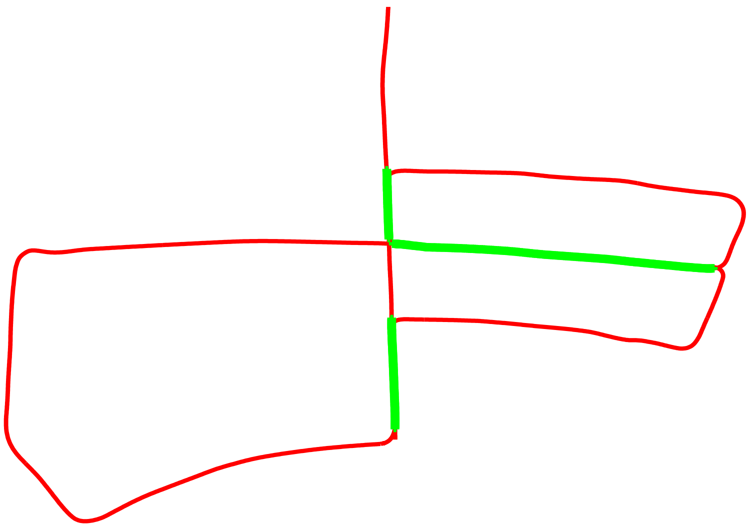


Figure A.7: Recognition of Re-visited Places in KITTI-05.

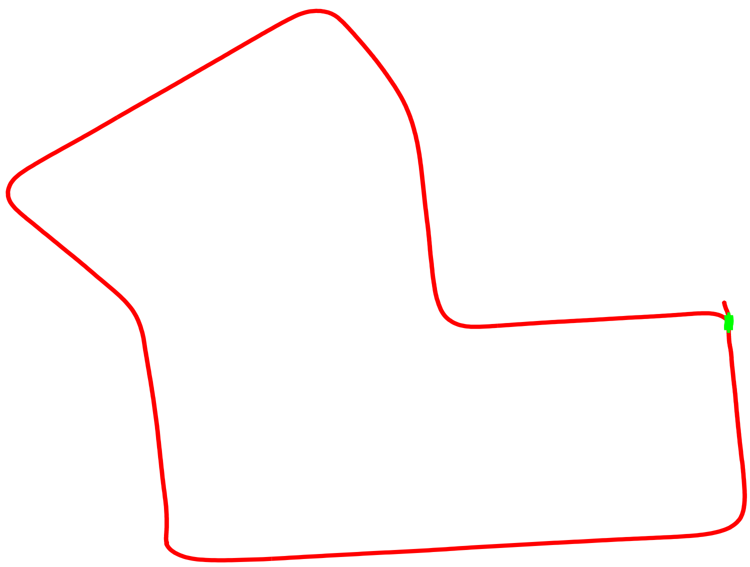


Figure A.8: Recognition of Re-visited Places in KITTI-07.

A.3. Infrared LED-based Autonomous Landing

Recently, monocular vision-based 6D pose estimation approaches have been presented in many works, e.g. (Faessler et al., 2014), (Breitenmoser et al., 2011). Inspired by the above works, a monocular vision-based method using Infrared (IR) LED has been developed for Unmanned Aerial Vehicle (UAV) to carry on the autonomous landing task at real-time frame rates in this work, as shown in Fig. A.9, the uEye camera [3] (type: UI-1221LE-C-HZ, Right-Up) with an Infrared (IR) filter [4] (type: MID-IBP850-D16, it only allows the special light, i.e. its wavelength matches the wavelength of light emitted by IR LEDs, to pass) has been fixed on the bottom of Asctec Pelican quadrotor platform, and IR Light Emitting Diodes (LEDs) have been configured and installed on a landing target (Right-Bottom) with known 3D positions related to the center $\{O\}$ of landing object. To reduce the ambiguity of visual pose estimation, IR LEDs have been set to **non-planar** and **non-symmetric**.

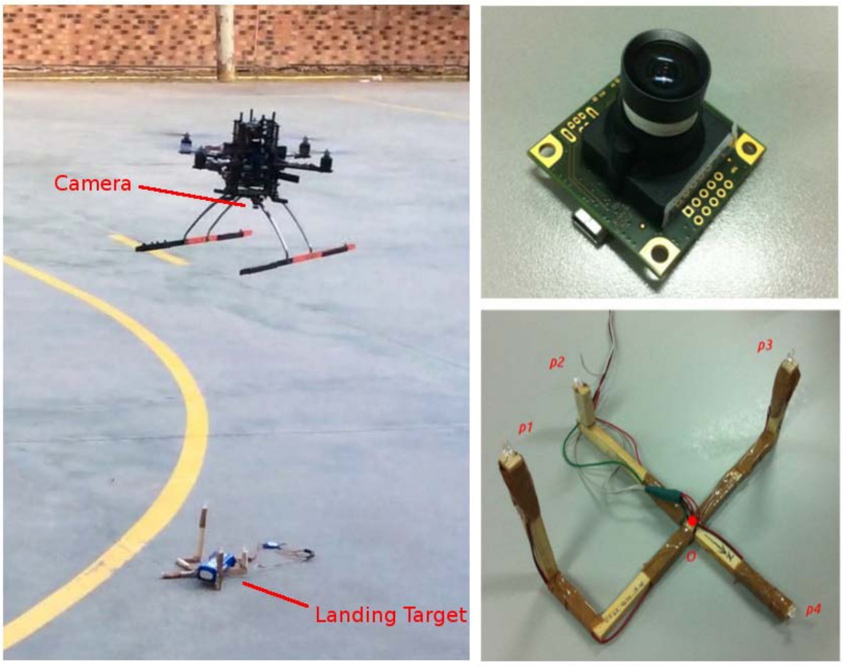


Figure A.9: Infrared LED-based Autonomous Landing for UAV.

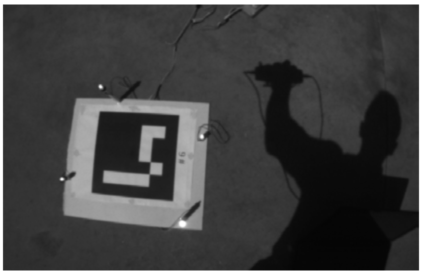
[3]https://en.ids-imaging.com
[4]http://midopt.com

The main steps utilized in the presented visual algorithm are listed:

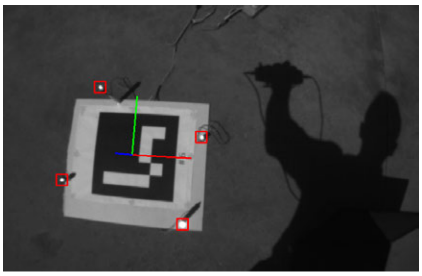
- **Segmenting** the LEDs in the input image frame (threshold the brightest pixels, gaussian filter for smoothing image, blob detection et al).
- **Searching** the correspondences of LEDs based on LED configuration with Perspective-3-Points (P3P) algorithm (Kneip et al., 2011).
- **Optimizing** the initial 6D pose estimation with Levenberg-Marquardt algorithm instead of the Gauss-Newton scheme utilized in the (Faessler et al., 2014).

An IR LED-based 6D pose estimation has been shown in Fig. A.10, and an Augmented Reality (AR) library, i.e. ArUco [5], has been utilized as the Ground Truth (GT) to compare the performance of 6D pose estimation:

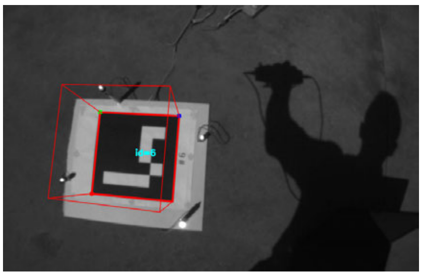
- IR-Pose: (-0.218418, 0.005998, 0.748139, 178.5175, -3.1235, 3.6492)
- AR-Pose: (-0.214517, 0.007943, 0.750164, 173.7716, -5.7299, 2.2774)



(a) Input image frame.



(b) IR LED-based pose estimation.



(c) ArUco-based pose estimation.

Figure A.10: The 6D pose estimation results from IR LED and ArUco.

[5]http://www.uco.es/investiga/grupos/ava/node/26

Appendix B

Publications

The publications, i.e. international journals, international book chapter and international conference papers, derived from this dissertation have been listed as follows. The related ranking information is provided by the 2014th version of the Journal Citation Report (JCR) data.

B.1. International Journals

- **Towards an Autonomous Vision-Based Unmanned Aerial System Against Wildlife Poachers**. MiguelA. Olivares-Mendez, Changhong Fu, Philippe Ludivig, Tegawende Bissyande, Somasundar Kannan, Maciej Zurad, Arun Annaiyan, Holger Voos, Pascual Campoy. Sensors. 2015. MDPI. ISSN: 1424-8220. **IF=2.245, Q1**
Under Review.

- **SIGS: Synthetic Imagery Generating Software for the De-**

velopment and Evaluation of Vision-based Sense-And-Avoid Systems. Adrian Carrio, Changhong Fu, Jean-Francois Collumeau, Pascual Campoy. Journal of Intelligent and Robotic Systems. 2015. Springer. ISSN: 0921-0296. **IF=1.178, Q3**
DOI = http://dx.doi.org/10.1007/s10846-015-0286-z

- **Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers**. Changhong Fu, MiguelA. Olivares-Mendez, Ramon Suarez-Fernandez, Pascual Campoy. Journal of Intelligent and Robotic Systems. 2014. Springer. ISSN:0921-0296. 73 (1-4): 513-533. **IF=1.178, Q3**
DOI = http://dx.doi.org/10.1007/s10846-013-9918-3

- **A General Purpose Configurable Controller for Indoors and Outdoors GPS-Denied Navigation for Multirotor Unmanned Aerial Vehicles**. Jesus Pestana, Ignacio Mellado-Bataller, JoseLuis Sanchez-Lopez, Changhong Fu, IvanF. Mondragon, Pascual Campoy. Journal of Intelligent and Robotic Systems. 2014. Springer. ISSN:0921-0296. 73 (1-4): 387-400. **IF=1.178, Q3**
DOI = http://dx.doi.org/10.1007/s10846-013-9953-0

B.2. Book Chapter

- **FuSeon: a Low-cost Portable Multi Sensor Fusion Research Testbed for Robotics**. JoseLuis Sanchez-Lopez, Changhong Fu, Pascual Campoy. ROBOT2015: Second Iberian Robotics Conference. Advances in Intelligent Systems and Computing. 2015. Springer. ISBN: 978-3-319-03652-6.
Accepted.

- **Floor Optical Flow Based Navigation Controller for Multirotor Aerial Vehicles**. Jesus Pestana, Ignacio Mellado-Bataller, JoseLuis Sanchez-Lopez, Changhong Fu, IvanF. Mondragon, Pascual Campoy. ROBOT2013: First Iberian Robotics Conference. Advances in Intelligent Systems and Computing. 2014. Springer. ISBN: 978-3-319-03652-6. 253: 91-106.
DOI = http://dx.doi.org/10.1007/978-3-319-03653-3_8

B.3. International Conference Papers

- **Efficient Visual Odometry and Mapping for Unmanned Aerial Vehicle Using ARM-Based Stereo Vision Pre-Processing System**. Changhong Fu, Adrian Carrio, Pascual Campoy. International Conference on Unmanned Aircraft Systems (ICUAS). 2015. Denver, Colorado, USA. June 9-12. Pages: 957-962.
  DOI = `http://dx.doi.org/10.1109/ICUAS.2015.7152384`

- **Using the Cross-Entropy Method for Control Optimization: A Case Study of See-And-Avoid on Unmanned Aerial Vehicles**. Miguel A. Olivares-Mendez, Changhong Fu, Somasundar Kannan, Holger Voos, Pascual Campoy. 22nd Mediterranean Conference of Control and Automation (MED). 2014. Palermo, Italy. June 16-19. Pages: 1183-1189.
  DOI = `http://dx.doi.org/10.1109/MED.2014.6961536`

- **Robust Real-Time Vision-Based Aircraft Tracking from Unmanned Aerial Vehicles**. Changhong Fu, Adrian Carrio, Miguel A. Olivares-Mendez, Ramon Suarez-Fernandez, Pascual Campoy. IEEE International Conference on Robotics and Automation (ICRA). 2014. Hong Kong, China. May 31-June 7. Pages: 5441-5446.
  DOI = `http://dx.doi.org/10.1109/ICRA.2014.6907659`

- **Online Learning-Based Robust Visual Tracking for Autonomous Landing of Unmanned Aerial Vehicles**. Changhong Fu, Adrian Carrio, MiguelA. Olivares-Mendez, Pascual Campoy. International Conference on Unmanned Aircraft Systems (ICUAS). 2014. Orlando, FL, USA. May 27-30. Pages: 649-655.
  DOI = `http://dx.doi.org/10.1109/ICUAS.2014.6842309`

- **A Ground-Truth Video Dataset for the Development and Evaluation of Vision-Based Sense-and-Avoid Systems**. Adrian Carrio, Changhong Fu, Jesus Pestana, Pascual Campoy. International Conference on Unmanned Aircraft Systems (ICUAS). 2014. Orlando, FL, USA. May 27-30. Pages: 441-446.
  DOI = `http://dx.doi.org/10.1109/ICUAS.2014.6842284`

- **UAS See-And-Avoid Strategy using a Fuzzy Logic Controller Optimized by Cross-Entropy in Scaling Factors and Membership Functions**. Changhong Fu, MiguelA. Olivares-Mendez, Pascual

Campoy, Ramon Suarez-Fernandez. International Conference on Unmanned Aircraft Systems (ICUAS). 2013. Atlanta, GA, USA. May 28-31. Pages: 532-541.
DOI = http://dx.doi.org/10.1109/ICUAS.2013.6564730

- **A General Purpose Configurable Navigation Controller for Micro Aerial Multirotor Vehicles**. Jesus Pestana, Ignacio Mellado-Bataller, JoseLuis Sanchez-Lopez, Changhong Fu, IvanF. Mondragon, Pascual Campoy. International Conference on Unmanned Aircraft Systems (ICUAS). 2013. Orlando, FL, USA. May 27-30. Pages: 557-564.
DOI = http://dx.doi.org/10.1109/ICUAS.2013.6564733

- **Toward Visual Autonomous Ship Board Landing of a VTOL UAV**. JoseLuis Sanchez-Lopez, Srikanth Saripalli, Pascual Campoy, Jesus Pestana, Changhong Fu. International Conference on Unmanned Aircraft Systems (ICUAS). 2013. Orlando, FL, USA. May 27-30. Pages: 779-788.
DOI = http://dx.doi.org/10.1109/ICUAS.2013.6564760

- **Real-Time Adaptive Multi-Classifier Multi-Resolution Visual Tracking Framework for Unmanned Aerial Vehicles**.
Changhong Fu, Ramon Suarez-Fernandez, MiguelA. Olivares-Mendez, Pascual Campoy. 2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS). 2013. Compiegne, France. November 20-22. Pages: 99-106.
DOI = http://dx.doi.org/10.3182/20131120-3-FR-4045.00010

- **Visual Identification and Tracking for Vertical and Horizontal Targets in Unknown Indoor Environment**. Changhong Fu, Jesus Pestana, Ignacio Mellado-Bataller, JoseLuis Sanchez-Lopez, Pascual Campoy. International Micro Air Vehicle Conference (IMAV). 2012. Braunschweig, Germany. July 3-6.

- **A Visual Guided Quadrotor for IMAV 2012 Indoor Autonomy Competition and Visual Control of a Quadrotor for the IMAV 2012 Indoor Dynamics Competition**. Jesus Pestana, Ignacio Mellado-Bataller, Changhong Fu, Jose Luis Sanchez-Lopez, Ivan Fernando Mondragon, Pascual Campoy. International Micro Air Vehicle Conference (IMAV). 2012. Braunschweig, Germany. July 3-6.

- **AR Drone Identification and Navigation Control at CVG-UPM**. Jesus Pestana, Jose Luis Sanchez-Lopez, Ignacio Mellado-Bataller, Changhong Fu, Pascual Campoy. XXXIII Jornadas Nacionales de Automatica.

B.4. Digital Media

Digital media related to this dissertation is available at the PhD Thesis Section of CVG-UPM official website:
`http://www.vision4uav.com/?q=phdthesis`

Appendix C

Project and Research Exchange

The obtained research results from different types of projects and international research exchange have contributed to this dissertation. They are listed as follows:

C.1. International Project

- **UECIMUAVS**: USA and Europe Cooperation in Mini UAVs. (Researcher)
  Funded by the IRSES project within the Marie Curie Program FP7.
  Date: January 2014-Pres.

- **OMNIWORKS**: Omnidirectional Vision for Human-UAV Co-working. (Project Leader)

Funded by the ECHORD Project in the European FP7.
Date: March-September 2013

C.2. National Project

- Computer Vision for UAV, from visual information to visual guidance. (Researcher)
  Funded by the Spanish Ministry of Science MICYT #DPI2010-20751-C02-01.
  Date: Febrary 2012-December 2013

C.3. Industry Technology Transfer

- **MeSOANTEN**: Improving Security for Full Operation of Unmanned Aircraft in Naval Environment. (Technical Leader)
  Funded by the National R&D Program RETOS-Colaboración RTC-2014-1762-8, with Unmanned Solutions S.L.
  Date: November 2014-Pres.

- **TAISAP-UAV**: Alternative Technologies to Increase Security in Precision Landing of UAV. (Technical Leader)
  Funded by the National R&D Program AEESD, ref. nr. TSI-100103-2014-177, contracted by Unmanned Solutions S.L.
  Date: November 2014-Pres.

- **E-Vision**: Computer Vision-based Intruder Detection for UAVs. (Researcher)
  Funded under AVANZA Program by Spanish Industry Ministry, contracted by Unmanned Solutions S.L.
  Date: February 2013 - December 2014.

C.4. Challenge and Competition

- **EuRoC**: European Robotics Challenges. (Researcher)
  Data: September-November 2014.

- **IMAV**: International Micro Air Vehicles Flight Competition. (Researcher)
  Date: October 2011-July 2012.

C.5. Research Exchange

- **Nanyang Technological University** (NTU), School of Mechanical and Aerospace Engineering (MAE), Flight Mechanics & Control (FMC) Laboratory, Singapore.
  Date: August-September, 2015

- **Arizona State University** (ASU), School of Earth and Space Exploration (SESE), Autonomous System Technologies Research & Integration Laboratory (ASTRIL), Tempe, Arizona, USA.
  Date: January-April, 2014

- **International Computer Vision Summer School** (ICVSS), Sicily, Italy.
  Date: July, 2012

Bibliography

Abdelkrim, N., Aouf, N., Tsourdos, A., and White, B. (2008). Robust nonlinear filtering for INS/GPS UAV localization. In *Control and Automation, 2008 16th Mediterranean Conference on*, pages 695–702.

Adam, A., Rivlin, E., and Shimshoni, I. (2006). Robust Fragments-based Tracking using the Integral Histogram. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 798–805.

Angelov, P. (2012). *Sense and Avoid in UAS: Research and Applications*. Wiley.

A.R.Babaei, M.Mortazavi, and M.H.Moradi (2011). Classic and fuzzy-genetic autopilot design for unmanned aerial vehicle. *Applied Soft Computing*, 11:365–372.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188.

Avidan, S. (2004). Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1064–1072.

Avidan, S. (2007). Ensemble Tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271.

Azrad, S., Kendoul, F., and Nonami, K. (2010). Visual Servoing of Quadrotor Micro-Air Vehicle Using Color-Based Tracking Algorithm. *Journal of System Design and Dynamics*, 4(2):255–268.

Babenko, B., Yang, M.-H., and Belongie, S. (2009). Visual tracking with online Multiple Instance Learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990.

Babenko, B., Yang, M.-H., and Belongie, S. (2011). Robust Object Tracking with Online Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632.

Bachrach, A., He, R., and Roy, N. (2009). Autonomous flight in unstructured and unknown indoor environments. In *in Proceedings of EMAV*.

Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A. S., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2012). Estimation, Planning, and Mapping for Autonomous Flight Using an RGB-D Camera in GPS-denied Environments. *International Journal of Robotics Research*, 31(11):1320–1343.

Bai, T. and Li, Y. (2012). Robust visual tracking with structured sparse representation appearance model. *Pattern Recognition*, 45(6):2390–2404.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.

Belhumeur, P. and Kriegman, D. (1996). What is the set of images of an object under all possible lighting conditions? In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 270–277.

Black, M. J. and Jepson, A. (1998). EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26(1):63–84.

Bouabdallah, S. and Siegwart, R. (2005). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2247–2252.

Brand, C., Schuster, M., Hirschmuller, H., and Suppa, M. (2014). Stereo-vision based obstacle mapping for indoor/outdoor SLAM. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1846–1853.

Breitenmoser, A., Kneip, L., and Siegwart, R. (2011). A monocular vision-based system for 6D relative robot localization. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 79–85.

Brockers, R., Hummenberger, M., Weiss, S., and Matthies, L. (2014). Towards Autonomous Navigation of Miniature UAV. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 645–651.

Brunelli, R. (2009). *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision (ECCV2010)*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer Berlin Heidelberg.

Campoy, P., Correa, J., Mondragon, I., Martinez, C., Olivares, M., Mejias, L., and Artieda, J. (2009). Computer Vision Onboard UAVs for Civilian Tasks. *Journal of Intelligent and Robotic Systems*, 54(1-3):105–135.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Carnie, R., Walker, R., and Corke, P. (2006). Image processing algorithms for UAV "sense and avoid". In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2848–2853.

Collins, R., Liu, Y., and Leordeanu, M. (2005). Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631–1643.

Coza, C. and Macnab, C. J. B. (2006). A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization. In *Fuzzy*

Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American, pages 454–458.

Dey, D., Geyer, C., Singh, S., and Digioia, M. (2010). Passive, Long-Range Detection of Aircraft: Towards a Field Deployable Sense and Avoid System. In *Field and Service Robotics*, volume 62 of *Springer Tracts in Advanced Robotics*, pages 113–123.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Doitsidis, L., Valavanis, K., Tsourveloudis, N., and Kontitsis, M. (2004). A framework for fuzzy logic based UAV navigation and control. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pages 4041–4046.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part I. *Robotics Automation Magazine, IEEE*, 13(2):99–110.

E.Haber, R., del Toro, R. M., and Gajate, A. (2010). Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process. *Information Sciences*, 180:2777–279.

Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision-ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 834–849.

Engel, J., Sturm, J., and Cremers, D. (2012). Accurate Figure Flying with a Quadrocopter Using Onboard Visual and Inertial Sensing. In *Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR) at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*.

Espinoza, E., Garcia, O., Lugo, I., Ordaz, P., Malo, A., and Lozano, R. (2014). Modeling and Sliding Mode Control of a Micro Helicopter-Airplane System. *Journal of Intelligent & Robotic Systems*, 73(1-4):469–486.

Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., and Scaramuzza, D. (2015). Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *Journal of Field Robotics*, pages 1–20.

Faessler, M., Mueggler, E., Schwabe, K., and Scaramuzza, D. (2014). A monocular pose estimation system based on infrared leds. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 907–913.

Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Forster, C., Faessler, M., Fontana, F., Werlberger, M., and Scaramuzza, D. (2015). Continuous On-Board Monocular-Vision-based Elevation Mapping Applied to Autonomous Landing of Micro Aerial Vehicles. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1–8.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). SVO: Fast Semi-Direct Monocular Visual Odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22.

Fossel, J., Hennes, D., Claes, D., Alers, S., and Tuyls, K. (2013). OctoSLAM: A 3D mapping approach to situational awareness of unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 179–188.

Fu, C., Carrio, A., Olivares-Mendez, M., Suarez-Fernandez, R., and Campoy, P. (2014a). Robust real-time vision-based aircraft tracking from Unmanned Aerial Vehicles. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5441–5446.

Fu, C., Carrio, A., Olivares-Mendez, M. A., Rubino, F., and Campoy, P. (2015). An Efficient Adaptive Visual Tracking with ARM Architecture System Pre-Processing for UAV Autonomous Landing. *Journal of Intelligent and Robotic Systems*.

Fu, C., Fernandez, R. A. S., Olivares-Mendez, M. A., and Campoy, P. (2013). In *Real-Time Adaptive Multi-Classifier Multi-Resolution Visual Tracking Framework for Unmanned Aerial Vehicles*, volume 2, pages 99–106.

Fu, C., Olivares-Mendez, M., Suarez-Fernandez, R., and Campoy, P. (2014b). Monocular Visual-Inertial SLAM-based Collision Avoidance

Strategy for Fail-Safe UAV using Fuzzy Logic Controllers. *Journal of Intelligent & Robotic Systems*, 73(1-4):513–533.

Fu, C., Pestana, J., Mellado-Bataller, I., Sanchez-Lopez, J., and Campoy, P. (2012). Visual Identification and Tracking for Vertical and Horizontal Targets in Unknown Indoor Environment. In *International Micro Air Vehicle Conference (IMAV2012)*.

Gálvez-López, D. and Tardós, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

Gandhi, T., Yang, M.-T., Kasturi, R., Camps, O., Coraor, L., and McCandless, J. (2000). Detection of obstacles in the flight path of an aircraft. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 304–311.

Garrido-Jurado, S., Munoz-Salinas, R., Madrid-Cuevas, F., and Marin-Jimenez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292.

Gehrig, S., Eberli, F., and Meyer, T. (2009). A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. In *Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 134–143.

Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient Large-scale Stereo Matching. In *The 10th Asian Conference on Computer Vision (ACCV2010)*, pages 25–38.

Geiger, A., Ziegler, J., and Stiller, C. (2011). Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968.

Gomez, J. and Jamshidi, M. (2011). Fuzzy Adaptive Control for a UAV. *Journal of Intelligent & Robotic Systems*, 62(2):271–293.

Grabner, H. and Bischof, H. (2006). On-line Boosting and Vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260–267.

Hall, P., Marshall, D., and Martin, R. (2002). Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. *Image and Vision Computing*, 20(13â14):1009–1016.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Herath, D., Kodagoda, S., and Dissanayake, G. (2006). Simultaneous Localisation and Mapping: A Stereo Vision Based Approach. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 922–927.

Hermann, S. and Klette, R. (2012). Evaluation of a New Coarse-to-Fine Strategy for Fast Semi-Global Stereo Matching. In *Advances in Image and Video Technology*, volume 7087 of *Lecture Notes in Computer Science*, pages 395–406. Springer Berlin Heidelberg.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814.

Hirschmuller, H. (2008). Stereo Processing by Semi-Global Matching and Mutual Information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341.

Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.

Hu, W., Li, X., Zhang, X., Shi, X., Maybank, S., and Zhang, Z. (2011). Incremental Tensor Subspace Learning and Its Applications toÂ Foreground Segmentation and Tracking. *International Journal of Computer Vision*, 91(3):303–327.

Huh, S. and Shim, D. (2010). A Vision-Based Automatic Landing Method for Fixed-Wing UAVs. In *2nd International Symposium on UAVs, 2009*, pages 217–231.

Jepson, A., Fleet, D., and El-Maraghi, T. (2003). Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1296–1311.

Juan, L. and Gwon, O. (2009). A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152.

Kadmiry, B. and Driankov, D. (2004). A fuzzy gain-scheduler for the attitude control of an unmanned helicopter. *Fuzzy Systems, IEEE Transactions on*, 12(4):502–515.

Kalal, Z., Mikolajczyk, K., and Matas, J. (2012). Tracking-Learning-Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages 506–513.

Kitt, B., Geiger, A., and Lategahn, H. (2010). Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486–492.

Klein, G. and Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*.

K.M.Zemalache and H.Maaref (2009). Controlling a drone: Comparison between a based model method and a fuzzy inference system. *Applied Soft Computing*, 9:404–418.

Kneip, L., Scaramuzza, D., and Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976.

Konolige, K. (1997). Small vision system: Hardware and implementation. In *Proc. of the Intl. Symp. of Robotics Research (ISRR*, pages 111–116.

Kurnaz, S., Cetin, O., and Kaynak, O. (2009). Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous Unmanned Aerial Vehicles. *Journal of Intelligent and Robotic Systems*, 54(1-3):229–244.

Kurnaz, S., Cetin, O., and Kaynak, O. (2010). Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Systems with Applications*, 37:1229–1234.

Kwon, J., Lee, K. M., and Park, F. (2009). Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 991–998.

Lai, J. S., Ford, J. J., Mejias, L., O'Shea, P. J., and Walker, R. A. (2012). See and avoid using on-board computer vision. In *Sense and Avoid in UAS: Research and Applications*, pages 265–294. John Wiley & Sons.

Lee, K.-C. and Kriegman, D. (2005). Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 852–859.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quart. J. Appl. Maths.*, II(2):164–168.

Levey, A. and Lindenbaum, M. (2000). Sequential Karhunen-Loeve basis extraction and its application to images. *Image Processing, IEEE Transactions on*, 9(8):1371–1374.

Li, G., Liang, D., Huang, Q., Jiang, S., and Gao, W. (2008). Object tracking using incremental 2D-LDA learning and Bayes inference. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1568–1571.

Limnaios, G. and Tsourveloudis, N. (2012). Fuzzy Logic Controller for a Mini Coaxial Indoor Helicopter. *Journal of Intelligent & Robotic Systems*, 65(1-4):187–201.

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *The 7th International Joint Conference on Artificial Intelligence (IJCAI1981)*, pages 674–679.

Martínez, C., Campoy, P., Mondragón, I. F., Sánchez-Lopez, J. L., and Olivares-Méndez, M. A. (2014). HMPMR Strategy for Real-time Tracking in Aerial Images, Using Direct Methods. *Machine Vision and Applications*, 25(5):1283–1308.

Martinez, C., Mondragon, I. F., Campoy, P., Sanchez-Lopez, J., and Olivares-Mendez, M. (2013). A Hierarchical Tracking Strategy for Vision-Based Applications On-Board UAVs. *Journal of Intelligent & Robotic Systems*, pages 517–539.

Martínez, C., Richardson, T., Thomas, P., du Bois, J. L., and Campoy, P. (2013). A vision-based strategy for autonomous aerial refueling tasks. *Robotics and Autonomous Systems*, 61(8):876–895.

Masselli, A., Yang, S., Wenzel, K., and Zell, A. (2014). A Cross-Platform Comparison of Visual Marker Based Approaches for Autonomous Flight of Quadrocopters. *Journal of Intelligent & Robotic Systems*, 73(1-4):349–359.

Mccandless, J. W. (1999). Detection of Aircraft in Video Sequences Using A Predictive Optical Flow Algorithm.

Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2009). A Constant-Time Efficient Stereo SLAM System.

Mei, X. and Ling, H. (2011). Robust Visual Tracking and Vehicle Classification via Sparse Representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2259–2272.

Mejias, L., Campoy, P., Saripalli, S., and Sukhatme, G. (2006a). A visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2503–2508.

Mejias, L., Saripalli, S., Campoy, P., and Sukhatme, G. (2006b). A visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *IEEE International Conference on Robotics and Automation 2006*, pages 2503–2508.

Mellinger, D., Michael, N., and Kumar, V. (2012). Trajectory generation and control for precise aggressive maneuvers with quadrotors. *International Journal of Robotics Research*, 31(5):664–674.

Merino, L., Caballero, F., MartĂnez-de Dios, J., Maza, I., and Ollero, A. (2012). An Unmanned Aircraft System for Automatic Forest Fire Monitoring and Measurement. *Journal of Intelligent & Robotic Systems*, 65(1-4):533–548.

Mian, A. (2008). Realtime Visual Tracking of Aircrafts. In *Digital Image Computing: Techniques and Applications (DICTA), 2008*, pages 351–356.

Michael, N., Mellinger, D., Lindsey, Q., and Kumar, V. (2010). The GRASP Multiple Micro-UAV Testbed. *Robotics Automation Magazine, IEEE*, 17(3):56–65.

Mondragón, I., Campoy, P., Martinez, C., and Olivares-Mendez, M. (2010). 3d pose estimation based on planar object tracking for uavs control. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 35–41.

Mur-Artal, Raúl, M. J. M. M. and Tardós, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *Submitted to IEEE Transaction on Robotics.*

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24.

Murphy, R., Steimle, E., Hall, M., Lindemuth, M., Trejo, D., Hurlebaus, S., Medina-Cetina, Z., and Slocum, D. (2011). Robot-Assisted Bridge Inspection. *Journal of Intelligent & Robotic Systems*, 64(1):77–95.

Murray, R. M., Sastry, S. S., and Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., 1st edition.

Ng, A. Y. and Jordan, M. I. (2002). On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, pages 841–848.

Nieuwenhuisen, M., Droeschel, D., Beul, M., and Behnke, S. (2014). Obstacle detection and navigation planning for autonomous micro aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 1040–1047.

Nutzi, G., Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent & Robotic Systems*, 61(1-4):287–299.

Olivares-Mendez, M., Campoy, P., Martinez, C., and Mondragon, I. (2009). A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2879–2884.

Olivares-Mendez, M., Mejias, L., Campoy, P., and Mellado-Bataller, I. (2013). Cross-Entropy Optimization for Scaling Factors of a Fuzzy Controller: A See-and-Avoid Approach for Unmanned Aerial Systems. *Journal of Intelligent and Robotic Systems*, 69:189–205.

Olivares-Mendez, M., MondragoĚn, I., Campoy, P., and Martinez, C. (2010). Fuzzy controller for UAV-landing task using 3D-position visual estimation. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8.

Olivares-Mendez, M. A., Fu, C., Kannan, S., Voos, H., and Campoy, P. (2014). Using the Cross-Entropy method for control optimization: A case study of see-and-avoid on unmanned aerial vehicles. In *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, pages 1183–1189.

Olivares-Mendez, M. A., Mejias, L., Campoy, P., Mellado-Bataller, I., and Mondragon, I. (2012). Uas See-and-Avoid using two different approaches of Fuzzy Control. In *2012 International Conference on Unmanned Aircraft Systems (ICUAS'12).*

Olivares-Mendez, M. A., Mondragon, I., Cervera, P. C., Mejias, L., and Martinez, C. (2011). Aerial object following using visual fuzzy servoing. In *First Workshop on Research, Development and Education on Unmanned Aerial Systems (RED-UAS 2011)*, pages 61–70.

Paz, L. M., Pinies, P., Tardos, J., and Neira, J. (2008). Large-Scale 6-DOF SLAM with Stereo-in-Hand. *Robotics, IEEE Transactions on*, 24(5):946–957.

Pestana, J., Mellado-Bataller, I., Sanchez-Lopez, J., Fu, C., Mondragón, I. F., and Campoy, P. (2014). A General Purpose Configurable Controller for Indoors and Outdoors GPS-Denied Navigation for Multirotor Unmanned Aerial Vehicles. *Journal of Intelligent & Robotic Systems*, 73(1-4):387–400.

Pizzoli, M., Forster, C., and Scaramuzza, D. (2014). REMODE: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616.

Pratt, K. S., Murphy, R., Stover, S., and Griffin, C. (2009). CONOPS and autonomy recommendations for VTOL small unmanned aerial system based on Hurricane Katrina operations. *Journal of Field Robotics*, 26(8):636–650.

Ross, D., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77(1-3):125–141.

Rosten, E. and Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. In *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer Berlin Heidelberg.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV2011)*, pages 2564–2571.

R.Y.Rubinstein and D.P.Kroese (2004). *The Cross-Entropy Method: A Unified Approach to Combinational Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Berlin, Germany.

Saffari, A., Leistner, C., Santner, J., Godec, M., and Bischof, H. (2009). On-line Random Forests. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1393–1400.

Sampedro, C., Martinez, C., Chauhan, A., and Campoy, P. (2014). A supervised approach to electric tower detection and classification for power line inspection. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 1970–1977.

Sanchez-Lopez, J., Saripalli, S., Campoy, P., Pestana, J., and Fu, C. (2013). Toward visual autonomous ship board landing of a VTOL UAV. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 779–788.

Santos, M., LoĔpez, V., and Morata, F. (2010). Intelligent fuzzy controller of a quadrotor. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pages 141–146.

Saripalli, S., Montgomery, J., and Sukhatme, G. (2003). Visually guided landing of an unmanned aerial vehicle. *Robotics and Automation, IEEE Transactions on*, 19(3):371–380.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. *Robotics Automation Magazine, IEEE*, pages 80–92.

Schauwecker, K. and Zell, A. (2014). Robust and efficient volumetric occupancy mapping with an application to stereo vision. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6102–6107.

Scherer, S. and Zell, A. (2013). Efficient onbard RGBD-SLAM for autonomous MAVs. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1062–1068.

Strasdat, H., Davison, A., Montiel, J., and Konolige, K. (2011). Double window optimisation for constant time visual SLAM. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359.

Teuliere, C., Eck, L., and Marchand, E. (2011). Chasing a moving target from a flying UAV. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4929–4934.

Tian, M., Zhang, W., and Liu, F. (2007). On-Line Ensemble SVM for Robust Object Tracking. In *Computer Vision-ACCV 2007*, volume 4843 of *Lecture Notes in Computer Science*, pages 355–364.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518.

Viola, P., Platt, J. C., and Zhang, C. (2006). Multiple instance boosting for object detection. In *Neural Information Processing Systems*, pages 1419–1426.

Wainwright, A. and Ford, J. (2012). Fusion of morphological images for airborne target detection. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1180–1187.

Wang, D., Lu, H., and wei Chen, Y. (2010). Incremental MPCA for Color Object Tracking. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1751–1754.

Wang, J., Chen, X., and Gao, W. (2005). Online selecting discriminative tracking features using particle filter. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1037–1042.

Wang, Q., Chen, F., Xu, W., and hsuan Yang, M. (2011). An Experimental Comparison of Online Object Tracking Algorithms. In *Proceedings of SPIE: Image and Signal Processing Track*.

Wang, T., Gu, I.-H., and Shi, P. (2007). Object Tracking using Incremental 2D-PCA Learning and ML Estimation. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages 933–936.

Wang, W., Song, G., Nonami, K., Hirata, M., and Miyazawa, O. (2006). Autonomous Control for Micro-Flying Robot and Small Wireless Helicopter X.R.B. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2906–2911.

Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874.

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., and Tardos, J. (2009). A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems*, 57(12):1188–1197.

Xu, G., Zhang, Y., Ji, S., Cheng, Y., and Tian, Y. (2009). Research on computer vision-based for UAV autonomous landing on a ship. *Pattern Recognition Letters*, 30(6):600–605.

Yang, F., Lu, H., Zhang, W., and Yang, G. (2012). Visual tracking via bag of features. *Image Processing, IET*, 6(2):115–128.

Yang, S., Scherer, S., Schauwecker, K., and Zell, A. (2014). Autonomous Landing of MAVs on an Arbitrarily Textured Landing Site Using Onboard Monocular Vision. *Journal of Intelligent & Robotic Systems*, 74(1-2):27–43.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object Tracking: A Survey. *ACM Comput. Surv.*, 38(4).

Yun, B., Peng, K., and Chen, B. (2007). Enhancement of GPS signals for Automatic Control of a UAV Helicopter System. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 1185–1189.

Zadeh, L. A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(1):28–44.

Z.Botev and D.P.Kroese (2004). Global likelihood optimization via the cross-entropy method with an application to mixture models. In *Proceedings of the 36th conference on Winter simulation (WSC2004)*, pages 529–535.

Zhang, C., Jing, Z., Tang, Y., Jin, B., and Xiao, G. (2013). Locally discriminative stable model for visual tracking with clustering and principle component analysis. *Computer Vision, IET*, 7(3).

Zhang, J., Kaess, M., and Singh, S. (2014). Real-time Depth Enhanced Monocular Odometry. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 4973–4980.

Zhang, K., Zhang, L., and Yang, M.-H. (2012). Real-time compressive tracking. In *Proceedings of the 12th European conference on Computer Vision (ECCV'12)*, pages 864–877.

Zheng, L. (1992). A practical guide to tune of proportional and integral (PI) like fuzzy controllers. In *Fuzzy Systems, 1992., IEEE International Conference on*, pages 633–640.

Zhou, S., Chellappa, R., and Moghaddam, B. (2004). Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, 13(11):1491–1506.