

**DEVELOPMENT AND APPLICATIONS OF A
VISION-BASED UNMANNED HELICOPTER**

LIN FENG

(M.Eng, Beihang University, China)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgments

First and foremost, I like to express my heartfelt gratitude to my supervisors, Professor Ben M. Chen and Professor Kai Yew Lum and Professor T. H. Lee. I will never forget it is Professor Chen who gives me this precious opportunity to pursue my PhD degree and introduces me to the marvellous research area on vision-based unmanned helicopters. To me, he is not only an advisor on research, but also a mentor on life. Professor Lum and Professor Lee provide me numerous constructive suggestions and invaluable guidance during the course of my PhD study. Without their guidance and support, it would have not been possible for me to complete my PhD program.

Special thanks are given to the friends and fellow classmates in our UAV research group in the Department of Electrical and Computer Engineering, National University of Singapore. Particularly, I would like to thank Dr. Kemao Peng, Dr. Guowei Cai, Dr. Miaobo Dong, Dr. Biao Wang, Dr. Ben Yu, and my fellow classmates Xiangxu Dong, Xiaolian Zheng, Fei Wang, Shiyu Zhao and Ali Karimoddini. Without their help and support, I would even not be able to make the vision-based unmanned helicopters fly.

Moreover, I am much grateful to Dr. Chang Chen of DSO National Laboratories, for his suggestions, generous help, and vast knowledge in the field of UAV research. I would also love to extend my sincere thanks to all of the friends in Control and Simulation Lab of the ECE Department, with whom I have enjoyed every minute during the last five years. I would like to give my special thanks to the lab officers, Mr. Hengwei Zhang and Ms. Sarasupathi for helping me process numerous purchasing issues. I would like to thank Dr. Kok Zuea Tang for patiently providing me technical support.

Another memorable thing that worth to be mentioned during the composition of this thesis is that I accidentally lost the thesis draft and some of the valuable raw data on a trip to Xia Men, China this June 2010. If not for the tremendous help from Dr. Sen Yan, Xiaolian Zheng and Xiangxu Dong, I would not be able to submit this thesis in time.

Last but certainly not the least, I owe a debt of deepest gratitude to my parents and my wife for their everlasting love, care and encouragement.

Contents

Acknowledgments	i
Contents	iii
Summary	vii
List of Tables	ix
List of Figures	x
Nomenclature	xiv
1 Introduction	1
1.1 Vision Systems for UAVs	2
1.2 Literature Review	3
1.2.1 Vision-Based Target Acquisition and Targeting	4
1.2.2 Vision-Based Flight Control	7
1.2.3 Vision-Based Navigation	8
1.3 Challenges in Vision-Based UAVs	12
1.4 Motivation and Contributions of This Research	13
1.5 Outline of This Thesis	15

2	Hardware Design of the Vision-Based Unmanned Helicopter	16
2.1	Introduction	16
2.1.1	Related Work	16
2.1.2	Requirements	20
2.2	Configuration of Hardware Components	21
2.2.1	Radio Controlled Helicopter	22
2.2.2	Flight Control System	25
2.2.3	Vision System	31
2.2.4	Ground Supporting System	39
2.3	Systematic Integration of the On-board System	40
2.3.1	Computer-Aided Virtual Design Environment	40
2.3.2	Virtual Design Methodology	41
2.3.3	Anti-Vibration Design	45
2.4	Ground Test Evaluation	53
2.5	Conclusion	54
3	Software System Design and Implementation	57
3.1	Introduction	57
3.2	Flight Control Software	58
3.3	Vision Software	59
3.3.1	Framework of Vision Software	60
3.3.2	Task Management	64
3.3.3	Computer Vision Library	65
3.4	Ground Station Software	67

3.5	Implementation of the Automatic Control	68
3.5.1	Dynamic Modeling and System Identification of the UAV	68
3.5.2	Automatic Flight Control System	71
3.5.3	Flight Tests	73
3.6	Conclusion	77
4	Vision-Based Ground Target Following	82
4.1	Introduction	82
4.2	Target Detection and Tracking in the Image	88
4.2.1	Target Detection	88
4.2.2	Image Tracking	102
4.3	Coordinate Systems	114
4.4	Camera Calibration	118
4.4.1	Camera Model	118
4.4.2	Intrinsic Parameter Estimation	120
4.4.3	Distortion Compensation	121
4.4.4	Simplified Camera Model	123
4.5	Target Following Control	125
4.5.1	Control of the Pan/Tilt Servo Mechanism	126
4.5.2	Following Control of the UAV	131
4.6	Experimental Results	134
4.7	Conclusion	137

5	Vision-Based Flight Control for the UAV	141
5.1	Introduction	141
5.2	Landmark Detection	144
5.3	Pose Estimation	149
5.4	Data Fusion	153
5.5	Experimental Results	156
5.6	Conclusion	159
6	Conclusions	163
6.1	Contributions	163
6.2	Future Works	165
	Bibliography	169
	Appendix: Publication List	186

Summary

Unmanned aerial vehicles (UAVs), especially unmanned helicopters, have achieved great success in both military and civil applications in the last two decades, and also aroused great interest in their potential in more complex and demanding environments. To extend their capabilities in such environments, unmanned helicopters have been equipped with advanced machine vision systems. This calls for in-depth research work on such vision-based unmanned helicopters, which is presented in this thesis.

This thesis begins with the hardware design and implementation of a vision-based small-scale unmanned helicopter. The on-board hardware system is built using embedded computer systems and Micro-Electro-Mechanical System (MEMS) technologies. A systematic and effective design methodology is summarized and presented in this thesis to construct UAVs with minimum complexity and time cost. This design methodology is also enhanced using a computer-aided design technique.

To ensure the overall vision-based unmanned helicopter system work harmoniously, an efficient software system is developed, which consists of three main parts: (1) the flight control software system, which performs multiple flight-control-related tasks such as device management, control algorithm execution, wireless communication and data logging; (2) the vision software system, which coordinates tasks such as image collection, image processing, target detection and tracking; and (3) the ground station software system, which is used to receive on-board information, send commands to the onboard system, and monitor the in-flight states of the UAV.

Next, research efforts are further focused on vision-based applications of the proposed vision-based UAV. An application of vision-based ground target following is presented in this thesis. To detect the target using the on-board camera, an advanced vision algorithm is proposed and implemented on board, which utilizes a robust feature-based target detection method and a hierarchical tracking scheme. The proposed vision algorithm is integrated with on-board navigation sensors to measure the relative distance between the target and the UAV. Taking advantage of the vision feedback, a two-layer target following control framework is utilized to control a pan/tilt servo mechanism to keep the target at the desired location in the image, and guide the helicopter to follow the motion of the target.

To further explore the potential of the proposed vision-based UAV, a sophisticated and systematic vision augmented approach is proposed to realize motion estimation and flight control of the UAV in GPS-denied conditions. This approach is composed of robust landmark detection and a core algorithm for vision-based motion estimation. A reference landmark is identified first, and then the key feature points on it are extracted, even under partially occluded conditions. Based on the extracted 2D image points and known corresponding 3D model, a pose estimation algorithm is proposed to estimate the relative position and angle of the UAV with respect to the ground reference. The velocity of the UAV is estimated with the measurement of the position, and improved by fusing with IMU measurements via a Kalman Filter in order to provide the necessary information for the hovering control of the UAV. The simulation and flight test results show that the proposed methodology is efficient and effective.

In conclusion, the development of a vision-based UAV is presented in this thesis. The vision-based ground target following and vision-based flight control in GPS-denied environments are conducted in flight to verify the proposed vision-based UAV system. Some prospective directions for future research are also included.

List of Tables

2.1	Main specifications of Raptor 90 SE	23
2.2	Main specifications of PC-104 ATHENA	26
2.3	Main specifications of MNAV100CA	28
2.4	Main Specifications of PC/104-Plus Cool RoadRunner III	38
2.5	Weight list of on-board hardware components	47
2.6	Power consumption list for SheLion	51
3.1	Test results of OpenCV functions	67
3.2	Physical meanings of the state and input variables.	72
4.1	Comparison of ϕ_1 between two normalization methods	96
4.2	Experimental results of target detection and tracking	114
4.3	Estimated intrinsic parameters of the on-board camera	121
4.4	Parameters of the pan/tilt servos	131
4.5	Experiment results of target detection and tracking in flight	135

List of Figures

2.1	Overview of the vision-based unmanned helicopter: SheLion	23
2.2	Radio controlled helicopter: Raptor 90 SE.	24
2.3	PC-104 embedded single board computer: ATHENA	26
2.4	Navigation sensor: MNAV100CA	29
2.5	Servo controller	31
2.6	On-board wireless transceiver	31
2.7	Ground wireless transceiver	31
2.8	Hardware configuration of the vision system.	33
2.9	On-board vision sensor	34
2.10	Pan/tilt servo mechanism	34
2.11	Frame grabber: Colory 104	36
2.12	Working principle of Colory 104	36
2.13	Capture mechanism of Colory 104	37
2.14	Vision computer: Cool RoadRunnerIII	38
2.15	Wireless video link	39
2.16	Virtual components created in SolidWorks.	42
2.17	Raptor 90 RC helicopter and its virtual counterpart.	43

2.18	MNAV100CA and its virtual counterpart.	43
2.19	Layout design procedure and the final on-board system.	46
2.20	3D views of the infrastructure and on-board system in SolidWorks.	48
2.21	3D views of the infrastructure and on-board system in physical world.	49
2.22	Anti-vibration design for the on-board system.	49
2.23	Working point of the selected wire rope isolators.	50
2.24	DC-to-DC convertor boards: JUPITER-MM	51
2.25	Power supply design for SheLion unmanned helicopter.	52
2.26	Execution time of the test loops of Flight Control CPU.	55
2.27	Output voltages of Lithium-Polymer batteries.	55
2.28	Sample result of comparison of vibrational amplitude.	56
2.29	Virtual and real unmanned helicopter: Shelion in flight.	56
3.1	Framework of the flight control software	59
3.2	Framework of the vision software	61
3.3	Task management of the vision software	66
3.4	Execution of multiple tasks of the vision software.	67
3.5	Framework of ground station software	69
3.6	User interface of ground station software	70
3.7	Framework of the autonomous flight control law	73
3.8	Simulation results of the autonomous flight control	74
3.9	Input signals in the manual flight test.	75
3.10	Velocity outputs in the manual flight test.	75
3.11	Angular rates in the manual flight test.	76

3.12 Euler angles in the manual flight test.	76
3.13 Input signals in the automatic hovering flight test.	77
3.14 Position outputs in the automatic hovering flight test.	78
3.15 Velocity outputs in the automatic hovering flight test.	78
3.16 Angular rates in the automatic hovering flight test.	79
3.17 Euler angles in the automatic hovering flight test.	79
3.18 Flight results of the autonomous flight control.	80
3.19 Samples of ground images captured by SheLion.	80
4.1 Illustration of the vision-based target following.	87
4.2 Flow chart of the ground target detection, tracking and following scheme. . .	88
4.3 Illustration of Segmentation.	91
4.4 Comparison of ϕ_1 using two normalization methods	97
4.5 Color histogram extraction.	98
4.6 Flow chart of image tracking.	104
4.7 Block diagram of the CAMSHIFT algorithm.	108
4.8 Image tracking using the CAMSHIFT algorithm.	111
4.9 Decision making using a finite state machine.	112
4.10 Target detection with occlusion.	113
4.11 The tracking errors of the pan/tilt servo in vertical and horizontal directions.	115
4.12 Coordinate systems.	116
4.13 Frontal pin-hole camera model.	119
4.14 Images for camera calibration.	122
4.15 Grid corner extraction for camera calibration.	123

4.16 Distortion compensation.	124
4.17 Block diagram of the tracking control scheme.	126
4.18 The demo of the vision-based target following.	135
4.19 The test result of the vision-based servo control.	136
4.20 The test result of the vision-based target following.	136
4.21 The test result of the vision-based target following in 3D.	137
5.1 Landmark design.	145
5.2 Flow chart of landmark detection.	146
5.3 Key point correspondence.	149
5.4 Illustration of vision-based motion estimation.	150
5.5 Vision-based position estimation using the simulation data	157
5.6 Vision-based velocity estimation using the simulation data	158
5.7 Vision-based position estimates using real images	159
5.8 Comparison of position estimation	160
5.9 Comparison of velocity estimation	161
5.10 The time cost of each thread in the flight	162

Nomenclature

Latin variables

\mathbf{a}_{nb}	the load acceleration in the body frame
A	the interior area of the object
\mathbf{B}	input matrix of the linearized model
B_B	velocity transformation matrix from body frame to NED frame
\mathbf{C}	output matrix in linearized model structure
f_x	vertical focal length
f_y	horizontal focal length
F	state feedback matrix
g	the acceleration of gravity
h	NED frame altitude
h	hue in the HSV color space
H	the color histogram
m_{pq}	the (p, q) -th order moment
o_x	the x coordinate of the principle point
o_y	the y coordinate of the principle point
p	body frame rolling angular velocity
\mathbf{p}_c	the coordinate of the point P in the camera frame
\mathbf{p}_i	the coordinate of the point P in the image frame
\mathbf{p}_n	the coordinate of the point P in the NED frame
\mathbf{p}_o	the coordinate of the point P in the object frame
\mathbf{p}_w	the coordinate of the point P in the world frame
q	body frame pitching angular velocity
r	body frame yawing angular velocity
r_{sp}	radius in the spherical coordinate system

\mathbf{R}_{oc}	rotation matrix from object frame to camera frame
\mathbf{R}_{sc}	rotation matrix from servo frame to camera frame
\mathbf{R}_{cw}	rotation matrix from camera frame to world frame
s	saturation in the HSV color space
$s\theta$	the skew factor
T_s	the sampling period of the vision software
u	body frame x axis velocity
v	value in the HSV color space
v	body frame y-axis velocity
\mathbf{v}_n	velocity vector in NED frame
w	body frame z axis velocity
\mathbf{x}	state vector in linearized model structure
X	position vector in NED frame
y	body frame y-axis position
\mathbf{y}	output vector in linearized model structure
z	body frame z-axis position

Greek variables

α	feature vector
β_c	compactness
η_{pq}	the (p, q) -th order central moment
θ	pitching angle in NED frame
θ_{sp}	azimuth angle in the spherical coordinate system
ρ	a density distribution function
ϕ	rolling angle in NED frame
ϕ_{sp}	elevation angle in the spherical coordinate system
ϕ_1	the first moment invariant

ϕ_2	the second moment invariant
ϕ_3	the third moment invariant
ϕ_4	the fourth moment invariant
ψ	yawing angle in NED frame
Ω	the region of the target
Ω_r	the region of interest
Ω_w	the region of the search window

Acronyms

A/D	analog-to-digital
CAD	computer-aided-design
CAMSHIFT	continously adaptive mean shift
CCD	charge-coupled device
CEP	circular error probable
CF	compact flash
CG	center of gravity
CMOS	complementary metal-oxide-semiconductor
CNF	composite nonlinear feedback
CPU	central processing unit
D/A	digital-to-analog
DC	direct current
DLT	direct linear transformation
DOF	degree-of-freedom
DSP	digital signal processing
EKF	extended Kalman filter
EMI	electromagnetic interference
FFT	fast Fourier transform

GPS	global positioning system
GUI	graphical user interface
INS	inertial navigation system
LQG	linear quadratic Gaussian
Li-Po	lithium-polymer
MEMS	Micro-Electro-Mechanical System
MIMO	multi-input/multi-output
NED	North-East-Down
NUS	National University of Singapore
OpenCV	open source computer vision
PnP	perspective- n -point problem
RC	radio-controlled
RPM	rotations per minute
SBC	single board computer
SISO	single-input/single output
SLAM	simultaneous localization and mapping
TANS	terrain aided navigation system
TERCOM	terrain contour matching
UAV	unmanned aerial vehicle
2D	two-dimensional
3D	three-dimensional

Chapter 1

Introduction

An unmanned aerial vehicle (UAV) is an aircraft that is equipped with the necessary data processing units, sensors, automatic control and communication systems in order to perform autonomous flight missions without an on-board crew [17]. During the last two decades, UAVs have aroused strong interest and made huge progress in the civil and industrial markets, ranging from industrial surveillance, agriculture, to wildlife conservation [100,37,76,19]. Particularly, unmanned rotorcrafts, such as helicopters, received much attention in the defense, security and research communities [2,13,45,95,117] due to their unique and attractive capabilities of vertical take-off, hovering and landing.

Although great progress has been achieved in the development of UAVs, it is still attractive and necessary to investigate the potential of UAVs, and extend their applications in future. It is undoubted that the latest trend in the UAV community is towards the creation of intelligent UAVs, such as a sophisticated unmanned helicopter equipped with a vision enhanced navigation system. The maneuvering capabilities of the helicopter and the rich information of visual sensors are combined to arrive at a versatile platform for a variety of applications.

In what follows of this chapter, an introduction of vision systems for UAVs is given in Section 1.1, and then a literature review of vision applications of UAVs is presented in

Section 1.2. The challenges of the vision systems for unmanned aerial vehicles are addressed in Section 1.3. Then, a general overview of the work achieved by our NUS UAV research team is presented in Section 1.4. Finally, the outline of this thesis is given in Section 1.5 for easy reference.

1.1 Vision Systems for UAVs

Vision systems have become an exciting field in academic research and industrial applications. By integrating vision sensors with other avionic sensors, functions of unmanned vehicles can be greatly extended to autonomously perform a variety of work, such as vision-based reconnaissance, surveillance and target acquisition.

Naturally, the sense of vision plays an essential role in the daily lives of animals and human beings. It is a great evolutionary advantage gained to make moving or hunting more efficient. Similarly, a UAV utilizes a vision system as its pair of eyes to obtain information of designated targets and environments.

Although many other simple range sensors, such as sonar and infrared sensors, are utilized in applications of unmanned systems, they are not sufficient to handle complex environments and provide accurate measurements. Sophisticated sensors such as radars or laser scanners can provide accurate relative distance to the target and the environment, but the cost and weight is not acceptable for low-cost and small-size unmanned vehicles. Furthermore, these kind of range sensors cannot identify targets and understand surrounding environments. In summary, compared to the aforementioned sensors, vision sensing technologies have the following features:

1. They are capable of providing rich information of objects of interest and the surrounding environments, including geometry of the scene, photometry of the object, and dynamics of the environment. So vision sensing is an indispensable part to develop intelligent unmanned vehicles.

2. They require only natural light and do not depend on any other signal source, such as beacon stations or satellite signals.
3. They are generally of low cost and light weight compared to other related sensing systems such as radars.
4. They do not emit any energy, so that the whole system is almost undetectable and safer in special conditions, such as battlefields.

Due to those advantages, vision sensors are suitable for small-size unmanned vehicles with limited space and payload. The main shortcoming of a vision systems is of computationally expensive for processing image sequences. Thanks to the rapid growth of computer and electronic technologies, light-weight, but powerful commercial processors become more and more feasible. A variety of vision systems using off-the-shelf components were reported [16, 41, 53], including the popular PC/104(-plus)-based single board computers, and other tiny single board computers, such as ARM-based Gumstix. Thus, researchers and developers do not need to rely on expensive and special hardware for image processing, and progress of developing vision systems is also speeded up. Moreover, these embedded single board computers require low power consumption, which is also a core concern for the vision system mounted on a UAV.

In the following section, we will discuss vision-based UAVs around world and their applications, as well as investigate novel ideas, concepts and technologies behind these applications.

1.2 Literature Review

In the last two decades, there are many explorations on vision-based UAV systems employed in different applications, such as vision-based stabilization [3, 48], air-to-air tacking [63], navigation in complex environments [56], vision-based pose estimation and autonomous

landing [107, 101], as well as localization and mapping [65, 81]. These applications can be roughly divided into several categories depending on how to use extracted vision information:

1. Vision-based Target Acquisition and Targeting: Vision information is used to search and identify the target of interest, and estimate the relative distance and orientation of the target with respect to the UAV. The estimated information is used to guide the UAV to follow the target.
2. Vision-based Flight Control: The purpose of the vision-based flight control is to use vision information to estimate relative motion of a UAV to the surrounding environment. Normally, such estimated motion is integrated with inertial sensors to obtain the displacement and velocity of the UAV, which are used in the feedback control to stabilize the UAV.
3. Vision-based Navigation: Vision-based navigation aims to estimate and control the location and motion of a UAV flying from one place to another by integrating vision sensing technologies with measurements of other navigation sensors.

In the following parts, these types of vision-based systems for UAVs will be surveyed in terms of their applications, and techniques adopted.

1.2.1 Vision-Based Target Acquisition and Targeting

Vision-based target acquisition and targeting approaches are widely using in many applications, including target tracking and following, autonomous landing and formation, and so on. In those applications, visual information is used to produce precise measurement of relative position between the target and the UAV, and the visual information is applied in feedback control. Different vision techniques adopted in these applications are presented in the following sections.

Vision-Based Target Detection and Tracking

Vision-based object detection and tracking is a fundamental task of advanced applications of vision-based unmanned helicopters. The authors in [80] presented a vision system for an unmanned helicopter to detect and track a specified building. Two feature tracking techniques were applied and analyzed. A model-based tracking algorithm was proposed based on a second order kinematic model and Kalman filtering technique. In [99], Ha et al, presented a real-time visual tracking approach based on a geometric active contour method, which was capable of realizing air-to-air tracking of a fix-wing airplane. However, the main focus of these applications is on the design of the tracking control law. In fact, robust and efficient vision-based detection and tracking schemes cannot be ignored due to their priority in a machine vision system.

In addition, a successful implementation of targeting for a UAV was presented in [122,91]. A small-size glider was developed to fly automatically to a specified target with information from an on-board vision sensors only. A fast image processing algorithm, executed in a ground station, was proposed to detect the target. An extended Kalman filtering technique was also used to estimate the states of the glider with information extracted from the captured images.

Although numerous vision-based detection and tracking technologies were proposed in machine vision societies [121,131], the issue of real-time processing constrained their implementation in vision-systems for UAVs [89]. Moreover, another challenge comes from the complex and dynamic environment surrounding UAVs.

Vision-Based Landing

Vision-based landing is considered to be a special case of target acquisition and targeting. The main challenge of vision-based landing is altitude change, which significantly change the scale of landmarks in image. Therefore, vision algorithms should be robust enough to

cope with the scaling of landmarks. Another challenge is altitude estimation during UAV landing.

A vision-based system for landing a UAV on a ground pad was reported in [107]. A differential ego-motion estimation approach was employed to observe the states of the UAV with known initial values, including the relative position and velocity. These estimates were integrated with the flight control as a state observer to realize autonomous landing. However, this work is mostly focused on simulation.

Another work from the same group presented a vision-based landing system for an unmanned helicopter [109]. This system was also composed of two single board computers: Pentium 233 MHz Ampro LittleBoard computers. One computer was used in the vision computer system, and another was used as the navigation computer system. A vision algorithm was proposed based on the corner detection approach to detect a well structured landing pad, and implemented in the vision computer. In order to estimate the pose of the helicopter with respect to the landing pad, both linear and nonlinear optimal algorithms were employed. The pose estimates obtained with the linear optimal algorithm were used as the initial values of the nonlinear optimal algorithm to obtain more precise pose estimates. However, this vision system could not provide the velocity estimates of the UAV. Moreover, the employed fast corner detection might not be robust enough in complex and dynamic environments.

Moreover, researchers in the University of Southern California designed a similar vision-based autonomous landing system, reported in [102]. This system was based on shape detection method and invariant moments. A gas-powered radio-controlled helicopter was chosen as the platform, equipped with a RT-20 DGPS, an IMU unit, a color CCD camera and a PC/104 stack. In order to realize autonomous landing, a vision based algorithm was proposed using moment invariants of geometric shapes of objects. Three lower-order invariant moments were used, which are invariant with respect to scaling, translation, and rotation. The position and orientation of the target relative to the helicopter was computed.

A hierarchical behavior-based controller was designed for the helicopter. Based on the vision feedback, autonomous landing of the unmanned helicopter on a specified landing pad was realized.

1.2.2 Vision-Based Flight Control

The earliest exploration on vision-based stabilization and flight control was reported in [3]. A system called “visual odometer” was presented to estimate the 3D motion of an unmanned helicopter by combining the lateral and longitudinal image displacements with the measured attitude of the helicopter. The image displacements were computed with an image template matching method. An initial target was selected in the first image, and then its location in subsequent image is detected and used to estimate the position of the helicopter. If the selected target moved outside the view field of the on-board cameras due to the movement of the helicopter, a new target would be selected and updated dynamically.

To overcome the changing of appearance of the target candidate as the helicopter adjusts the altitude, a pair of target templates and a small baseline were selected. Before matching templates, the scale and orientation of the target template pair were corrected using the magnitude and angle of the baseline from the previous match. Experiment results showed that “visual odometer” was able to stabilize motion of a small-size model helicopter by integrating information from a set of inexpensive angular sensors. However, “visual odometer” is unable to provide the velocity estimate that is important for closed-loop control. In addition, the proposed vision system is constructed based on TI DSPs, which may require more time and effort to develop the embedded software.

Instead of using the DSP-based processors, a low-weight and low-power FPGA-based vision system was reported in [41]. Harris corner detection and template matching algorithms were implemented in the custom-made FPGA hardware. The vision feedback was combined with the *Kestrel Autopilot* Inertial Measurement Unit (IMU) developed by BYU MAGICC Lab [8] to realize the drift-free control for a Micro UAV in indoor environments.

The IMU was used to stabilize the attitude of an aerial vehicle, but it would not eliminate the drift caused by uncertain air flow and sensor drift. An on-board vision system was utilized to correct such drift, which was not detected by the on-board IMU. The correction was estimated by measuring feature movement through consecutive images captured by an on-board camera. In this application, the vision and on-board navigation sensors are closely coupled to realize the drift-free hovering of a micro aerial vehicle. The special hardware, FPGA, was used to realize time-consuming vision algorithms, such as the corner detection and template matching. But the custom-made hardware systems may also require longer development period.

Although numerous applications of vision-based stabilization were reported, vision-based stabilization is still a challenge for both indoor and outdoor UAVs in the GPS-denied and landmark less environments. The fundamental problems include hardware implementation, and fast but robust velocity estimation techniques. The detailed discussion will be given in Chapter 5.

1.2.3 Vision-Based Navigation

Many research teams focused on vision-based navigation systems applied in unmanned vehicles, such as obstacle detection and avoidance, urban/indoor navigation, simultaneously localization and mapping, and mapping. We will discuss them in detail in the following part

Obstacle Detection and Avoidance

An autonomous exploration method was proposed in [111] for navigating UAVs in unknown urban environments. A local obstacle map was built by detection of surrounding area using an on-board laser rangefinder. Based on the local obstacle map, a model predictive control framework was addressed to generate a conflict-free flight trajectory in real time. The updated trajectory was sent to the position tracking layer in the UAV avionics. In

addition, researchers at the University of California, Berkeley were also leading military-funded research into development of swarms and formations of unmanned aircraft able to navigate in and around building and cityscapes. The effort involved high-level autonomy, multi-sensor integration and multi-aircraft coordination.

A vision-based navigation system was addressed in [56] to guide a UAV fly through urban canyons. Optic-flow was proposed to work together with a stereo vision algorithm. Optic flow from a pair of sideways-looking cameras was used to keep the UAV centered in a canyon and initiate turns at junctions, while the stereo vision sensing from a forward-facing stereo head was used to avoid obstacles in front. They claimed that the combination of stereo and optic-flow (stereo-flow) was more effective at navigating urban canyons than either technique alone.

Vision based feature detection and tracking in an urban environment was investigated for an autonomous helicopter in [80]. The rectangular features in structured environments were detected by using on-board vision system, and tracked by using a GPS navigation system. Template matching based method was proposed to search and track rectangular features, such as windows in an urban environment.

For the navigation of UAVs in urban environments, most research focused on collision avoidance, obstacle sensing and evasion, target detection, as well as optimal path planning with the available GPS signal. However, the GPS signal is not always available in urban areas due to urban canyons. Low-cost GPS/INS systems, widely used in UAV applications, become more dependent on the availability and quality of GPS signal. If GPS signal is blocked even a short-term dropout, the performance of an inertial navigation system would be negatively affected significantly. Thus, the navigations system, which can also work without GPS, is crucial in the UAV navigating in urban environments. It is necessary to consider certain algorithms to combine measures of multiple sensors to achieve autonomous navigation and localization in such conditions.

Obstacle detection and avoidance methods are also important topics, which are inves-

tigated in detail in the following parts. Target detection can be achieved by using a laser range finder or a camera. For a laser range finder, a standard map generation method can construct the geometry environment, which is used for the obstacle avoidance and the path planning.

Another approach is to use the time-to-collision estimation to realize visual collision detection, where an image sequence from a forward looking camera is employed to compute the time to collision for surfaces in a scene [128]. Although it cannot find the absolute depth information, optical flow can tell us the time-to-collision, which is also useful information to avoid potential collisions. Flow divergence methods will be focused, which rely on the observation that objects on a collision course with a monocular image sensor exhibit expansion or looming.

Based on the generated map, we can perform the path planning and avoid the obstacles. There have been many studies on UAV path planning using various algorithm approaches, such as Dijkstra Algorithm, A* algorithm, Genetic Algorithm, Ant Colony Algorithm, Probability Roadmap, Potential Fields, Rapidly-exploring Random Trees and etc. These traditional computational geometry-based approaches to path planning can be classified into three basic categories; the cell decomposition method, the roadmap method, and the potential field method.

Simultaneously Localization and Mapping

An augmented system with a GPS/INS navigation system and a Simultaneous Localization and Mapping (SLAM) was presented in [65]. The vision-based landmark detection algorithm was used to generate a landmark-based map with GPS/INS signals when GPS signals were available. If GPS signals were lost, the landmark-based map was used to reduce the measurement error of INS.

One solution of navigation in a GPS denied environment is by using Terrain Aided Navigation System (TANS) which can relieve the dependency on GPS navigation system. This

type of navigation system typically makes use of on-board sensors and a preloaded terrain database. Terrain Contour Matching (TERCOM) system has been successfully applied in cruise missile navigation in [7]. However, it usually requires some sort of space-borne or air-borne mapping infrastructure as it is typically built from high resolution satellite or radar images of the mission area. Furthermore, it has a constrained degree of autonomy since the mission is bound to the knowledge of the terrain database.

In order to extend TANS, a new concept of terrain-aided navigation, known as Simultaneous Localization and Mapping (SLAM) [65, 66, 35, 6], employed to augment the existing GPS/INS system. SLAM was firstly addressed in the paper by Smith and Cheeseman in [115]. Contrary to TANS, SLAM does not require any pre-surveyed map database. It builds a map incrementally by sensing environments and uses the built map to localize the vehicle simultaneously, which results in a truly self-contained autonomous system.

SLAM algorithm is a landmark based terrain aided navigation system that has a capability for online map building, and simultaneously utilizing the generated map to bound the errors in the Inertial Navigation System (INS). The mathematical framework of the SLAM algorithm is based on an estimation process, when given a kinematic/dynamic model of the vehicle and relative observations between the vehicle and landmarks, estimates the structure of the map and the position of vehicles, velocity and orientation within that map. In [65] and [66], a SLAM-augmented GPS/INS system was proposed based on certain landmarks for GPS denied environments, which can be used to build the local map and estimate the states of the aircraft simultaneously. If GPS signal was available, it worked like normal GPS/INS navigation system and built the landmark based map. If GPS signal was not available, the INS error was constrained by the generated landmark based map, and the map was also updated on-line in real time.

1.3 Challenges in Vision-Based UAVs

In the last three decades, vision sensors have been extensively explored in control systems because of their unique advantages, which can provide a huge amount of information on objects and surrounding environments. By analyzing visual information, relative positions of objects and situation of the surrounding environment can be obtained and applied to control and navigation. Although such integration of vision and robots achieved remarkable success in the last two decades, the machine vision is still a challenge due to inherent limitations [52]:

1. The way that biological vision works is still largely unknown and therefore hard to emulate on computers, and
2. Attempt to ignore biological vision and to reinvent a sort of silicon-based vision has not been as successful as initially expected.

Additionally, when information from visual sensors is applied in real-time control systems, many difficulties has to be solved due to the huge amount of image data such as,

1. Automated image interpretation and object recognition are important tasks in numerous vision-based control systems. The objective of the tasks is to establish the model-to-data correspondence with one or a couple of images in real time or online;
2. Precise measurement of the relative position and motion of objects in the images is necessary by fusing information from the images and other sensors. With the precise measurement, the designated objects have to be held in the vision field by controlling orientation of the camera so that image capture of the designated objects can be carried out efficiently;
3. Fusion of attitude dynamics of the camera and kinematics of its carrier such as aircraft and cars is needed, so that control design can be based on a complete system;

4. Reconstruction of the 3D structure of the environment by fusing the information from vision and other sensors is important to realize the autonomous navigation in an unknown and dynamic environment.
5. Moving platform and moving target, which may cause large motion of background in the image, as well as significant changes of shape, size and appearance of targets in the image. That may caused many tracking algorithms fail.
6. Real-time and on-board processing of vision algorithms.

Such challenges are to be overcome in future work to implement ideal integration of visual information and those from other sensors adopted. Numerical computation will play an important role in overcoming those challenges.

1.4 Motivation and Contributions of This Research

It is noted that most of the works reported in the literature, however, focus only on certain parts of vision systems for UAVs, such as hardware construction or vision algorithms. Many of these are adopted from those designed for ground robots, which are not very suitable for applications on UAVs. To the best of our knowledge, there is hardly any systematic documentation in the open literatures dealing with the complete design and implementation of a vision-based unmanned helicopter, which includes architectural and algorithmic design of real-time vision systems. In addition, although target tracking in video sequences has already been studied in a number of applications, there has been very little research related to the implementation of vision-based target following for UAVs, and motion estimation in GPS-denied environments.

1. In this thesis, the design and implementation of a comprehensive real-time embedded vision system for an unmanned rotorcraft is presented, which includes an on-board

embedded hardware system, a real-time software system and a mission-based vision algorithm. More specifically, the on-board embedded hardware system is designed to fulfill the on-board image acquisition, real-time processing and tracking control requirements by using off-the-shelf commercial products, such as PC/104 embedded modules. A comprehensive design methodology is proposed for the design of the hardware system. The hardware construction of the vision system is optimized using a novel computer-aided technique. Anti-vibration design is considered due to the demanding working environment during the flight of unmanned helicopters;

2. Based on the on-board vision hardware system, a real-time vision software is developed, which is running on the real-time operating system QNX. As an embedded microkernel operating system, QNX requires less computation resources and can be tailored to suit most embedded systems. Under the QNX operating system, a multiple-thread is implemented to coordinate multiple tasks, such as image acquisition, processing, communication, and pan/tilt servo mechanism control;
3. An advanced vision algorithm is then proposed and implemented to realize ground target following, which utilizes robust feature detection and tracking scheme. This proposed vision scheme is integrated with on-board navigation sensors to estimate the relative distance between the target and the UAV. Finally, using the vision feedback, a two-layer target tracking control framework is utilized to control a pan/tilt servo mechanism to keep the target in the center of the image, and guide the UAV to follow the motion of the target. The overall vision system has been tested in actual flight missions, and the results obtained show that the proposed system is very robust and efficient;
4. In addition, a sophisticated and systematic vision-augmented approach is presented to realize motion estimation of the UAV in GPS-denied conditions. This approach is composed of robust landmark detection and a core algorithm for vision-based motion

estimation, which is primary contribution of the work. In this thesis, a well-structured landmark is used as the reference. To realize robust key feature point extraction and correspondence, a hierarchical detection scheme is employed. The pattern structure will be identified first, and then the key feature points will be extracted from it, even in partially occluded conditions. Special feature point correction procedure is used to eliminate impact of the noise in feature point extraction to obtain optimal extraction results;

Based on the 3D model and corresponding 2D image points, the pose estimation algorithm is proposed to estimate the relative position and angle of the aircraft with respect to the ground reference. The velocity of the aircraft is estimated with the measurement of the position, and can be improved by fusing IMU measurements using a Kalman filter, which can provide the necessary information for hovering control of the unmanned helicopters.

1.5 Outline of This Thesis

The remainder of this thesis is organized as follows: The design and implementation of hardware and software of the vision-based unmanned helicopter is presented in Chapters 2 and 3 respectively. The vision-based ground target following algorithms and flight test results are detailed in Chapter 4, and was verified in actual flight tests. A systematic design and implementation of a vision aided motion estimation approach for an unmanned helicopter in the GPS-denied condition is given in Chapter 5, and experimental results of the vision system obtained through actual flight tests are presented. Finally, some concluding remarks are drawn in Chapter 6.

Chapter 2

Hardware Design of the Vision-Based Unmanned Helicopter

2.1 Introduction

This chapter presents the hardware development of a vision-based unmanned helicopter, SheLion. It is noted that the flight control system and the ground station for SheLion are similar to the previously reported platform, HeLion [18]. In what follows, the new feature of SheLion: the vision system, will be focused. Moreover, to realize systematic integration of the vision system with the unmanned helicopter, a comprehensive design methodology for the vision-based unmanned helicopter is presented, including hardware components selection, design and integration, as well as experimental evaluation.

2.1.1 Related Work

Here, vision systems used in various applications of UAVs are presented, and ideas and technologies behind these applications are also investigated.

To perform vision processing in flight, two main integration modes for vision systems and UAVs are widely used: 1) the ground processing mode, and 2) the on-board processing

mode. The main advantage of the ground processing mode is that a light-weight on-board system can be established. But vision signals had to be transmitted to ground stations for processing, and then the results were sent back to UAVs. One example has been reported in [122, 91]. A small-size glider equipped with an on-board vision sensor was guided by a ground vision system to fly automatically to a specified target. The ground vision system was used to execute a fast image processing algorithm to estimate the states of the glider to the target. Another work on using the ground processing was also proposed in [53] to realize target tracking and obstacle avoidance. However, such transmission-decision-transmission manner causes many problems in the vision-based control, including extra noise in the images, and transmission latency. This integration mode greatly limits the operating range of UAVs, and the responsiveness of UAVs in highly dynamic environments [41].

To increase flexibility of vision-based UAVs, the on-board processing mode has attracted much interest recently. Thanks to the rapid development of computer technologies, on-board and real-time vision processing becomes feasible for small-scale UAVs by using embedded processing modules.

To realize the on-board processing, it is necessary to select suitable hardware components. Generally speaking, the hardware development of the avionics costs a lot of time and effort regardless the processing modes. Therefore, for the research-based applications, off-the-shelf hardware components are strongly recommended, such as single-board-computers, commercial navigation sensors, industrial CCTV cameras, standard power supplies, and more. Integration and debugging based on such standard hardware components is generally easier compared to constructing all the components from scratch. Such vision systems can also provide acceptable and reliable performance.

Based on such a concept, for instance, a vision system using an 850 MHz Pentium III Embedded PC with 2GB Flash Drive was proposed in [63]. The vision algorithm can be executed up to 10 frame per second on-line. Another vision system based on the PC-104 single board computers with the similar performance was also proposed in [102] to realize au-

tonomous landing. A gas-powered radio-control (RC) helicopter was chosen as the platform. The UAV system was equipped with a RT-20 DGPS, an IMU unit, a color downward-looking CCD camera and a PC-104 stack using Tiny886ULP 800 MHz Crusoe based processor board. The video signals were transferred to the ground station for monitoring.

To realize advanced vision algorithms, powerful processors are definitely required. Single board computers with the Atom 1.1 GHz or 1.6 GHz processors have been widely used recently. But power consumption will increase significantly. A vision system using a Lipert CoreExpress 1.6 GHz Intel Atom board with a wifi link was proposed in [1]. Feature detection and frame to frame motion estimation algorithms were implemented to realize autonomous navigation of a quad-rotor helicopter in indoor environments.

On the other hand, for micro UAVs, super light-weight and small-size hardware components are expected, such as the Gumstix-like single board computers and tiny sensors. The authors in [87] presented a vision system developed based on the Gumstix Overo fire 600 MHz and a webcam to realize the vision-aided indoor navigation. A novel and efficient vision algorithm was proposed to realize the robust landmark tracking and path generation on board. The flight tests verified the robustness and efficiency of the proposed system.

As mentioned above, off-the-shelf hardware components are used to save time and effort in the development stage. However, to reduce the size and weight of a vision system or execute time-consuming algorithms in certain applications, custom-made hardware modules are expected.

For instance, the “visual odometer” system, reported in [3] and [127], consisted of a pair downward looking black/white cameras and a custom-made vision system with six TI C44 DSPs. The proposed system was used to realize the vision-based feedback control, target detecting and tracking for an unmanned helicopter. An appearance based template detection was employed to detect and track ground objects. Each of the TI C44 DSPs was used to execute a 32×32 pixel template matching. To provide the helicopter position, an approach was proposed to estimate the helicopter position with the visual information from

the on-board cameras. In addition, a FPGA based vision system was proposed in [41] to realize the drift-free control for a Micro-UAV in indoor environments. This vision system was called Helios, composed of SDRAM, SRAM, a Virtex-4 FPGA, and USB connectivity. Harris corner detection and template matching algorithms were implemented in the custom-made vision system, which was used to detect the drift of the helicopter in X- and Y-axis. However, the custom-made systems require more skills and effort to design the hardware modules, and develop drivers for them.

In addition, the configuration of using two separated embedded computers in an on-board system is recommended: one for flight control, and another one for machine vision algorithms. This configuration is recommended due to the following reasons:

1. The computation consumption of flight control task and vision program are very heavy, which can hardly be carried out together in a single embedded computer;
2. The sampling rate of the flight control law is faster than the operation of vision algorithms, since the faster sampling rate is required to stabilize the unmanned helicopter;
3. The two-computer structure reduces the negative effect of data blocking caused by the vision program and the flight control system, and thus makes the overall system reliable.

The two-computer configuration, for example, was employed in the avionics of the vision-based UAV [109], which utilized two single board computers: Pentium 233 MHz Ampro LittleBoard computers, for navigation and vision processing respectively. A Yamaha R-50 helicopter was used as the platform, which can provide the sufficient payload of 20 Kg for the avionics. The vision approaches were developed to realize the application of landing the unmanned helicopter autonomously.

Layout design of the on-board system, including the flight control system, the vision system, the power supply system and the anti-vibration system, is another critical step

that requires extensive time and effort. To speed up the design procedure, the on-board system can be built virtually in a computer, before physically constructing the hardware components. A CAD software is chosen as a virtual design tool, such as SolidWorks, which is easy to use, and has powerful 3D and 2D design features [16].

In summary, here, several important rules on hardware design of the vision systems for UAVs are proposed:

1. On-board vision processing : On-board processing can significantly increase the flexibility of UAVs in the demanding applications, though it may lead to the challenges of hardware development and efficient algorithms;
2. Using off-the-shelf hardware components : Constructing vision system using the off-the-shelf hardware can greatly save the effort and time in the development;
3. Two-computer configuration : Such configuration can make the entire system more reliable and stable, and it is also easy to upgrade the individual system in future;
4. Virtual design : Using virtual design tools to speed up the iterative design procedure in the hardware design.

In the following part, the detailed design and implementation of the hardware platform of the vision-based unmanned helicopter will be presented.

2.1.2 Requirements

The vision system built in the project is applied to provide visual information to implement autonomous vision-based applications of UAVs. For instance, unmanned helicopters can follow a certain path, detect and track objects of interest on the ground or in the sky, as well as estimate the relative pose and location. In order to complete these tasks effectively and robustly, the vision-based UAV should hold the following functions and properties:

1. To capture the designated targets and collect images of the targets;
2. To analyze the image data on-board in real time;
3. To carry out data fusing to complete more advanced tasks, such as vision-based stabilization, control, and scene re-construction;
4. To communicate with the ground supporting system, and send back the visual information for monitoring;
5. The weight and size of the avionic system should be suited for small-size unmanned helicopters;
6. The avionic system should be anti-vibration and has less effect to movement of the mass center of helicopters;
7. The cost of the avionic system is a bargain.

2.2 Configuration of Hardware Components

In this project, to fulfill the above requirements, a vision-based unmanned helicopter, named Shelion, was developed. The schematic diagram of SheLion is shown in Figure 2.1, which consists of several main parts as follows:

1. Platform : A small-size radio-controlled (RC) helicopter is used as the platform, which is fully equipped for manual operation;
2. On-board Flight Control System : It is mainly composed of a flight control computer, navigation and inertial measurement units, as well as communication units. The flight control computer is embedded the main program and control law to achieve autonomous flight. Navigation and GPS sensors: integrated in MNAV100CA, provide the states of the UAV used in the flight controller. The onboard wireless modem,

which communicates with the ground station. This wireless modem is a duplex radio, which is able to send the states of the UAV to the ground state and receive commands from the ground station at the same time;

3. On-board Vision System : It includes a vision sensor, a pan/tilt servo mechanism, an image acquisition module, a vision processing module, and video-link. This set provides necessary components to achieve autonomous target detection and tracking in the image. The vision algorithm is embedded in the vision computer;
4. Ground Supporting System : To provide user interface and high-level command, as well as telemetry, a ground supporting system is developed. It includes a ground wireless transceiver and a laptop computer, which is able to monitor the states of the unmanned helicopter in real-time in a friendly user interface, and send the command to the unmanned helicopter through this interface.

In the following parts of this chapter, the details in design and assembling of the unmanned vision-based helicopter SheLion will be presented.

2.2.1 Radio Controlled Helicopter

A high quality radio controlled (RC) helicopter, Raptor 90 SE, is selected as the basic rotorcraft of SheLion to carry the avionics, which is shown in Figure 2.2. Some key physical parameters of the helicopter are listed in Table 2.1. Five onboard servo actuators are used to drive the helicopter. More specifically, the aileron, elevator and collective pitch servos are in charge of tilting the swash plate to realize the rolling motion, pitching motion and to change the collective pitch angle of the main rotor. The throttle servo, cooperated with a hobby purpose RPM governor, is used to control the engine power. One high-speed digital servos, associated by a low cost yaw rate gyro, is employed to control the yaw motion. Digital servos, a digital receiver and a digital gyro provide extremely fast response time. The commonly used stabilizer bar, which acts as a damper to reduce the over-sensitive

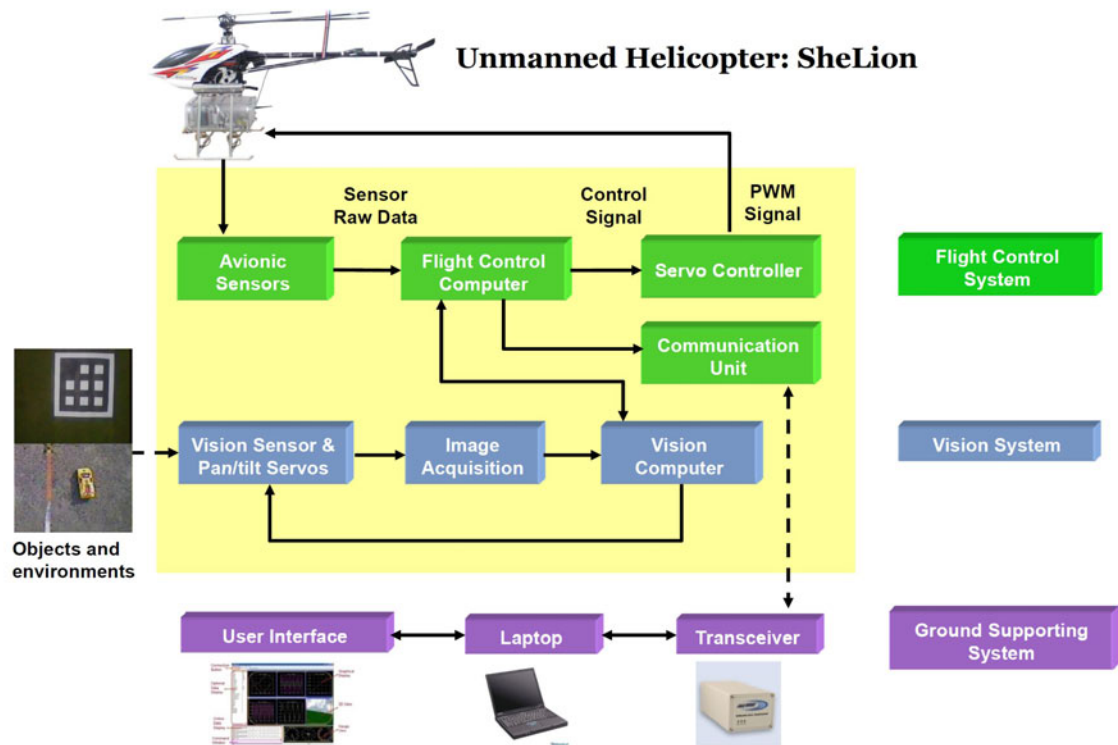


Figure 2.1: Overview of the vision-based unmanned helicopter: SheLion

Table 2.1: Main specifications of Raptor 90 SE

Full Length of fuselage	1410mm
Full width of fuselage	190mm
Total height	476mm
Main rotor diameter	1605mm
Tail rotor diameter	260mm

aerodynamic forces caused by the ultra small size of helicopter, is also equipped to facilitate manual control.

In summary, the Raptor 90 helicopter is well suited to the platform of the project because of the following three reasons:

1. Great maneuverability :



Figure 2.2: Radio controlled helicopter: Raptor 90 SE.

Raptor 90 helicopter is originally designed for F-3D acrobatic flight. Its agility and maneuverability are both famous in the RC hobby flight circle. Correspondingly the upgraded unmanned helicopter holds more control flexibility compared to those upgraded from RC helicopters developed for F-3C stable flight;

2. Large payload :

The equipped OS-91SX engine is capable of generating 3.1 ps at 15,000 rpm, resulting in the maximum taking off weight up to 11 kg. Since the dry weight of helicopter is about 4.9 kg, the effective payload is up to 6 kg, which suits well with the budget of the weight (i.e., 3.5 kg) for the onboard computer system and provides sufficient room for future upgrading;

3. Low cost but high performance :

Compared with other expensive but same size RC helicopters such as Hirobo-90 and Bergen Industrial Twin, Raptor 90 helicopter provides the same high quality flight performance but at a half price.

2.2.2 Flight Control System

As illustrated in Figure 2.1, the Flight Control System consists of several main parts: a flight control computer, avionic sensors, a servo controller, and a communication unit. We will describe them in the following parts.

Flight Control Computer

The onboard computer processor stack, which is a combination of multiple processor boards, acts like the brain of the unmanned helicopter system. For the construction of SheLion, PC-104 embedded single board computer (SBC): Athena, is adopted as the control computer because of the following three features: (1) the small but uniform size ($96\text{ mm} \times 90\text{ mm} \times 10\text{ mm}$); (2) light weight (normally less than 200 g); and (3) anti-vibration structural design (pin-and-socket bus connection method). The most challenging issue we are facing is to ensure the working efficiency while strictly avoiding computational overloading and software crash during actual flight tests.

The main tasks of the flight control computer (see Figure 2.1) are: (1) collecting data from INS/GPS and RPM sensors, sonar and servo controllers; (2) analyzing collected data; (3) implementing flight control laws and driving servo actuators through servo controllers; (4) logging the in-flight data to CF Card for post-flight analysis; (5) communicating with the vision computer; and (6) communicating with the ground supporting system. Although there are multiple flight missions involved, it has been proved in [31] that the computational load for normal flight tests is fairly light (less than 23% in the CPU usage at the peak) for a 600 MHz CPU board used in HeLion. Consideration for the selection of the flight control computer for SheLion is focused more on reducing the weight and power consumption while maintaining the system safety and working efficiency. We choose a PC-104 ATHENA, which has four RS-232 serial ports, a 16-pin analog-to-digital (A/D) port, two counters/timers and runs at 600 MHz. PC-104 ATHENA is a 3-IN-1 board, which integrates all of the necessary



Figure 2.3: PC-104 embedded single board computer: ATHENA

Table 2.2: Main specifications of PC-104 ATHENA

Processor	Low-power Pentium-3 equivalent
Speed	400MHz
Power consumption	10 watts
Cooling	Heat sink
Memory	128MB soldered on board
Serial ports	4 \times RS232
USB ports	4, version 1.1
Analog inputs	8 channels, 16 bits, 100KHz maximum
Analog outputs	4 channels, 12 bits
Power supply	+5VDC
Dimensions	106 \times 113 mm
Weight	150 gram

functions of a main processor board, a serial communication board used for data exchange with INS/GPS, servo controller and wireless modem, and a data acquisition board used for data exchange with a RPM sensor. As a result, the weight and power consumption is greatly reduced to 30% and 50%, respectively. PC-104 ATHENA is shown in Figure 2.3. Main specification is listed in Table 2.2.

Avionic Sensors

Avionic sensors are responsible for collecting all of the information of the unmanned helicopter during the flight. For SheLion, three avionic sensors are adopted: (1) an inertial navigation system/global positioning system (INS/GPS) measuring all of the necessary helicopter states; 2) an ultrasonic sonar measuring the altitude in the near ground level; and 3) an RPM sensor recording the RPM of the main rotor. Their selections are based on the following.

The core navigation sensor, i.e., INS/GPS, is selected in accordance with the requirements on its output signals:

1. The essential signals that the INS/GPS is to provide are three-axis angular rates, three-axis accelerations, three-axis magnetics and three-axis positions. The first three are in body frame of the UAV and the last one resides in the north-east-down (NED) frame. It is noted that the three-axis Euler angles are not necessary since they can be estimated by using an extended Kalman filter (EKF) as reported in [62] and complementary filtering reported in [129].
2. The measuring ranges of the three-axis accelerations, three-axis angular rates and three-axis magnetics are set as ± 2 g, $\pm 150^\circ$ and ± 0.7 Gauss, respectively, in accordance with the specifications of the commonly used commercial products. The selected threshold values are reasonable since we do not intend to cover the extreme or acrobatic flight conditions. As a result, the acceleration, angular rate and magnetics are not to change dramatically during flight tests. Based on this setting, we need to carefully perform an anti-vibration design to avoid the measurement saturation caused by various vibration sources associated with the UAV. This is to be addressed in Section 2.3.
3. On the basis of meeting all of above mentioned requirements, the size, weight and power consumption of the adopted INS/GPS should be minimized.

Table 2.3: Main specifications of MNAV100CA

	Requirements	MNAV100CA
Acceleration Range X/Y/Z (g)	± 2	± 2
Angular Rate Range (deg)	± 150	± 200
Magnetometer Range (G)	± 0.7	± 0.75
GPS Accuracy in CEP (m)	≤ 3	3
Sampling Rate (Hz)	≥ 50	{1, 25, 50, 75, 100} selectable
Size (mm)	$\leq 76 \times 97 \times 76$	$57 \times 45 \times 11$
Weight (g)	≤ 580	33
Power Consumption (W)	≤ 5	≤ 0.8

A compact INS/GPS, namely, MNAV100CA, shown in Figure 2.4 along with the virtual counterpart, is selected for SheLion. The key specifications of this sensor are listed in Table 2.3, which clearly shows that all of the requirements are satisfied. Furthermore, by using MNAV100CA, the weight and power consumption of the INS/GPS sensor is greatly reduced to 5.6% and 16%, respectively, compared to those of the fully integrated INS/GPS, NAV420CA, installed on HeLion. Note that MNAV100CA adopts the same sampling rate as NAV420CA, i.e., 50 Hz, which has been proven sufficient for small-scale unmanned helicopter control by many successful results (see, for example [13, 44, 67, 112]). Section 2.4 will demonstrate that MNAV100CA can yield the similar level of working performance as the NAV420CA, which is more expensive.

The ultrasonic sonar is capable of providing altitude signal in near ground level. Due to the inaccuracy of the GPS signals, the altitude signal generated by the sonar is the key reference for automatic takeoff and landing processes. SheLion adopts an ultrasonic sonar, namely, SNT-UPK2500, with a resolution in the mm range and a weight of 50 g. The effective range is up to 2 m, which is sufficient for automatic takeoff and landing.

The RPM sensor, Futaba GV-1, is a commercial product which is commonly used in the RC hobby flight circle. To simplify the overall design, we retain this product in SheLion but with necessary modifications to obtain the RPM number, which is originally set as an

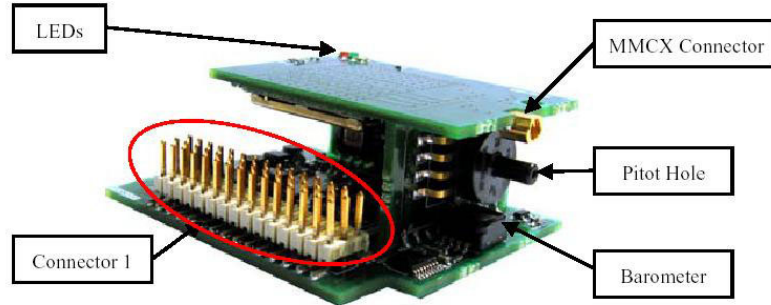


Figure 2.4: Navigation sensor: MNAV100CA

internal signal. More specifically, we connect the RPM sensor to a Schmidt Trigger and the output of the Schmidt Trigger is then sent to a counter/timer port resided in PC-104 ATHENA processor board.

Servo Controller

Servo controller is used to realize smooth switching between the manual control mode and automatic control mode, which is shown in Figure 2.5. The requirements for the servo controller are listed as follows:

1. Reliable switching function: The switching between automatic control and manual control should be both fast and smooth. A particular channel must be assigned to ensure the reliability.
2. Sufficient input/output channels: For most RC helicopters, five onboard servos are equipped to drive the helicopter. Adding an extra channel for switching function and some necessary redundancy, the input/output number must not be less than 7.

3. Capability of recording servo actuator's input signal: This function is important in initial manual flight tests. The recorded input data are essential for deriving the dynamical model of the UAV and for evaluating control performance.
4. High resolution: Substantially the input-recording and servo-driving function are the A/D and D/A procedure. The resolution should be sufficiently high to ensure the data consistency and accuracy.

The final selection of the servo controller, an HBC-101, is a 8-input/8-output digital signal processing (DSP) board with a resolution of 0.009° . RS-232 serial protocol is used to exchange data with PC-104 ATHENA. Input channels 2-6 and serial port are assigned to receive the manual input signals and ATHENA-generated auto input signals, respectively. Channels 7 and 8 are currently not in use. Channel 1 is preoccupied by the switching function. The switching signal comes from the manual control joystick. By doing so the pilot owns the highest authority to determine which side (automatic or manual input) is mapped to the output. Such a *piloted-highest-control* design is especially important during some unexpected situations since the pilot can immediately retrieve back the manual control to avoid accident or crash.

Communication Unit

The communication unit includes a pair of the FreeWave wireless data transceivers. This pair of transceivers establish communication between the onboard system and the ground station. They are configured to operate in point-to-point mode and working in 2.400 to 2.4835GHz. The transceiver used in the onboard system is set as a point-to-point slave, and connected to flight control computer board. The transceiver in ground station is set as point-to-point master, and connected to the ground laptop.

The wireless communication between SheLion and the ground supporting system is realized by a pair of serial wireless radio modems (one installed on the UAV and the other

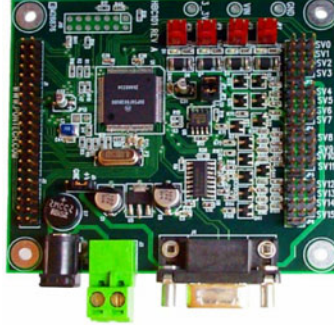


Figure 2.5: Servo controller

on the ground supporting system). We select the FreeWave IM-500 wireless modem system with a light weight (75 g), high throughput (115.2 kbps), wide range (up to 32 km in the open field environment) and a working frequency at 2.4 GHz. The onboard and ground transceiver is shown in Figure 2.6 and 2.7.



Figure 2.6: On-board wireless transceiver

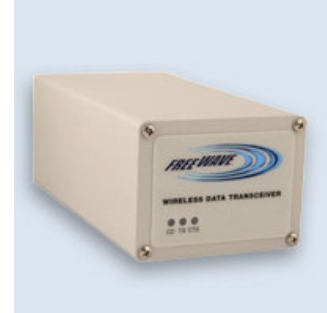


Figure 2.7: Ground wireless transceiver

2.2.3 Vision System

In this part, the vision system employed in SheLion is presented, which serves a wide range of vision applications as mentioned before. The hardware configuration of the vision system

consists of the following five main parts, which is illustrated in Figure 2.8:

1. The vision sensor, which is employed on-board to obtain in-flight visual information of the target and surrounding environments;
2. The pan/tilt servo mechanism, which aims to keep the target objects in the field of view of the camera to increase the flexibility of the vision system;
3. The image acquisition module to digitalize the analog video signals before being processed;
4. The vision processing module that is the key unit of the vision system. It coordinates the overall vision system, such as image processing, vision algorithms, and communicating with the flight control computer, and so on;
5. The video- and data-link that is used to provide ground operators clear visualization in monitoring the work that the on-board vision is processed during flight tests, the video captured by the on-board camera is transmitted and displayed in the ground control station.

In these applications, the vision system is required to have the capabilities of working in an environment full of vibrations and perform required tasks in real time such as image acquisition, image processing and data logging. The components of the vision system will be described in detail in the following sections.

Vision Sensor

A vision sensor is employed on-board to obtain visual information of the surrounding environments of the UAV in-flight. Interested visual information is composed of static and dynamic features, such as the color and shape of landmarks, and motions of vehicles. Typically, a single camera will be used in the most of the vision applications, such as target

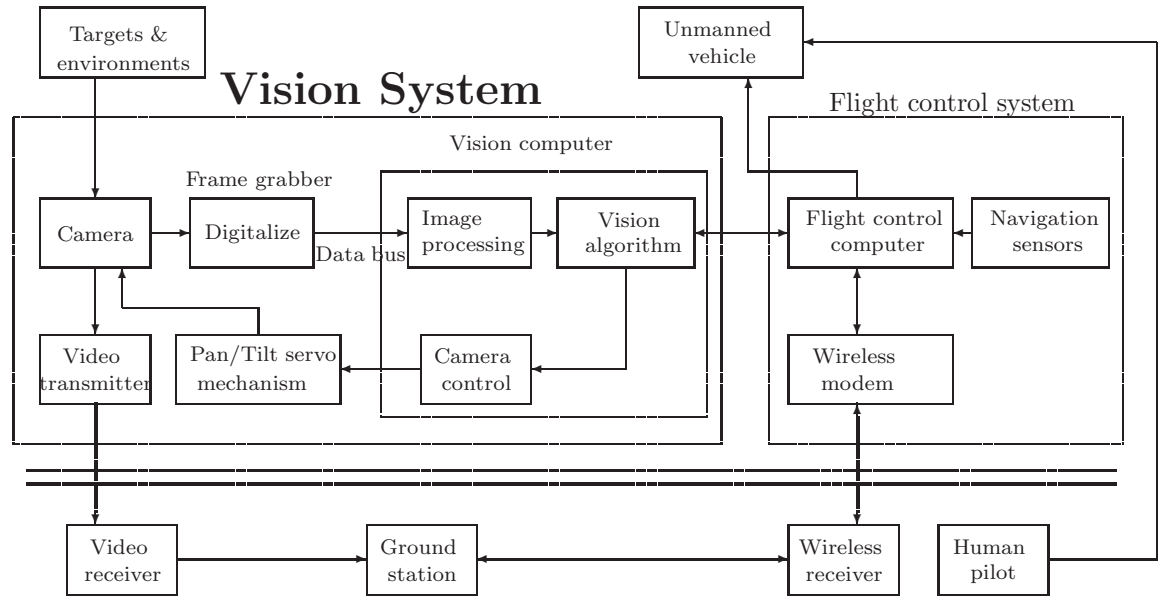


Figure 2.8: Hardware configuration of the vision system.

tracking, identification, etc., but in the applications that the relative distance measurement is required, the stereo vision will be considered. For an individual camera, several aspects of the camera need to be considered, including sensor, interface.

The purpose of a vision sensor is to convert an optical image to electric signals in a digital camera. There are two kinds of the main sensors are widely used: a charge-coupled device (CCD) and a complementary metal-oxide-semiconductor (CMOS). In fact, they are doing the same thing that converts light into electrons. With the rapid growth in the electronic and semiconductor technologies, the difference between these two kind of sensors is not significant. Both of them can provide similar image quality. The slight difference is that the CMOS sensor will provide faster readout than CCD and require less power, as well as less expensive in the market due to fewer components. By contrast, CCD sensors use more mature technologies compares to the CMOS in most aspects of manufacturing.

In the vision system, a color video camera is selected as the on-board visual sensor, which has a compact size and a weight less than 30 g, as well as 380 TV line resolution and 40-degree field of view. This video camera uses a CCD sensor that is capable of providing higher quality and sharper image compared to that offered by CMOS sensors. The sharpness of the image is important for feature extraction, especially for color feature extraction. The on-board CCD video camera outputs analog video signal in the PAL TV standard.



Figure 2.9: On-board vision sensor

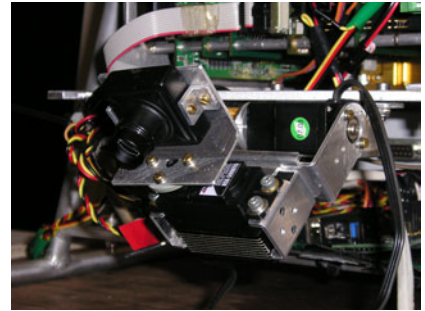


Figure 2.10: Pan/tilt servo mechanism

Pan/Tilt Servo Mechanism

In the vision-based navigation, it is required to keep the target objects, such as the landmark, in the field of view of the camera to increase the flexibility of vision-based tracking. As such, we select to mount the camera on a pan/tilt servo mechanism that can be rotated in the horizontal and vertical directions as shown in Figure 2.10. A custom-made pan/tilt servo mechanism has a weight less than 200 gram and two degrees of freedom with a resolution of 0.04 degrees. It is small in size and less expensive compared to most of the air photography gimbals available in the market, and is well suited for small-scale helicopters with limited payload and space.

To reduce the effect of vibration to the camera, the entire pan/tilt servo mechanism is mounted under the infrastructure of the unmanned helicopter. This infrastructure stays on

four spring-roll vibration isolators. Moreover, the pan/tilt servo mechanism is equipped with two digital servos using titanium alloy gears and ultra hardness shaft such that the overall structure of the mechanism is more stable and less sensitive to vibration. These pan/tilt servos are controlled by a small-size and high-accurate servo controller: Pololu serial 8-servo controller that is able to convert control signals given by the vision computer to PWM signal for controlling the servos. The tracking control algorithm for the pan/tilt servo mechanism is to be presented later in Chapter 4, which aims to realize efficient and robust tracking based on vision information.

Frame Grabber

The video signal captured by the CCD camera must be digitalized before being processed. An on-board frame grabber: Colory 104, is used to digitalize, scale and format the analogy video signal, which is shown in Figure 2.11. RGB and YUV formats of the output video data are supported by such a frame grabber. The PC/104(-Plus) bus is used in this Colory 104, which takes full advantages of the bus bandwidth provided by the new generation processor, and can satisfy the requirement of high-speed data transfer, such as video signals.

The block diagram of the frame grabber is shown in Figure 2.12. The Colory board consists of a video interface, a processing module and a PCI-interface. The video interface receives the analog video signals of the camera. These signals will be digitized and decoded. Next, in the processing module, the digitalized video signals can be scaled, formatted to RGB etc. After the processing, the PCI interface transfers the formatted video data ('captures') to the system memory of the host computer.

For the video capture mechanism, two tasks of the frame grabber: Task A and B, are used to covert the image signal into two different formats, which is illustrated in Figure 2.13. Task A is configured to convert the video signal to the RGB format and transfer to the memory of the vision computer through PCI bus by DMA mode for the further image processing. Task B is configured to compress the video signal to the JPEG format and



Figure 2.11: Frame grabber: Colory 104

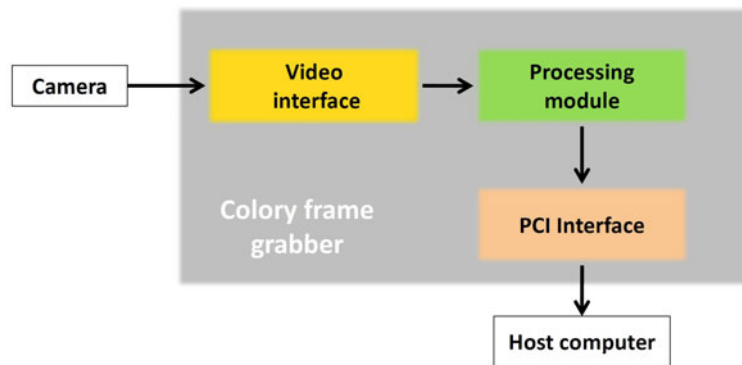


Figure 2.12: Working principle of Colory 104

save it to the compact flash memory card for off-line analysis. Four buffers are created in the memory of the vision computer, and each task uses two buffers to store the converted image data temporally. It is noted that the proposed image acquisition structure, i.e., a video camera plus a frame grabber, gives us more freedom to choose off-the-shelf cameras to fulfill the requirement of applications, since most of video cameras available in markets support analog output.

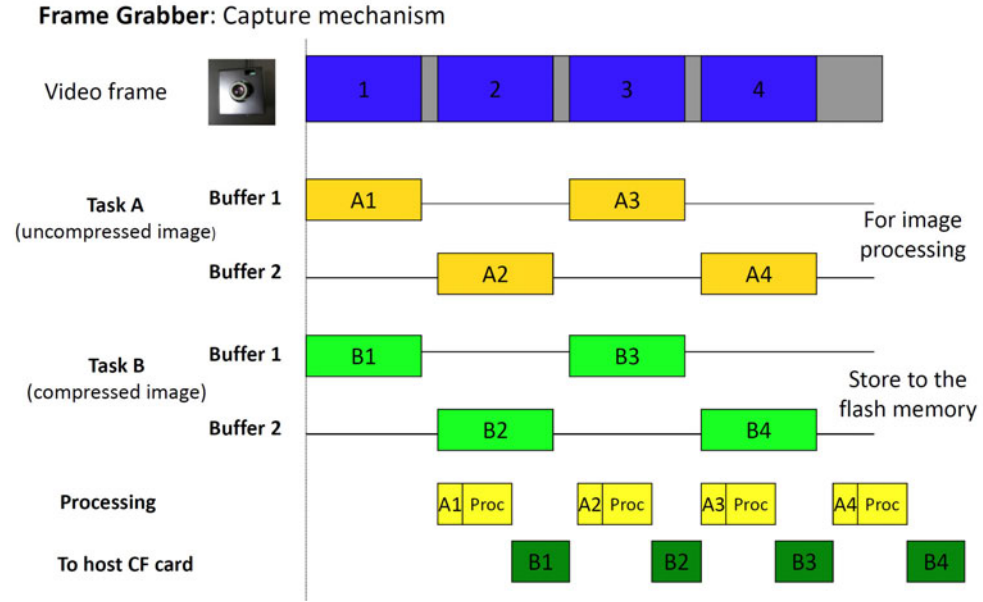


Figure 2.13: Capture mechanism of Colory 104

Vision Computer

As shown in Figure 2.8, the digitalized visual signals provided by the frame grabber is transferred to the on-board vision computer that is the key unit of the vision system. The vision computer coordinates the overall vision system, such as image processing, target tracking, and communicating with the flight control computer, which is to be described in detail later in Chapter 4 and 5. As mentioned before, in this work, the configuration of using two separated embedded computers in the on-board system for UAVs are proposed. A separated on-board PC/104(-Plus) embedded computer, Cool RoadRunner III, is employed to process the digitalized video signal and execute the vision algorithms, which is shown in Figure 2.14. The core of the board is an Intel LV Pentium-III processor running at 933 MHz. A compact flash memory card is used to save the captured images. The main specifications are listed in Table 2.4.



Figure 2.14: Vision computer: Cool RoadRunnerIII

Table 2.4: Main Specifications of PC/104-Plus Cool RoadRunner III

Processor	Intel Pentium III
Board Format	PC/104-Plus
Speed	933 MHz
Power consumption	10 watts
Cooling	Fan
Memory	512MB soldered on board
Serial ports	2 \times RS232
USB ports	2, version 1.1
Power supply	+5 VDC
Dimensions	106 \times 113 mm
Weight	145 gram

Wireless Video Transition

As shown in Figure 2.8, to provide ground operators clear visualization in monitoring the work when the on-board vision is processing during flight tests, the video captured by the on-board camera is transmitted and displayed in the ground control station.

For analogy transmission of the video signal: The RF and FM demodulation techniques are widely used in the light-weight and low-cost analog video transmission. The wireless modules are running on 1.2 or 2.4 GHz. Since most of countries implement strict regulation

on the frequency of the wireless transition, the unlicensed frequency band: 2.4GHz is used by many suppliers. Therefore, an airborne 2.4 GHz wireless video link is used to transmit the live video captured to the ground control station, which is shown in Figure 2.15. In addition, to make the vision system more robust, operators can set and reset parameters for the on-board vision algorithms through the ground control station. The information of the vision algorithm is transferred to the flight computer first, and then sent to the ground control station through the wireless modem.



Figure 2.15: Wireless video link

2.2.4 Ground Supporting System

The last essential part of the overall unmanned system is the ground station. Its main responsibility is to realize effective communications between the avionic system and the ground users and pilots. To fulfill this aim, the ground station is generally required to have capabilities such as 1) displaying and monitoring the in-flight states, 2) displaying images captured by the on-board system, 3) generating and updating flight trajectories, 4) sending control commands to the avionic system, 5) facilitating the ground piloted control or automatic control, especially in unexpected occasions such as emergency landing and cruise, and 6) logging in-flight data, to name a few. Some other features such as displaying the reconstruction of the actual flight status in a 3D virtual environment can be very helpful to the ground users when the UAV is flying out of sight (see, for example, [32]).

2.3 Systematic Integration of the On-board System

Based on the hardware components selected in Section 2.2, we now proceed to carry out a systematic integration of those components for the SheLion on-board system.

2.3.1 Computer-Aided Virtual Design Environment

In the proposed design methodology, the first step for constructing an unmanned helicopter is to choose a suitable virtual design environment. When HeLion is first instructed, we are mainly based on two-dimensional computer-aided-design (2D CAD) blueprints. The lack of a powerful 3D design environment causes great difficulty in layout design and the integration of hardware components. As a result the design and integration procedure has to be iterated for quite a number of times, which prolongs the total constructing time for months. To avoid such a problem, from the construction of SheLion, a powerful virtual design environment, SolidWorks, is adopted. Its main advantages are listed as follows:

1. Easy to use: Users can be familiar with the necessary functions in a short time through learning several key examples.
2. Powerful 3D and 2D design: In SolidWorks, the virtual counterpart can be modeled to be identical with the real hardware component, both in shape and color. When the 3D design is finished, the corresponding 2D views will be generated at the same time for the convenience of mechanical manufacturing.
3. Physical description: Each virtual component can be parameterized with necessary physical parameters such as density and weight. The center of gravity (CG) can be either calculated by SolidWorks or arbitrarily specified. Such a function is especially useful in the layout design of the on-board computer system of the UAV.
4. Animation function: For certain components, which can move or rotate, we can emulate their motions by using an animation function. This function is essential when

some complicated devices, such as a 2-DOF camera frame, are needed to be mounted on board.

Such a virtual-design-software-facilitated design concept is one of the most remarkable features of our proposed unmanned helicopter design methodology and will be closely followed throughout the design procedure.

Specifically in the following Section 2.3.2, the virtual counterpart, which reflects all of the key features including: (1) the location and dimension of its mounting hole; (2) the center of gravity; (3) the dimension of the object; and (4) the weight, will be created for each of the selected hardware components. In Section 2.3.2, each of the design steps is to be tuned virtually till it is fully determined. It is noted that the SheLion is carefully built up in the virtual design environment, which provides an excellent backup of our design process. Through using such a software-facilitated design procedure, we have successfully avoided unnecessary iterations and greatly shortened the design period.

2.3.2 Virtual Design Methodology

All the hardware components will be mounted onto a RC helicopter to achieve the autonomous flight and vision-based tasks. An infrastructure is used to house all the components. Adding such an infrastructure and on-board equipments will significantly increase the size and weight of the helicopter, as well as change the outline and gravity center of the helicopter. Such changes will affect air dynamics and performance of the unmanned helicopter. Due to the importance of the infrastructure design, many aspects have been well considered during the design.

Virtual Component Design

First, all the main components of the unmanned helicopter and the on-board system are designed in SolidWorks, including PC/104 boards, IMU, wireless modem, batteries, servo

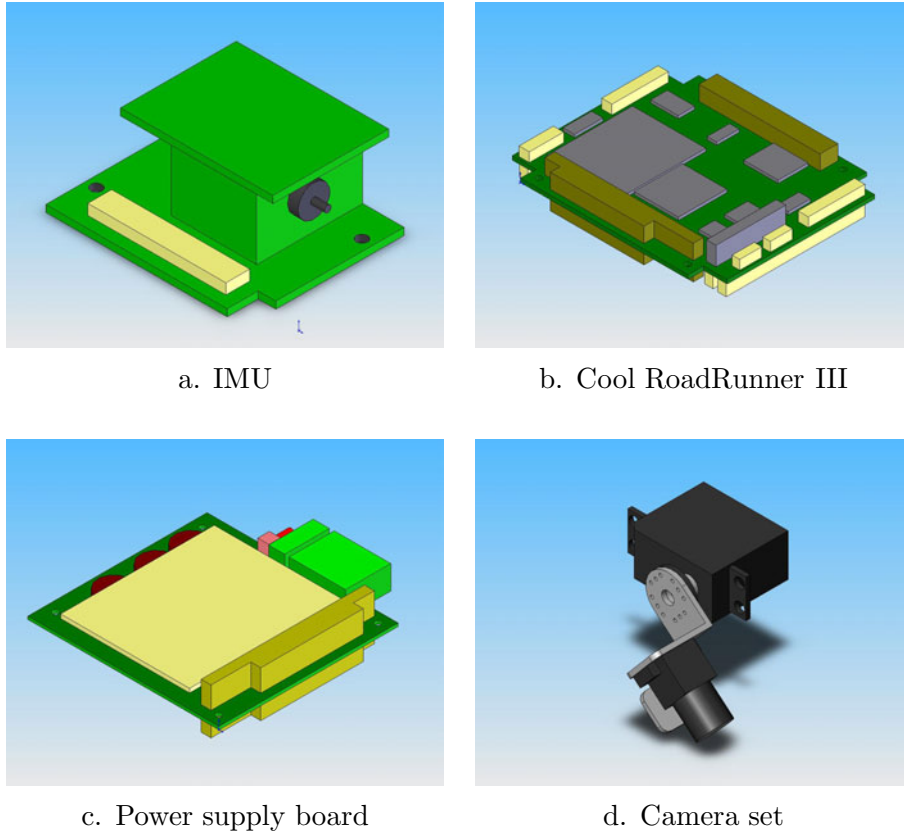


Figure 2.16: Virtual components created in SolidWorks.

controller and so on. Samples of these virtual components are shown in Figure 2.16. The hardware components and their virtual counterparts are illustrated in Figure 2.17 and 2.18. Based on these virtual components, we can assemble the on-board system. Another advantage of using SolidWorks is that these models can be reused when we design the third or fourth generation helicopters.

On-board Layout Design

Layout design for on-board computer systems is a challenging issue for small-scale UAV helicopters. There is no systematic methods reported in the literature to date. In what follows we propose a simple and uniform layout design approach, which is independent of the hardware components used and can be easily adopted to construct any small-scale UAV



Figure 2.17: Raptor 90 RC helicopter and its virtual counterpart.

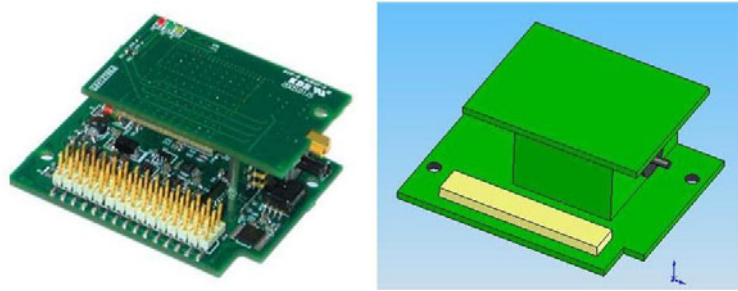


Figure 2.18: MNAV100CA and its virtual counterpart.

helicopter. This design approach includes four steps and a visual illustration for the case of SheLion is shown in Figure 2.19.

1. Step I: Determining the location of INS/GPS

The essential rule of this step is to mount the INS/GPS as close as possible to the CG of the UAV helicopter to minimize the so-called lever effect, which can cause bias on the measured accelerations when the UAV performs rotatory motions. Based on the experience we gained from the construction of our earlier version UAV, HeLion, we find that it is easier to control the UAV when the on-board system is mounted underneath the bare helicopter. For such a layout, the general guideline is to line up the CGs of the INS/GPS, the on-board computer system and the basic helicopter along the z-axis of the body frame. Since the CG location of the bare helicopter is fully known using pendulum test introduced in [51], the mounting location of the INS/GPS in x-y plane of body frame can be determined. The offset between the CG of the UAV

helicopter and that of the INS/GPS is only in z-axis and unavoidable. However, it can be minimized by carefully considering the height of on-board system and adding necessary space between the bare helicopter and the on-board system for bumping avoidance.

2. Step II: Determining the location of the camera and laser pointer

The on-board camera and laser pointer are employed for ground target tracking and attacking, their mounting locations should have a good eyesight and sufficient moving space. To fulfill these requirements, they are both mounted at the most front part of the on-board system. To simplify the design, these two components are bound with each other in parallel and attached to a digital servo, which is capable of providing motion in pitch direction. In searching or attacking a ground target, the yaw direction movement is to be controlled and accomplished by the UAV itself.

3. Step III: CG balancing

The locations of the following four components, i.e., the two PC-104 processor boards, the servo controller, the wireless modem, and the battery packs, have also to be carefully selected. In general, the PC-104 processor boards and servo controller board are to be mounted at the front part for the convenience of cable/wire connection and the wireless modem is mounted on the back for the ease of wireless communications. The battery packs are also placed on the back to balance the overall CG of the on-board system. Furthermore, we also guarantee that the CG of the on-board system coincides with the CG of the INS/GPS, and the on-board system is symmetrical in both longitudinal and lateral directions. According to the size and weight of all on-board electrical components, we roughly divide the on-board system into three parts: front, median and rear part. The weights of the main components are listed in Table 2.5.

4. Step IV: Locating the remaining light-weight components

The remaining light-weight (less than 50 g) components include ultrasonic sonar and

toggle panel, for which anti-pollution and short circuit avoidance are the main consideration. At the end, we decide to place the sonar on the landing skid and the toggle panel along with the plastic cover opposite the muffle of the helicopter.

In summary, the assembled on-board system and the infrastructure in SolidWorks and in reality are shown, respectively, in Figure 2.20 and 2.21 with different views.

2.3.3 Anti-Vibration Design

For any small-scale UAV upgraded from RC helicopter, there are three main vibration sources which should be taken into account: (1) the rotation of the main rotor; (2) the power engine; and (3) the rotation of the tail rotor. These frequencies for SheLion are 30.8 Hz, 260.5 Hz and 143.4 Hz, respectively, based on a governed motor speed at 1850 rpm. The combined vibration has an amplitude about 2g, i.e., 19.6 m/s^2 , along all of the three body axes of SheLion. It has potential to introduce bias to measurement data and to cause loose-of-connection of mechanical components. For this reason, an anti-vibration design is necessary to ensure the overall on-board system working properly.

For SheLion, four wire-rope isolators are carefully selected to realize an anti-vibration aim. They are mounted symmetrically around the CG of the on-board system (see Figure 2.22), and their working features are as follows:

1. 45°-Compression mounting: Such mounting method provides the same stiffness in both the horizontal and vertical directions.
2. Good transmitting rate: The transmitting rate is defined as the ratio of the output vibration level to the input vibration level. According to the selection rules provided by the manufacturer (see Figure 2.23 for the characteristic of the wire-rope isolators), we choose a natural frequency and a cutoff frequency around 9.5 and 13.4 Hz, respectively, which are sufficient to ensure that a satisfactory transmitting rate for the vibration

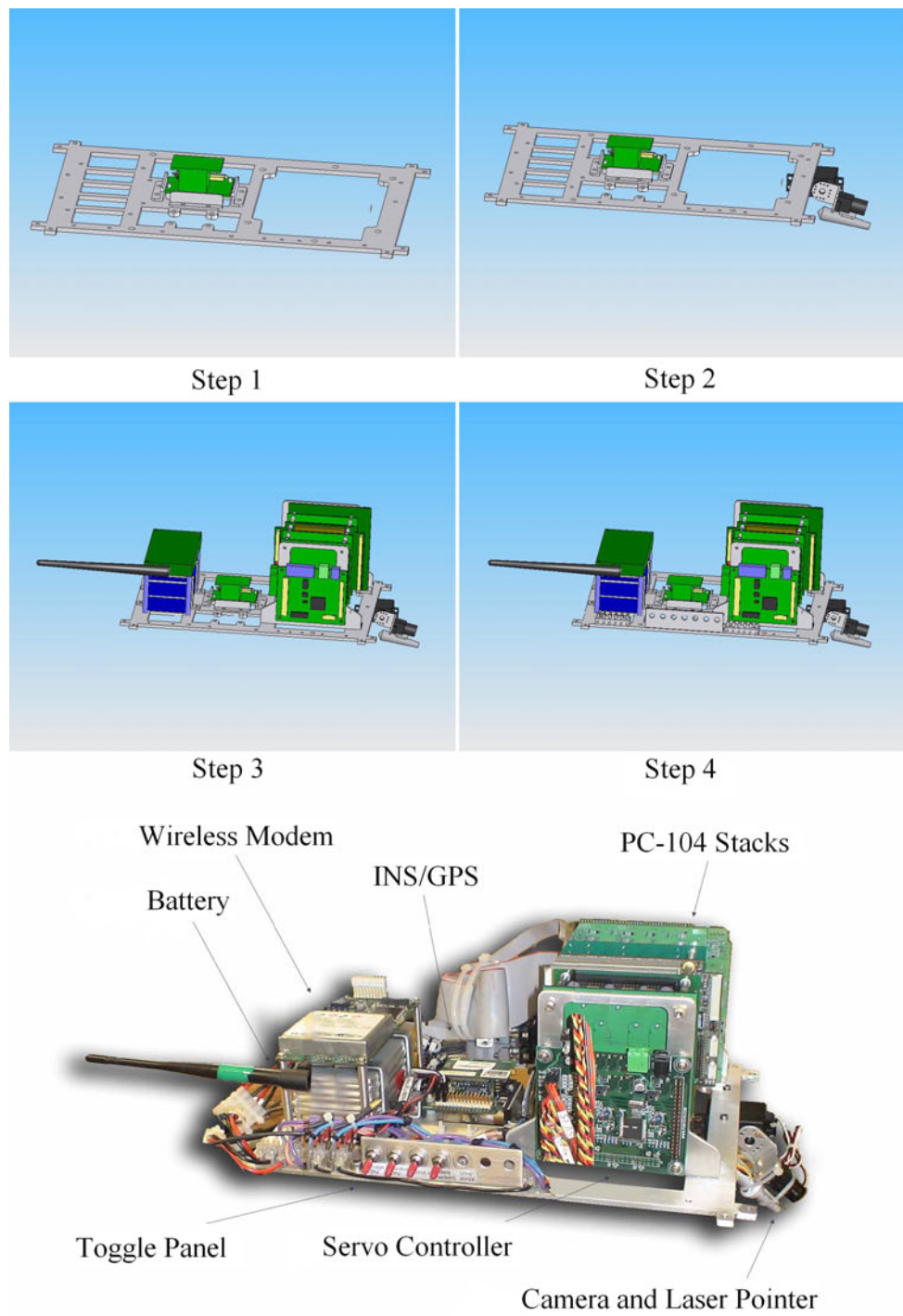
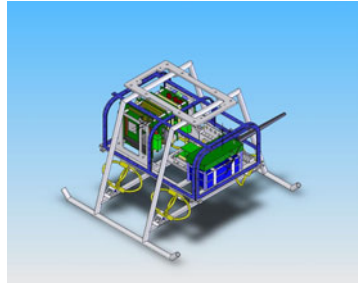


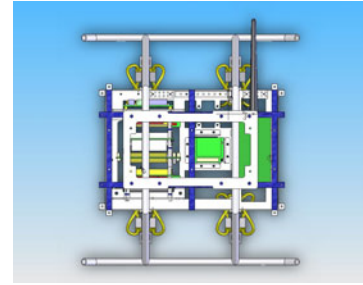
Figure 2.19: Layout design procedure and the final on-board system.

Index	Item	Weight(g)
	Front part	
1	On-board camera and pan/tilt servo	85
2	Control computer (Diamond Systems: Athena)	270
3	Vision computer (Lippert: Cool Road Runner)	470
4	Servo controller (PONTECH: HBC101)	65
5	Bracket for two CPU units	120
	Median part	
6	IMU(Crossbow: MNAV100)	55
	Rear part	
7	Battery pack (Thunder power: 3800mAh 4s2p 14.8v)	550
8	Wireless modem(FreeWave: 2.4 GHz Radios)	200
	Others	
9	Accessories (screws)	200
10	Wire, switch, connector and so on	300
11	Bottom plate	320
12	Main cover	60
13	Side cover(front)	20
14	Side cover(rear)	20
15	Plate beneath the helicopter	130
16	Uclip×4	36
	Landing ski and isolators	
17	Isolators (Enidine: wire rope)	260
18	45 deg mount×4	80
19	Landing ski	460
	Total weight	3151

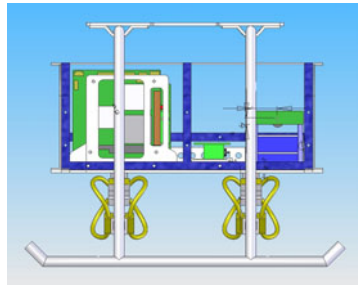
Table 2.5: Weight list of on-board hardware components



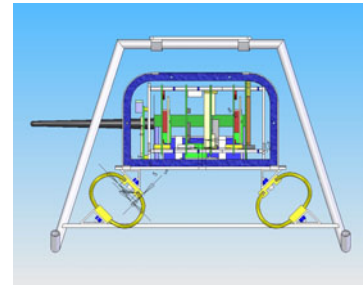
Isometric



Top



Front



Left

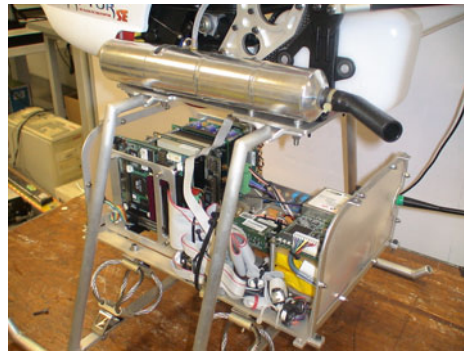
Figure 2.20: 3D views of the infrastructure and on-board system in SolidWorks.

source with the lowest frequency, i.e., 30.8 Hz. More specifically, about 80% of the vibration at this frequency is suppressed.

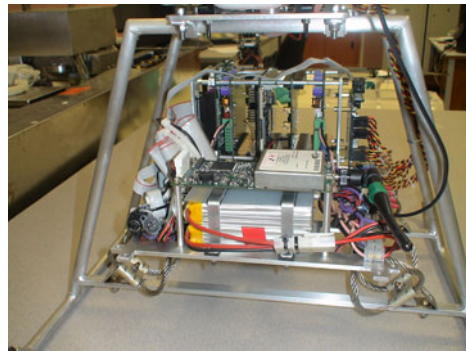
Such an anti-vibration design has demonstrated to effectively reduce the harmful raw vibration and increase the overall safety. Its actual performance is to be further examined in Section 2.4.

Power Supply Design

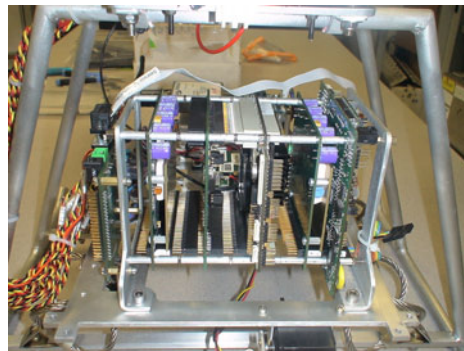
The main consideration in the power supply design is to meet the overall experimental requirement and overall system safety. Based on the detailed power consumptions of the hardware components on-board given in Table 2.6 and the consideration of safety issues, we



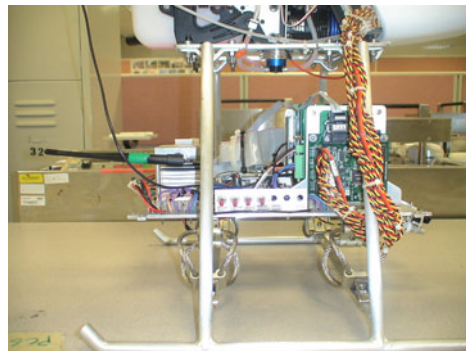
Isometric



Rear

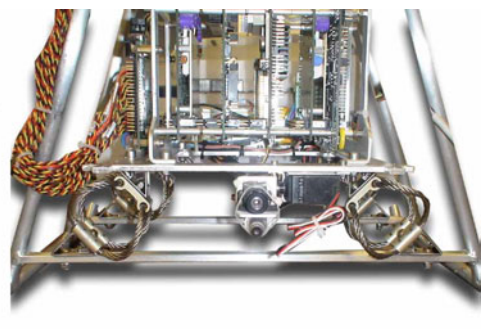
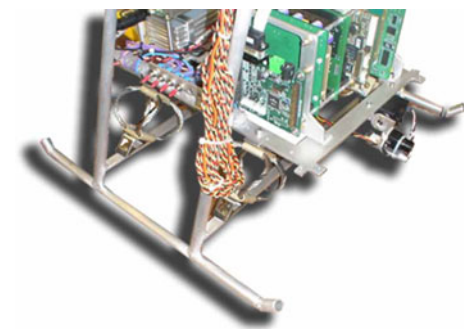


Front



Right

Figure 2.21: 3D views of the infrastructure and on-board system in physical world.

Figure 2.22: Anti-vibration design for the on-board system.
(left: side view, right: front view)

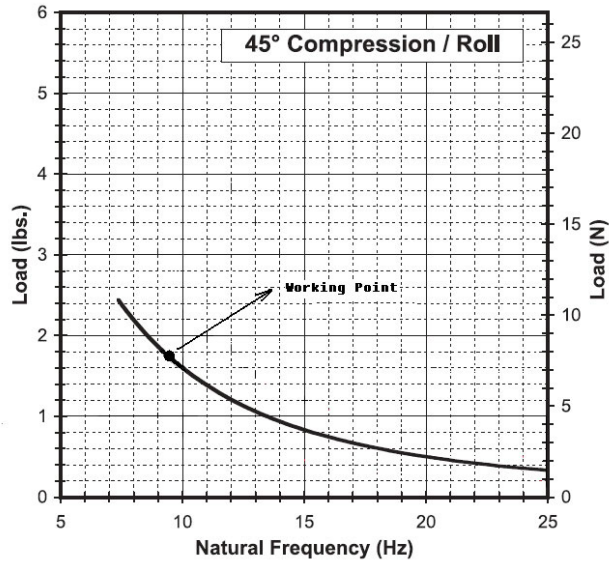


Figure 2.23: Working point of the selected wire rope isolators.

come out with a power supply scheme for SheLion, which is shown in Table 2.6, in which batteries 1 and 2 with an output voltage of 4.8 V, a power capacity of 17.5 Wh and a weight of 90 g, are used to power the on-board servos and servo controllers. Although a single battery is sufficient to power the components on board, two batteries have been chosen instead to enhance the overall safety of the system. The system can still run smoothly and guarantee manual maneuvering even if one of the batteries is out of order. Another feature of our design is to include the servo controller with the RC helicopter as its function is extremely important for both manual and automatic flight. With such a configuration, the servo controller can still work during unexpected events, such as the breakdown of the on-board system, so that the ground pilot would still have chances to guide the helicopter.

To avoid the potential conflict of the power supply between the flight control CPU and the image processing CPU, two separate batteries (batteries 3 and 4) are used to provide power supply to these two units. To accommodate for the different input voltage levels of each individual hardware components, two high-efficiency DC-to-DC convertor boards with a transferring rate of 92% are used to convert the output voltages of batteries 3 and 4-5

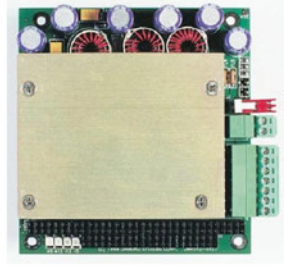


Figure 2.24: DC-to-DC convertor boards: JUPITER-MM

Table 2.6: Power consumption list for SheLion

Hardware Component	Power Consumption (W)
Flight Control Computer	12.5 (at 5V DC)
INS/GPS	0.5 (at 5V DC)
Servo Controller	1 (at 5V DC)
Wireless Modem	3.9 (at 12V DC)
Ultrasonic Sonar	1 (at 12V DC)
Image Processing Computer	19.5 (at 5V DC)
Frame Grabber	0.5 (at 5V DC)
CMOS Camera	0.6 (at 12V DC)

and 12 V, respectively, which are shown in Figure 2.24. It can be observed from Table 2.6 that the total power consumptions of the flight control unit and the vision processing unit are quite similar. We thus select two identical batteries with an output voltage of 8.4 V, a power capacity of 35 Wh and a weight of 190 g, for batteries 3 and 4.

EMI Shielding Design

Electromagnetic interference (EMI) is a serious issue for small-scaled UAV helicopters as all of the highly integrated electronic components are required to be mounted in a very limited

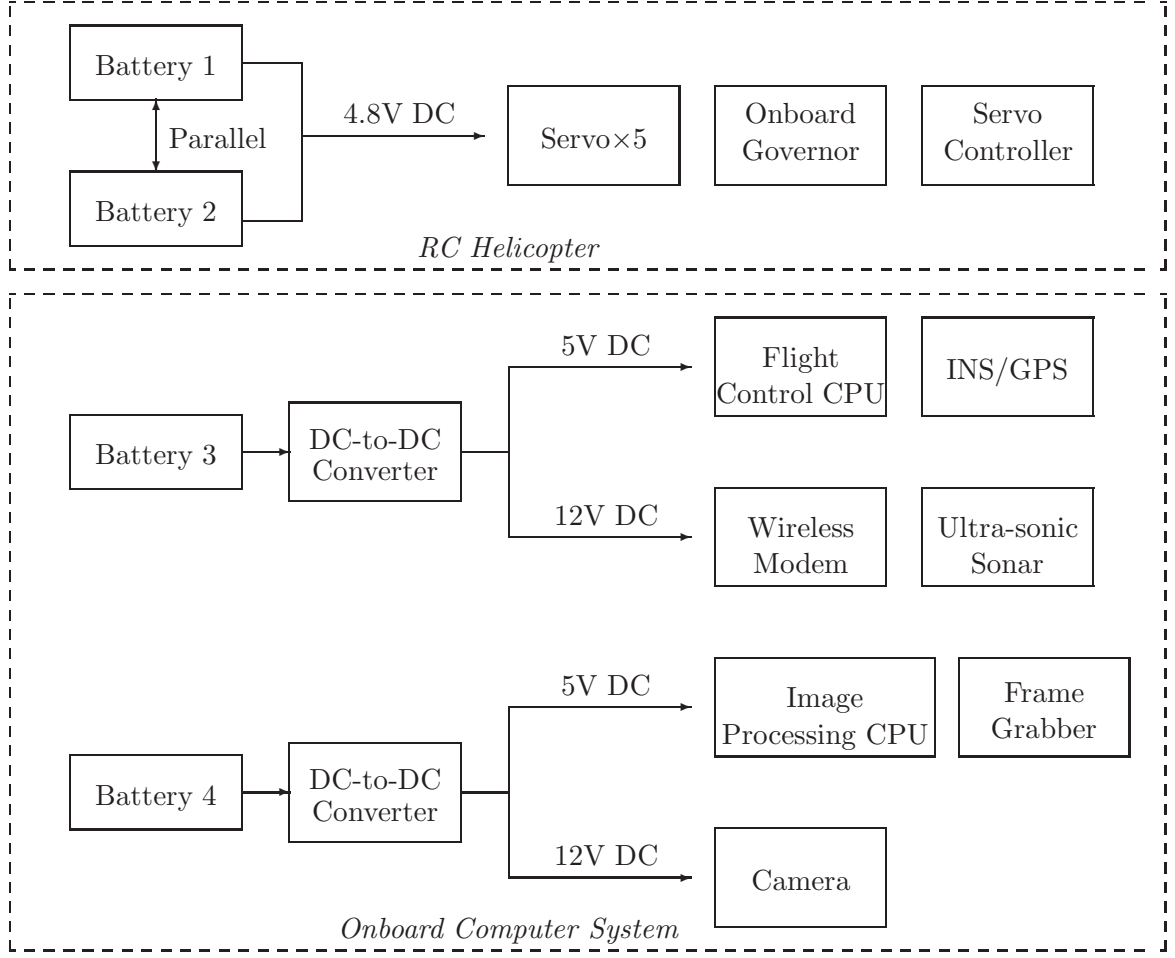


Figure 2.25: Power supply design for SheLion unmanned helicopter.

space. The main problems aroused by EMI include: (1) reducing the effective range of RC manual control; (2) generating errors in INS/GPS measurements; and (3) causing data losses in wireless communications. These problems have to be eliminated or reduced to minimum before conducting actual flight tests. In SheLion, we use aluminum boxes and foil to isolate the necessary electronic components. More specifically, the key hardware components such as the servo controller board, RC receiver, MNAV100CA and wireless modem are kept in separate aluminum boxes, and the on-board system is protected with aluminum foil. As a result, we have successfully maintained the original manual control range (50 m without extending the antenna of the joystick), and the reliability of the MNAV100CA and wireless

modem.

2.4 Ground Test Evaluation

The last step of our proposed design methodology is conducting a series of ground tests and actual flight tests to evaluate the performance and reliability of the overall unmanned helicopter. For SheLion, the performed ground and flight tests are introduced as follows.

During the ground tests, SheLion is placed on a level ground with its engine running at 85% of the hovering RPM, which is set to be 1850 for SheLion. The ground supporting system is placed about 500 m away from SheLion. Each ground test lasts more than 12 min. More specifically, the following items are thoroughly examined:

1. Flight control computer.

For the flight control computer, we run the on-board software system of [31] to execute iteratively all of the tasks listed in Section 2.2.2. We set the execution time for each iteration loop to be 20 ms, which coincides with the sampling rate of the INS/GPS. Figure 2.26 shows the actual CPU execution time of all the loops tested. Clearly, the actual time consumption of each loop is in the neighborhood of 20 ms. The bias is mainly caused by the inaccuracy of the internal clock of the PC-104 ATHENA processor.

2. Wireless communication system.

The wireless communication system between SheLion and the ground supporting system is tested through transmitting some pre-set data. Our test shows that the communication channel between the ground station and the UAV is perfect.

3. Power consumption.

For this item, a special ground test which lasts 50 min is performed. The input voltages for both the flight control unit and the image processing unit are recorded periodically

with a time interval of 5 min. The resulting output voltages of the batteries are plotted in Figure 2.27. As expected, the output voltages of both units drop but with the reasonable slopes. The final values stay, respectively, at 7.72 V for the flight control part and 7.31 V for the image processing part, which are within the safety level for the overall system. This result indicates that the selected batteries have sufficient power to continuously supply the overall on-board system during the whole experimental period.

4. Anti-vibration system.

To examine the efficiency of the anti-vibration system, two small-size vibration detecting sensors are used, of which one (vibration sensor 1) is stucked on a lever of the landing skit and the other (vibration sensor 2) is attached underneath the aluminum plate of the on-board system. Figure 2.28 shows a test sample of the z-axis acceleration measured by the two sensors and the measured acceleration data of the INS/GPS. With the wire-rope isolators, the resulting vibration transmitting rate is in the range of 20-25%, which indicates that our anti-vibration design is successful. Similar results are also obtained for the other two axes. We note that the remaining 20-25% vibration can be further eliminated through using the Bessel filters.

After these ground tests, SheLion is ready for flight tests. In addition, the virtual and real SheLion in air are shown in Figure 2.29.

2.5 Conclusion

In this chapter, a complete and systematic design procedure is proposed for the hardware configuration of the vision-based unmanned helicopter. The design procedure includes a hardware selection method, a design and integration scheme, and a virtual design procedure. Test results show that the proposed methodology is efficient and effective. In order to realize autonomous flight and vision applications, software development is investigated in the next chapter based on the proposed hardware configuration,

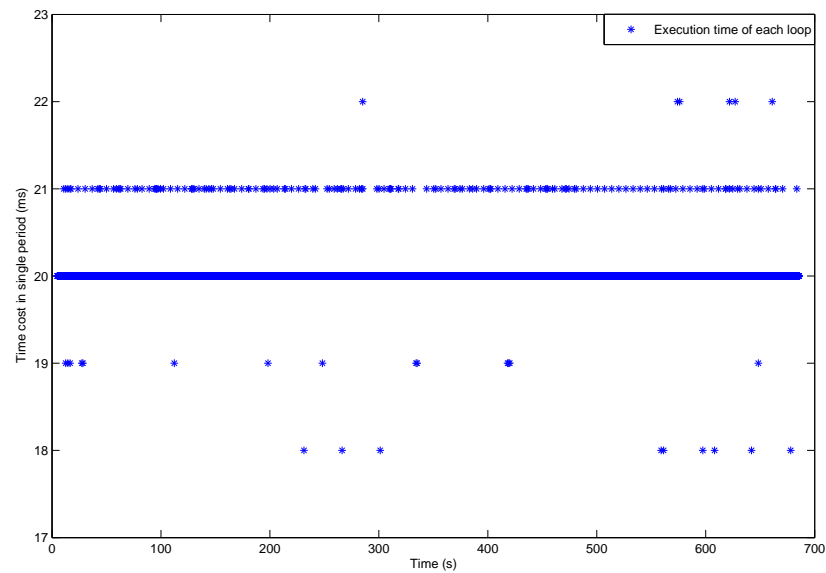


Figure 2.26: Execution time of the test loops of Flight Control CPU.

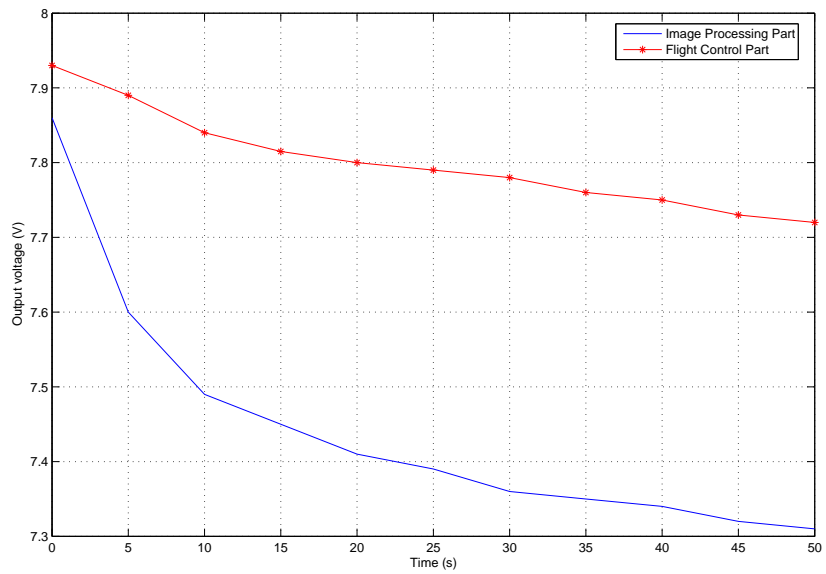


Figure 2.27: Output voltages of Lithium-Polymer batteries.

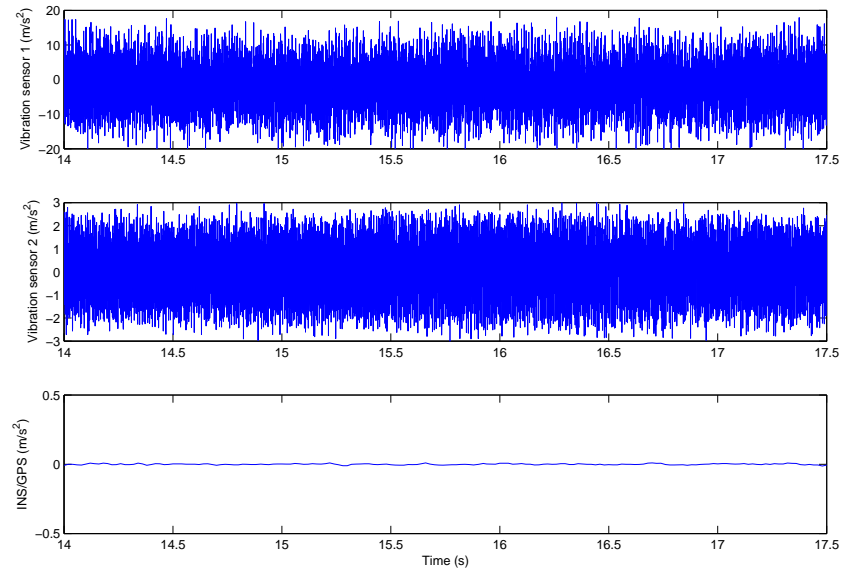


Figure 2.28: Sample result of comparison of vibrational amplitude.



Virtual



Real

Figure 2.29: Virtual and real unmanned helicopter: Shelion in flight.

Chapter 3

Software System Design and Implementation

3.1 Introduction

A sophisticated software system is required to ensure all of the hardware components for a UAV system to work properly and effectively, as well as to ensure good communications and coordinations between the onboard system and the ground station [17].

In this work, the on-board processing tasks are required to be executed strictly and precisely in every execution cycle. As such, the development of avionic software system are dominantly carried out in a real-time operating system (RTOS) environment, which can effectively guarantee the final system executed in a deterministic behavior, based on certain scheduling, intertask communications, resource sharing, interrupt handling, and memory allocation algorithms [139]. Currently, three most popular real-time operating systems adopted in the UAV development are the QNX Neutrino [137], VxWorks [143], and RTLinux [140].

Since the on-board software system targets for real-time applications and runs in an embedded PC/104(-Plus) single board computer, QNX Neutrino [137], a real-time embedded operating system, is employed as the developing platform. QNX Neutrino has a microkernel

that requires fewer system resources, and performs reliably and efficiently for embedded systems during runtime compared to the traditional monolithic kernel.

The entire software system can naturally be divided into three main parts for the vision-based unmanned helicopter: 1) onboard control software, 2) onboard vision software, and 3) ground station software. Their functions will be presented in the following sections.

3.2 Flight Control Software

The purpose of on-board flight control software is to coordinate all the hardware components on-board in an appropriate sequence [17]. For most of the UAV systems, the essential tasks onboard include:

1. Navigation data collection,
2. Flight control algorithm execution, and
3. Servo actuation driving.

Shown in Figure 3.1 is a general framework of the on-board flight control system adopted in [31], which employs a multi-thread structure and consists of several blocks with each of them being designed for a specific device and task. More specially,

1. IMU is a block interacting with the navigation sensors and collecting necessary measurement data;
2. DAQ is to read additional information from the peripheral sensors;
3. CTL is to implement the automatic flight control laws;
4. SVO is to drive the servo actuators;
5. CMM is to communicate between the avionic system and the ground station through the wireless links;

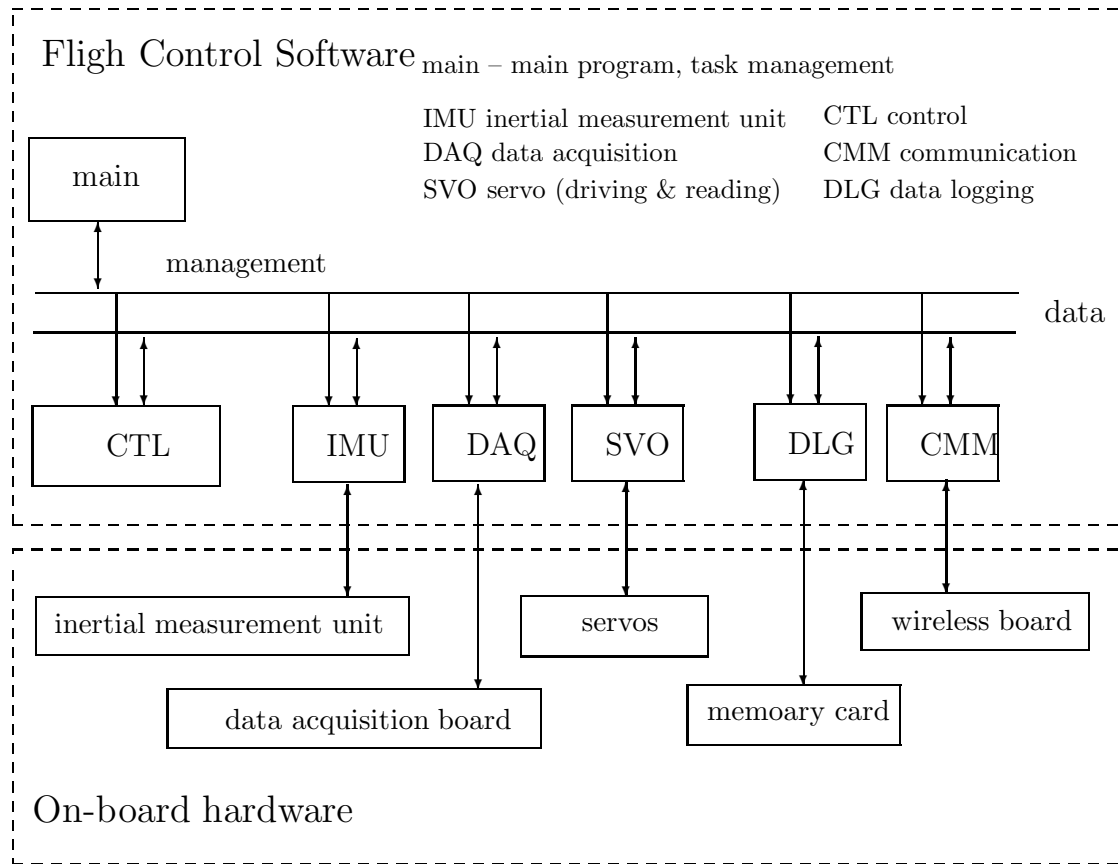


Figure 3.1: Framework of the flight control software

6. DLG is for data logging, which is usually designed as a background task saving all necessary in-flight data.
7. Finally, the main block is to manage all tasks.

3.3 Vision Software

In addition to the on-board flight control software system, an on-board vision software system also plays an important role to coordinate the tasks of on-board vision system. The essential tasks include:

1. Capturing video signals,
2. Executing vision algorithms,
3. Controlling pan/tilt servo mechanism, and
4. Communicating with flight control system and the ground supporting system.

As presented in Section 3.2, QNX Neutrino [137], a real-time embedded operating system, is employed as the developing platform. The configuration of the vision software system is briefly described in following subsections, including two main parts: 1) the architecture of the vision software system, and 2) task management.

3.3.1 Framework of Vision Software

The main purpose of the vision software is to coordinate multiple tasks mentioned before. The general framework of the vision software is illustrated in Figure 3.2. Such framework employs a multi-thread structure and consists of several blocks with each of them being designed for a specific device and task. All the blocks share the same global memory to transfer data. The main advantages of such structure are to make software system clear and easy to be developed under the multi-thread framework. The functions of these tasks are presented as follows:

1. CAM Block : The main purpose of CAM block is to read the image data captured by the frame grabber. Before reading data, the configuration of the frame grabber is set up according to the requirements of the project. The adopted frame grabber can support a sustained frame rate up to 31 frame per second (FPS) for uncompressed RGB 24-bits image with full resolution (720×576). Such capability is sufficient for typical analog video inputs in the real-time operation. In fact, the real capturing rate is limited by the complex of the vision algorithms and other tasks involved in writing

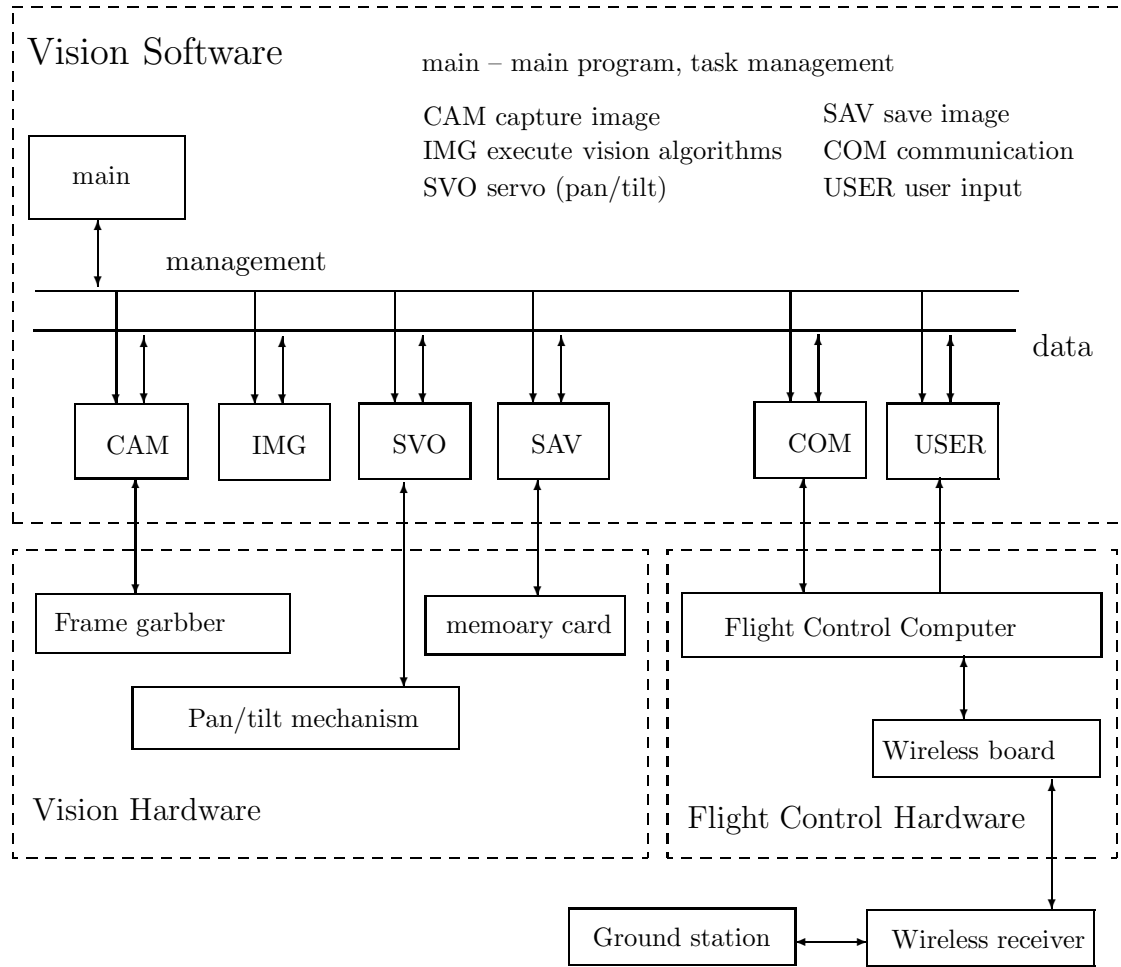


Figure 3.2: Framework of the vision software

operations, such as writing processed images to a solid-disk. In the project, the frame rate of the vision system is set to 10 frames per second.

Two image data buffers are used to store the captured images by the frame grabber alternatively in order to reduce the risk of damaging the image data. Moreover, the caching function is enabled to realize the fast access of the mapped memory regions for the image data, which can increase the image data transfer in the memory. The transmission of the image data from the buffers of the frame grabber to the shared

memory of the vision program will take about $2 \sim 3$ ms. The constant time delay of the image data is caused by capturing frame, which is around 40 ms.

2. **IMG Block** : The main vision processing algorithms are implemented in IMG block, which is a critical part in the vision software. The main functions of the IMG block include processing the image data, and making the decision in terms of the information from multiple sensors. The image data is obtained from shared memories, which is transferred from CAM block. The specified algorithm will be executed in this block, such as image processing, object segmentation, feature extraction, pattern recognition, and automatic tracking and camera control. The processed image will be saved to global shared memory for further saving. The decision will be made in terms of the vision information and outputs of other sensor, then the action will be taken to control the camera servo and send the command to the flight control computer. Since the IMG block is pure calculation, the time consumption of this block only depends on the complex of the vision algorithms used.
3. **SVO Block** : The SVO block is utilized to control the rotation of the pan/tilt servo mechanism to keep the interest target in a certain location of the image, typically the center of the image. The control input of the pan/tilt servo mechanism is calculated according to the vision information of the target and the state of the helicopter. The servo control is realized through a RS232 serial port with the baud rate of 9600.
4. **SAV Block** : The main function of SAV block is to save the captured or processed images to a high-speed compact flash. A compact flash (CF) card is employed as the storage media for the onboard system instead of a hard disk. That is because of hard disk's size and vulnerability against vibration which is unavoidable on the helicopter. A 4GB high speed compact flash is used, which has the maximum speed 45 MB/s. However, the bandwidth of direct motherboard connection of the compact flash is often limited to 33 MB/sec, since the IDE (Integrated Drive Electronics) to CF

adapters cannot support high speed ATA (66 MB/sec plus) cable support. We then can estimate the required saving time of a frame.

$$t = \frac{s_{\text{im}}}{r_{\text{tf}}}$$

where

$$s_{\text{im}} = W \times H \times d$$

s_{im} is the size of the image, and r_{tf} is the transfer rate of the IDE bus; W and H are the width and height of a frame; d is the depth of the image;

In our project, a frame size of 360×288 (WxH) at a color depth of 24bits is employed, the EIDE computer bus is employed with a throughput of 33 MB/s. The estimated transfer time is:

$$\begin{aligned} s_{\text{im}} &= 360 \times 288 \times 24 = 2.48832 \text{ Mbits} = 0.31104 \text{ Mbytes} \\ t &= 0.31104/33 = 0.009425 \text{ sec} = 9.424\text{ms} \end{aligned}$$

Thus, the time of saving a frame to the compact flash will take about 10 ms, which already reach the limit assigned for the saving block. So if saving larger images or multiple processed images was required, the compressed format of the images will be considered.

5. COM Block : The main function of COM block is to communicate with the flight control computer. The flight control computer sends the states of the unmanned helicopter to the vision computer, and the vision computer sends the obtained vision information to the flight control computer to guide the flight of the unmanned helicopter. The flight control computer also sends commands of the operator from the ground control station to the vision computer.

6. **USER Block** : The USER block provides a mean for users to control the vision program such as running and stopping the tracking as well as changing the parameters of the vision algorithms. That block can facilitate the program debugging.
7. **MAIN Block** : Managing and scheduling the work of the entire vision software system.

3.3.2 Task Management

A scheme of multiple threads is used to implement these blocks in the image processing unit based on the approach proposed in [31]. Each block is implemented in a thread, named task thread. The main thread will activate every thread periodically in a specified rate by a timer. In every period, these task threads are scheduled to run in a designed order through a mechanism of synchronization. This mechanism prevents the main thread and other threads from blocking each other. This is the advantage of using the scheme of multiple threads. Task scheduling is performed in the main thread. At every period, the main thread sends pulse messages to all task threads one by one. The pulse message plays a role of activating signal for the task thread. It wakes up the task thread to perform work.

A timer is designed in the program to manage the execution of the main program. The timer sends a pulse signal at every specified period to active the main task processing. The main program has a waiting and processing loop. In this loop, the main program keeps idle until it receives the pulse signals from the timer. Once the pulse signal is received, the user command is read and executed, and the following task threads are executed. The loop will be terminated if an exit command is given. In each cycle, the main program will send activate pulses to every task threads in a specified order. After all task threads are executed, the main program returns to the loop and waits for the next pulse signal from the timer.

The executions of task threads are carried out in this way. Every task thread has a waiting and processing loop, which waits for a pulse signals from the main program. Once the pulse signals is received, all subsequent steps scheduled in the thread are executed. The

loop is terminated when an exit pulse is sent from the main thread. After the thread is successfully processed, a notification signal will be returned to the main program. Then the main program processes the next task thread. The waiting scheme for each task thread sets up the thread exclusive occupation of the CPU resource and prevents the main program from terminating the thread when it is running.

The tasks of the image software are shown in Figure 3.3. The execution structure is like a lotus. Each single node stands for a task thread. The center part stands for the main thread. Solid arrows in the figure stand for the task thread activation or notification of accomplishment and indicate the exchanges of the processes. Dotted round arrows in the center of the task nodes denote the direction of processing. Along the processing direction of main thread, task threads are activated one after another.

Time Consumption Measurement

In each cycle, a thread needs a period of time to execute the specified task. The time consumption of each thread is measured in tests and shown in Table 3.4. From the results shown in Table 3.4, we can observe that the vision algorithms costs more than 80% of the computational resource, and the complexity of the vision algorithms severely affects the real-time performance of the vision system.

3.3.3 Computer Vision Library

To realize the proposed vision algorithms, existing computer vision libraries are employed in the research project. In this section, the Open Source Computer Vision (OpenCV) library [136] is introduced. OpenCV is an open source library of programming functions which is free for both academic and commercial use. With minor modifications, it can be installed in other platforms, such as QNX. Since the OpenCV library is developed mainly for the real-time processing, it is very suitable for our embedded vision applications.

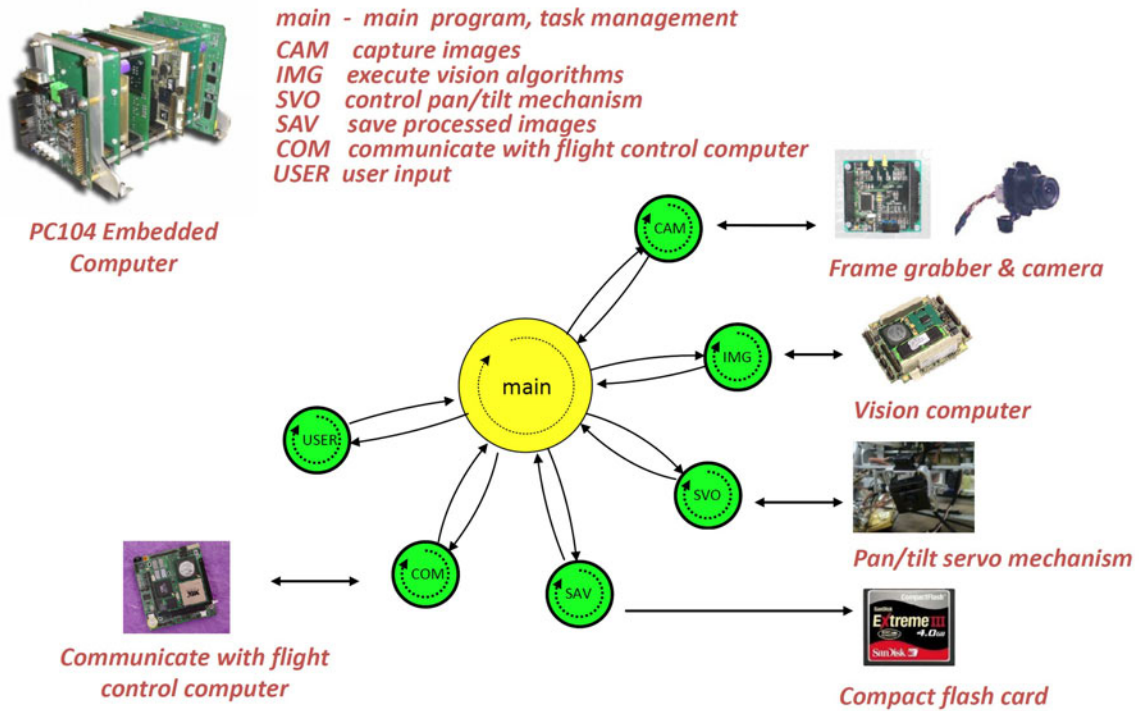


Figure 3.3: Task management of the vision software

Although OpenCV library is not a commercial vision library, it is still very powerful for the machine and robot vision applications. OpenCV library provides more than five hundreds of optimized and commonly used algorithms, ranging from the image processing, pattern recognition, computer vision, to linear algebra algorithms. With minor modifications, the most of algorithm functions of the OpenCV library 1.0, are installed on the embedded QNX operating system. The test results of the commonly used vision algorithms in a desktop and the embedded single board computers are shown in Table 3.1. An image with the size of 256×256 pixels is used in the tests.

In addition, specially thanks to the contribution of the researchers from all of the world, the OpenCV version 2 was released on October 2009, which includes more new vision algo-

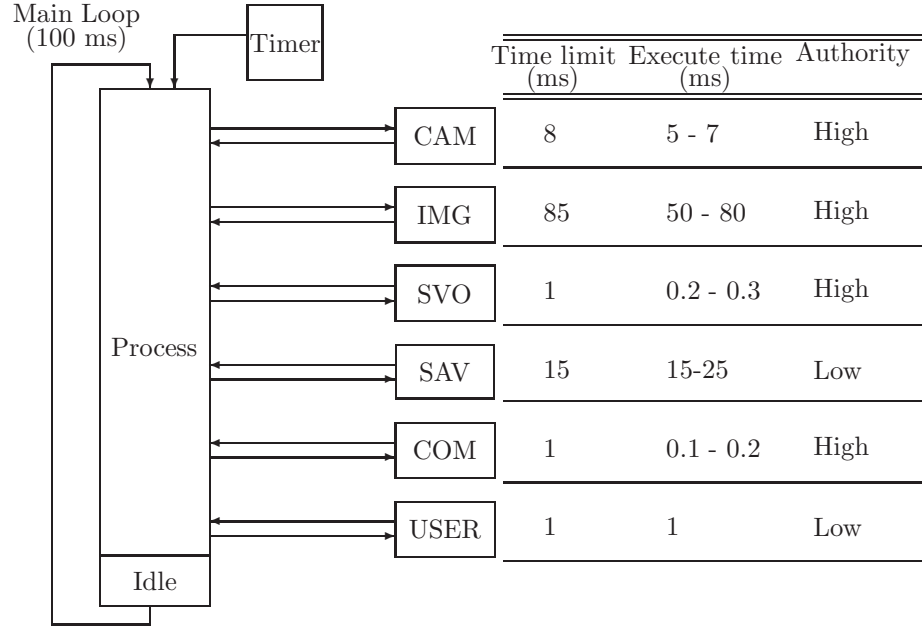


Figure 3.4: Execution of multiple tasks of the vision software.

Table 3.1: Test results of OpenCV functions

	Laptop (Intel 2.8G)	PC/104 CRR III (Intel 933MHz)	Gumstix (OMAP3530 720MHz)
Operating system	Window 7	QNX	QNX
Canny edge detector:	7 ms	12.29 ms	15.89 ms
Laplace edge detector	7 ms	9.74 ms	118.16 ms
Harris corner detector	12 ms	17.57 ms	164.88 ms
Median filter	5 ms	7.87 ms	15.61 ms

rithms, functions and implementations. The version 2 will be employed in future research.

3.4 Ground Station Software

Compared with the on-board flight control software, the real-time feature for the ground station software system is preferable but not strictly compulsory. As such, many ground

station software systems, particularly for the scientific research and commercial purposes, are not developed under an RTOS environment. Instead, other powerful programming environments with rich interface capacities, such as Windows-based Visual C++ [142], are commonly adopted. Shown in Figure 3.5 is a framework of the ground station software system employed in [31]. Generally, the ground station software system consists of two layers, i.e., the background and foreground layers. Data transferring usually runs in the background layer, through the wireless channel with the avionic system (i.e., the CMM task thread), receiving data from and sending commands to the avionic system. In the dual-layer configuration, two separate threads are required for receiving and sending data. The receiver thread keeps reading the port connected to the wireless device, whereas the sender thread keeps waiting for commands issued by the ground users and writing it to the port once a command is captured. The foreground interface is capable of displaying windows for showing in-flight data and for issuing flight commands, which is illustrated in Figure 3.6.

3.5 Implementation of the Automatic Control

In this section, based on the proposed software system, we proceed to describe the design and implementation of the flight controller for the unmanned helicopter: Shelion. Unlike HeLion, a twin brother of SheLion, the latter is equipped with an onboard camera and image processing system. But they have the similar shape and structure, and the controller design approach for HeLion can be applied to SheLion too. The procedure of the controller design follow the design of a twin brother of HeLion is described in [13, 15].

3.5.1 Dynamic Modeling and System Identification of the UAV

Before designing the controller, it is crucial to obtain a fairly comprehensive model of a UAV if one wishes to design an advanced automatic flight control system by incorporating multi-variable control techniques such as LQR and H_∞ control, and nonlinear control. Modeling

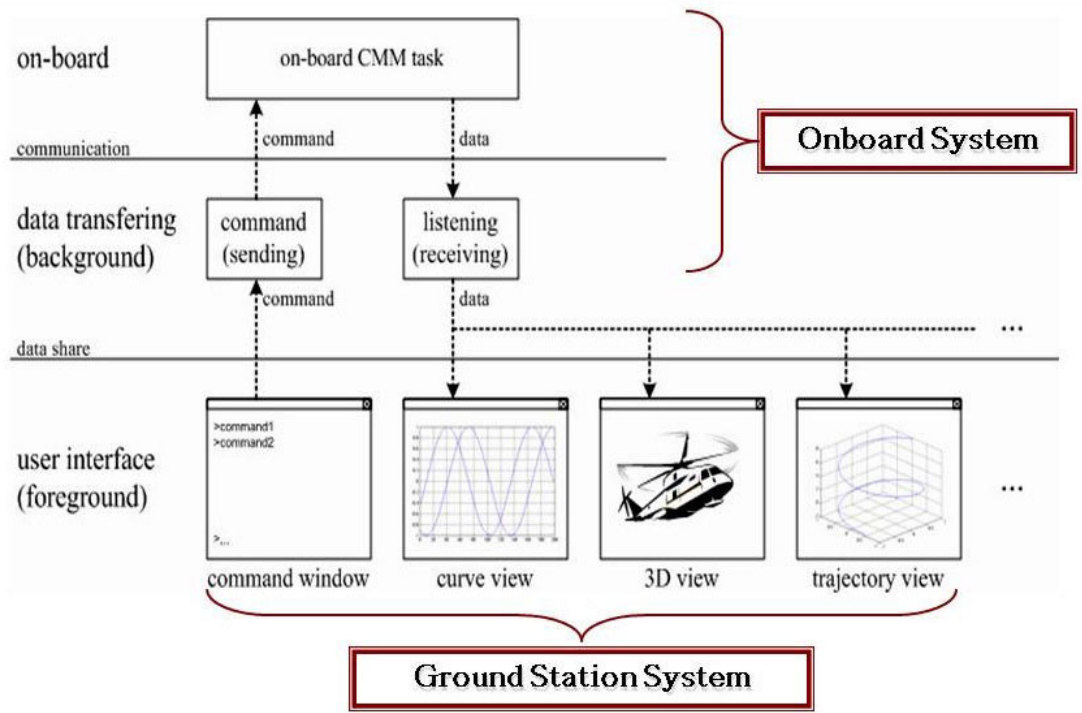


Figure 3.5: Framework of ground station software

of a mini rotorcraft UAV, especially for its full flight envelope dynamics, is an extremely challenging task. Due to the nature and physical structure of the rotorcraft UAV, dynamic modeling with in-flight data and with parameter identification approach has been proven to be an ideal choice to derive a fairly accurate model of the flying vehicle.

Based on the modeling work we have done in [13,15], we would like to conclude this section by noting that a fairly comprehensive and accurate nonlinear model has been obtained in [15] for HeLion, a Raptor 90 based UAV helicopter [138], which can also be used for SheLion with minor changing of parameters. Through the combination of both the first-principles approach and the system and parameter identification method, the result has been successfully used in designing an advanced flight control system.

In order to formulate the inner-loop controller design into the framework of state space

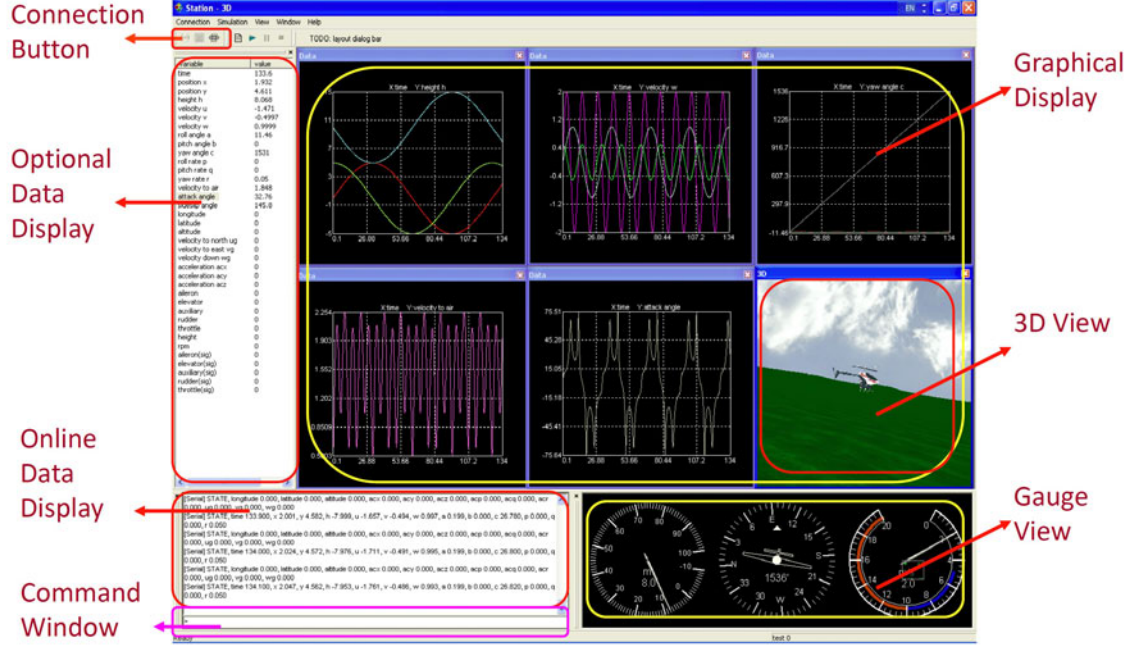


Figure 3.6: User interface of ground station software

representation, we proceed to linearize the obtained model of SheLion at the hover and near hover flight condition [14], which can be expressed as

$$\begin{pmatrix} \dot{\mathbf{P}}_n \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} B_B & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{V}_b \\ r \end{pmatrix} \quad (3.1)$$

and

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3.2)$$

$$\mathbf{A} = \begin{bmatrix} -0.03621 & 0 & 0 & 0 & 0 & -9.7807 & -9.7807 & 0 & 0 & 0 & 0 \\ 0 & -0.04386 & 0 & 0 & 9.7807 & 0 & 0 & 9.7807 & 0 & 0 & 0 \\ -1.0577 & -1.2651 & 0 & 0 & 0 & 0 & 74 & 401 & 0 & 0 & 0 \\ 0.5121 & -0.3908 & 0 & 0 & 0 & 0 & 257.7 & -39.48 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -5.357 & 2.456 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -0.2141 & -5.357 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.733 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.309 & -3.652 & 94.89 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.9928 & -7.304 \end{bmatrix} \quad (3.3)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.095 & 2.523 & 0 & 0 \\ 1.932 & 0.097 & 0 & 0 \\ 0 & 0 & 12.19 & 0 \\ 0 & 0 & 9.655 & -94.89 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.4)$$

where $\mathbf{x} = (u \ v \ p \ q \ \phi \ \theta \ a_s \ b_s \ w \ r \ r_{fb})^T$; $\mathbf{u} = (\delta_{\text{lat}} \ \delta_{\text{lon}} \ \delta_{\text{col}} \ \delta_{\text{ped}})^T$; $\mathbf{V}_b = (u \ v \ w)^T$; B_B denotes velocity transformation matrix from body frame to NED frame. The physical meanings of the states and input variables are explained in Table 3.2.

3.5.2 Automatic Flight Control System

The classical single-input/single-output (SISO) feedback control method (i.e., PD or PID control) is one of the most common choices because its simplicity in structure with less requirement on the accuracy on the dynamical model of the UAV. Examples include the CMU-R50 UAV helicopter [82], in which a SISO PD control law is adopted and further optimized using CONDUIT for both hovering and forward flight, and the Ursa Major 3 UAV helicopter [112], in which a SISO PID control is implemented for automatic hovering. To improve flight control performance, many researchers have devoted to the study of implementing more advanced control techniques on mini rotorcraft UAVs. For example, a flight control system using a MIMO H_∞ control approach has been designed and implemented for their mini rotorcraft UAVs in [124]. It is reported that the resulting system has clearly

Table 3.2: Physical meanings of the state and input variables.

Variable	Physical meaning	Unit	Measurability
u	Velocity vector along body-frame x-axis	m/s	Yes
v	Velocity vector along body-frame y-axis	m/s	Yes
p	Roll angular rate	rad/s	Yes
q	Pitch angular rate	rad/s	Yes
ϕ	Roll angle	rad/s	Yes
θ	Pitch angle	rad/s	Yes
a_s	Longitudinal tip-path-plane (TPP) flapping angle	rad	No
b_s	Lateral TPP flapping angle	rad	No
w	Velocity vector along body-frame z-axis	m/s	yes
r	Yaw angular rate	rad/s	yes
r_{fb}	Yaw rate gyro feedback	NA	No
δ_{lat}	Normalized aileron servo input (-1 ~ 1)	NA	Yes
δ_{lon}	Normalized elevator servo input (-1 ~ 1)	NA	Yes
δ_{col}	Normalized collective pitch servo input (-1 ~ 1)	NA	Yes
δ_{ped}	Normalized rudder servo input (-1 ~ 1)	NA	Yes

outperformed the classical method. Other cases reported in the literature include systems designed by using: (i) decentralized decoupled model predictive approach [113], (ii) neural network method [133], [38], [123], (iii) adaptive control technique [90], (iv) fuzzy logic approach [64], (v) μ -synthesis [125], (vi) approximate linearization method [69], (vii) non-linear feed-forward method [12], (viii) differential geometry technique [61], (ix) H_∞ static output-feedback control [43], (x) learning control technique [39], and (xi) intelligent control methods [141], to name a few. Although there are vast of works that have been done along the line, many of them, however, are still in the simulation stage. They are far being ready for actual implementation onto the real platform.

The automatic flight control system is essential for a UAV to carry out flight missions with minimal or even without interference from human pilots. Recently, Peng et al. [86] and Cai et al. [14] have proposed a flight control scheme consisting of three parts, namely, the inner-loop control law, outer-loop and flight scheduling, which is depicted in Figure 3.7.

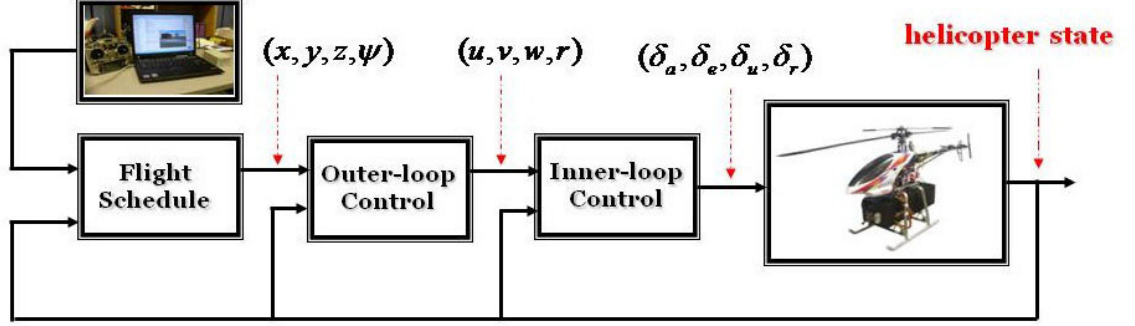


Figure 3.7: Framework of the autonomous flight control law

In [14], the function of the inner-loop control law, designed using the H_∞ control approach, is to guarantee the asymptotic stability of the aircraft motion with respect to the surrounding air and to have good disturbance rejection with respect to wind gusts. The role of the outer-loop is to produce flight commands or references to the inner-loop control layer, and finally the task of the flight scheduling part is to generate the flight references for pre-scheduled flight missions. It is showed that the closed inner-loop, equivalently the open outer-loop, is decoupled under the H_∞ controller. The similar approach proposed in [14] is employed to design the inner- and outer-loop controller of SheLion. The simulation results of the flight controller is shown in Figure 3.8.

3.5.3 Flight Tests

After successfully completing the ground tests in Chapter 2, we next move on to test the overall UAV system in the sky under both the manual control mode and automatic control mode with the developed softwares. For manual mode, we have conducted a series of perturbation tests. More specially, we first command SheLion to be stabilized at a hovering flight condition and then inject a frequency-sweep signal to the input channels to produce

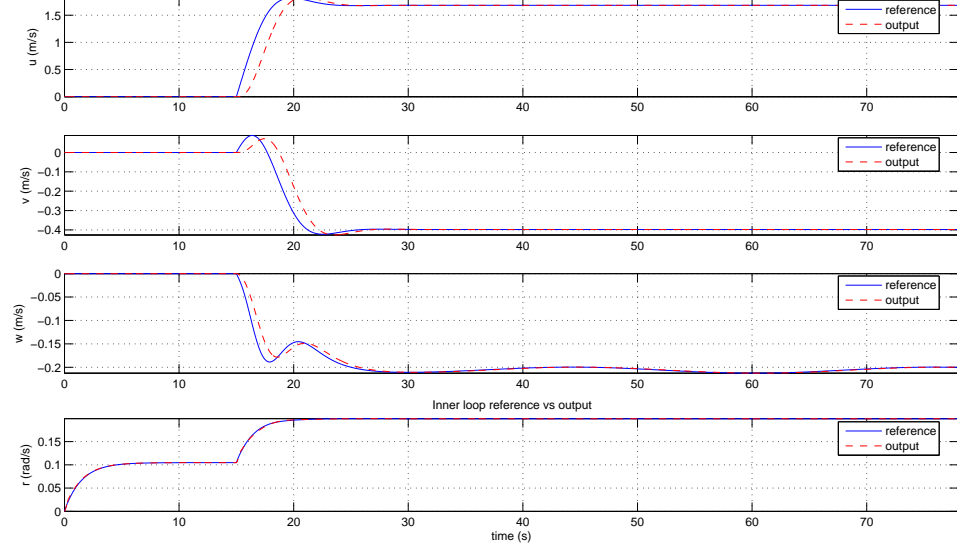


Figure 3.8: Simulation results of the autonomous flight control
(Velocity in hovering condition)

perturbations up to 30° in rolling, pitching and yawing angles, respectively. The main aim for this kind of tests is to evaluate the performance and feasibility of the UAV hardware components in drastic flight actions. Figure 3.9 to 3.12 show the resulting manual flight test results. It is proved that the constructed SheLion has been working properly in such a severe flight condition.

Automatic hovering flight is used to test the automatic mode of the integrated UAV system. After manual hovering is achieved, SheLion is commanded to switch to the automatic mode and its onboard system takes over the control authority to continue performing the hovering flight test. The results shown in Figure 3.13 to 3.17 are obtained from an automatic hovering test, which clearly indicate that SheLion is capable of hovering stably around the desired position $(-19.5, 30, 14.5)$ m without drifting. The constructed UAV helicopter can thus be further utilized for other developments. In the flight test, we have also activated

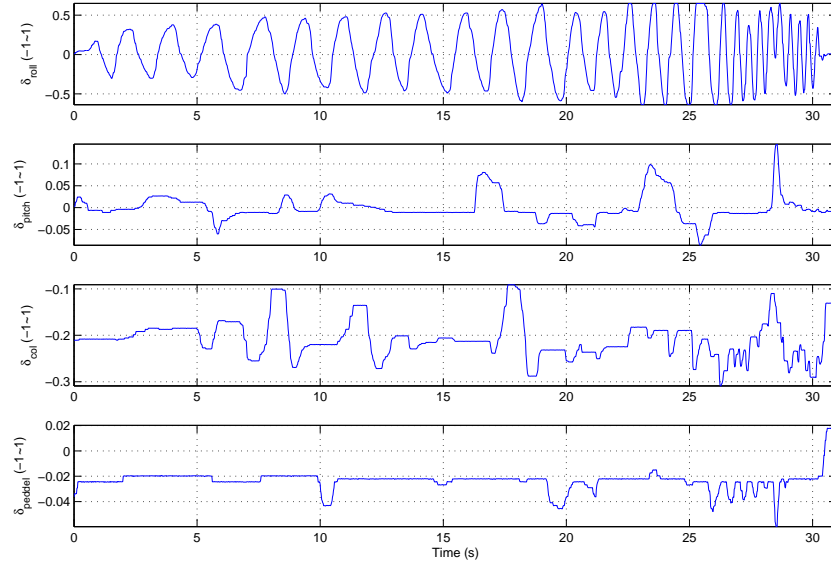


Figure 3.9: Input signals in the manual flight test.

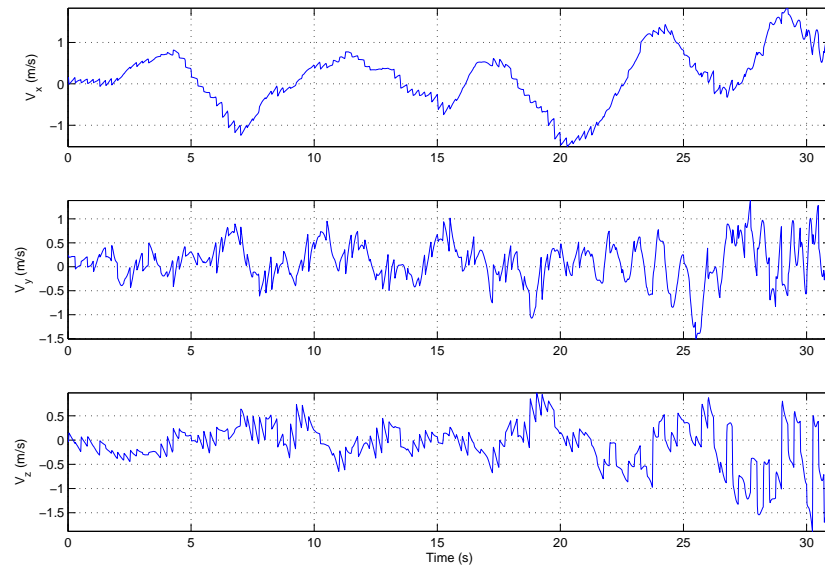


Figure 3.10: Velocity outputs in the manual flight test.

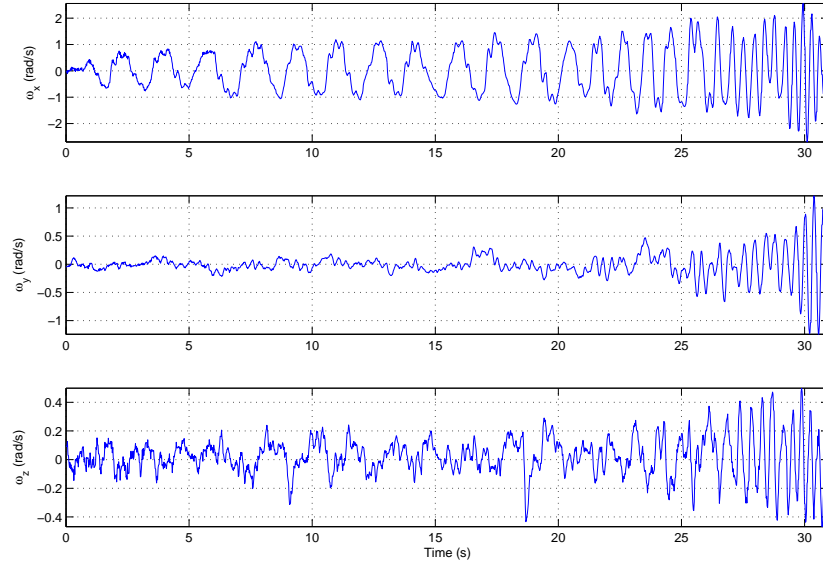


Figure 3.11: Angular rates in the manual flight test.

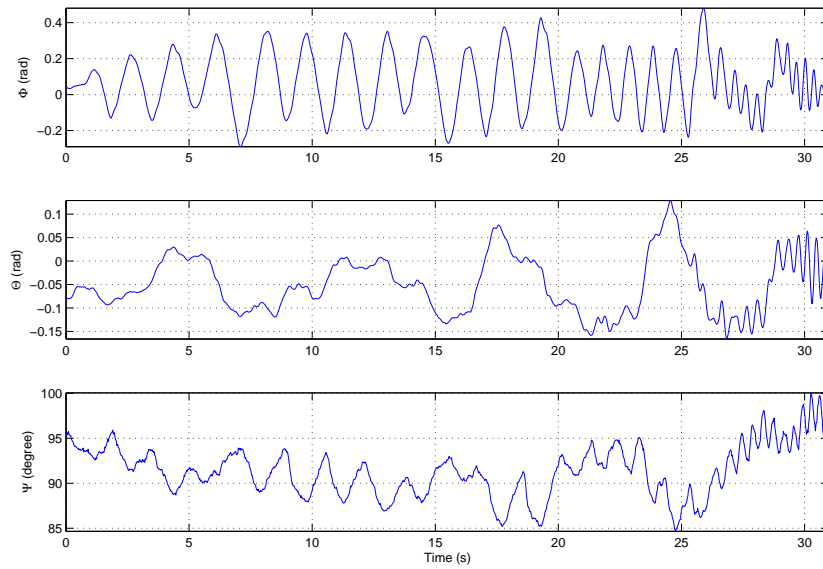


Figure 3.12: Euler angles in the manual flight test.

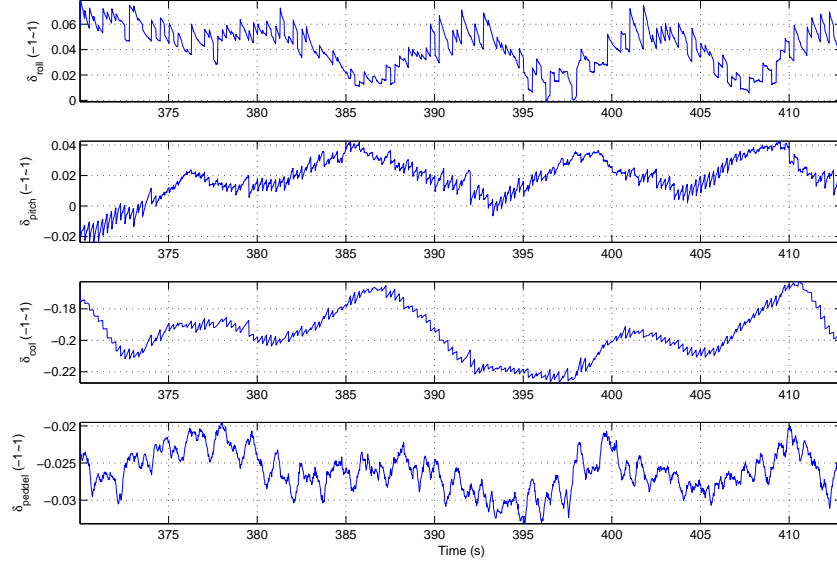


Figure 3.13: Input signals in the automatic hovering flight test.

the image processing unit and commanded the system to capture ground images when the UAV is hovering steadily.

In addition, more advanced flight tests were also performed. The test results of SheLion in the spiral up flight with 2 m/s are presented in Figure 3.18, which are used to show the capability of autonomous flight.

To test the performance of the vision system, Figure 3.19 shows a pair of images captured during this process. Our post-flight examination on the mechanical components of the UAV and the data obtained clearly indicates that SheLion is very reliable in all categories tested.

3.6 Conclusion

In this chapter, the software systems for the vision-based unmanned helicopter has been presented in detail, including the on-board flight control and vision softwares, as well as the

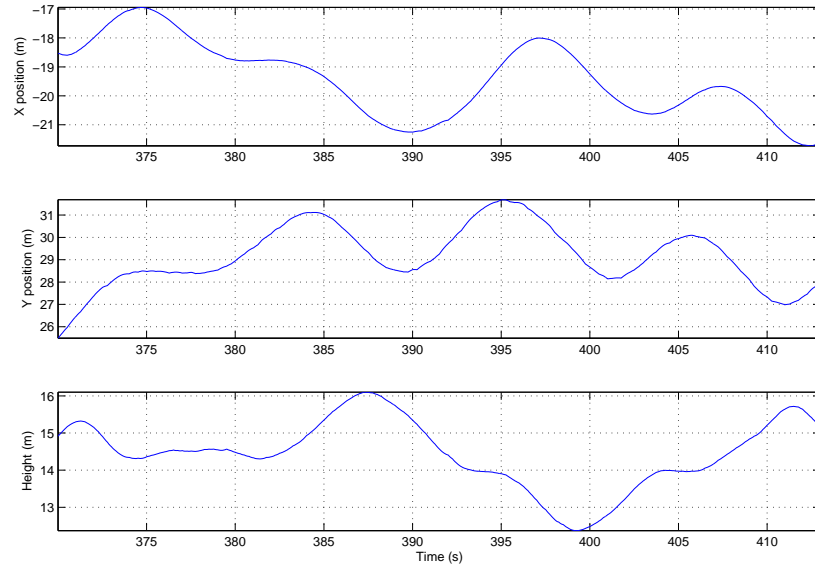


Figure 3.14: Position outputs in the automatic hovering flight test.

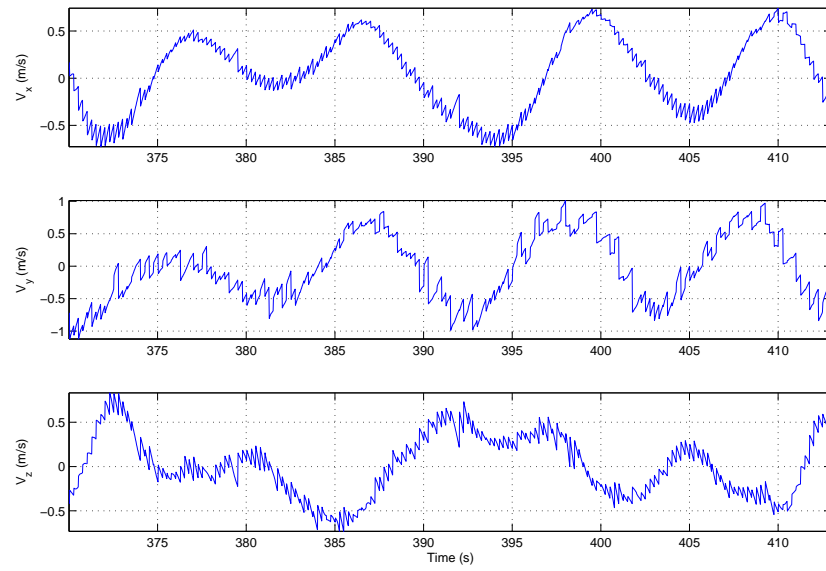


Figure 3.15: Velocity outputs in the automatic hovering flight test.

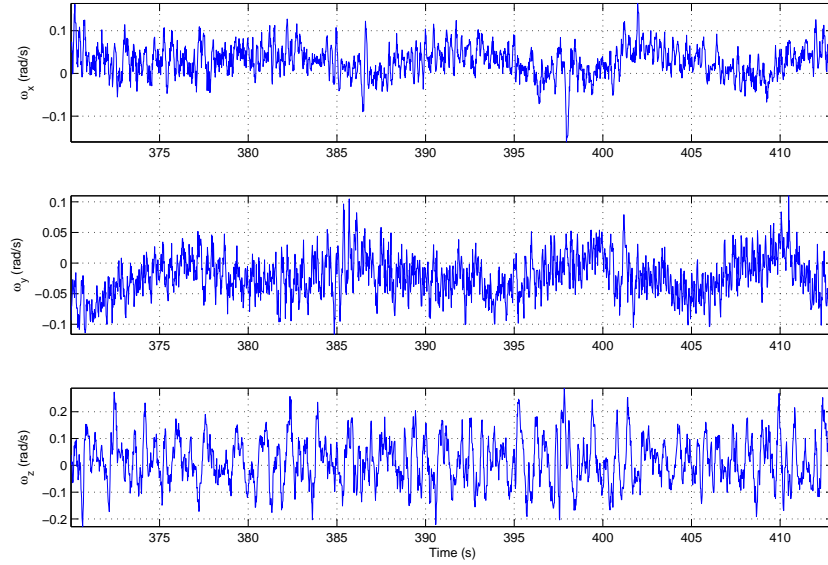


Figure 3.16: Angular rates in the automatic hovering flight test.

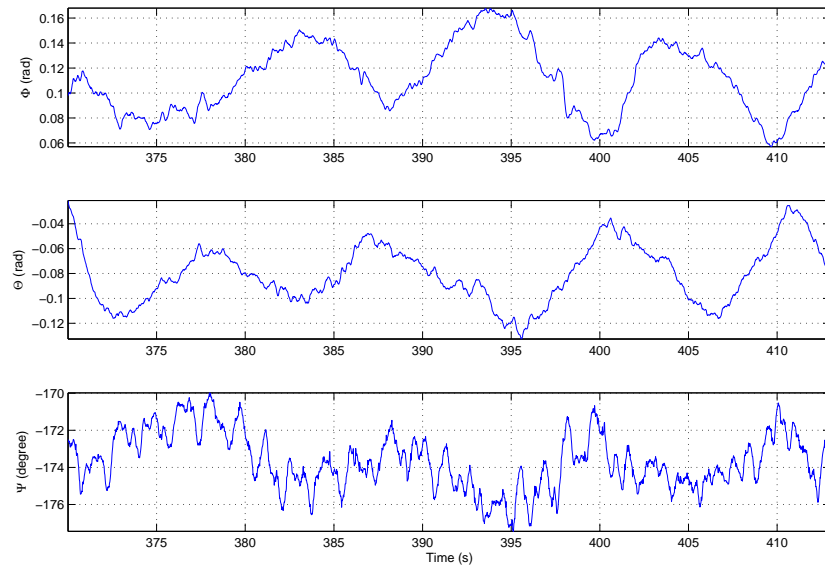


Figure 3.17: Euler angles in the automatic hovering flight test.

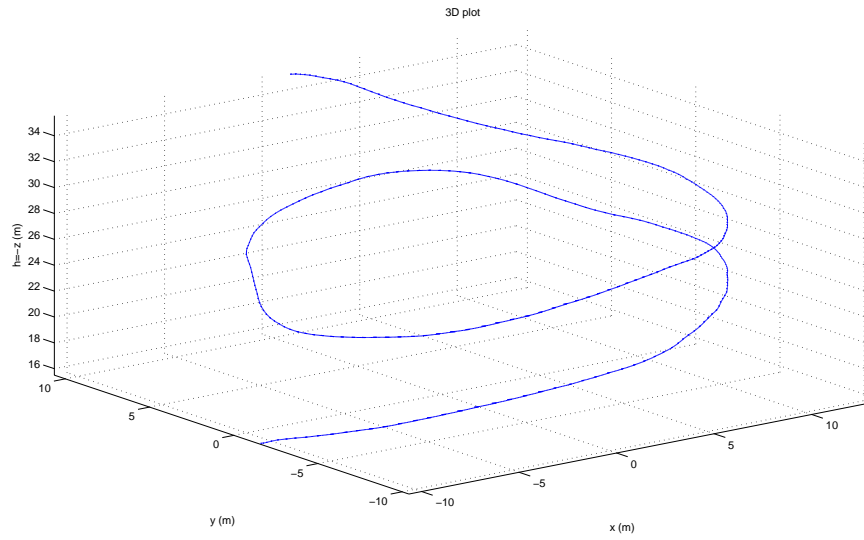


Figure 3.18: Flight results of the autonomous flight control.
(Spiral up with 2 m/s)

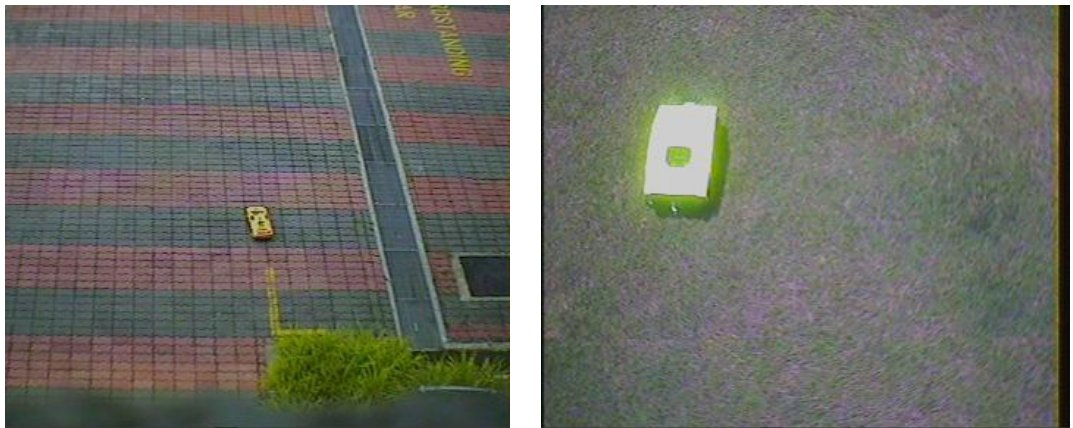


Figure 3.19: Samples of ground images captured by SheLion.

ground station software. As for the onboard part, the multi-thread-based framework and the time allocation scheme have been applied to both the flight control and vision softwares. The automatic flight schemes have been designed and implemented in the on-board control software. Moreover, vision software have been developed to coordinate the necessary tasks. Any required vision algorithms for different applications can be implemented in this vision software system efficiently. The ground station software has been developed based on a two-layer (i.e., data transferring in background and information visualization in foreground) framework. The proposed software systems have been thoroughly verified through actual flight tests. The software systems presented in this chapter can be regarded as a baseline which can be easily ported to other vision-based unmanned aerial or ground vehicles with minimal modifications.

Chapter 4

Vision-Based Ground Target Following

In order to explore capabilities of the proposed vision-based unmanned helicopter, this chapter presents a ground target following application using the proposed visual sensing techniques. To realize autonomous vision sensing, a sophisticated vision-based target detection method and tracking scheme is proposed, which employs robust feature descriptors and efficient image tracking techniques. Based on the vision sensing and navigation sensors, the relative distance to the target is estimated. Such vision feedback is integrated with the flight control system to guide the helicopter to follow the ground target in flight.

4.1 Introduction

Autonomous target following based on vision sensing is one of the most important capabilities for unmanned vehicles and robots in various applications, including search and rescue, vision-based landing, surveillance, vehicle monitoring, sensing and navigation, and so on. The key point of visual sensing is the detection and tracking of objects or features in the image.

To be specific, target detection can be considered as a pattern recognition problem. A straightforward approach is to identify appearance, color and shape signatures of an object

captured in image by comparing it with templates in a large library that includes all potential views of the target or its components, in different sizes and orientations. A more sophisticated method is to develop a composite template filter, consisting of the principal components of all potential templates [97]. For instance, the “visual odometer” is addressed in [3], based on template matching method, which requires special digital signal processors to perform correlation calculation. In [80], template matching is proposed to search a small-size patch defined by the operators in a region of interest in successive frames. However, these template matching based approaches remain sensitive to scaling, rotation and translation, and perform poorly against occlusion and changing backgrounds as a result of target movement. Moreover, they involve repeated evaluation of image correlations and are therefore computationally costly.

Another popular target detection method used in most surveillance system is the background subtraction method, which separates the foreground objects from the background using a template of the latter [58]. The background template is updated, during the processing of each image frame, using a certain number of preceding frames. The accuracy of object detection by background subtraction can be affected by camera movement, which is usually assumed to be stationary or moving at a low and constant speed [47]. Otherwise, egomotion estimation is required to construct the background image in mobile camera applications [120]. However, using motion segmentation methods to detect the moving targets is not suitable for a UAV platform due to the strong coupling of target and the on-board camera movements.

An increasingly popular approach for automatic detection and tracking is the feature-based approach. This involves segmentation of the object based on its texture, color and appearance, and extraction of object features by some high-level abstraction, yielding certain descriptors of the object, which are more or less independent of the size, orientation and other distortions of the object in the image [97], and tracking of the object by matching its features [58]. For instance, a vision-based autonomous helicopter is reported in [109] to

land onto a well defined target, which is used to simplify the corner detection based vision algorithm. The detected corners were used to estimate the relative pose of the aerial vehicle relative to the ground target. In [80], two feature-based tracking algorithms were proposed in conjunction with a Kalman filter to estimate the velocity of the helicopter relative to the target, where the features used were polyline description of contours and optical flow. The estimated velocity was used as a feedback signal for helicopter-position control in order to track the target.

In fact, the target detection methods mentioned above are normally used to initialize image tracking in many vision applications. An effective combination of detection and motion tracking is able to speed up the processing, as well as deal with moving targets and uncertainties of outdoor environments, such as variations in lighting, background, etc. Image tracking typically involves mathematical tools such as the Kalman filter, Bayesian network, etc [63, 58, 9, 131]. This image tracking method can be referred to as *Filtering and Data Association approach*, considered as a top-down process. The purpose of image tracking is to find the corresponding region or point to the given targets. For example, in [131], when the target was occluded, a nearby object at the predicted position via a motion model and a Kalman filter was selected as the target. However, this over-emphasis of the importance of spatial information can be problematic with unmodeled target maneuvers, which may cause the algorithm to lock onto a wrong object.

Beside using the motion prediction to find the corresponding region or point, *Target Representation and Localization* is considered to be another efficient way, which is referred to as the bottom-up approach. Among various searching methods, the mean shift approach using the density gradient is commonly used [11], which searches for the peak value of the object probability density. Continuously Adaptive Mean Shift (CAMSHIFT) algorithm was proposed in [11] to cope with noise and partial occlusion in images. This algorithm uses mean shift search method to efficiently obtain the optimal location of the target in a search window. The detected target is verified by comparing with the adaptive target

template. The mean shift procedure is used to carry out the optimization. However, the motion of the target between successive frames cannot be too large for successive search, and iterative search is not preferred in real-time applications. Therefore, mean shift-based tracking methods need to be integrated with other trackers in demanding applications.

Another image tracking approach, called active contour-based tracking, tracks objects by representing their outlines as contours, and updating these contours dynamically in successive frames via some optimization routine. It is computationally simple, thus allowing efficient implementation in real-time. Ha et al. presented a real-time visual tracking approach based on geometric active contour in [99], which is capable of realizing air-to-air tracking of a fix-wing airplane. However, this approach is limited by its sensitivity to initialization, which makes it difficult to automatically detect and track objects [58].

To realize target following, the relative distance to the target can be calculated based on the target location in the image and certain assumptions. This relative distance is used as the reference of the unmanned vehicle in the target following control. Classical control techniques can be used to design the following control law. Such work has been reported in many applications. In [20] the kinematic model was proposed for pose tracking, which tried to put certain geometric structure in the image. In order to realize tracking of a moving target, the dynamic model of the servo system should be considered. In [98], a time delay model of the RCA servos are considered and nonlinear controller was designed to achieve the expected performance.

Among the aforementioned applications, several main challenges in the applications of vision technologies to UAVs need to be highlighted:

1. Real-time and on-line processing : Normally the computational consumption of machine vision algorithms is very heavy, and powerful processors is required, especially in the real-time applications. However, powerful processors may not satisfy the payload and space constraints of UAV platforms. Thus, efficient vision algorithms are definitely preferred in such applications;

2. Moving platform : The moving platform will cause large motion in the image, and also lead to significant changes in shape and appearance of targets in the image. The feature descriptors should be insensitive to such changes, and such motion need to be compensated in the image tracking algorithm;
3. Autonomous initialization : Since it is hard to manually select a target in flight, the autonomous target initialization is required in vision applications of UAVs.

In fact, to satisfy the requirements of UAV applications, any existing vision algorithms still need to be improved. Vision sensing plays an important role in these applications. Therefore, a general framework for robust vision-based target detection, tracking and following for the vision applications on UAVs is proposed, which is shown in Figure 4.1. An advanced target detection approach is proposed and implemented in this work, which utilizes robust feature descriptors, such as color histogram, compactness, and improved moment invariants. The moment invariant method has the significant advantages of simple calculation and invariance under translation, rotation and scaling of the object in images, caused by the movements of the UAV and the target. However, the original formulation incurs high computation costs. Many alternative approaches to reduce the computational costs have been proposed in the literature (see, for example, [34, 49]). Chen and Tsai presented improved moment invariants in [24], which were computed along the boundary of a shape and greatly reduced the computation costs. In this work, the approach of [24] has been improved further for better performance.

In addition to target detection based on robust feature descriptors, a hierarchical image tracking scheme is proposed, which is composed of model-based tracking and mean shift-based tracking. First, motion-model based Kalman filtering technique is employed to provide accurate prediction of the position and velocity of a single target, which is also referred to as dynamic information. The target is then detected by using the data association method. But if the model based tracker fails to find the target, the mean shift-based image tracking method will be used to make the image tracking more robust.

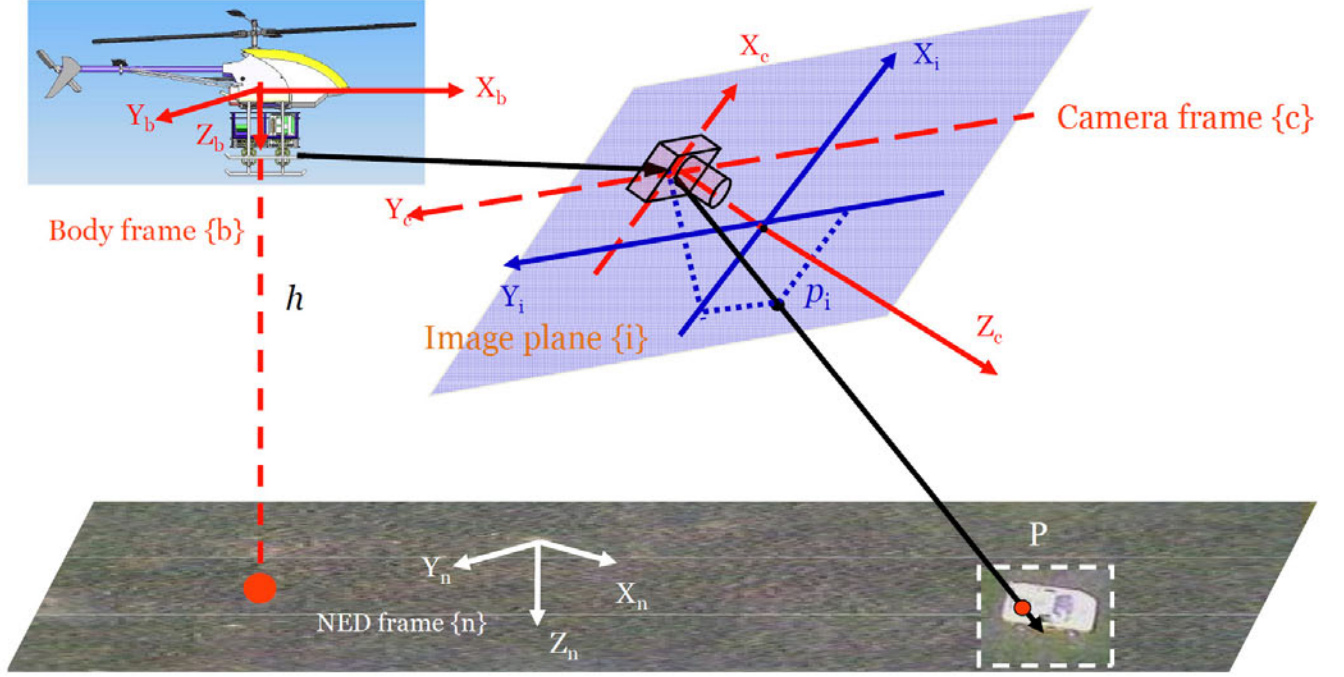


Figure 4.1: Illustration of the vision-based target following.

To achieve target following control, vision information is integrated with the on-board navigation sensors to estimate the relative distance to the target. Based on this vision feedback, a two-layer target tracking control framework is utilized to control a pan/tilt servo mechanism to keep the target in the center of the image, and guide the helicopter to follow the motion of the target.

In this application, to reduce computation load, it is assumed that only one target will appear in the image, and it has distinguished color and shape compared to the background and other objects in environments.

The remainder of this chapter is organized as follows: Section 4.2 details the vision-based target detection and tracking algorithm aided by the motion estimation. Section 4.3 describes the definitions of the coordinate systems. Section 4.5 describes the target following control scheme based on the vision information. Section 4.6 shows experimental results in both off-line tested and actual flight tests. Finally, some concluding remarks are drawn in

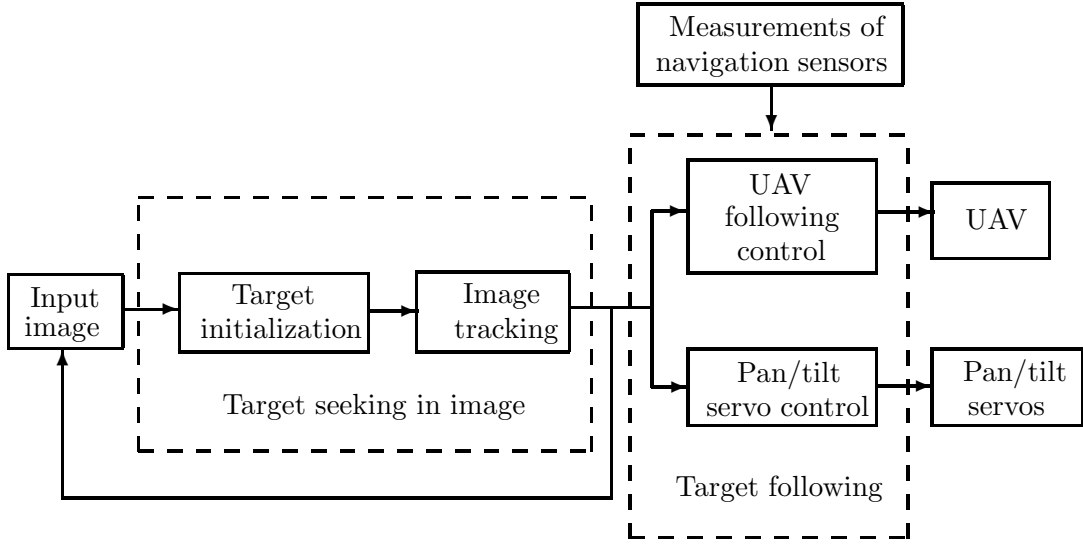


Figure 4.2: Flow chart of the ground target detection, tracking and following scheme.

Section 4.7.

4.2 Target Detection and Tracking in the Image

In this section, a systematic approach is proposed to realize robust and efficient ground target detection and tracking. As illustrated in Figure 4.2, in the target initialization (or target detection part), a pre-defined target is identified automatically by using color segmentation and feature-based pattern recognition methods. A hierarchical tracking scheme is then used to track the target in the image. The target detection and tracking in the image will be detailed in the following parts.

4.2.1 Target Detection

The purpose of target detection is to identify the target of interest from the image automatically based on a database of the interest targets. A toy car is chosen as the ground target. A classical pattern recognition procedure is used to identify the target automatically, which includes three main steps: segmentation, feature extraction, and pattern recognition.

Segmentation

The segmentation step aims to separate the interest objects from background. To simplify the processing, some assumptions are made. First, the target and environments are assumed to exhibit *Lambertian* reflectance. In other words, their brightness is unchanged regardless of the viewing directions. Second, the target is assumed to have a distinct color distribution compared to the surrounding environment.

1. STEP 1: THRESHOLD IN COLOR SPACE. Based on the above assumptions, a fast threshold approach is applied to the input color image. To make the surface color of the target constant and stable under the varying lighting condition, the color image is represented in HSV space, which stands for hue (*hue*), saturation (*sat*) and value (*val*) as originally introduced by Smith [114].

In fact, there are a number of color models used in various applications involving color image processing, such as CIE, RGB, YUV, HSV and CMYK. The HSV color space is selected in this work due to two useful facts, i.e., 1) the intensity component is decoupled from chrominance information represented as hue and saturation; and 2) the hue and saturation are intimately related to the way in which humans perceive chrominance [46, 88, 110].

Given each pixel with a set of RGB values $\{r, g, b\}$, defined $max_{rgb} = \max\{r, g, b\}$ and $min_{rgb} = \min\{r, g, b\}$ the HSV color model, conveniently represented by the hexcome

model [40], is defined as follows:

$$hue = \begin{cases} \text{undefined,} & \text{if } max_{rgb} = min_{rgb} \\ 60^\circ \times \frac{g - b}{max_{rgb} - min_{rgb}} + 0^\circ, & \text{if } max_{rgb} = r \text{ and } g \geq b \\ 60^\circ \times \frac{g - b}{max_{rgb} - min_{rgb}} + 360^\circ, & \text{if } max_{rgb} = r \text{ and } g < b \\ 60^\circ \times \frac{b - r}{max_{rgb} - min_{rgb}} + 120^\circ, & \text{if } max_{rgb} = g \\ 60^\circ \times \frac{r - g}{max_{rgb} - min_{rgb}} + 240^\circ, & \text{if } max_{rgb} = b \end{cases} \quad (4.1)$$

$$sat = \begin{cases} 0, & \text{if } max_{rgb} = 0 \\ 1 - \frac{min_{rgb}}{max_{rgb}}, & \text{otherwise} \end{cases} \quad (4.2)$$

and

$$val = max_{rgb} \quad (4.3)$$

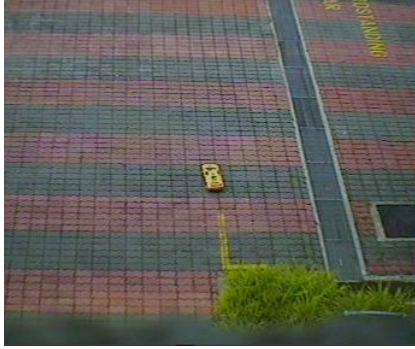
Pre-calculated threshold ranges are applied to the *hue*, *sat*, and *val* channels:

$$hue_r = [h_1, h_2], \quad sat_r = [s_1, s_2], \quad val_r = [v_1, v_2] \quad (4.4)$$

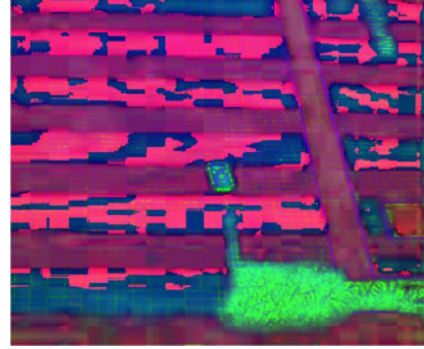
Only the pixel values falling in these color ranges are identified as the foreground points, and pixels of the image that fall out of the specified color range are removed. The procedure of the image pre-processing is illustrated in Figure 4.3. The raw image is converted into HSV color space, which is shown in Figure 4.3.b. The converted image is then segmented by using the method mentioned earlier based on *hue_r*, *sat_r*, and *val_r* three channels shown in Figure 4.3.c.

2. Step II: Morphological Operations

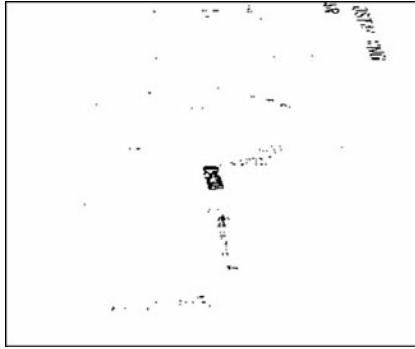
As shown in Figure 4.3, normally, the segmented image is not smooth and has image



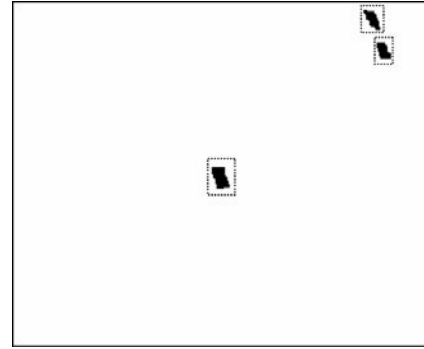
a. Original image



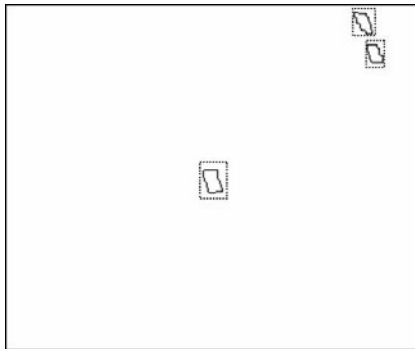
b. The image in HSV color space



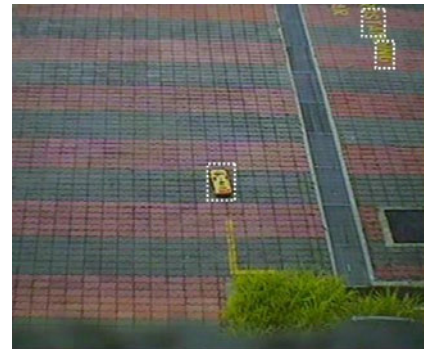
c. The image after threshold



d. The image after morphological operations



e. The image after contour detection



f. Regions of interest in detected in the image

Figure 4.3: Illustration of Segmentation.

noise. Morphological operations are then employed to smooth the contour, filter out noise, fuse narrow breaks and gulfs, eliminate small holes, and fill gaps in the contour. Next, a contour detection approach is used to obtain the complete boundary of the objects in the image, which will be used in the feature extraction step.

Feature Extraction

Generally speaking, multiple objects will appear in the segmented images, including the desired target and false objects. Instead of extensively matching based on the templates of the target, features are extracted from these objects, and feature descriptors are constructed. In fact, many feature extraction approaches and descriptors are proposed by researchers, such as the intensity, edge, corner, color, boundary, location, orientation, gradient, and so on. Among these features, good features should have the following characteristics as follows:

1. Distinctiveness : They should be able to identify the target correctly from other objects;
2. Robustness : They should be sensitive to the noise and disturbance in the image, such as sensor noise, illumination changes, rotation, scaling, and so on;
3. Efficiency : They should be extracted easily. The computation cost of the feature descriptor is not heavy. The matching to the database is not costly. Efficiency is very important for real-time applications.

Due to such requirements, a set of descriptors are constructed. They are invariant under rotation, translation and scaling of the target, including 1) compactness, 2) moment invariants, and 3) color histogram. The detailed discussion is given as follows:

1. Compactness :

Compactness of a shape is measured by the ratio of the square root of the area and

the perimeter (in [131], a slightly different definition was given), and is a useful feature descriptor for recognition, which is given by

$$\text{Compactness: } \beta_c = \frac{\sqrt{A}}{C} , \quad (4.5)$$

where A is the interior area of the object; C is the boundary curve of the object. It can be easily proven that compactness is invariant with respect to translation, scaling and rotation. In addition, to reduce the computation time, an increment method is used to calculate area and perimeter of a shape.

2. Moment invariants :

Moment invariants are widely used in vision applications, such as pattern recognition, image coding and orientation estimation. Hu defined the continuous two-dimensional (p, q) -th order moments of a density distribution function $\rho(x, y)$ in [57], which is given by

$$m_{pq} = \int_0^\infty \int_0^\infty x^p y^q \rho(x, y) dx dy , \quad (4.6)$$

where $\rho(x, y)$ is assumed to be a piecewise continuous and bounded function, and has nonzero values only in the finite portion of the XY plane. It can then be proven that the moments of all orders exist and are unique. In particular, we have the following so-called uniqueness theorem [57].

Theorem 4.2.1. *The double moment sequence $\{m_{pq}\}$ is uniquely determined by $\rho(x, y)$; and, conversely, $\rho(x, y)$ is uniquely determined by $\{m_{pq}\}$.*

The central moments were defined by Hu as:

$$\mu_{pq} = \int_0^\infty \int_0^\infty (x - \bar{x})^p (y - \bar{y})^q \rho(x, y) dx dy , \quad (4.7)$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. It can be easily proven that central moments are invariant to translation. However, they are variant with respect to scaling. In [57], Hu defined normalized central moments, which are invariant with respect to scaling, as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad (4.8)$$

where $\gamma = (p + q)/2 + 1$, $p + q = 2, 3, \dots$. These normalized central moments are invariant with respect to translation and scaling, but variant with respect to rotation. From the normalized central moments, Hu further constructed a set of moment invariants that are invariant to translation, scaling, and rotation [57]. The four lowest moment invariants are given by

$$\phi_1 = \eta_{20} + \eta_{02} \quad (4.9)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (4.10)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \quad (4.11)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \quad (4.12)$$

While these moment invariants may be used to identify a given shape, their calculation requires every pixel value in the interior and on the boundary of the shape. Thus, the computational cost increases dramatically with an increasing object size.

To reduce the cost of moment computation, in [24], Chen and Tsai presented a new set of moments that only uses the pixels on the boundary of a shape. These moments are given by:

$$m_{pq}^c = \int_C x^p y^q \rho(x, y) ds, \quad p, q = 0, 1, 2, \dots \quad (4.13)$$

where C is the boundary curve of the shape, \int_C is a line integral along C , and $ds =$

$\sqrt{(dx)^2 + (dy)^2}$. Since $\rho = 1$ on the boundary and $\rho = 0$ elsewhere, eq. (4.13) can be simplified as:

$$m_{pq}^c = \int_C x^p y^q ds, \quad p, q = 0, 1, 2, \dots \quad (4.14)$$

Next, Chen and Tsai defined the new central moments and new normalized central moments as:

$$\begin{aligned} \mu_{pq}^c &= \int_C (x - \bar{x})^p (y - \bar{y})^q ds, \quad p, q = 0, 1, \dots \\ \eta_{pq}^c &= \frac{\mu_{pq}^c}{(\mu_{00}^c)^{p+q+1}} \end{aligned} \quad (4.15)$$

where

$$\bar{x} = \frac{m_{10}^c}{m_{00}^c}, \quad \bar{y} = \frac{m_{01}^c}{m_{00}^c}$$

and

$$\mu_{00}^c = \int_C ds = |C| = \text{length of curve } C.$$

It was shown in [24] that, substituting into eq. (4.9)–(4.12) the new normalized central moments in eq. (4.15) instead of eq. (4.8), the resulting moment invariants are also invariant to translation, rotation and scaling.

The formulation of Chen and Tsai can significantly reduce the moment computation cost. However, it was found that the normalization factor μ_{00}^c in eq. (4.15) is very sensitive to noise. Actually, μ_{00}^c is the length of the boundary curve, compared to which the area of the object is less sensitive to noise, i.e., the variance of the area is smaller than that of the length of the boundary. Hence, a variant of Chen and Tsai's formulation is proposed by using the area of the object as the normalization factor.

Table 4.1: Comparison of ϕ_1 between two normalization methods

	Normalize by μ_{00}	Normalize by area
Mean value	0.01627	1.6763
Standard deviation	0.002119	0.14028
$\frac{\text{Standard deviation}}{\text{Mean value}}$	13.7598%	8.3682%

Thus, the improved normalized central moments are defined as:

$$\eta_{pq}^m = \frac{\mu_{pq}^c}{A^{(p+q+1)/2}} \quad (4.16)$$

where A is the interior area of the object, for $p + q = 2, 3, \dots$

In Appendices, the similar lines of reasoning as in [24] are adopted to show that the moment invariants ϕ_i in eq. (4.9)–(4.12), calculated by using η_{pq}^m as normalization factor, are indeed invariant to translation, scaling and rotation.

As an example, the moment invariant ϕ_1 of a target, captured repeatedly in actual flight, are calculated using the normalized central moments in eq. (4.15) and (4.16), respectively. The test results are shown in Figure 4.4, from which we can see that the distribution of ϕ_1 calculated using η_{pq}^m is narrower than that calculated using η_{pq}^c . The statistical parameters of these two distributions, namely the mean value and standard deviation, are shown in Table 4.1.

3. Color feature :

To make the target detection and tracking more robust, color histogram is also employed to represent the color distribution of the image area of a target, which is not only independent of the target orientation, position and size, but also robust to partial occlusion of the target and easy to implement. Due to the stability of color features in outdoor environments, only *hue* and *val* are employed to constructed the color histogram for object recognition. The color histogram is defined as: $H(i, j)_{i=1, \dots, N_{hue}; j=1, \dots, N_{val}}$.

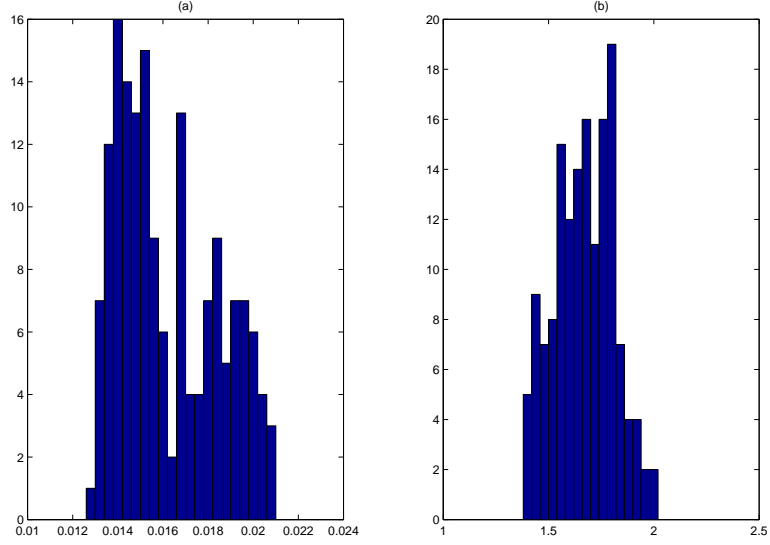


Figure 4.4: Comparison of ϕ_1 using two normalization methods
 (a) Distribution of ϕ_1 calculated using η_{pq}^c as normalization factor;
 (b) Distribution of ϕ_1 calculated using η_{pq}^m .

N_{hue} , N_{val} are the partition numbers of *hue* and *val* color channels.

$$H = \{hist(i, j)\}_{i=1, \dots, N_{hue}; j=1, \dots, N_{val}} , \quad (4.17)$$

where

$$hist(i, j) = \sum_{(x, y) \in \Omega} \delta \left(i, \left[\frac{hue(x, y)}{N_{hue}} \right] \right) \delta \left(j, \left[\frac{val(x, y)}{N_{val}} \right] \right) , \quad (4.18)$$

Ω is the region of the target, $[\cdot]$ is the nearest integer operator, and $\delta(a, b)$ is the Kronecker delta function given by

$$\delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \quad (4.19)$$

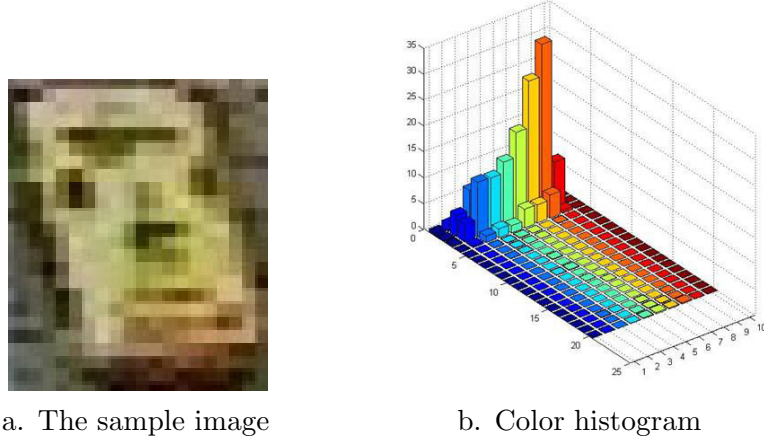


Figure 4.5: Color histogram extraction.

Object Representation

The purpose of object representation is to arrange features of an object in a compact and identifiable form [89]. A straightforward way is to convert these features in a high dimensional vector. For example, the feature vector of an object is given by

$$\alpha = [\beta_c \ \phi_1 \ \phi_2 \ \phi_3 \ \phi_4 \ H] = \{\alpha_k\}, \ k = 1 \dots d \quad (4.20)$$

where d is the dimension of the feature vector. Theoretically, all the states or the features of an object template in a database should be updated based on a certain rule. To reduce computational cost, the appearance of objects is assumed to be constant.

Pattern Recognition

The purpose of pattern recognition is to identify the target from the extracted foreground objects in terms of the extracted features in (4.20). The straightforward classifier is to use the nearest-neighbor rule. It calculates a metric or a “distance” between an object and a template under a feature space, and assign an object to a class with the highest scope. To take advantage of *a priori* knowledge of feature distributions, the pattern recognition

problem in this project is formulated under a model-based framework, and solved by using a probabilistic classifier. The pattern recognition algorithm will be described in the following parts.

1. Step I. Pre-filter :

Before classifying objects, a pre-filter is carried out to remove objects whose feature values are outside certain regions determined by *a priori* knowledge. This step aims to improve robustness of pattern recognition and also speed up computation.

2. Step II. Discriminant Function :

A discriminant function, derived from Bayes' theorem is employed to determinate the target based on measured feature values of each object and known distribution of features obtained from training data, which is given by.

$$P(\omega_j|\alpha) = \frac{p(\alpha|\omega_j)P(\omega_j)}{p(\alpha)} \quad (4.21)$$

where ω_j is a discrete class j , $j = 1, 2, \dots, n$, which indicates the target class in the database; $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)^T$ is a continuous feature vector of an object in d -dimensional, which includes a set of continuous feature variables given in (4.20) ; $P(\omega_j)$ is the *a priori* probability, which is the probability that the target class be present; $P(\omega_j|\alpha)$ is the *a posteriori* probability, which is the probability that α be assigned to class ω_j ; $p(\alpha|\omega_j)$ is the class-conditional probability density function of the features, given the target class j , which is denoted in low case and represents a function of continuous variable; $p(\alpha)$ is a probability density function for the continuous feature variables.

Actually, the purpose of equation (4.21) is to decide the *a posteriori* probability $P(\omega_j|\alpha)$ based on the *a priori* probability $P(\omega_j)$ and a measured feature vector α . Since, given α , the denominator $p(\alpha)$ in (4.21) does not depend on the classes, it is therefore constant and can be viewed as a scale factor. Thus, we ignore $p(\alpha)$ and

only consider the numerator in (4.21), and define Bayesian discriminant function and Bayesian classifier as:

$$\begin{cases} f_j(\alpha) = P(\omega_j|\alpha) = p(\alpha|\omega_j)P(\omega_j) \\ h(\alpha) = \begin{cases} \arg \max_j f_j(\alpha), & \max f_j(\alpha) \geq \Delta_j \\ \text{Undetectable object}, & \max f_j(\alpha) < \Delta_j \end{cases} \end{cases} \quad (4.22)$$

The purpose of the classifier in (4.22) is to assign an object with the feature value α to the class corresponding to the largest value of the discriminant function in (4.22), if the discriminant function is less than a predefined threshold Δ_j .

3. Step III. Simplified Discriminant Function :

The Bayesian classifier in (4.22) needs to know the distribution of class ω_j : $P(\omega_j)$ and the conditional probability density function of features with the given class ω_j : $p(\alpha|\omega_j)$, where $P(\omega_j)$ is the probability of class ω_j . To simplify the discriminant function, it is assumed that only one target will appear in the image, and $P(\omega_j)$ in (4.22) is a constant value .

In addition, it is unreasonable to assume that all the features are equally important for the target detection. It is observed in the experiment that some features are relatively sensitive to the disturbances, such as distortion of shape, noise in the image and calculation error, while others are not. According to our experience, we thus assign different weightings to the features in the simplified discriminant function.

Furthermore, it is assumed that features are independent, and the probability densities of each feature approximately fulfill normal distributions, which can be estimated using the maximum likelihood method from the training data captured in flight. Based on the assumption of distribution of $p(\alpha|\omega_j)$, we take logarithm of the discriminate

function in (4.22), and ignore the constant factors, and yield:

$$f(\alpha) = \sum_{k=1}^5 w_k \left(\frac{\alpha_k - \mu_k}{\sigma_k} \right)^2 + \sum_{p=1}^{N_{hue}} \sum_{q=1}^{N_{val}} w_{p,q} \left(\frac{H(p,q) - \mu_{p,q}}{\sigma_{p,q}} \right)^2 \quad (4.23)$$

To reduce computation consumption, instead of directly using the color histogram in 4.23, the color histogram intersection [119] is employed to match the color histogram of each object with the pre-defined target template G , which is given by

$$d_c(H, G) = \frac{\sum_{p=1}^{N_{hue}} \sum_{q=1}^{N_{val}} \min(H(p,q), G(p,q))}{\min(|H|, |G|)}, \quad (4.24)$$

where $|H|$ and $|G|$ are the numbers of the pixels in the image region H and G . The advantage of this distance formula is that the colors that are not present in the defined target histogram do not contribute to the intersection distance. The effect of the background to the intersection distance can be reduced in this way. Thus the simplified discriminant function and classifier are given by

$$f(\alpha) = \sum_{k=1}^5 w_k \left(\frac{\alpha_k - \mu_k}{\sigma_k} \right)^2 + w_6 \left(\frac{d_c(H, G) - \mu_6}{\sigma_6} \right)^2 \quad (4.25)$$

and

$$h(\alpha) = \begin{cases} \text{target,} & \text{if } f(\alpha) \leq \Delta \\ \text{Undetectable object,} & \text{if } f(\alpha) > \Delta \end{cases} \quad (4.26)$$

where w_1 to w_6 are the weighting scalars of the corresponding features. Δ is a threshold value to be chosen empirically. Based on this classifier. An object is assigned to the target class, if $f(\alpha) \leq \Delta$. Otherwise it is assigned to the undetectable object class. If there are multiple objects in the target class, the one with the smallest value of the discriminant function is selected as the target. While, if the target class is empty,

there is no target in the current frame.

4.2.2 Image Tracking

As shown in Figure 4.2, after initialization, the image tracking techniques are employed. The purpose of image tracking is to find the corresponding region or point to the given target. Unlike detection, the objects are not required to be classified, although verification is required. Thus, the processing speed of image tracking is faster than detection. The image tracking problem can be solved by using two main approaches: 1) *Filtering and Data Association*, and 2) *Target Representation and Localization* [27], which are presented in the following parts.

1. Filtering and Data Association :

The Filtering and Data Association approach can be considered as a top-down process. The purpose of filtering is to estimate the state of the target, such as appearance and location. Typically, the state estimation is achieved by using filtering techniques [126, 131]. “Most tracking algorithm are model based because a good model-based tracking algorithm will greatly outperform any model-free tracking algorithm if the underlying model turns out to be a good one.” [71]. If the measurement noise satisfies the Gaussian distribution, the optimal solution can be obtained by Kalman filtering [10]. In the more general case, particle filters are more suitable and robust [60]. However the computational cost will increase and sample degeneracy is also a problem.

When multiple targets are tracked in the image sequence, validation and association of the measurements become a critical issue. Association techniques, such as Probabilistic Data Association Filter (PDAF) and Joint Probabilistic Data Association Filter (JPDAF) are widely used [121].

2. Target Representation and Localization :

Besides using motion prediction to find the corresponding region or point, *Target*

Representation and Localization is considered as another efficient way, which is referred to as the bottom-up approach. Among the search methods, the mean shift approach using the density gradient is commonly used [11], which searches for the peak value of the object probability density. However, the efficiency will be limited when the movement of the target becomes significant.

To take advantages of the aforementioned approaches, multiple-tracker methods were widely adopted in applications of image tracking. In [121], a tracking scheme that integrates motion, color and geometric features was proposed to realize robust image tracking. In conclusion, combining the motion filtering and advanced searching algorithms will definitely make the tracking processing more robust, but the computational consumption will increase.

In this work, instead of using multiple trackers simultaneously, a hierarchical tracking scheme is proposed to balance the computational cost and performance, which is illustrated in Figure 4.6. In model-based image tracking, the Kalman filtering technique is employed to provide accurate estimation and prediction of the position and velocity of a single target, referred to as dynamic information. If the model-based tracker fails to find the target, a mean shift-based image tracking method will be activated to retrieve the target in the image.

Model-based Image Tracking

Model-based image tracking is a top-down method. It predicts the possible location of the target in the subsequent frames, and then perform data association based on a updated likelihood function. Several methods are employed to make the tracking more robust and efficient, which are given by:

1. Narrow the search window in terms of the prediction of the Kalman filter;
2. Integrate the spatial information with appearance and set the different weightings for the discriminant function. For example, it is meaningful to give higher weightings to

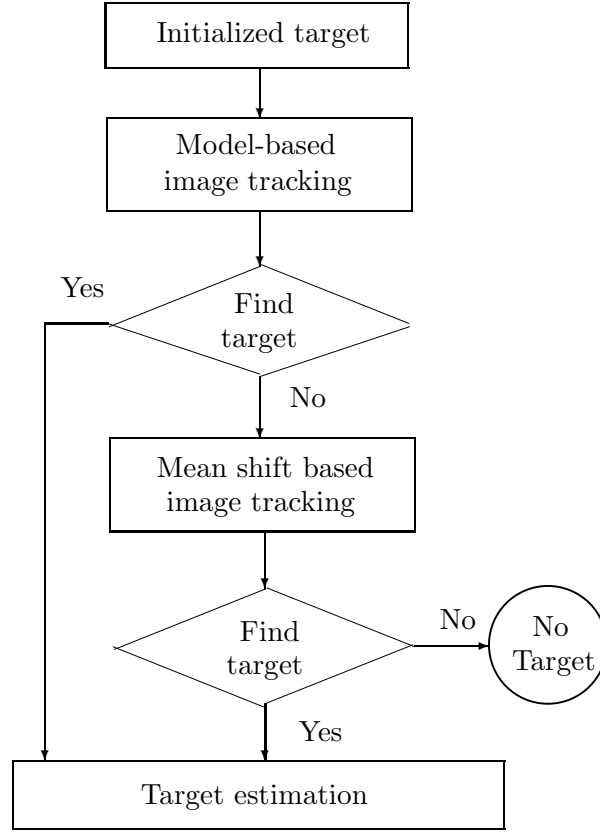


Figure 4.6: Flow chart of image tracking.

the spatial information, when the geometric features may change significantly under noise conditions.

To predict the target location in the image, a motion model is required. It is well known that the motion of a point mass in the two-dimensional plane can be defined by its two-dimensional position and velocity vector. Let $\mathbf{x} = [\bar{x}, \dot{\bar{x}}, \bar{y}, \dot{\bar{y}}]^T$ be the state vector of the centroid of the tracked target in the Cartesian coordinate system. Non-maneuvering motion of the target is defined by it having zero acceleration: $[\ddot{\bar{x}}, \ddot{\bar{y}}]^T = [0, 0]^T$. Strictly speaking, the motion of the intended ground targets may be maneuvering with unknown input. Nevertheless, we assume the standard 4-th order non-maneuvering motion model by setting the acceleration as $[\ddot{\bar{x}}, \ddot{\bar{y}}]^T = \mathbf{w}(t)$, where $\mathbf{w}(t)$ is a white noise process [71]. The

resulting continuous-time model is:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w} \\ \mathbf{z} &= \mathbf{C}\mathbf{x} + \mathbf{v}\end{aligned}$$

where \mathbf{w} and \mathbf{v} denote the input and measurement zero-mean Gaussian noises, and

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The motion of the centroid of the target, $\mathbf{x} = [\bar{x}, \dot{\bar{x}}, \bar{y}, \dot{\bar{y}}]^T$, in the two-dimensional image coordinate is tracked using a standard 4th-order Kalman filter, which predicts the possible location of the target in successive frames. The discrete-time model of the target motion can be expressed as

$$\begin{aligned}\mathbf{x}(k|k-1) &= \Phi\mathbf{x}(k-1) + \Lambda\mathbf{w}(k-1), \\ \mathbf{z}(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k),\end{aligned}\tag{4.27}$$

where

$$\begin{aligned}\Phi &= \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix}, \\ \mathbf{H} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},\end{aligned}$$

T_s is the sampling period of the vision software. A Kalman filter can then be designed based on the above motion model to estimate the states of the target in the image plane. The filter consists of the following stages:

1. Predicted state

$$\hat{\mathbf{x}}(k|k-1) = \Phi \hat{\mathbf{x}}(k-1),$$

2. Updated state estimate

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)),$$

where $\mathbf{K}(k)$ is the optimal Kalman gain.

The Euclidean distance between the location of an object z and the predicted location of the target \hat{z} is employed in the likelihood function, which is defined by

$$\tilde{\mathbf{z}} = \mathbf{z}(k) - \hat{\mathbf{z}}(k) = \mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1).$$

Both of the static and dynamic features of an object are employed in the image tracking. Thus, the updated likelihood function, which includes the appearance and spatial information, is shown as follows:

$$\begin{cases} f(\alpha) &= \sum_{k=1 \dots 5} w_k \left(\frac{\alpha_k - \mu_k}{\sigma_k} \right)^2 + w_6 \left(\frac{d_c(H, G) - \mu_6}{\sigma_6} \right)^2 + w_7 \left(\frac{\|\tilde{\mathbf{z}}\| - \mu_7}{\sigma_7} \right)^2 \\ h(\alpha) &= \begin{cases} \text{Target,} & \min f(\alpha) \leq \Delta \\ \text{Undetectable object,} & \min f(\alpha) > \Delta \end{cases} \end{cases}$$

Most of time, the model-based tracker can lock the target in the image sequence, but sometime it may fail due to the noise or disturbance, such as partial occlusion. Thus, a

scheme is required to check whether the target is still in the image, and then activate other trackers.

Switching mechanism

The purpose of the switching mechanism is to check whether the target is still in the image at the moment when the target is lost by the model-based tracker. If the target is still in the image, the mean shift tracker will be activated. The loss of the target can be attributed to poor feature match due to noise, distortion, or occlusion in the image. An alternative reason may be the maneuvering motion of the target, and the target is out of the image. Therefore, in order to know the reason and take the special way to find target again, it is necessary to formulate the decision making as the following hypothesis testing problem:

H_0 : The target is still in the image;

H_1 : The target is not in the image due to maneuvers.

The estimation error is considered as a random variable, which is defined by:

$$\begin{aligned}\varepsilon &= \|\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1}\|_{\Sigma^{-1}}^2 \\ &= (\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1})' \Sigma^{-1} (\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1})\end{aligned}$$

where $\mathbf{H}\hat{\mathbf{x}}_{k-1} - \mathbf{z}_{k-1}$ is assumed to be $N(0, \Sigma)$ -distributed. ε is Chi-square distributed with 2-degrees of freedom (x and y directions) under H_0 .

$$\begin{cases} \varepsilon < \lambda = \chi_2^2(\alpha), & H_0 \text{ is true} \\ \varepsilon \geq \lambda = \chi_2^2(\alpha), & H_1 \text{ is true} \end{cases}$$

where $1 - \alpha$ is the level of confidence, which should be sufficient high (for our project, $1 - \alpha = 99\%$). If H_0 is true, the Chi-square testing-based switching declares the target is

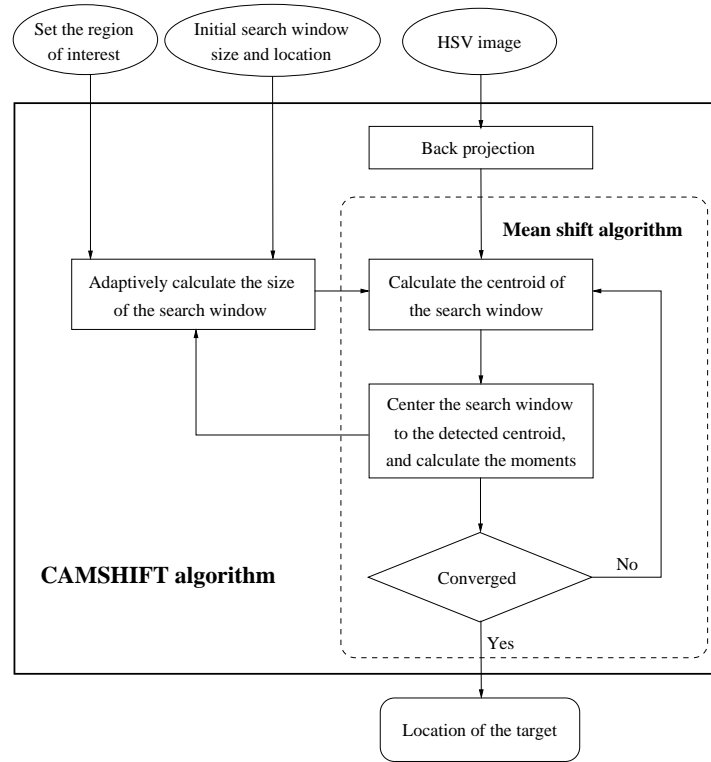


Figure 4.7: Block diagram of the CAMSHIFT algorithm.

still in the image and enables the mean shift- based tracker.

Mean Shift-Based Image Tracking

If the target is still in the image, the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm [11] is employed, which is shown in Figure 4.7. This algorithm uses mean shift search method to efficiently obtain the optimal location of target in the search window. The principle idea is to search the dominated peak in the feature space based on the previous information and certain assumption. The detected target is verified by comparing with an adaptive target template. The CAMSHIFT algorithm consists of three main steps: back projection, mean shift searching, and search window adaptation, which is illustrated in Figure 4.7.

1. Step I. Back projection :

In order to search the target in the image, the probability distribution image needs to be constructed based on the color distribution of the target. The color distribution of the target defined in the *hue* channel is employed and given by

$$hist_{tg}(i) = \sum_{(x, y) \in \Omega} \delta \left(i, \left\lceil \frac{hue_{tg}(x, y)}{N_{hue}} \right\rceil \right), \quad i = 1, \dots, N_{hue}.$$

where Ω is the region of the target. Based on the color model of the target, the back projection algorithm is employed to convert the color image to the color probability distribution image. The probability of each pixel $\mathbf{I}_p(x, y)$ in the region of interest Ω_r is calculated based on the model of the target, which is used to map the histogram results and given by

$$\mathbf{I}_p(x, y) = hist_{tg} \left(\left\lceil \frac{\mathbf{I}_{hue}(x, y)}{N_{hue}} \right\rceil \right), \quad (x, y) \in \Omega_r \quad (4.28)$$

where \mathbf{I}_{hue} is the pixel values of the image in the hue channel.

2. Step II. Mean shift algorithm :

Based on the obtained color density image, robust non-parametric method, mean-shift algorithm, is used to search the dominated peak in the feature space. The mean-shift algorithm is an elegant way of identifying these locations without estimating the underlying probability density function [25].

Recall the discrete 2D image probability distributions in (4.28), the mean location (the centroid) of the search window is computed by

$$\begin{aligned} x_c(k) &= \frac{m_{10}}{m_{00}}; \\ y_c(k) &= \frac{m_{01}}{m_{00}}; \end{aligned}$$

where

$$\begin{aligned} m_{00} &= \sum_{(x, y) \in \Omega_w} \mathbf{I}_p(x, y), \\ m_{10} &= \sum_{(x, y) \in \Omega_w} \mathbf{I}_p(x, y)x, \\ m_{01} &= \sum_{(x, y) \in \Omega_w} \mathbf{I}_p(x, y)y, \end{aligned}$$

where k is the iteration index; Ω_w is the region of the search window; m_{00} is the zero-th order moment; m_{10} and m_{01} are the first order moments for x and y , respectively. The search window is centered at the mean location $\mathbf{c}(k) = (x_c(k), y_c(k))$. Then, Step 2 will be repeated until $\|\mathbf{c}(k) - \mathbf{c}(k-1)\| < \varepsilon$, so the algorithm converges.

3. Step III. Search window adaptation :

The region of interest is calculated dynamically using the motion filtering given in Section 4.2.2. To improve the performance of the CAMSHIFT algorithm, multiple search windows in the region of interest are employed. The initial locations and sizes of the searching windows are adopted from the centers and boundaries of the foreground objects respectively. These foreground objects are obtained using the color segmentation in the region of interest. In the CAMSHIFT algorithm, the size of the search window is dynamically updated according to the moments of the region inside the search window [11]. Generally, more than one target candidate will be detected due to multiple search windows adopted.

To identify the true target, the similarity between the target model and the detected target candidate is measured using the intersection comparison (4.24). This verification can effectively reduce the risk of detecting the false target.

In this work, CAMSHIFT is only utilized to handle special cases where the target is partially occluded or light condition is changed, since iterative calculation is involved in the mean shift. It is not easy to define the stop criterion of the iteration in the real-time applications.

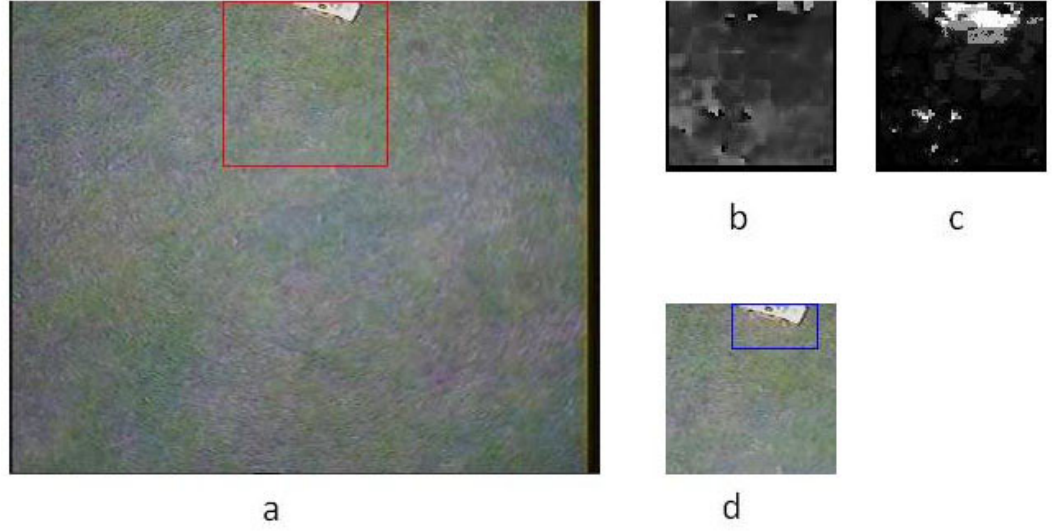


Figure 4.8: Image tracking using the CAMSHIFT algorithm.

(a) The image with the search window; (b) The hue channel of the image inside the search window; (c) The back projection image; (d) The searched target using the CAMSHIFT algorithm.

One example of the target tracking using the CAMSHIFT algorithm in the partial occluded condition is given in Figure 4.8.

Supervisor

A finite state machine is employed as the supervisor in the proposed vision scheme, as depicted in Figure 4.9. This finite state machine dynamically chooses necessary features and gives different weightings to each feature in the discriminant function under different tracking conditions. The detailed functions of each state in the finite state machine are illustrated as follows.

State 0 (S0): Since there is no target found in the image, only static features are used in discriminant function of (4.26) to identify the target in the entire image.

State 1 (S1): The same target has continuously been found by the algorithm less than n frames, thus, the target cannot be locked with confidence. The discriminant function of

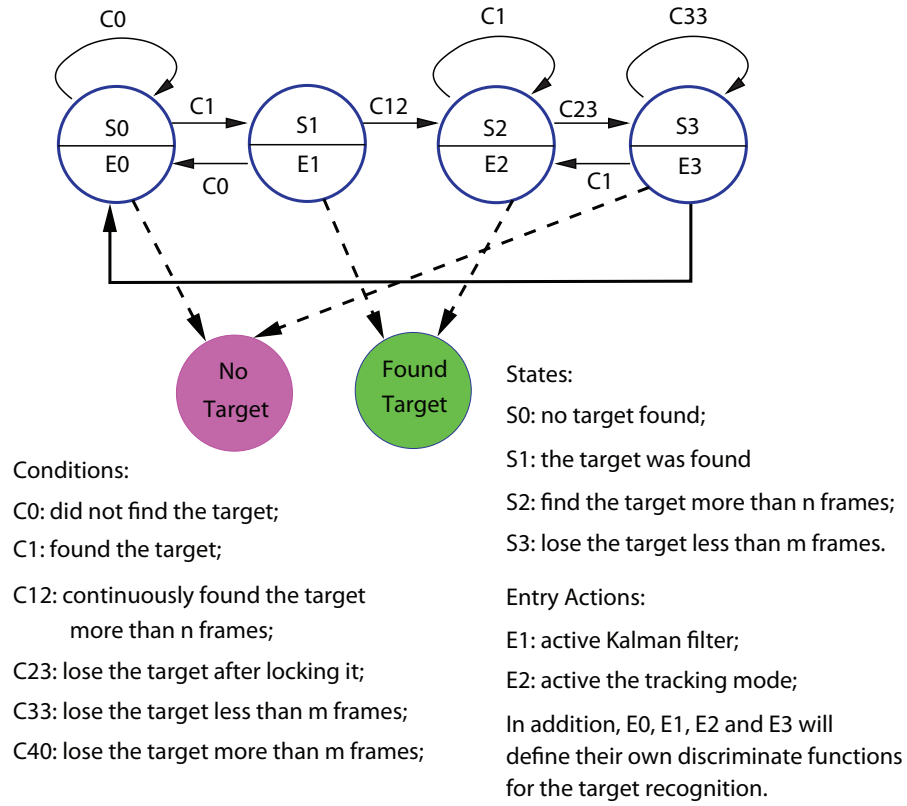


Figure 4.9: Decision making using a finite state machine.

(4.26) still uses static features in the pattern recognition, but enables a Kalman filtering to estimate the possible location of the target in the next frame.

State 2 (S2): The same target has continuously been found by the algorithm more than n frames. We then have confidence to decide that it is the real target and activate model-based image tracking to lock the target in the successive frames.

State 3 (S3): The target is lost by the model-based tracking approach. If the partial occlusion detection, based on Chi-square test, indicates that the target is still in the image and may be partially occluded. The mean shift-based tracker will be activated to iteratively search the target.

Shown in Figure 4.10 is an example of the tracking of a toy car using the proposed vision detection algorithm in the ground test. The solid window is the measured location

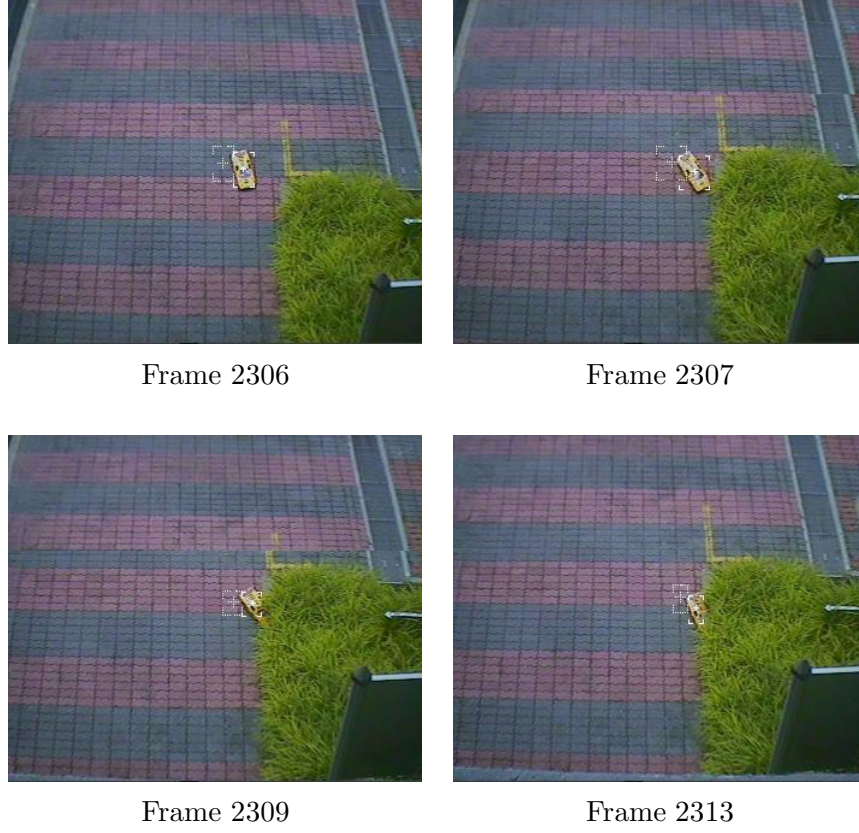


Figure 4.10: Target detection with occlusion.

of the target, and the dash window is the predicted location of the target in the image plane. The vision detection algorithm automatically initializes the detection, then track the target. When the target is partially occluded, the vision algorithm gives high weightings to the dynamic and color features in the discriminant function, and also activates the mean shift-based tracking approach. Thus, the target still can be identified, even though it is partially occluded.

Experimental Results of Target Detection and Tracking in Video Sequence

The proposed vision-based tracking algorithm is implemented in the on-board system of the unmanned helicopter, SheLion. The processing rate of the algorithm is 10 frames per second (FPS). During the real flight tests, the helicopter is manually controlled to hover at

Table 4.2: Experimental results of target detection and tracking

Times	Total frame	Detected frames	Accuracy
1	219	191	87.21%
2	284	209	73.59%
3	703	538	76.53%
4	375	295	78.67%
5	676	508	75.15%
6	431	311	72.16%
7	108	91	84.26%
8	1544	1162	75.26%
9	646	529	81.89%

a fixed position 10 meters above the flat ground, and the on-board visual tracking system automatically identify and track the ground moving target: a toy car, which is manually controlled to randomly move in the flat ground.

We performed nine times of visual tracking tests and the tracking results are shown in Table 4.2. During these tests, the visual tracking system can successfully track the ground target. One example of the tracking errors in vertical and horizontal direction is shown in Figure 4.11, which indicates that the tracking error is bounded. The experimental results demonstrate the robustness and effectiveness of the visual tracking system, which can automatically identify and track the moving target in flight.

4.3 Coordinate Systems

Before presenting the vision-based ground target following algorithm, the coordinate systems involved in this thesis are introduced, which are depicted in Figure 4.12.

1. The local north-east-down (NED) coordinate system (labeled with a subscript ‘n’) is an orthogonal frame on the surface of the earth, whose origin is the launching point of the aircraft on the surface of the earth. The X_n - Y_n plane of the NED frame is tangent

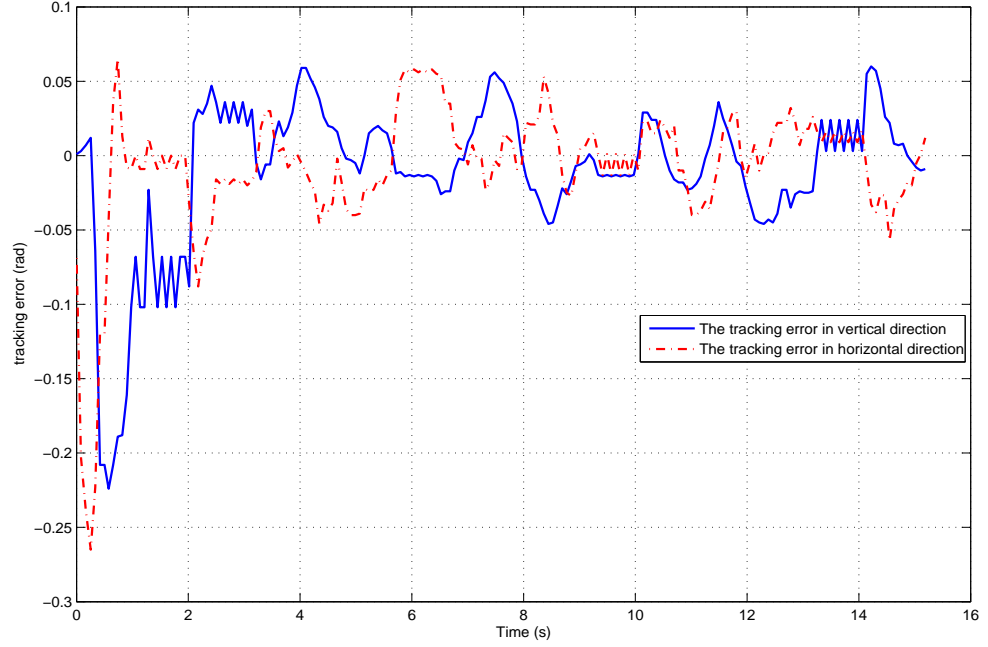


Figure 4.11: The tracking errors of the pan/tilt servo in vertical and horizontal directions.

to the surface of the earth at the origin of the NED frame. The coordinate X_n -, Y_n - and Z_n -axes of the NED frame are specified to point towards the north, east and down (towards into the earth, vertically to the surface of the Earth), respectively.

2. The world coordinate system (labeled with a subscript 'w') or the world reference frame, which is not shown in Figure 4.12, is the projection of the body frame of the aircraft on the ground.
3. The body coordinate system (labeled with a subscript 'b') is aligned with the shape of the fuselage of the aircraft. The X_b -axis points through the nose of the aircraft, and the Y_b -axis points to the right of the X_b -axis (facing in direction of view of the pilot), perpendicular to the X_b -axis. Finally, the Z_b -axis points down through the bottom of the craft, perpendicular to the X_b - Y_b plane.
4. The servo-base coordinate system (labeled with a subscript 's') is attached to the base

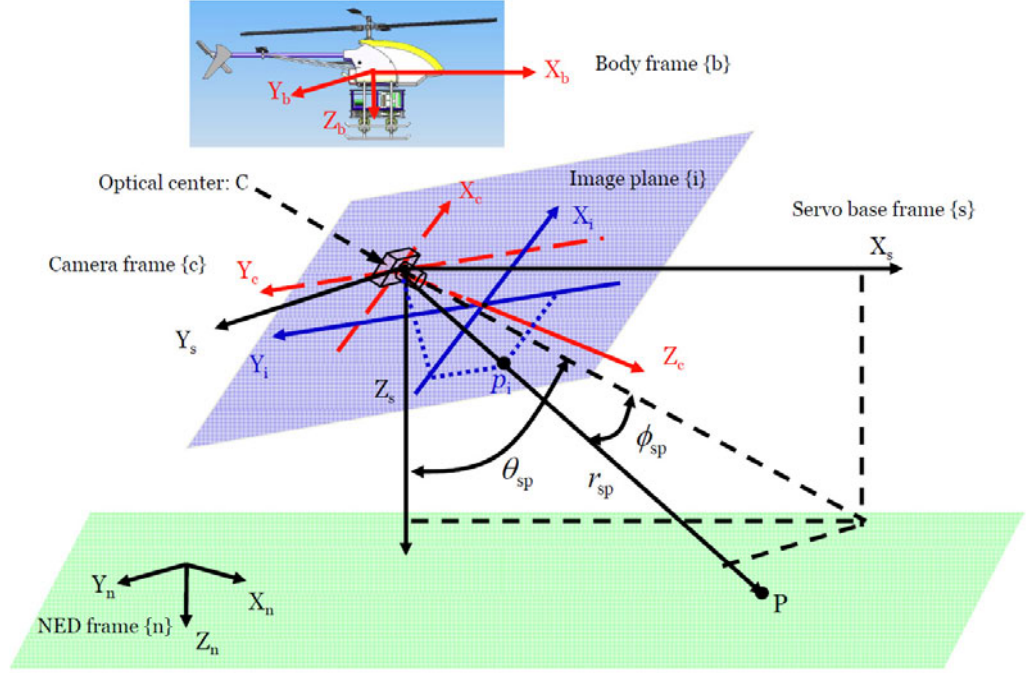


Figure 4.12: Coordinate systems.

of the pan/tilt servo mechanism, which is aligned with the body coordinate system of the UAV. The rotation between the origin of the servo-based coordinate system and the body coordinate system can be ignored, and the translation is fixed.

5. The spherical coordinate system (labeled with a subscript ‘sp’) is also attached to the base of the pan/tilt servo mechanism. It is used to define the orientation of the pan/tilt servo mechanism, and the target. Given a generic point P in the servo-base coordinate system, say

$$\mathbf{p}_s = \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} \quad (4.29)$$

its position can be defined in the spherical coordinate system by three numbers: radius

r_{sp} , azimuth angle θ_{sp} and elevation angle ϕ_{sp} , which is given by:

$$\mathbf{p}_{\text{sp}} = \begin{pmatrix} x_{\text{sp}} \\ \theta_{\text{sp}} \\ \phi_{\text{sp}} \end{pmatrix} \quad (4.30)$$

where

$$r_{\text{sp}} = \sqrt{x_s^2 + y_s^2 + z_s^2}, \quad \theta_{\text{sp}} = \tan^{-1} \frac{x_s}{z_s}, \quad \phi_{\text{sp}} = \sin^{-1} \frac{y_s}{r_{\text{sp}}} \quad (4.31)$$

6. The camera coordinate system (labeled with a subscript ‘c’) or the camera frame is to describe the orientation of the camera, which is attached to the end of the pan/tilt servo mechanism, with the pan/tilt rotation with the pan/tilt rotation with respect to the servo-base frame

$$\mathbf{x}_{\text{c,sp}} = \begin{pmatrix} \phi_{\text{c,sp}} \\ \theta_{\text{c,sp}} \end{pmatrix}. \quad (4.32)$$

The origin of the camera coordinate system is the optical center of the camera. The Z_{c} -axis is aligned with the optical axis of the camera and points from the optical center C towards the image plane. When the camera frame coincides with the servo-base frame, $\mathbf{x}_{\text{c,sp}} = 0$. Otherwise, $\mathbf{x}_{\text{c,sp}} = [\phi_{\text{c,sp}}, \theta_{\text{c,sp}}]^T$ is corresponding to the rotation of the camera frame first about the Y_{s} -axis by an angle of θ_{sp} , and then about the X_{s} -axis by an angle of ϕ_{sp} . This rotation sequence is defined in terms of the structure of the pan/tilt servo mechanism.

7. The object coordinate system (labeled with a subscript ‘t’) or the object frame is attached to the fixed ground landmark.
8. The image frame (or principle image coordinate system) (labeled with a subscript ‘i’) has the origin at the principal point. The coordinate axes, X_{i} and Y_{i} , are aligned with

the camera coordinate axes, X_c and Y_c , respectively.

4.4 Camera Calibration

This sections present the calibration of the camera adopt. As a measurement sensor, the camera is used to map the points in the 3D space to the projected points in the 2D image frame. Such mapping can be described using a suitable mathematic model with intrinsic and extrinsic parameters. The computation of the intrinsic and extrinsic parameters is called camera calibration. Calibration results strongly affect performance of a vision system.

In this section, we first introduce a camera model, and then explain how to estimate the parameters of the model. Finally, a compensation approach is utilized to cope with the distortion in the model.

4.4.1 Camera Model

The frontal pinhole projection [79] is a commonly used camera model, illustrated in Figure 4.13. It is simple and convenient by placing the image plane in front of the optical center. Given the generic point P , utilizing the model of the pinhole projection and geometric structure of the pan/tilt servo mechanism, the transformations among the camera coordinate system, the image coordinate system and the object coordinate system (see Figure 5.4) are given by

$$\lambda \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & s_\theta & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{c/t} & \mathbf{t}_{c/t} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{p}_t \\ 1 \end{pmatrix}, \quad (4.33)$$

or in the matrix format

$$\lambda \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \mathbf{K}_s \mathbf{\Pi}_0 \mathbf{G} \begin{pmatrix} \mathbf{p}_t \\ 1 \end{pmatrix} = \mathbf{K}_s \mathbf{\Pi}_0 \begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix}, \quad (4.34)$$

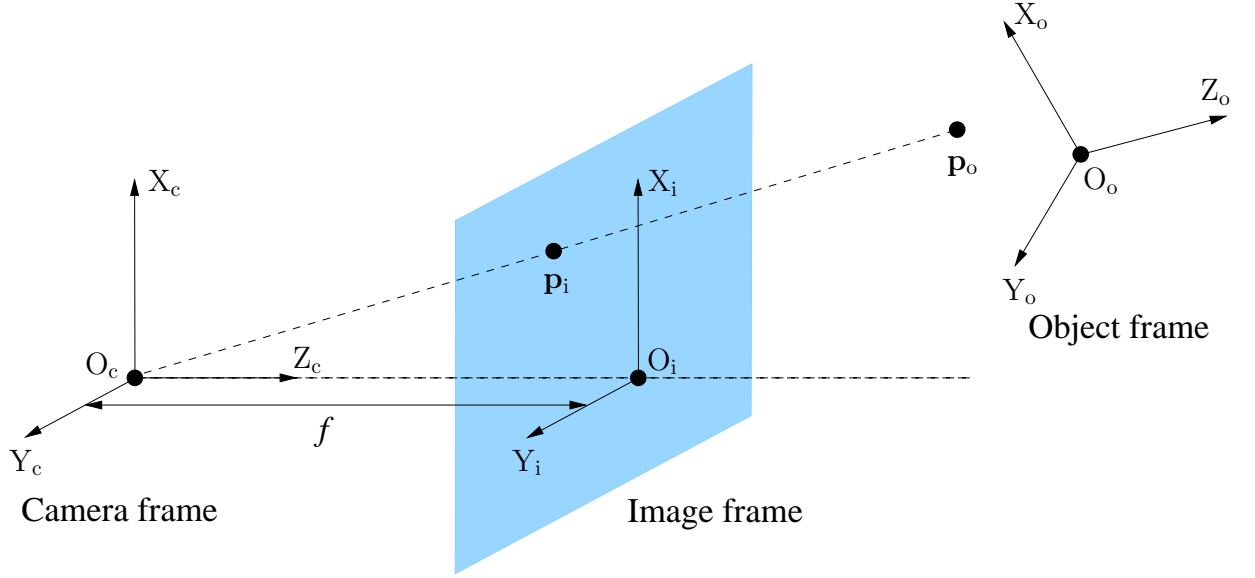


Figure 4.13: Frontal pin-hole camera model.

where

$$\mathbf{p}_t = \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix}, \quad \mathbf{p}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}, \quad \mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (4.35)$$

\mathbf{p}_t and \mathbf{p}_c are respectively the coordinates of \mathbf{P} with respect to the object frame and the camera frame. \mathbf{p}_i is the projection of \mathbf{P} in the image plane; $\mathbf{R}_{t/c}$ and $\mathbf{t}_{t/c}$ are respectively the rotation matrix and the translation vector, which define the rigid-body transformation from the object frame to the camera frame; λ is a scaling factor, which is the depth of the point \mathbf{P} in the camera coordinate system; s_θ is the so-called skew factor, which is very close to zero; $[o_x, o_y]^T$ are the coordinate of the principal point in pixels; and finally $f_x = f s_x$ and $f_y = f s_y$ are respectively the vertical and horizontal focal lengths measured in pixels (f is the focal length measured in metric units, and s_x and s_y are the numbers of pixel per unit length of the sensor along the x and y directions).

The pinhole projection mode in (4.33) is an approximation of the real image formation process. However, this model is not accurate enough in the demanding situations, since

the image distortion introduced by optical lenses always severely affects the accuracy of the camera model. So a more precise pinhole projection model, extended by using radial and slight tangential distortion [54], is given as

$$\left\{ \begin{array}{l} \begin{pmatrix} \mathbf{p}_p \\ 1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \frac{1}{\lambda} \mathbf{\Pi}_0 \begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix} = \frac{1}{\lambda} \mathbf{\Pi}_0 \mathbf{G} \begin{pmatrix} \mathbf{p}_t \\ 1 \end{pmatrix} \\ \mathbf{p}_d = (1 + k_1 d^2 + k_2 d^4 + k_5 d^6) \mathbf{p}_p + \mathbf{K}_d \\ d = \sqrt{x_p^2 + y_p^2} \\ \mathbf{K}_d = \begin{pmatrix} 2k_3 x_p y_p + k_4 (d^2 + 2x_p^2) \\ k_3 (d^2 + 2y_p^2) + 2k_4 x_p y_p \end{pmatrix} \\ \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \mathbf{K}_s \begin{pmatrix} \mathbf{p}_d \\ 1 \end{pmatrix} \end{array} \right. \quad (4.36)$$

where \mathbf{p}_p is the normalized projected image point, and \mathbf{p}_d is the normalized distorted image point; k_1, k_2, k_5 are radial distortion coefficients, k_3, k_4 are tangential distortion coefficients.

The intrinsic parameters f_x, f_y, o_x, o_y and the distortion coefficients $k_1 \dots k_5$ are referred to as the physical camera parameters, and they have a explicit physical meaning. The objective of the calibration is to search for the optimal values of these parameters in terms of the image observations of known feature points in 3D world.

4.4.2 Intrinsic Parameter Estimation

Based on the camera model given in (4.36), the camera calibration toolbox proposed in [134] is used to estimate the intrinsic parameters. The procedure of the calibration includes three steps as follows:

1. Capturing images of chessboard patten using the on-board camera with the varying rotation and the translation of the chessboard with respect to the camera. The calibration images are shown in Figure 4.14;

Table 4.3: Estimated intrinsic parameters of the on-board camera

Focal Length $\begin{pmatrix} f_x \\ f_y \end{pmatrix}$:	$\begin{pmatrix} 605.67766 \\ 661.84569 \end{pmatrix} \pm \begin{pmatrix} 2.65276 \\ 2.91701 \end{pmatrix}$
Principal point $\begin{pmatrix} o_x \\ o_y \end{pmatrix}$:	$\begin{pmatrix} 176.26743 \\ 141.21784 \end{pmatrix} \pm \begin{pmatrix} 3.33796 \\ 3.18607 \end{pmatrix}$
Skew s_θ :	0.00000 ± 0.00000
Distortion $k_i, i = 1 \dots 5$:	$\begin{pmatrix} -0.42687 \\ 0.50745 \\ 0.00349 \\ -0.00146 \\ 0.00000 \end{pmatrix} \pm \begin{pmatrix} 0.02702 \\ 0.34637 \\ 0.00097 \\ 0.00123 \\ 0.00000 \end{pmatrix}$

2. Extracting grid corners of the chessboard pattern using the Calibration Toolbox in Matlab. The extracted corners are illustrated in Figure 4.15;
3. Calculating the intrinsic parameters of the camera using the Calibration Toolbox in Matlab, which employs a two-step optimization approach to minimize the total re-projection error of all calibration parameters [130]. First, the direct linear transformation (DLT) algorithm is used to obtain a linear solution. Then, using this solution as the initial values, an optimal solution can be achieved by an iterative searching. The calibration parameters are comprised of 9 DOF intrinsic parameter and $n \times 6$ DOF extrinsic parameters (n is the number of the images). For the detailed description of the camera calibration using the Calibration Toolbox, please refer to [134].

The final calibration results of 9 intrinsic parameters of the on-board camera after optimization (with uncertainties) is described in Table 4.3.

4.4.3 Distortion Compensation

The distorted image will affect the performance the vision system. Thus, the original images will be corrected based on the known intrinsic parameters of the camera. Since the equation in (4.36) is not analytical solvable from distorted image to a pinhole projection, the iteration



Figure 4.14: Images for camera calibration.

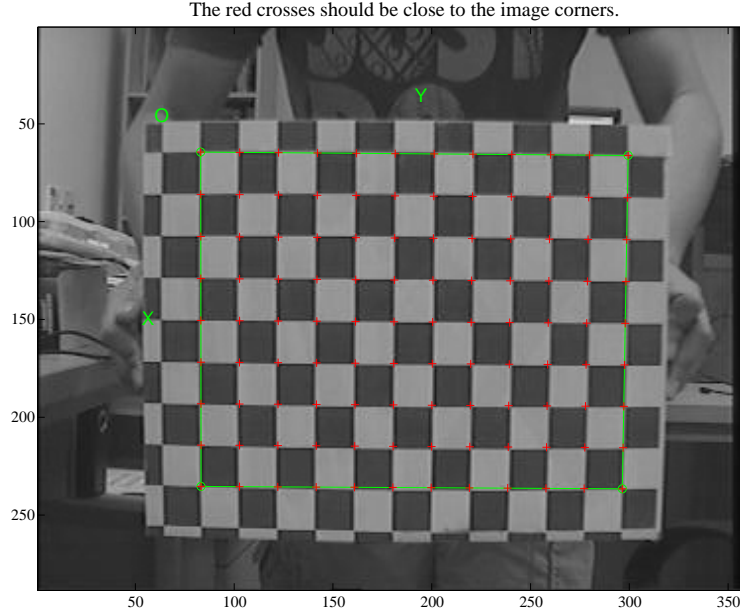
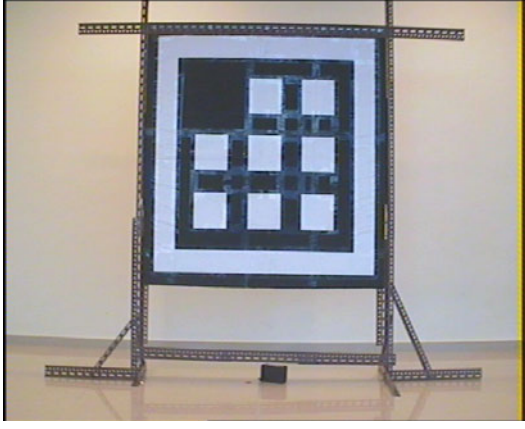


Figure 4.15: Grid corner extraction for camera calibration.
(The red crosses indicate the extracted corners)

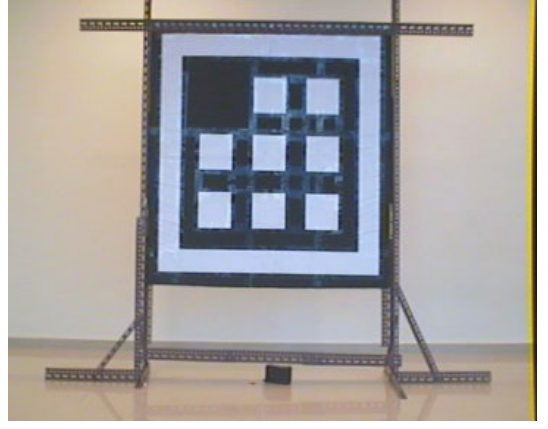
is required. To speed up the processing in the real-time application, the solution for each pixel is pre-calculated and saved to a look-up table. The experiments of the distortion compensation were conducted using the intrinsic parameters of the camera in Table 4.3. The comparison of the images before and after the correction is shown in Figure 4.16, in which it can be found that the corrected one has less distortion than the uncorrected one.

4.4.4 Simplified Camera Model

In the image processing part of our algorithm, the distortion of the lens is compensated, and the origin of the image plane is set to the principle point. Thus, o_x and o_y can be set to zero. Moreover, since $s_\theta = 0$, we can obtain a simplified pinhole projection model based on (4.34), which is given as



a. The original image



b. The image after distortion compensation

Figure 4.16: Distortion compensation.

$$\lambda \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \mathbf{K}_f \mathbf{\Pi}_0 \begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix}, \quad (4.37)$$

where

$$\lambda = z_c, \quad \mathbf{K}_f = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.38)$$

This simplified pinhole camera model is adopted throughout the rest of this chapter. Following from (4.37), the representation of \mathbf{p}_c in terms of the location of P in the image plane is given by

$$\mathbf{p}_c = \lambda \begin{pmatrix} \bar{x}_i \\ \bar{y}_i \\ 1 \end{pmatrix}, \quad (4.39)$$

where

$$\bar{x}_i = \frac{x_i}{f_x}, \quad \bar{y}_i = \frac{y_i}{f_y} \quad . \quad (4.40)$$

In the rest of this thesis, this simplified pinhole project model is employed. In addition, since s_θ is very close to zero, Equation (4.34) can be re-written and yield:

$$\begin{cases} \frac{x_i - o_x}{f_x} = \frac{x_c}{\lambda} = \frac{x_c}{z_c} \\ \frac{y_i - o_y}{f_y} = \frac{y_c}{\lambda} = \frac{y_c}{z_c} \end{cases} \quad (4.41)$$

The transformation between the camera coordinate system and the object coordinate system is given by:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \mathbf{R}_{c/t} \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} + \mathbf{t}_{c/t} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (4.42)$$

which together with (4.41), and yields the collinearity equation, which is given by:

$$\begin{cases} \frac{x_i - o_x}{f_x} = \frac{r_0 x_t + r_1 y_t + r_2 z_t + t_x}{r_6 x_t + r_7 y_t + r_8 z_t + t_z} \\ \frac{y_i - o_y}{f_y} = \frac{r_3 x_t + r_4 y_t + r_5 z_t + t_y}{r_6 x_t + r_7 y_t + r_8 z_t + t_z} \end{cases} \quad (4.43)$$

The collinearity equation is popularly used in many vision-based applications, such as the pose estimation.

4.5 Target Following Control

This section presents the design of a comprehensive target following control. It consists of two main layers, the pan/tilt servo mechanism control and the UAV following control. The overall structure of the target following control is depicted in Figure 4.17. As mentioned earlier, a pan/tilt servo mechanism is employed in the first layer to control the orientation

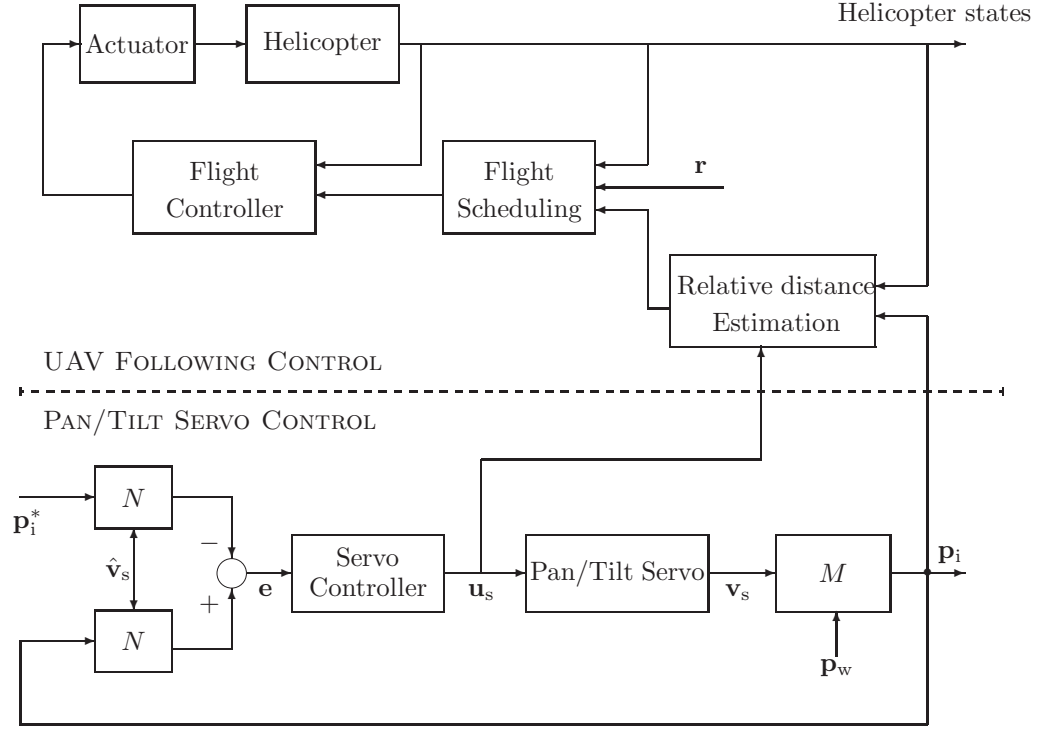


Figure 4.17: Block diagram of the tracking control scheme.

of the camera to keep the target in an optimal location in the image plane, which makes target tracking in the video sequence more robust and efficient. The parameters associated with the pan/tilt servo control in Figure 4.17 are to be introduced in detail later. In the second layer, the UAV is controlled to maintain the constant relative distance between the moving target and the UAV in flight.

4.5.1 Control of the Pan/Tilt Servo Mechanism

The purpose of the control of the pan/tilt servo mechanism is to adjust the orientation of the on-board camera to keep the target in an optimal view (for example, make the optical axis aligned with the line-of-sight vector of the target) in terms of the vision feedback, namely eye-in-hand visual servoing [20, 21]. A robust closed-loop tracking control scheme for the pan/tilt servo mechanism is designed using the vision information.

As depicted in Figure 4.17, given a generic point P , \mathbf{p}_i and \mathbf{p}_i^* are the measured and desired locations of the projected point of P in the image plane, respectively. To make the vision-based detection more robust, $\mathbf{p}_i^* = [x_i^*, y_i^*]^T$ is set as the center of the image frame, which in general is not necessarily the principle point of the camera. $\mathbf{e} = [e_\phi, e_\theta]^T$ is the tracking error, $\mathbf{u}_s = [u_\phi, u_\theta]^T$ is the output of the tracking controller, $\mathbf{v}_s = [v_\phi, v_\theta]^T$ is the output of the pan/tilt servo mechanism. M is a nonlinear mapping between the coordinates of the point P in the world frame and in the image plane under the current \mathbf{v}_s . N is a nonlinear mapping between the location of the point P in the image plane and its orientation with respect to the UAV under the current \mathbf{v}_s . As mentioned in the definitions of the coordinate systems, the orientation of P with respect to the UAV can be defined using azimuth and elevation angles in the spherical coordinate system, which is described by two rotation angles $\mathbf{p}_e = [p_\phi, p_\theta]^T$.

Nonlinear Mappings M and N

Given the generic point P , recall (4.37), the transformations among the camera coordinate system, the image coordinate system and the world coordinate system (see Figure 5.4) are given by

$$\begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix} = \frac{1}{\lambda} \mathbf{K}_f \mathbf{\Pi}_0 \begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix}, \quad (4.44)$$

and

$$\begin{pmatrix} \mathbf{p}_w \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R}_{w/c} & \mathbf{t}_{w/c} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix}, \quad (4.45)$$

where $\mathbf{R}_{w/c}$ and $\mathbf{t}_{w/c}$ are respectively the rotation matrix and the translation vector, which define the rigid-body transformation from the camera frame to the world frame, and which

are to be described in detail in Section 4.5.2. Thus, M is defined as

$$\mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \mathbf{M}(\mathbf{p}_w, \mathbf{v}_s) = \frac{1}{\lambda} \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\mathbf{R}_{w/c}^{-1} \mathbf{p}_w - \mathbf{R}_{w/c}^{-1} \mathbf{t}_{w/c} \right).$$

Next, to derive the nonlinear mapping N , the transformation between the camera coordinate system and the servo-base coordinate system is defined as

$$\begin{pmatrix} \mathbf{p}_c \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R}_{c/s}(\mathbf{v}_s) & \mathbf{t}_{c/s} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{p}_s \\ 1 \end{pmatrix}, \quad (4.46)$$

where

$$\mathbf{R}_{c/s}^{-1}(\mathbf{v}_s) = \mathbf{R}_{s/c}(\mathbf{v}_s) = \begin{bmatrix} \cos v_\theta & 0 & \sin v_\theta \\ 0 & 1 & 0 \\ -\sin v_\theta & 0 & \cos v_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos v_\phi & -\sin v_\phi \\ 0 & \sin v_\phi & \cos v_\phi \end{bmatrix}, \quad (4.47)$$

\mathbf{p}_s is the coordinate of the point P with respect to the servo-base coordinate system; $(\mathbf{R}_{c/s}, \mathbf{t}_{c/s})$ describes the rigid-body transformations from the servo-base frame to the camera frame. Since only the rotation of the pan/tilt servo mechanism is considered in the above transformations, $\mathbf{t}_{c/s}$ is equal to zero, we can then simplify (4.46) as

$$\mathbf{p}_c = \mathbf{R}_{c/s}(\mathbf{v}_s) \mathbf{p}_s. \quad (4.48)$$

It follows from (4.34) with all the necessary assumptions that

$$\lambda \begin{pmatrix} \frac{x_i}{f_x} \\ \frac{y_i}{f_y} \\ 1 \end{pmatrix} = \mathbf{p}_c, \quad (4.49)$$

which together with (4.48) and the fact that $\mathbf{R}_{\text{sc}}(\mathbf{v}_s) = \mathbf{R}_{\text{cs}}^{-1}(\mathbf{v}_s)$ imply

$$\lambda \mathbf{R}_{\text{s/c}}(\mathbf{v}_s) \begin{pmatrix} \frac{x_i}{f_x} \\ \frac{y_i}{f_y} \\ 1 \end{pmatrix} = \mathbf{p}_s. \quad (4.50)$$

We then define

$$\bar{\mathbf{p}}_s = \begin{pmatrix} \bar{x}_s \\ \bar{y}_s \\ \bar{z}_s \end{pmatrix} = \mathbf{R}_{\text{cs}}(\mathbf{v}_s) \begin{pmatrix} \frac{x_i}{f_x} \\ \frac{y_i}{f_y} \\ 1 \end{pmatrix}, \quad (4.51)$$

which together with (4.50) yield

$$\lambda \bar{\mathbf{p}}_s = \mathbf{p}_s. \quad (4.52)$$

Recall the definitions of azimuth and elevation angles in (4.30), the nonlinear mapping N is then given as

$$\mathbf{p}_e = \begin{pmatrix} p_\phi \\ p_\theta \end{pmatrix} = N(\mathbf{p}_i, \mathbf{v}_s) = \begin{pmatrix} \sin^{-1} \frac{y_s}{r_{\text{sp}}} \\ \tan^{-1} \frac{x_s}{z_s} \end{pmatrix} = \begin{pmatrix} \sin^{-1} \frac{\bar{y}_s}{\bar{r}_{\text{sp}}} \\ \tan^{-1} \frac{\bar{x}_s}{\bar{z}_s} \end{pmatrix}, \quad (4.53)$$

where

$$\bar{r}_{\text{sp}} = \sqrt{\bar{x}_s^2 + \bar{y}_s^2 + \bar{z}_s^2} \quad (4.54)$$

Pan/Tilt Servo Control

The pan/tilt servo mechanism, shown in Figure 2.10, can be approximately considered as two decoupled servo motors, which regulate the visual sensor for horizontal and vertical rotation, respectively. The dynamic model of the servo motor can be described by using

a standard second order system. To identify the parameters of the model, we inject step signals with different values to the pan/tilt servo mechanism. The parameters of the models of the vertical and horizontal servos are given in Table 4.4. Before proceeding to design the control law for the pan/tilt servo mechanism, the tracking error function is defined as

$$\mathbf{e}(k) = \begin{pmatrix} e_\phi \\ e_\theta \end{pmatrix} = \mathbf{p}_e - \mathbf{p}_e^* = N(\mathbf{p}_i(k), \mathbf{v}_s(k)) - N(\mathbf{p}_i^*, \mathbf{v}_s(k)), \quad (4.55)$$

where \mathbf{p}_e and \mathbf{p}_e^* denote the light-of-sight angles of the target and the center of the image plane relative to the UAV respectively. The control inputs will be sent to the pan/tilt servos after the vision-based target detection algorithm, which generally cost about one sampling period. To track the moving target efficiently, the pan/tilt servo control inputs are calculated using the predicted location of the target in the subsequent frame, which is derived from (4.27) and given by

$$\hat{\mathbf{p}}_i(k+1) = \begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} = \hat{\mathbf{z}}(k+1|k) = \mathbf{H}\hat{\mathbf{x}}(k+1|k). \quad (4.56)$$

In the implementation, it is not easy to measure the output of the pan/tilt servo \mathbf{v}_s in (4.55). It is assumed that the bandwidth of the pan/tilt servo mechanism is much faster than that of the control system. Then, the transient of the pan/tilt servos can be ignored, and considered as scaling factors with one step delay. The estimate of \mathbf{v}_s is defined as

$$\hat{\mathbf{v}}_s(k) = \mathbf{K}_d \mathbf{u}_s(k-1). \quad (4.57)$$

Replacing \mathbf{v}_s and \mathbf{p}_i with $\hat{\mathbf{v}}_s$ and $\hat{\mathbf{p}}_i$ in (4.55), the modified error function yields

$$\mathbf{e}(k) = N(\hat{\mathbf{p}}_i(k+1), \hat{\mathbf{v}}_s(k)) - N(\mathbf{p}_i^*, \hat{\mathbf{v}}_s(k)). \quad (4.58)$$

The purpose of the design of the tracking control law is to minimize the tracking error function given in (4.58) by choosing a suitable control input $\mathbf{u}_s(k)$. Since the dynamics model

Table 4.4: Parameters of the pan/tilt servos

Parameters	Tilt servo	Pan servo
DC gain: K_d	1.1198	1.2945
Damping ratio: ζ	0.8143	0.8814
Natural frequency: ω_n	123.2525	130.676

of the pan/tilt servos is relatively simple, a discrete-time proportional-integral controller (see, for example, [42]) is employed, which is structurally simple but fairly robust. It is very suitable for our real-time application. The incremental implementation of the PI controller is given by

$$\Delta \mathbf{u}_s(k) = K_p [\mathbf{e}(k) - \mathbf{e}(k-1)] + \frac{K_p T_s}{T_i} \mathbf{e}(k),$$

where the proportional gain and the integral time are chosen as $K_p = 0.65$ and $T_i = 0.8$, respectively. It is noted that two identical controllers are respectively used for the pan and tilt servos, since the dynamics of the two servos are very close.

4.5.2 Following Control of the UAV

As illustrated in Figure 4.17, to realize the UAV following control, a geometric approach is employed to estimate the relative distance, which uses the measured target location in the image plane and the pose of the UAV based on the flat ground assumption.

As illustrated in Figure 5.4, to estimate the relative distance between the target and the UAV, the transformation in (4.45) is recalled, and simplified as

$$\mathbf{p}_w = \mathbf{R}_{w/c} \mathbf{p}_c + \mathbf{t}_{w/c}, \quad (4.59)$$

which together with (4.49) and the fact $z_c = \lambda$, and generates the overall geometric model

from an ideal image to world frame:

$$\mathbf{p}_w = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \mathbf{R}_{w/c} \begin{pmatrix} \frac{x_i}{f_x} \\ \frac{y_i}{f_y} \\ 1 \end{pmatrix} z_c + \mathbf{t}_{w/c}. \quad (4.60)$$

It is assumed that the ground is flat, and the altitude of the helicopter to the ground: h is known. In addition, X_c - and Y_c -axis translations from camera frame to world frame are smaller compared to the height and can be ignored; and X_b - and Y_b -axis rotations from the body frame to the world frame is smaller compared to the rotation of the pan/tilt servos and can be ignored too.

$$\mathbf{R}_{w/c} = \mathbf{R}_{s/c}(\mathbf{v}_s) = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \quad \mathbf{t}_{w/c} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix}, \quad (4.61)$$

which together with (4.60) yield

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{pmatrix} \frac{x_i}{f_x} \\ \frac{y_i}{f_y} \\ 1 \end{pmatrix} z_c + \begin{pmatrix} 0 \\ 0 \\ -h \end{pmatrix}. \quad (4.62)$$

Based on the assumption that the target is on the ground, z_w is equal to zero. We can then rewrite the last row in (4.62) and derive z_c as

$$z_w = \left(r_7 \frac{x_i}{f_x} + r_8 \frac{y_i}{f_y} + r_9 \right) z_c - h = 0, \quad (4.63)$$

$$z_c = \frac{h}{r_7 \frac{x_i}{f_x} + r_8 \frac{y_i}{f_y} + r_9}, \quad (4.64)$$

which together with (4.62) yield

$$\begin{pmatrix} x_{\text{tg}} \\ y_{\text{tg}} \\ z_{\text{tg}} - h \end{pmatrix}_{\text{b}} = \begin{pmatrix} x_{\text{w}} \\ y_{\text{w}} \\ z_{\text{w}} \end{pmatrix} = \begin{pmatrix} h \frac{r_1 x_{\text{i}} f_y + r_2 y_{\text{i}} f_x + r_3 f_x f_y}{r_7 x_{\text{i}} f_y + r_8 y_{\text{i}} f_x + r_9 f_x f_y} \\ h \frac{r_4 x_{\text{i}} f_y + r_5 y_{\text{i}} f_x + r_6 f_x f_y}{r_7 x_{\text{i}} f_y + r_8 y_{\text{i}} f_x + r_9 f_x f_y} \\ 0 \end{pmatrix}, \quad (4.65)$$

where $[x_{\text{tg}} \ y_{\text{tg}} \ z_{\text{tg}}]_{\text{b}}^{\text{T}}$ is the coordinate of the target in the body frame. Due to the same reasons mentioned in Section 4.5.1, we replace \mathbf{v}_{s} and \mathbf{p}_{i} with $\hat{\mathbf{v}}_{\text{s}}$ and $\hat{\mathbf{p}}_{\text{i}}$, and rewrite (4.65) as

$$\begin{pmatrix} x_{\text{tg}} \\ y_{\text{tg}} \\ z_{\text{tg}} - h \end{pmatrix}_{\text{b}} = \begin{pmatrix} x_{\text{w}} \\ y_{\text{w}} \\ z_{\text{w}} \end{pmatrix} = \begin{pmatrix} h \frac{\hat{r}_1 \hat{x}_{\text{i}} f_y + \hat{r}_2 \hat{y}_{\text{i}} f_x + \hat{r}_3 f_x f_y}{\hat{r}_7 \hat{x}_{\text{i}} f_y + \hat{r}_8 \hat{y}_{\text{i}} f_x + \hat{r}_9 f_x f_y} \\ h \frac{\hat{r}_4 \hat{x}_{\text{i}} f_y + \hat{r}_5 \hat{y}_{\text{i}} f_x + \hat{r}_6 f_x f_y}{\hat{r}_7 \hat{x}_{\text{i}} f_y + \hat{r}_8 \hat{y}_{\text{i}} f_x + \hat{r}_9 f_x f_y} \\ 0 \end{pmatrix},$$

where

$$\mathbf{R}_{\text{cs}}(\mathbf{v}_{\text{s}}) \doteq \mathbf{R}_{\text{cs}}(\hat{\mathbf{v}}_{\text{s}}) = \begin{bmatrix} \hat{r}_1 & \hat{r}_2 & \hat{r}_3 \\ \hat{r}_4 & \hat{r}_5 & \hat{r}_6 \\ \hat{r}_7 & \hat{r}_8 & \hat{r}_9 \end{bmatrix}.$$

As shown in Figure 4.17, the relative distance between the target and the UAV is estimated, which is employed as the reference signal to guide the UAV to follow the motion of the target. The tracking reference for the UAV is defined as

$$\begin{pmatrix} x_{\text{uav}} \\ y_{\text{uav}} \\ z_{\text{uav}} \\ \psi_{\text{uav}} \end{pmatrix}_{\text{ref}} = \begin{pmatrix} \begin{pmatrix} x_{\text{tg}} \\ y_{\text{tg}} \end{pmatrix}_{\text{n}} - [e_1, e_2]^{\text{T}} \mathbf{R}_{\text{b/n}}^{\text{T}} \begin{pmatrix} c_{\text{x}} \\ c_{\text{y}} \\ 0 \end{pmatrix} \\ h_0 \\ \psi_0 \end{pmatrix} = \begin{pmatrix} [e_1, e_2]^{\text{T}} \mathbf{R}_{\text{b/n}}^{\text{T}} \left(\begin{pmatrix} x_{\text{tg}} \\ y_{\text{tg}} \\ 0 \end{pmatrix}_{\text{b}} - \begin{pmatrix} c_{\text{x}} \\ c_{\text{y}} \\ 0 \end{pmatrix} \right) \\ h_0 \\ \psi_0 \end{pmatrix}.$$

where c_{x} and c_{y} are the desired relative distance between the target and the UAV in the X_{b} - and Y_{b} -axis respectively; h_0 is the pre-defined height of the UAV above the ground; ψ_0

is the predefined heading angle of the UAV; \mathbf{R}_{nb} is the rotation matrix from body frame to NED frame, which can be calculated in terms of the output of the on-board navigation sensors.

4.6 Experimental Results

The images of the target were captured by the on-board camera before the tests, and analyzed off-line to estimate the feature model of the target using the proposed algorithm in Section 4.2 and 4.5, to realize the automatic target detection in flight.

To verify the proposed vision system, multiple tests of the complete system were conducted. During these tests, SheLion was hovering autonomously at a certain position. If the moving target entered into the view of the on-board camera, the target would be identified and tracked in the video sequence by the vision system automatically. Based on the vision information, the pan/tilt servo mechanism was controlled to keep the target in a certain position in the image as described in Section 4.5.1. The operator can then command the UAV to enter into the target following mode, in which the UAV followed the motion of the target autonomously based on the estimated relative distance, using the algorithm proposed in Section 4.5.2.

The experimental results of the vision-based target detection and tracking in flight are shown in Table 4.5, which indicate that the proposed vision algorithm could effectively identify and track the target in the video sequence in the presence of unknown motion disturbance between the UAV platform and the target. One example of the pan/tilt servo tracking control in flight is also shown in Figure 4.19. The solid line in Figure 4.19 indicates the expected position of the target in the image, and the dash line indicates the actual location of the target in image during the flight test. It can be observed from Figure 4.19 that in spite of the unknown motion between the UAV and the target, the pan/tilt servo mechanism can effectively control target in a box-like neighborhood of the center point of the image by employing vision-based pan/tilt servo control.



Figure 4.18: The demo of the vision-based target following.

Table 4.5: Experiment results of target detection and tracking in flight

Test	Total time (s)	Total frames	The target detected frames	Accuracy
1	101.8	761	728	95.66%
2	77.2	591	518	87.65%
3	50.4	388	382	98.45%
4	75.3	572	501	87.59%
5	86.4	662	645	97.43%

One example of the ground target following is described in Figure 4.20 and 4.21, in which the target was manually controlled to move randomly on the flat ground and the UAV followed the motion of the target automatically based on the scheme proposed in Section 4.5.2. It was observed from Figure 4.20 that the UAV can follow the trajectory of the target and keep the constant relative distance between the UAV and the target. The experimental results of moving ground target following of the UAV indicate the efficiency and robustness of the presented vision-based following scheme. The videos of the vision-based target following test can be downloaded from the NUS UAV Research website [135].

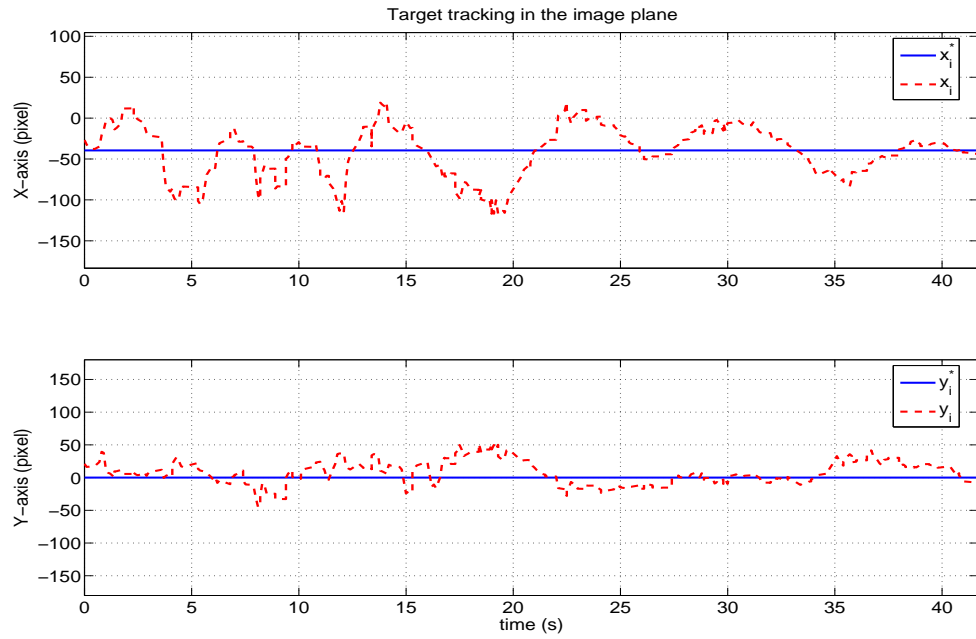


Figure 4.19: The test result of the vision-based servo control.

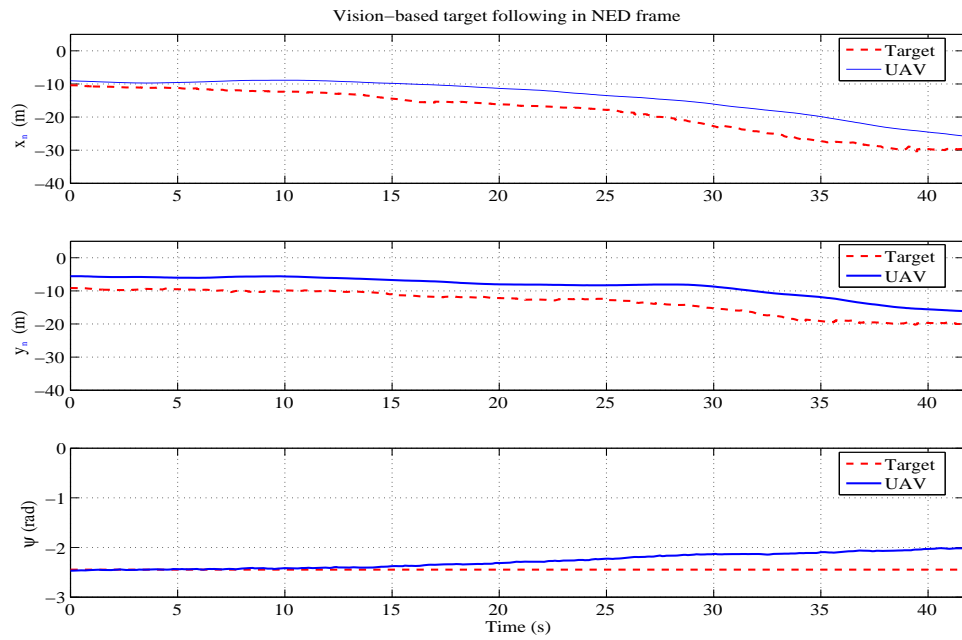


Figure 4.20: The test result of the vision-based target following.

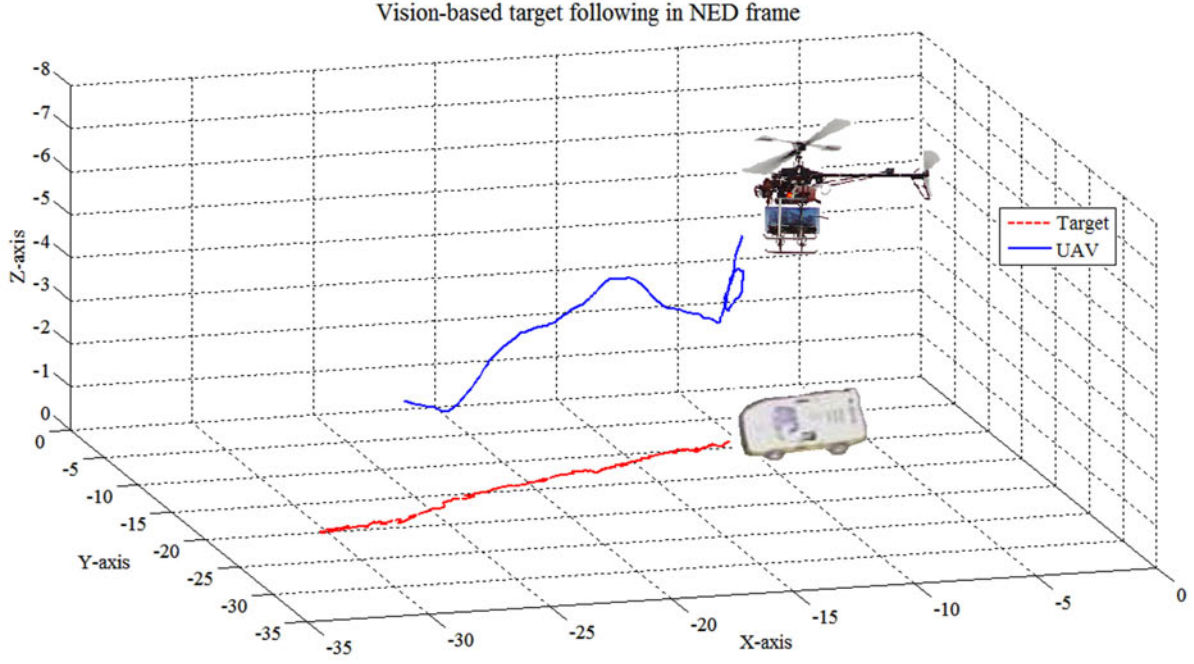


Figure 4.21: The test result of the vision-based target following in 3D.

4.7 Conclusion

In this chapter, the comprehensive design and implementation of a vision system for an UAV has been presented. Multiple flight tests were conducted to verify the proposed vision system. The experimental results show that this vision system is not only able to automatically detect and track the pre-defined ground target in the video sequence, but also able to guide the UAV to follow the motion of the target in flight. The robustness and efficiency of the developed vision system for UAVs could be achieved by the current system. Based on the presented work, vision-based motion estimation for a maneuvering target is currently focused, which will be integrated with the flight control system of the UAV. With such improvement, a better vision-based detection, tracking and following system for the maneuvering target can be achieved.

Appendix: Proof of Moment Invariants Remain Unchanged under Scaling

Theorem 4.7.1. *Suppose a shape has smooth boundary C , and this shape has been homogeneously rescaled by a factor r , then we get new boundary C' . Then*

$$(\eta_{pq}^m)' = \eta_{pq}^m \quad (4.66)$$

where

$$\begin{aligned} (\eta_{pq}^m)' &= \frac{(\mu_{pq}^c)'}{(A')^{(p+q+1)/2}} \\ \eta_{pq}^m &= \frac{\mu_{pq}^c}{A^{(p+q+1)/2}} \end{aligned}$$

and where A is the area of the original object, and A' is the area of the rescaled object.

Proof. The lines of reasoning in the following proof follow from ref. [24]. For any $r > 0$, we can get the relationship between $(\mu_{pq}^c)'$ and μ_{pq}^c , which is given by

$$\begin{aligned} (\mu_{pq}^c)' &= \int_{C'} [x(s')]^p [y(s')]^q ds' \\ &= \int_{C'} [rx(s)]^p [ry(s)]^q d(rs) \\ &= r^{p+q+1} \mu_{pq}^c \end{aligned}$$

Note that the relationship between area of the original object and area of the rescaled object is given by

$$A' = r^2 A \quad (4.67)$$

Thus, for any $r > 0$, we have

$$(\eta_{pq}^m)' = \frac{(\mu_{pq}^c)'}{(A')^{(p+q+1)/2}} = \frac{r^{p+q+1} \mu_{pq}^c}{(r^2 A)^{(p+q+1)/2}}$$

$$= \frac{\mu_{pq}^c}{A^{(p+q+1)/2}} = \eta_{pq}^m$$

which shows that η_{pq}^c are indeed invariant with respect to homogeneous scaling.

Proof of Moment Invariants Remain Unchanged under Rotation

Theorem 4.7.2. *Suppose C is a smooth boundary curve of the original object and C' is the boundary curve obtained by rotating the object an angle θ , then*

$$\phi'_i = \phi_i, \quad 1 \leq i \leq 4$$

where ϕ_i and ϕ'_i is defined in eq. (4.9) to (4.12) calculated by using η_{pq}^m and $(\eta_{pq}^m)'$ for $p + q = 2, 3, \dots$

Proof. Note that

$$\begin{aligned} (\mu_{pq}^c)' &= \int_{C'} [x(s')]^p [y(s')]^q ds' \\ &= \int_C [x(s) \cos \theta - y(s) \sin \theta]^p \\ &\quad [y(s) \sin \theta + x(s) \cos \theta]^q ds \end{aligned}$$

Since $ds' = ds$, and the area of the object does not change under the rotation, i.e., $A' = A$, from eq. (4.9), we have

$$\begin{aligned} (A')^{1.5} \phi'_1 &= (A')^{1.5} ((\eta_{20}^m)' + (\eta_{02}^m)') \\ &= (\mu_{20}^c)' + (\mu_{02}^c)' \\ &= \int_C [x(s) \cos \theta - y(s) \sin \theta]^2 \end{aligned}$$

$$\begin{aligned}
& +[x(s) \sin \theta + y(s) \cos \theta]^2 ds \\
= & \int_C [\cos^2 \theta + \sin^2 \theta] [x(s)]^2 \\
& + [\cos^2 \theta + \sin^2 \theta] [y(s)]^2 ds \\
= & \int_C [x(s)]^2 + [y(s)]^2 ds \\
= & \mu_{20}^c + \mu_{02}^c = (A)^{1.5} (\eta_{20}^m + \eta_{02}^m) = (A')^{1.5} \phi_1
\end{aligned}$$

Therefore

$$\phi_1' = \phi_1$$

Similarly, the same can be proven for $\phi_i' = \phi_i$, for $i = 2, \dots, 4$.

Chapter 5

Vision-Based Flight Control for the UAV

5.1 Introduction

Despite the rapid progress of UAVs in the academic research and industrial applications [48, 63, 65, 56], the navigation systems for unmanned aerial vehicles still pose significant challenge. Most of them are constructed based on the scheme of INS/GPS.

Generally, GPS receivers are applied to implement global autonomous navigation. However, in certain applications and situations, GPS-based systems are not reliable to sustain autonomous navigation for unmanned systems such as UAVs due to the risk of the loss of GPS signals. They may degrade dramatically due to the lost of the GPS signal. Moreover, GPS-based systems may not be applicable to local accurate navigation because the positioning accuracy of GPS receivers is too large compared to the small measured fields.

To enhance the performance of navigation systems in such conditions, the vision augmented system become a necessary and promising solution. The core issues of such system are the vision aided motion estimation techniques, which are capable of recovering position and velocity of a UAV with respect to local reference.

According to the knowledge in computer vision, the most straightforward approach of range measurement is the stereo vision technique. Such work has already been reported

in [103]. A stereo vision technology was employed to augment a traditional sensor system for a UAV. However, fixed base line of a stereo camera constrains the measurement range, and the computational cost of processing stereo images also limits its usages in applications of UAVs with limited payload and space.

Optic-flow techniques are also widely used for the motion estimation. Such a technique was presented in [56] to navigate a UAV through urban canyons. Both the optic-flow approach and stereo vision technique were employed to hold the UAV in the center of the canyons safely, and avoid obstacles detected. Although the optical flow method is suitable for the motion estimation of UAVs in the forward flight condition, it cannot estimate the absolute position, which is required in applications, such as the drift-free hover.

To obtain the absolute position of the UAV with respect to the local environment, approaches based on model-to-data correspondence has been explored in numerous applications of the vision-based navigation, especially the autonomous landing of UAVs. Vision-based landing system was presented in [109, 107] to implement landing unmanned helicopters. A differential ego-motion estimation approach in [107] was employed to observe the states of the UAV with the known initial values. The multiple view geometry technique was used in [108] to obtain accurate estimation of the real position of the UAV. However, in this application, the vision information cannot be effectively used to compute the velocity of the carrier.

In addition, the classical visual servo control approach was applied in [48] to the under-actuated system, the rotorcraft to estimate its motion and stabilize it. The error function for the visual servoing was defined in the image frame, which is not easy to be used for the classical control systems. As we know, the application of visual information in feedback control becomes one of challenges in under-actuated systems because the visual approach is not easy to estimate the velocity and angular velocity used in the feedback control of the unmanned systems, especially for UAVs. Due to such challenges, the visual information has to be fused with those from other sensors adopted, such as inertial measurement units, to

realize the feedback control and autonomous navigation for unmanned systems.

A template matching based navigation system was proposed in [41], which combined the vision information with the inertial measurements to realize the drift-free control for a Micro-UAV.

Instead of the classical displacement estimation approaches using template match, here, a more sophisticated and systematic vision augmented approach is proposed to realize motion estimation of a UAV in the GPS-denied condition. This approach is composed of the robust landmark detection and core algorithm for vision-based motion estimation, which is the primary contributions of the work.

In this chapter, a well-structured landmark is used as the reference. To realize the robust key feature point extraction and correspondence, a hierarchical detection scheme is employed. The pattern structure is identified first, and then key feature points are extracted even in partially occluded conditions. A special feature point correction procedure is used to eliminate impact of noise in the feature point extraction to obtain optimal extraction results. Based on the 3D model and corresponding 2D image points, a pose estimation algorithm is proposed to estimate the relative position and angle of the aircraft with respect to the ground reference. The velocity of the aircraft is estimated with the measurement of the position, and can be improved with the Kalman filter fusing measurements of IMU, which can provide the necessary information for the hovering control of the unmanned helicopters.

The purpose of the vision-aided motion estimation is to use the camera to estimate the relative pose between the local reference and the camera. Such information is integrated with the INS to obtain the displacement and velocity of the UAV. The displacement and velocity is used in the feedback control to stabilize the UAV.

The remainder of this chapter is organized as follows: Section 5.2 details the vision-based landmark detection approach. Section 5.3 describes the motion estimation by the Kalman filtering. Section 5.5 shows experiment results of the vision-based motion estimation

approach using the data in actual flight tests. Finally, concluding remarks and discuss the future work are presented in Section 5.6.

5.2 Landmark Detection

In this section, a vision algorithm is proposed to identify a pre-defined landmark and find the correspondence of the key feature points. To speed up the vision processing, in this chapter, an artificial landmark with the simple pattern is employed as the local reference. Although the artificial landmark used, the presented general framework for the pose estimation can also be applied for the natural landmarks. Inspired by the pattern design in [109], a landmark with the square pattern on it is proposed, which is shown in Figure 5.1. It is composed of eight equal-size squares that are enclosed with a big white border, and has the following advantages:

1. High Contrast: the white surface of the squares can give the highest contrast to the background, which is assumed to be dark. It can make the segmentation processing robust and efficient;
2. Simple Shape: Squares are used as the basic shape on the landmark, which can be identified easily;
3. Rich Information: This pattern can give rich information for the motion estimation. The eight equal-size squares give us the highest feature density in the same area. The centers of the landing pad and the corners of the squares are chosen as the feature point, up to 48 points. At least four point and no three collinear are required to estimate the pose and motion of the UAV relative to the landmark, since the pose of a calibrated camera can be uniquely determined from a minimum of four coplanar but noncollinear points [105]. The special design of the pattern can provide the heading information efficiently;

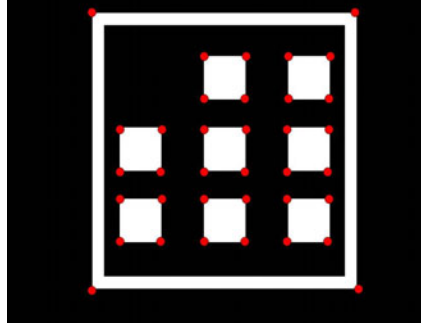


Figure 5.1: Landmark design.

A hierarchical scheme is proposed to efficiently search and identify this landmark during the flight. This scheme identifies the pattern structure first, and then extract the key feature points, including corners and centers of squares from the pattern structure. An algorithm, using the contour moments and the angles of join edges, is proposed to detect the squares on the image. The extracted squares are used to identify the pattern structure based on certain rules. If the pattern structure is detected, the correspondence between the squares on the landmark and their locations in the image is established. At last, the key points, including corners and centers of the squares, are extracted and related to their 3D points in the landmark for the pose estimation. The purpose of applying such detection scheme instead of doing the corner detectors is to improve the robustness of the key point detection, and avoid the false detection in the outdoor cluttered environments. The flow chart of the landmark detection algorithm is illustrated in Figure 5.2. The main functions of each component of the algorithm are described in the following part.

1. Noise Reduction:

A 3×3 median filter is employed in the image processing algorithm to filter out as much as possible noises on captured images. It is to get rid of speckle noises, and salt and pepper noises in an image. In addition, the distortion of the image is also corrected before the further processing. The detailed explanation have been presented

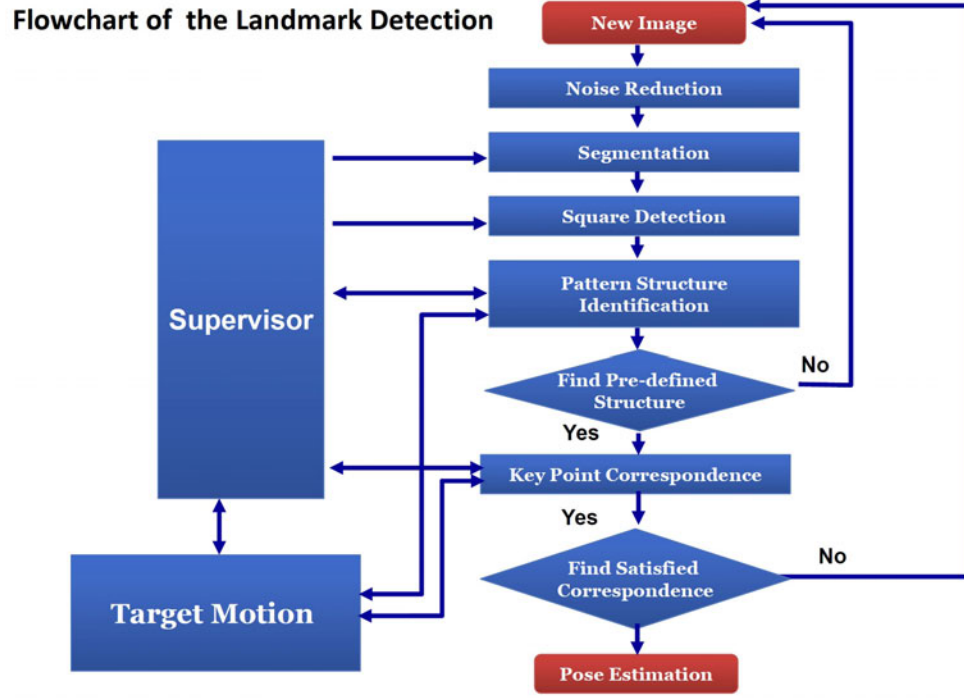


Figure 5.2: Flow chart of landmark detection.

in Chapter 4.

2. Segmentation:

The purpose of the segmentation is to separate the foreground objects from the background. To obtain better threshold result, the image is converted to the intensity image, and normalized in terms of the minimum and maximum values of the image intensity. This normalization procedure can minimize the effect of changing of lighting conditions in the outdoor. A threshold approach is then used to convert the intensity image to the binary image. To calculate the optimal threshold value, the intensity distributions of the landmark and the background are modeled by using the image database captured in flight. Based on these estimated models, the optimal threshold value is calculated by using the discriminant functions defined as

$$g_1(x) = \ln p(x|\omega_1) + \ln P(\omega_1) ,$$

$$g_2(x) = \ln p(x|\omega_2) + \ln P(\omega_2) ,$$

where $p(x|\omega_1)$ is the probability density of the background; $p(x|\omega_2)$ is the probability density of the landmark; $P(\omega_1)$ is the probability of the background; $P(\omega_2)$ is the probability of the landmark. The optimal threshold value x_0 can be calculated as the solution of the following equation:

$$g_1(x) = g_2(x) ,$$

After the image segment by using the optimal threshold value, a contour detection algorithm is employed to find the boundary of shapes of all the foreground objects. The contour detection algorithm is to obtain the outlines of foreground objects that provide rich geometry information of the objects.

3. Square detection:

To detect all the squares in the image robustly, two kind of geometry features are employed:

- (a) The cosines of the angles between the joint edges at the corners of the polygon derived from contours of the shapes;
- (b) The moment invariants of the shapes, which are proposed in Chapter 4.

These feature descriptors are invariant to rotation, translation and scaling, which are suitable for the applications using moving platforms. Based on the probability densities of the features, a likelihood function is used to identify the squares in the image. The distributions of these two geometry feature descriptors are computed based on the captured image database off-line.

4. Pattern structure identification:

Generally, multiple squares are detected in the clustered environment. To decide the

expected pattern structure, several decision rules are proposed, which are defined as

- (a) The ratio of the area of each small square and the small and big square must fall in a certain range;
- (b) The distance between each small squares must fall in a certain range;

5. Key point correspondence:

If the designed pattern structure is identified, it is necessary to find the correspondence between the squares on landmark and their location in the image. Such correspondence is obtained by using the similarity transform, since the roll and pitch angle is small during the flight tests. The similarity transform is illustrated as follows:

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix}_j = \begin{bmatrix} s \cos \theta_s & -s \sin \theta_s & x_s \\ s \sin \theta_s & s \cos \theta_s & y_s \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}_j ,$$

where s is the scaling factor; θ_s is the rotation angle; $[x_s \ y_s]$ is the translation vector. $[x_i, \ y_i]_j$ is the coordinate of the center of the square j in the initial frame; $[x'_i, \ y'_i]$ is the estimated coordinate of the j -th square in the current image frame. Based on the similarity transform, each square will be labeled according to their distance to the predicted location of the pattern structure. After labeling the squares in the image, it is not difficult to find correspondence of the key feature points, including the corners and the centers of the squares.

6. Supervisor

As the supervisor, the finite state machine plays a critical role in the vision algorithm, which dynamically chooses necessary features and gives different weightings to each feature in the discriminant function under different conditions. In addition, the rule for the pattern structure identification is varied according to the detection state to retrieve the landmark even in the partial occluded situation.

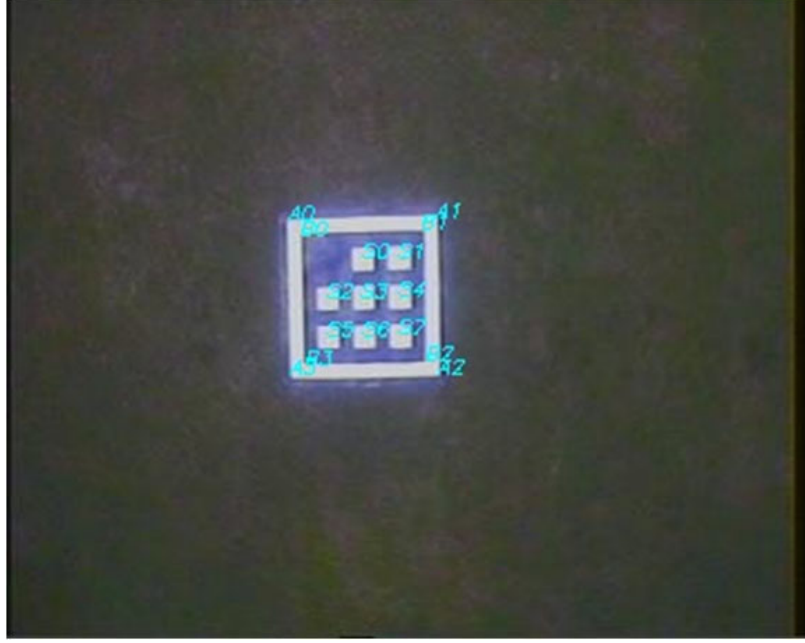


Figure 5.3: Key point correspondence.

5.3 Pose Estimation

Among various applications of visual feedback for mobile robots or unmanned vehicles, an important commonality is to recover the 3D structure of the scene, and estimate the motion of camera relative to the scene from a sequence of image inputs. The former one will be discussed in future. Regarding the motion estimation, both the image point position and image point velocity can be employed as the input, which are referred to as discrete method and differential method [107]. In this chapter, the vision-based motion estimation is investigated based on image point position of the fixed feature points in the scene.

With the detected key feature points of the landmark, and the prior knowledge of the structure model of the corresponding points in the scene, the motion estimation of the unmanned helicopter can be regarded as the model based camera pose estimation or perspective- n -point problem(PnP) in the computer vision. The objective of pose estimation is to determine the geometry transformation that relates the camera to the scene structure

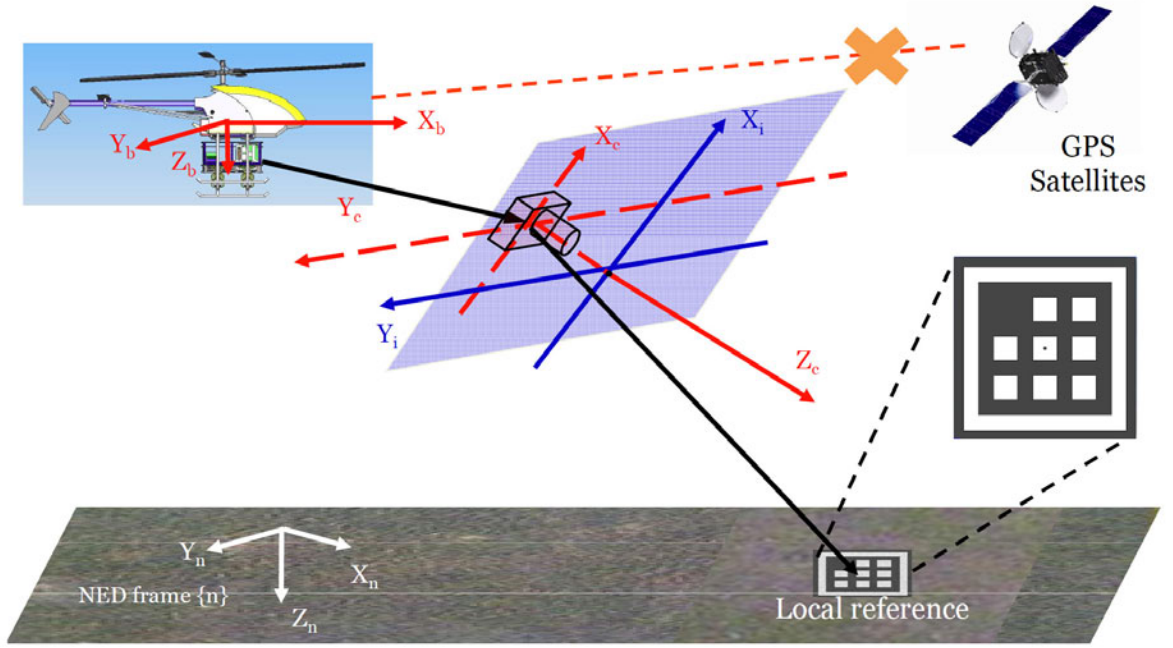


Figure 5.4: Illustration of vision-based motion estimation.

with the known intrinsic parameters of the camera, known geometry structure and sufficient feature points in the image, as depicted in Figure 5.4.

For three or four noncollinear points, closed form solutions of the pose of the camera can be obtained [30]. For instance, up to four possible solutions can be obtained in the P3P problem. Furthermore, in the P4P problem, two solution may be obtained. Especially, for four noncollinear points, it was proved that in an ordinary condition the unique theoretical solution of the pose estimation problem can be obtained [55]. For five and six noncollinear points, the polynomial system can be formulated in terms of geometrical information, and the problem is solved by finding roots of the polynomial system. However, these analytic solutions cannot be applied to more than six points, and these closed form solutions are also sensitive to the noise in the image and not precise enough. Therefore, to achieve robust and accurate pose estimation, more feature points should be used.

To solve more than six matched points in linear formulation, the orthogonality constraint of the rotation matrix is relaxed or the perspective model of the camera is simplified [29].

A linear (unconstrained) solution is computed first, and then the solution is fit to the "closest" orthogonal matrix. Although this linear solution is already close enough to the true value, linear method may not guarantee the convergence of the algorithm, and achieve high precision. The linear solution can be employed as a good initial value for the nonlinear method.

To search the global optimal solution in the pose estimation for more than four points, it is formulated as a nonlinear least-squares problem and try to solve it by using nonlinear optimization algorithms, such as the Gauss-Newton method or the Levenberg-Marquardt method [73, 85, 75, 104]. The pose estimation problem can be defined as the problem of estimating six exterior parameters of the camera: orientation $\mathbf{R}_{c/t} = f(\phi_c, \theta_c, \psi_c) = [r'_1{}^T \ r'_2{}^T \ r'_3{}^T]$ and position $\mathbf{t}_{c/t} = [t_x, t_y, t_z]^T$ of the camera with known scene structure and known interior parameters of camera. In another word, the pose estimation is to find the $\hat{\mathbf{R}}_{c/t}$ and $\hat{\mathbf{t}}_{c/t}$ that minimize an error function, which can be defined in two ways: image-space error and object-space error [75]. Nonlinear optimization algorithms based on these two definitions can achieve high precision, but sensitive to the initial values.

In this project, a two-step method for pose estimation is used. First, a linear approach is employed to compute an initial value by using collinearity equation and the direct linear transformation (DLT) method. Then, an optimal solution is searched by using Gauss-Newton method based on an error function. Recall the collinearity equation for one point in (4.43), which is given by

$$\begin{cases} r_0x_t + r_1y_t + r_2z_t + t_x - \frac{x_i - t_x}{f_x}(r_6x_t + r_7y_t + r_8z_t + t_z) = 0 \\ r_3x_t + r_4y_t + r_5z_t + t_y - \frac{y_i - t_y}{f_y}(r_6x_t + r_7y_t + r_8z_t + t_z) = 0 \end{cases} \quad (5.1)$$

In this project, all the feature points are assumed to be on the ground, so $z_o = 0$. Thus, the above homogeneous linear equation can be simplified and yield:

$$\begin{cases} r_0x_t + r_1y_t + t_x - \frac{x_i - o_x}{f_x}(r_6x_t + r_7y_t + t_z) = 0 \\ r_3x_t + r_4y_t + t_y - \frac{y_i - o_y}{f_y}(r_6x_t + r_7y_t + t_z) = 0 \end{cases} \quad (5.2)$$

Normally, many more feature points are extracted from the images, and more collinearity equations can be obtained. Thus, by stacking these equations, a homogeneous linear system yields

$$\mathbf{F}\mathbf{x} = \mathbf{0}$$

where

$$\begin{aligned} \mathbf{F} &= [\mathbf{F}_1^T, \dots, \mathbf{F}_m^T]^T, \\ \mathbf{F}_i &= \begin{bmatrix} x_{t,i} & 0 & -\bar{x}_{i,i} & x_{t,i} & y_{t,i} & 0 & -\bar{x}_{i,i} & y_{t,i} & 1 & 0 & -\bar{x}_{i,i} \\ 0 & x_{t,i} & -\bar{y}_{i,i} & x_{o,i} & 0 & y_{t,i} & -\bar{y}_{i,i} & y_{t,i} & 0 & 1 & -\bar{y}_{i,i} \end{bmatrix}, \\ \bar{x}_i &= \frac{x_{i,i} - o_x}{f_x}, \\ \bar{y}_i &= \frac{y_{i,i} - o_y}{f_y}, \\ \mathbf{x} &= [a_0 \ a_3 \ a_6 \ a_1 \ a_4 \ a_7 \ t_x \ t_y \ t_z]^T, \\ \mathbf{x} &= [r_0 \ r_3 \ r_6 \ r_1 \ r_4 \ r_7 \ t_x \ t_y \ t_z]^T. \end{aligned}$$

$\mathbf{F} \in \mathcal{R}^{2m \times 9}$, m is the number of the correspondences. As proved in [55], if there are at least 4 points such that no three are collinear, then $\text{rank}(\mathbf{F}) = 8$, a unique solution can be obtained together with the orthogonality of the rotation matrix [109].

To refine the solution obtained using the linear method, an optimal search method is employed. The image space error function for the optimal search is given by

$$EI(\hat{\mathbf{R}}_{c/t}, \hat{\mathbf{t}}_{c/t}) = \sum_{i=1}^m [EX_i^2 + EY_i^2], \quad (5.3)$$

where

$$\begin{aligned} EX_i &= x_{i,i} - \frac{r_1'^T \mathbf{p}_{t,i} + t_x}{r_3'^T \mathbf{p}_{t,i} + t_z}, \\ EY_i &= y_{i,i} - \frac{r_2'^T \mathbf{p}_{t,i} + t_y}{r_3'^T \mathbf{p}_{t,i} + t_z}. \end{aligned}$$

Given m key feature points extracted from the image and $m \times 2$ functions EX_i and EY_i of n variables $\beta = (t_x, t_y, t_z, \phi_c, \theta_c, \psi_c)$, the Gauss-Newton method is used to find the minimum of Equation (5.3). Starting with an initial guess β^0 for the minimum, the method proceeds by iterations [109].

$$\beta^{s+1} = \beta^s + \delta\beta, \quad (5.4)$$

with the increment $\delta\beta$ satisfying the normal equations

$$(J_r^T J_r) \delta\beta = -J_r^T r$$

where r is the vector of functions $[EX_i \ EY_i]^T$, J_r is the $2m \times n$ Jacobian matrix of r with respect to β , both evaluated at β^s . In addition, the error function can also be defined based on object-space:

$$EO = \sum_{i=1}^m \|(I - \hat{S}_i)(\hat{\mathbf{R}}_{c/t} \mathbf{p}_{t,i} + \hat{\mathbf{t}}_{c/t})\|,$$

where

$$\begin{aligned} \hat{S}_i &= \frac{\hat{s}_i \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i}, \\ \hat{s}_i &= [x_{i,i} \ y_{i,i} \ 1]^T. \end{aligned}$$

The similar optimal search algorithms mentioned above can be used to find the optimal solution.

5.4 Data Fusion

To stabilize the aircraft motion, at least measurement of the position and velocity of the aircraft are needed. The position of the aircraft can be identified with information from the

vision based sensor. The algorithms to identify the position of aircraft have been developed. However, it is difficult to identify velocity of aircraft individually with information from the vision based sensor or IMU. The data fusing to estimate velocity of aircraft has to be considered. Those data fusing algorithms based on Kalman filtering have been developed.

Kalman Filtering

A discrete-time linear system is formulated as follows,

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(\mathbf{u}(k) + \mathbf{w}(k)) , \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathbf{v}(k), \end{cases} \quad (5.5)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^m$ are state, input and measured variables. $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$ are system matrices with appropriate dimensions. $\mathbf{w} \in \mathbb{R}^p$ and $\mathbf{v} \in \mathbb{R}^m$ are input and measurement noises, which are zero-mean Gaussian noise. The objective of Kalman filtering is to present $\hat{\mathbf{x}}(k|k)$ at the step k with the measurement, $\mathbf{y}(k)$, input, $\mathbf{u}(k-1)$ and the estimated $\hat{\mathbf{x}}(k|k-1)$. We need to assume that (5.5) is observable. Kalman filtering is given as follows:

Time update

$$\begin{aligned} \hat{\mathbf{x}}(k|k-1) &= \mathbf{A}\hat{\mathbf{x}}(k-1) + \mathbf{B}\mathbf{u}(k-1) , \\ \mathbf{P}(k|k-1) &= \mathbf{A}\mathbf{P}(k-1)\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T \end{aligned} \quad (5.6)$$

Measurement update

$$\begin{aligned} \mathbf{H}(k) &= \mathbf{P}(k|k-1)\mathbf{C}^T (\mathbf{C}\mathbf{P}(k|k-1)\mathbf{C}^T + \mathbf{R})^{-1} , \\ \hat{\mathbf{x}}(k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{H}(k)(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)), \\ \mathbf{P}(k) &= (\mathbf{I} - \mathbf{H}(k)\mathbf{C})\mathbf{P}(k|k-1) \end{aligned} \quad (5.7)$$

where $\mathbf{H}(k|k-1)$ is a feedback gain matrix, and $\mathbf{P}(k|k-1)$ is the covariance of the state estimation error that is defined as

$$\begin{aligned}\mathbf{P}(k|k-1) &= E\{[\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)][\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)]^T\}, \\ \mathbf{R}(k) &= E\{\mathbf{v}(k)\mathbf{v}^T(k)\}, \quad \mathbf{Q}(k) = E\{\mathbf{w}(k)\mathbf{w}^T(k)\}.\end{aligned}$$

$E\{\star\}$ denotes expectation.

Displacement and Ground Velocity Estimation

The position of the vehicle relative to the landmarks is measured with the vision based sensors and additional information on the landmarks. We need to identify the ground velocity and to improve the measured position of the vehicle, which can be achieved using Kalman filtering. The data fusing algorithm is developed based on the kinematical model of the aircraft. We firstly convert the model into the standard discrete-time version so that Kalman filtering can be easily applied.

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}(\mathbf{u}(k) + \mathbf{w}(k)), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k),\end{aligned}\tag{5.8}$$

where

$$\begin{aligned}\mathbf{x} &:= \begin{pmatrix} \mathbf{p}_n \\ \mathbf{v}_n \end{pmatrix}, \quad \mathbf{y} := \mathbf{p}_n, \quad \mathbf{u} := \mathbf{R}_{b/n}^T \mathbf{a}_{nb} + g \mathbf{e}_3, \\ \mathbf{A} &= \begin{bmatrix} \mathbf{I} & \mathbf{T}_s \mathbf{I} \\ 0 & \mathbf{I} \end{bmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{\mathbf{T}_s^2}{2} \mathbf{I} \\ \mathbf{T}_s \mathbf{I} \end{pmatrix}, \quad \mathbf{C} = [\mathbf{I} \quad 0],\end{aligned}$$

\mathbf{T}_s is the sampling period. $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are assumed to be zero-mean Gaussian noise. With (5.8), we can follow computation of Kalman filtering in (5.6) and (5.7) to calculate $\hat{\mathbf{x}}(k+1|k)$ and $\hat{\mathbf{x}}(k|k)$. The parameters of the Kalman filtering are given by

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}(0) = \mathbf{BQB}^T, \quad \hat{\mathbf{x}}(-1) = \begin{pmatrix} \mathbf{p}_n(0) \\ \mathbf{0}_{3 \times 3} \end{pmatrix}, \quad \mathbf{a}_{nb}(-1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In some special cases, an open-loop estimation scheme is employed as follows

1. If the target is suddenly lost: DetectState == 0, the state estimation in (5.7) will not be executed.
2. If the error between the estimated values and measured values are large than certain threshold values: $\|e_1\| > \Gamma_1$ or $\|e_2\| > \Gamma_2$ or $\|e_3\| > \Gamma_3$, the state estimation (5.7) will not be executed.

$$\mathbf{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1) \quad (5.9)$$

5.5 Experimental Results

In this section, the proposed vision-based motion estimation approach is evaluated. The main focus of this approach is to estimate the position and velocity of the UAV with respect to the landmark, though the rotation matrix can be estimated simultaneously. The proposed approach is evaluated by varying the translation between the landmark and the camera. The image data collected on the ground and in the flight are used to test the proposed motion estimation approach. The vision algorithms are implemented by using the open source computer vision library (OpenCV).

To verify the proposed motion estimation approach, the simulation of the pose estimation was conducted using the computer-generated image data. The error of the translation vector and the Euler angles are shown in Figure 5.5 and 5.6. These results show that very accurate estimates can be achieved in the noise-less condition.

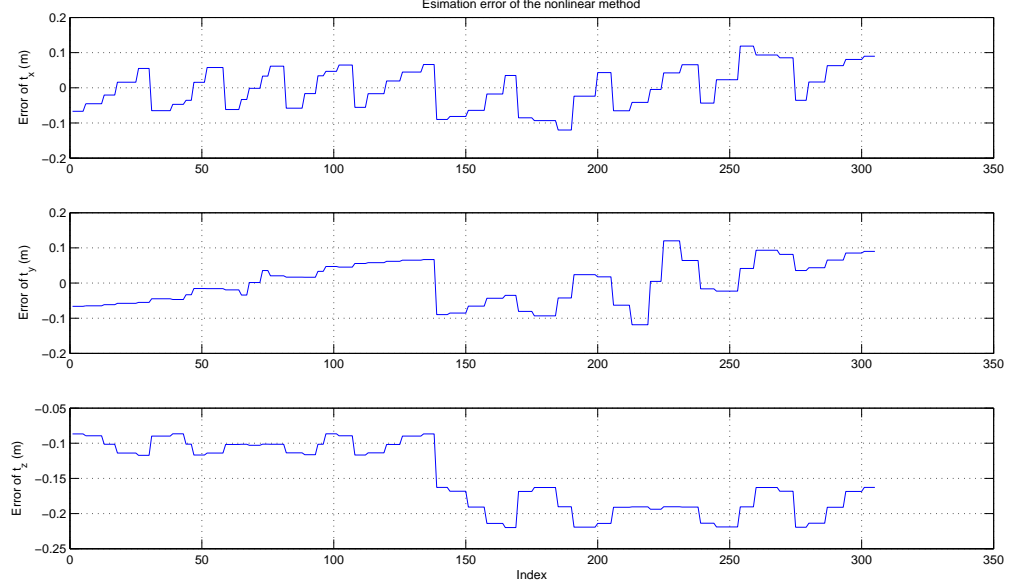


Figure 5.5: Vision-based position estimation using the simulation data

In addition to evaluation using the computer-generated image data, pose estimation is performed using real images captured on the ground. In the experiment, the translation vector is varied with increasing the distance in the axis- Z_o and $-Y_o$. The results in Figure 5.7 show that the translation error increases, as the camera moves far away the landmark. That is due to the image capturing error in the long distance.

After presenting the whole system for the vision-based stabilization, the flight tests have been conducted to verify the proposed system. In the flight tests, the landing pad was place on the flat ground, and the detected information is used to estimate the relative position and velocity of the UAV in the NED frame.

To deal with the illumination change in the outdoor environments, an searching scheme is adopted to select the threshold value in the image processing. At the initial stage, a set of different threshold values in a certain range $[h_1, h_2]$ will be tried. If the target was found by using several different threshold values, we will choose the median value of them in the

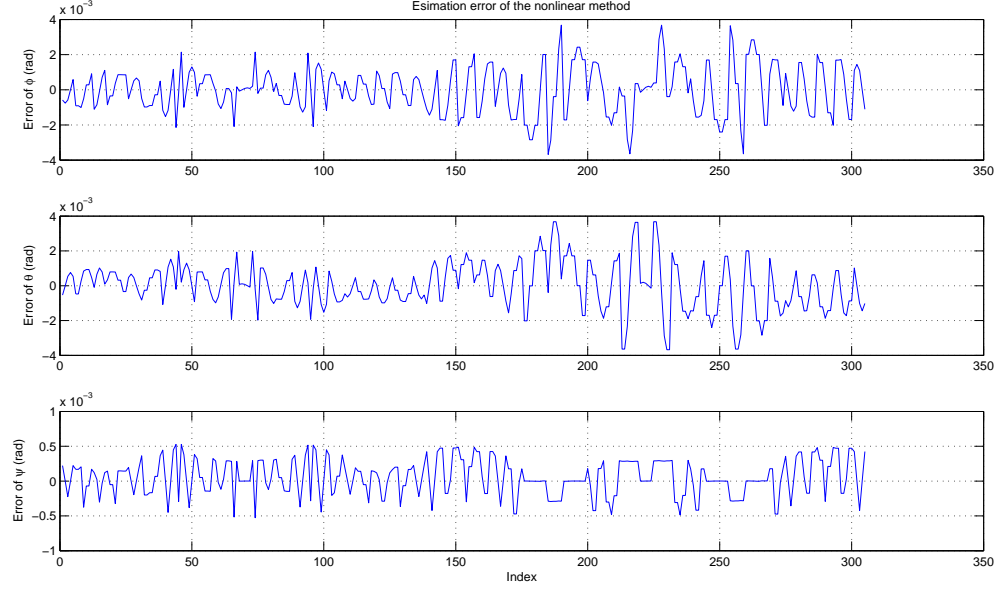


Figure 5.6: Vision-based velocity estimation using the simulation data

subsequent processing. Otherwise, the searching will be continued during the flight.

In the flight tests, first, the UAV is manually hovered at a position where the landing pad can be observed. After the landing pad is detected automatically, the vision-based autonomous flight control is started. GPS signals are not employed in the flight control. The test results are shown in Figure 5.8 and 5.9. In the tests, the vision aided INS technique can provide equivalent position and velocity estimation compared with the widely used GPS aided INS technique.

In each cycle, a thread needs a period of time to execute the specified task. The time consumption of each thread is measured in tests and shown in Table 5.10. From the results shown in Figure 5.10, we can observe that the vision algorithms cost more than 80% of the computational resource, and the complex of the vision algorithms will severely affect the real-time application of the vision system.

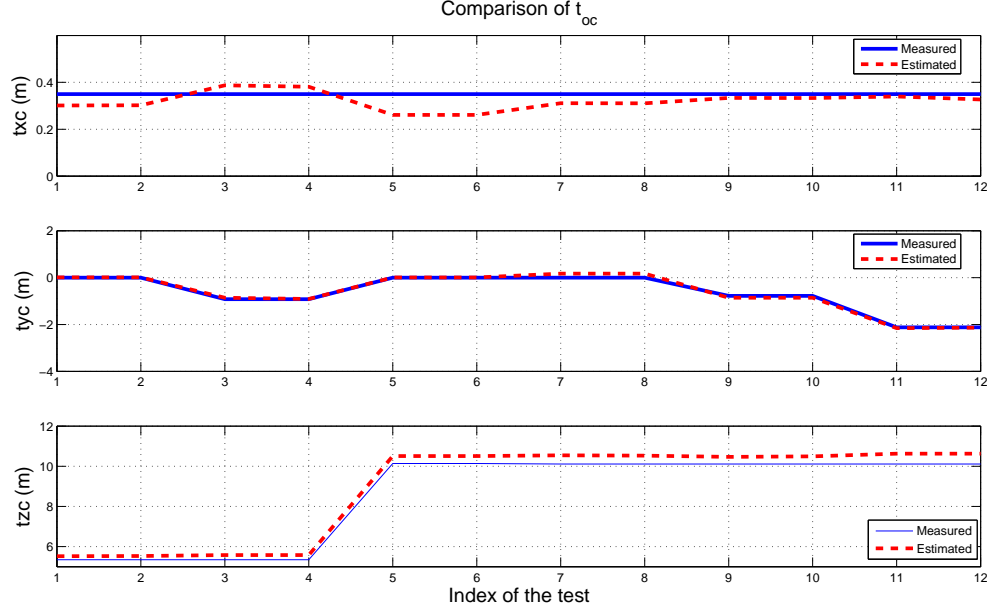


Figure 5.7: Vision-based position estimates using real images

5.6 Conclusion

This chapter has presented the comprehensive design and implementation of a vision aided motion estimation approach for UAVs under the GPS-denied environment. The sophisticated landmark detection and feature correspondence algorithm has been addressed. Moreover, the relative motion estimation scheme is proposed by Kalman filter fusing vision information with the outputs of the IMU. The preliminary tests of the vision aided system to execute relative positioning autonomously was made. The test results demonstrated that the positioning accuracy can be improved by the vision aid system compared to GPS receivers.

The proposed approach is integrated with the flight controller to realize the drift-free hovering. In addition, more advanced feature extraction approaches will be studied and implemented in order to realize the applications of the motion estimation using more general landmarks.

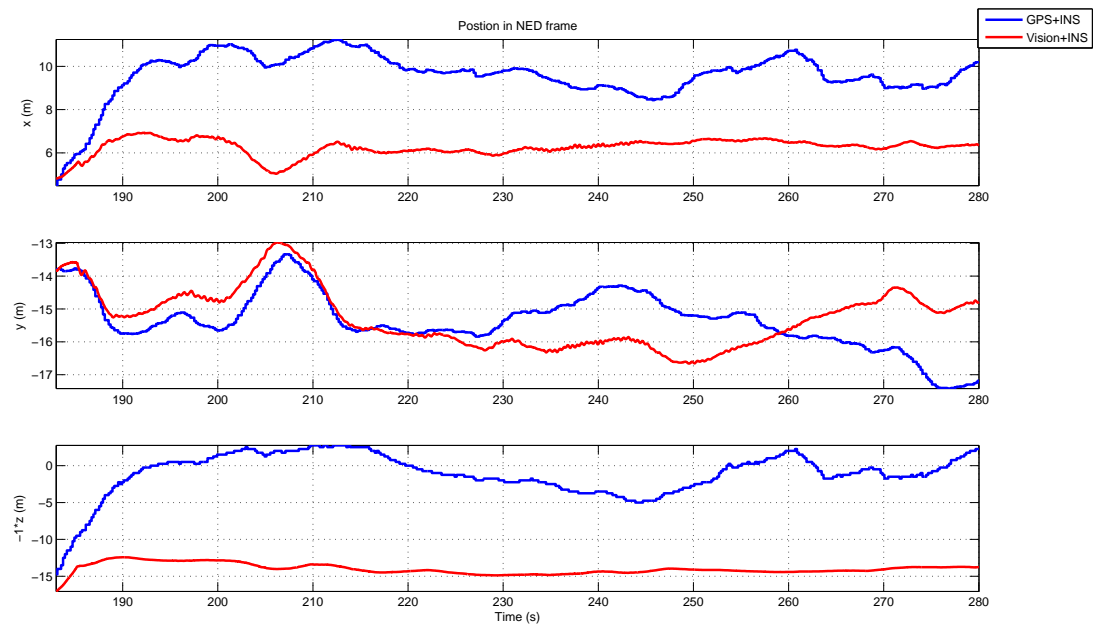


Figure 5.8: Comparison of position estimation

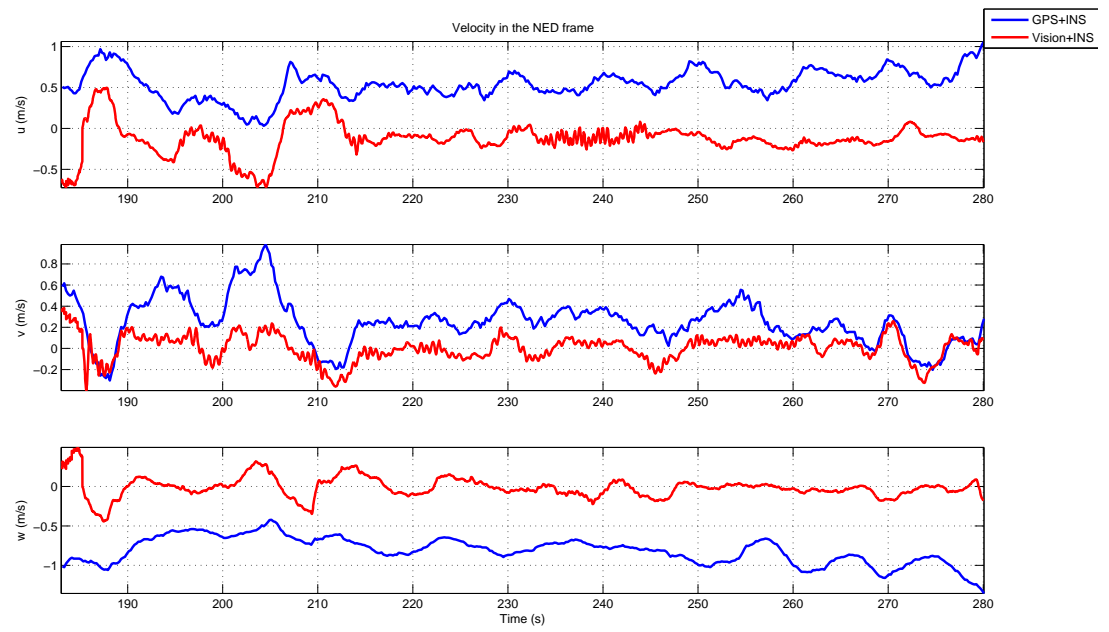


Figure 5.9: Comparison of velocity estimation

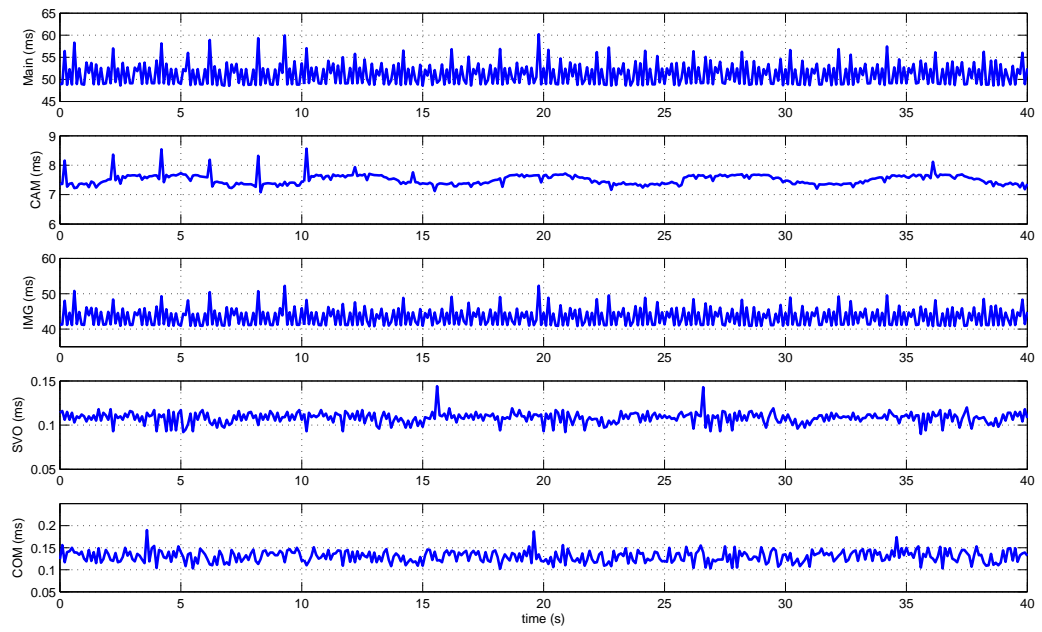


Figure 5.10: The time cost of each thread in the flight

Chapter 6

Conclusions

The aim of this research is to explore the potentials of UAVs in the vision-based applications. Specifically, the comprehensive design and implementation of a vision-based unmanned helicopter has been presented in this thesis, including hardware construction and software development. Based on this fully autonomous unmanned helicopter, the advanced ground target following scheme has been proposed by using the on-board vision sensing. Multiple flight tests were conducted to verify the presented vision-based target following of the autonomous UAV. In addition, we have presented the comprehensive design and implementation of a vision-aided motion estimation and flight control approach for UAVs in GPS-denied environments. The preliminary tests of the vision system to execute relative positioning autonomously was made.

6.1 Contributions

The research work contributes towards the investigation on the potential capabilities of vision-based unmanned helicopter in the following four aspects:

1. First, we have proposed a comprehensive methodology for designing a small-scale vision-based helicopter platform construction. As introduced in Chapter 2, the construction of the vision-based helicopter involves multiple disciplines, including control theory, communications, aerodynamics, navigation, computer vision, and artificial intelligence. The construction is generally challenging, and time consuming. Since there is no comprehensive and effective design method can be found in the literature, researchers have to spend a lot of time to repeat the literature survey to investigate the design procedure. Therefore, a comprehensive survey of the vision system for UAVs has been presented in Chapter 2 of this thesis. Moreover, a standard design procedure has been proposed. Based on such a procedure, a vision-based helicopter: SheLion was constructed using the computer aided design method. This design method can also be extended to develop other ground vehicles or robots.
2. Second, a highly efficient software system for the vision-based unmanned helicopter has been developed in this study. The proposed software consists of three main parts: on-board flight control software, on-board vision software, and ground support software. The software is highly modularized and can be easily (1) upgraded to a more advanced version and (2) extended to other platforms, such as the fixed-wing UAV or unmanned ground vehicle (UGV). Particularly for the onboard flight control and vision softwares, the design concepts on (1) multi-thread framework, (2) scheduling of task management and (3) coordination of the flight control and vision system are very useful for normal unmanned vehicle and robots. The design concepts and methodologies can also benefit the researchers who are intended to design their software system for unmanned vehicles or robots.
3. Third, systematic approaches to realize the robust and efficient ground target following based on the vision sensing have been proposed. To realize the vision sensing, the feature-based target detection and the hierarchical tracking scheme have been presented. The feature-based detection can efficiently identify the target and initialize

the tracking scheme. Then, model-based tracking and mean-shift based tracking can be employed hierarchically to achieve robust performance and save processing time. The vision information is used as the reference for the UAV to realize the vision-based target following. Multiple real flight tests are conducted to verify the presented target following application for the vision-based UAV. The experimental results show that this vision system is not only able to automatically detect and track the pre-defined ground target in the video sequence, but also able to guide the unmanned helicopter to follow the motion of the target in flight. Based on the proposed approaches, more advanced applications can be achieved, such as vision-based landing and air-to-air tracking.

4. Finally, for UAVs in GPS-denied environments, the comprehensive design and implementation of a vision aided motion estimation approach has been presented in Chapter 5. This approach is composed of a sophisticated landmark detection, feature correspondence algorithms, as well as a relative motion estimation scheme by Kalman filter fusing vision information with the outputs of the IMU. The preliminary tests of the vision aided system to execute relative positioning autonomously is made. The test results have demonstrated that the positioning accuracy can be improved by the vision aid system compared to GPS receivers. This approach is integrated with the flight controller to realize the drift-free hovering in a GPS-denied condition. Advanced feature extraction approaches will be studied and implemented in order to realize the applications of the motion estimation in unknown environments.

6.2 Future Works

Although a comprehensive study on vision-based unmanned helicopters has been carried out, it is only the beginning of our vision-based UAV research. Considering the requirements on various practical implementations, extensive contributions could be achieved by extending

the small-scale UAV research in the following directions.

Vision-based Formation Flight and Motion Coordination

The research completed in the last five years mainly focused on single small-scale unmanned helicopter. Currently, the practical flight missions are becoming more and more complicated. In many situations, multiple helicopters are required to cooperate with each other or, further, form a group to cooperate with other type of unmanned/manned vehicles to fulfill the tasks. The vision sensing can be used as important feedback information in formation control of multiple UAVs.

Vision-based Maneuver Target Tracking

In the real applications, maneuver targets are required to be tracked, which involves unknown and fast motion, such as suddenly turn. That causes the severe change of appearance and scale of the target. Special techniques are required to cope with these problems.

Urban Area Implementation

Urban area generally has the features of limited space, complicated environment, and various uncertainties. As such, a small-scale unmanned aerial vehicle is the most suitable platform for the surveillance purpose. To facilitate implementation in this aspect, we need to combine the current small-scale unmanned helicopters with research results in other regions, such as visual-based navigation and obstacle-avoidance, to achieve an advanced unmanned system.

Vision-aided Navigation System in GPS-Denied Environments

The vision-aided navigation systems in GPS-denied environments, such as indoor and urban areas, gain strong interest from civilian to military areas. To extend the functions of

unmanned vehicles in complex and dynamic environments, the intelligent vision-aided navigation system is required to explore and understand the scene and structure of such environments. Moreover, by integrating the vision information with other sensor measurements, such as laser and IMU, the vision-aided navigation system is able to carry out high-level operations, such as localization and mapping, obstacle detection, ego-motion estimation, path planning and more. Enhanced by such vision-aided navigation system, unmanned vehicles can perform more advanced applications, such as indoor navigation, vision-based autonomous landing and so on. However, such navigation systems arouse many challenges from various aspects such as the robust vision information understanding, data fusion, on-board implementation, and so on. Although facing such challenges, the whole NUS UAV research team is currently carrying out the development of the qualified and suitable micro-aerial-vehicle (MAV) helicopter platforms and practical navigation methodologies for indoor navigation and further practical implementations. In addition, for the vision-based outdoor applications, we also focus on the approach of vision-based automatic landing of a UAV on a moving platform in GPS-denied environments, such as the vision-based landing of a UAV on a ship.

Information Extraction and Object Representation

In fact, the vision sensing plays a critical important role in aforementioned vision applications. The robust and efficient information extraction approaches can extensively improve the performance of the vision algorithms. Based on the extracted information or features, the objects and scene can be represented, and understood based on a well-developed vision database. Therefore, the state-of-art technologies in machine vision will be investigated. Especially, feature extraction, object representation and establishment of model-to-data correspondence will be kernel parts of the future work.

In conclusion, we have successfully carried out the comprehensive study on the vision-based unmanned helicopters, based on our self-instrumented helicopter platform. The overall

research procedure and achievements, as well as concepts and ideas of the research work have been sequentially documented in Chapter 2 to 5. To any researchers who are on the way to the research of vision-based UAVs or UGVs, our experience is greatly instrumental. With the effort and passion, we hope vast contributions could be made on the research of vision-based UAVs.

Bibliography

- [1] M. Achtelik, A. Bachrach, R. He, S. Prentice and N. Roy, “Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments,” *The First Symposium on Indoor Flight Issues 2009*, Mayagez, Spanish, 2009.
- [2] M. J. Adriaans, R. C. Cribbs, B. T. Ingram and B. P. Gupta, “A160 Hummingbird unmanned air vehicle development program,” *Proceedings of the AHS International Specialists’ Meeting — Unmanned Rotorcraft: Design, Control and Testing*, Chandler, USA, 2007.
- [3] O. Amidi, T. Kanade and R. Miller, “Vision-based autonomous helicopter research at Carnegie Mellon Robotics Institute 1991-1997,” *Proceedings of American Helicopter Society International Conference*, Gifu, Japan, pp. 1–12, 1998.
- [4] C. Arth and H. Bischof, “Real-time object recognition using local features on a DSP-based embedded system,” *Journal of Real-Time Image Processing*, vol. 3, pp. 233–253, 2008.
- [5] S. Avidan, “Ensemble tracking,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, pp. 494–501, 2005.
- [6] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): part II,” *IEEE Robotics & Automation Magazine*, vol. 13, pp. 108–117, 2006.

- [7] W. R. Baker and R. W. Clem, "Terrain contour matching (TERCOM) premier," *Aeronautical Systems Division, Wright-Patterson*, 1977.
- [8] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, M. Goodrich, "Autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, pp. 92–108, 2005.
- [9] A. Betser, P. Vela and A. Tannenbaum, "Automatic tracking of flying vehicles using geodesic snakes and kalman filtering," *Proceedings of 43rd IEEE Conference on Decision and Control*, Atlantis, Bahama, pp. 1649–1654, 2004.
- [10] Y. Boykov and D. P. Huttenlocher, "Adaptive bayesian recognition in tracking rigid objects," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, pp. 697–704, 2000.
- [11] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal Q2 '98*, pp. 1–15, 1998.
- [12] G. Cai, A. K. Cai, B. M. Chen and T. H. Lee, "Construction, modeling and control of a mini autonomous UAV helicopter," *Proceedings of the IEEE International Conference on Automation and Logistics*, Qingdao, China, September 2008.
- [13] G. W. Cai, B. M. Chen, K. M. Peng, M. B. Dong and T. H. Lee, "Modeling and control system design for a UAV helicopter," *Proceedings of the 14th Mediterranean Conference on Control and Automation*, Ancona, Italy, pp. 1–6, 2006.
- [14] G. W. Cai, B. M. Chen, X. Dong and T. H. Lee, "Design and implementation of robust flight control system for a small-scale UAV helicopter," *Proceedings of the 7th Asian Control Conference*, Hong Kong, China, pp. 691–697, 2009.

- [15] G. W. Cai, B. M. Chen, T. H. Lee and K. Y. Lum, “Comprehensive nonlinear modeling of an unmanned-aerial-vehicle helicopter,” *Proceedings of the 2008 AIAA Guidance, Navigation and Control Conference*, Honolulu, Hawaii, 2008.
- [16] G. W. Cai, F. Lin, B. M. Chen and T. H. Lee, “Systematic design methodology and construction of UAV helicopters,” *Mechatronics*, vol. 18, pp. 545–558, 2008.
- [17] G. W. Cai, F. Lin, B. M. Chen and T. H. Lee, “Development of fully functional miniature unmanned rotorcraft systems,” *To be presented at the 29 Chinese Control Conference*, Beijing, China, 2010.
- [18] G. W. Cai, K. M. Peng, B. M. Chen and T. H. Lee, “Design and assembling of a UAV helicopter system,” *Proceedings of the 5th International Conference on Control and Automation*, Budapest, Hungary, pp. 697–702, 2005.
- [19] M. E. Campbell and W. W. Whitacre, “Cooperative tracking using vision measurements on seascan UAVs,” *IEEE Transactions on Control Systems Technology*, vol. 15, pp. 613–626, 2007.
- [20] F. Chaumette and S. Hutchinson, “Visual servo control part I: Basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, pp. 82–90, 2006.
- [21] F. Chaumette and S. Hutchinson, “Visual servo control part II: Advanced approaches,” *IEEE Robotics & Automation Magazine*, vol. 14 pp. 109–118, 2007.
- [22] B. M. Chen, T. H. Lee, K. Peng and V. Venkataramanan, “Composite nonlinear feedback control for linear systems with input saturation: Theory and an application,” *IEEE Transactions on Automatic Control*, vol. 48, pp. 427–439, 2003.
- [23] B. M. Chen, T. H. Lee, K. Peng and V. Venkataramanan, *Hard Disk Drive Servo Systems*, 2nd edn. New York, Springer, 2006.

- [24] C. C. Chen and T. I. Tsai, “Improved moment invariants for shape discrimination,” *Pattern Recognition*, vol. 26, pp. 683–686, 1993.
- [25] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 790–799, 1995.
- [26] D. Comaniciu, V. Ramesh and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, pp. 142–149, 2000.
- [27] D. Comaniciu, V. Ramesh and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, 2003.
- [28] R. V. Dell’Aquila G. Campa, M. R. Napolitano and M. Mammarella, “Real-time machine-vision-based position sensing system for UAV aerial refueling,” *Journal of Real-Time Image processing*, vol. 1, pp. 213–224, 2007.
- [29] D. F. Dementhon and L. S. Davis, “Model-based object pose in 25 lines of code,” *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995.
- [30] M. Dhome, M. Richetin, J. T. Lapreste and G. Rives, “Determination of the Attitude of 3-D Objects from a single perspective view,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1256–1278, 1989.
- [31] M. B. Dong, B. M. Chen, G. W. Cai and K. M. Peng, “Development of a real-time onboard and ground station software system for a UAV helicopter,” *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 933–955, 2007.
- [32] M. B. Dong, B. M. Chen and C. Cheng, “Development of 3D monitoring for an unmanned aerial vehicle,” *Proceedings of the 1st International Conference on Computer Science and Education*, Xiamen, China, pp. 135–140, 2006.

- [33] R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification." 2nd edn, John Wiley & Sons, Inc, 2001.
- [34] S. A. Dudani, K. J. Bredding and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Transactions on Computer*, vol. C-26, pp. 39–46, 1977.
- [35] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): part I," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 99–110, 2006.
- [36] A. Elgammal, D. Harwood and L. Davis "Non-parametric model for background subtraction," *Proceedings of European Conference on Computer Vision*, Dublin, Ireland, pp. 751–767, 2000.
- [37] B. Enderle, "Commercial applications of UAV's in Japanese agriculture," *Proceedings of the AIAA 1st UAV Conference*, Portsmouth, Virginia, 2002.
- [38] R. Enns, and J. Si, "Helicopter flight control design using a learning control approach," *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 1754-1759, Sydney, Australia, 2000.
- [39] R. Enns, and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Transactions on Neural Networks*, Vol. 14, pp. 929-939, 2003.
- [40] J. D. Foley, A. Vandam, S. K. Feiner and J. F. Hughes, "Fundamentals of Interactive Computer Graphics." Addison Wesley, Reading, MA 1990.
- [41] S. G. Fowers, D. J. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis, and J. K. Archibald, "Vision Aided Stabilization and the Development of a Quad-Rotor Micro UAV," *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, USA, pp. 143-148, 2007.

- [42] G. Franklin, J. D. Powell and A. E. Naeini, *Feedback Control of Dynamic Systems*, 4th edn. Upper Saddle River, New Jersey: Prentice-Hall, 2002.
- [43] J. Gadewadikar, F. L. Lewis, K. Subbarao, K. Peng and B. M. Chen, “ H_∞ static output-feedback control for rotorcraft,” *Journal of Intelligent and Robotic Systems*, vol. 54, pp. 629–646, 2009.
- [44] V. Gavrillets, B. M. Mettler, and E. Feron, “Nonlinear model for a small-size acrobatic helicopter,” *Presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, 2001.
- [45] V. Gavrillets, A. Shterenberg and M. A. Dahleh, “Avionics system for a small unmanned helicopter performing aggressive maneuvers,” *Proceedings of the 19th Digital Avionics Systems Conferences*, Philadelphia, USA, pp. 1–10, 2000.
- [46] G. Gonzalez, and R. E. Woods, “Digital Image Processing,” Addison Wesley, Reading, MA, 1992.
- [47] W. E. L. Grimson, C. Stauffer, R. Romano and L. Lee, “Using Adaptive Tracking to Classify and Monitor Activities in a Site,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, pp. 22–29, 1998.
- [48] N. Guenard, T. Hamel and R. Mahony, “A practical visual servo control for an unmanned aerial vehicle,” *IEEE Transactions on Robotics*, vol. 24, pp. 331–340, 2008.
- [49] L. Gupta and M. Srinath, Invariant planar shape recognition using dynamic alignment *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Dallas, USA, pp. 217–220, 1987.
- [50] J. Ha, E. N. Jonhson and A. Tannebaum, “Real-Time Visual Tracking using Geometric Active Contours for the Navigation and Control of UAVs,” *Proceedings of the 26th American Control Conference*, New York, USA, pp. 365–370, 2007.

- [51] C. Harris, *Shock and Vibration Handbook* (edited), McGraw-Hill, New York, 1996.
- [52] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd edn., The Edinburgh Building, UK: Cambridge University Press, 2003.
- [53] R. He, A. Bachrach, M. Achtelik, A. Geramifard, D. Gurdan, S. Prentice, J. Stumpf and N. Roy “On the design and use of a micro air vehicle to track and avoid adversaries,” *The International Journal of Robotics Research*, vol. 29, pp. 529–546, 2010.
- [54] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction ,” *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition* , San Juan, Puerto Rico, pp. 1106–1112, 1997.
- [55] R. Horaud, B. Conio and O. Le Boulleux, “An analytic solution for the perspective 4-point problem,” *Proceedings of Computer Vision, Graphics, and Image Processing*, pp. 33-44, 1989.
- [56] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher and J. Roberts, “Combined optic-flow and stereo-based navigation of urban canyons for a UAV,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3309-3316, 2005.
- [57] M. K. Hu, “Visual pattern recognition by moment invariants,” *IEEE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.
- [58] W. M. Hu, T. N. Tan, L. Wang and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Transactions on Systems, man, and cybernetics*, vol. 34, pp. 334–352, 2004.
- [59] S. Hutchinson, G. D. Hager and P. I. Corke, “A tutorial on visual servo control, ” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651-670, 1996.
- [60] M. Isard and A. Blake, “Condensationconditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

- [61] A. Isidori, L. Marconi and A. Serrani, “Robust nonlinear motion control of a helicopter,” *IEEE Transactions on Automatic Control*, vol. 48, issue 3, pp. 413-426, 2003.
- [62] J. S. Jang and D. Liccardo, “Small UAV automation using MEMS”, *Aerospace and Electronic Systems Magazine, IEEE*, vol 22, pp. 30–34, 2007.
- [63] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha and J. C. Neidhoefer, “Real-time vision-based relative aircraft navigation,” *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 707–738, 2007.
- [64] B. Kadmiry, “Fuzzy control for an autonomous helicopter,” Thesis No. 938 for the degree of Licenciante of Engineering, Linkoping, Sweden, May 2002.
- [65] J. Kim and S. Sukkarieh, “SLAM aided GPS/INS navigation in GPS denied and unknown environments,” *Proceedings of the 2004 International Symposium on GNSS/GPS*, Sydney, Australia, 2004.
- [66] J. Kim and S. Sukkarieh, “Autonomous Airborne Navigation in Unknown Terrain Environments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 1031–1045, 2004.
- [67] S. K. Kim and D. M. Tilbury, “Mathematical modeling and experimental identification of an unmanned helicopter robot with flybar dynamics”, *Journal of Robotic Systems*, No. 3, Vol. 21, pp. 95-116, 2004.
- [68] M. Kolsch and S. Butner, “Hardware considerations for Embedded vision systems,” *Embedded Computer Vision*, pp. 3-26, 2009.
- [69] T. J. Koo and S. Sastry, “Output tracking control design of a helicopter model based on approximate linearization,” *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, 1998.

- [70] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao; Y. L. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen and P. Burt, "Aerial video surveillance and exploitation," *Proceedings of the IEEE* vol. 89, pp.1518 - 1539, 2001.
- [71] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, Part I: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, pp. 1333-1364, 2003.
- [72] F. Lin, K. Y. Lum, B. M. Chen and T. H. Lee, "Development of a vision-based ground target detection and tracking system for a small unmanned helicopter," *Science in China – Series F: Information Sciences*, vol. 52, pp. 2201-2215, 2009.
- [73] D. G. Lowe, "Three-dimensional object recognition form single two-dimensinal images," *Artificial Intelligence*, vol. 31, pp. 355-395, 1987.
- [74] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [75] C. P. Lu, G. D. Hager and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 610-622, 2000.
- [76] B. Ludington, E. Johnso and G. Vachtsevanos, "Augmenting UAV autonomy," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 63–71, 2006.
- [77] R. Lukac and K.N. PlataniotisColor, "Color Image Processing: Methods and Applications," 1st edn, CRC Press, 2006.
- [78] Y. Ma, J. Kosecka and S. S. Sastry, "Vision Guided Navigation for a Nonholonomic Mobile Robot," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 521-536, 1999.

- [79] Y. Ma, S. Soatto, J. Kosecka and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, 1th edn. 175 Fifth Avenue, New York: Springer-Verlag, 2004.
- [80] L. Mejias, S. Saripalli, P. Cervera and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, pp. 185-199, 2006.
- [81] M. Meingast, C. Geyer and S. Sastry, "Vision based terrain recovery for landing unmanned aerial vehicles," *Proceedings of IEEE Conference on Decision and Control*, Atlantis, Bahamas, pp. 1670-1675, 2004.
- [82] B. Mettler, *Identification, Modeling and Characteristics of Miniature Rotorcraft*, Kluwer Academic Publishers, Boston, MA, 2002.
- [83] K. Molen, "Feature tracking using vision on an autonomous airplane," Autonomous Systems Lab, Swiss Federal Institute of Technology, Feb 2004. [Online]. Available: <http://asl.epfl.ch/research/projects/VtolIndoorFlying/rapports/>
- [84] K. Nordberg, P. Doherty, G. Farneb, P. Forssén, G. Granlund, A. Moe and J. Wiklund, "Vision for a UAV helicopter," *Proceedings of IROS'02, workshop on aerial robotics*, Lausanne, Switzerland, pp. 1-6, 2002.
- [85] D. Oberkampf, D. F. Dementhon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Computer Vision and Image Understanding*, vol. 63, pp. 495-511, 1996.
- [86] K. Peng, M. Dong, B. M. Chen, G. Cai, K. Y. Lum and T. H. Lee, "Design and implementation of a fully autonomous flight control system for a UAV helicopter," *Proceedings of the 26th Chinese Control Conference*, Zhangjiajie, Hunan, China, pp. 662-667, 2007.

- [87] S. K. Phang, J. J. Ong, R. T. C. Yeo, B. M. Chen and T. H. Lee, "Autonomous mini-UAV for indoor flight with embedded on-board vision processing as navigation system," *Proceedings of the IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Irkutsk Listvyanka, Russia, pp. , 2010.
- [88] K. N. Plataniotis and A. N. Venetsanopoulos, "Color Image Processing and Applications." 1st edn, Springer, 2000.
- [89] F. Porikli, "Achieving real-time object detection and tracking under extreme conditions," *Journal of Real-Time Image Processing*, vol. 1, pp. 33-40, 2006.
- [90] J. V. R. Prasad, A. J. Calise and J. E. Corban, "Implementation of adaptive nonlinear controller for flight test on an unmanned helicopter," *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, 1998.
- [91] A. A. Proctor and E. N. Johnson, "Vision-Only Aircraft Flight Control Methods and Test Results," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, Rhode Island, pp. 1-16, 2004.
- [92] A. A. Proctor, S. K. Kannan, C. Raabe, H. B. Christophersen and E. N. Johnson, "Development of an Autonomous Aerial Reconnaissance System at Georgia Tech," *Proceedings of the Association for Unmanned Vehicle Systems International Unmanned Systems Symposium and Exhibition*, Baltimore, Maryland, pp. 1-9, 2003.
- [93] A. Puri, "A Survey of Unmanned Aerial Vehicles (UAV) for Traffic Surveillance," Department of Computer Science and Engineering, University of South Florida, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.108.8384>
- [94] , "Tracking Deforming Objects Using Particle Filtering for Geometric Active Contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1470-1475, 2007.

- [95] J. M. Roberts, P. Corke and G. Buskey. “Low-cost flight control system for a small autonomous helicopter.” *Proceedings of the 2002 Australian Conference on Robotics and Automation*, Auckland, New Zealand, pp. 546-551, 2002.
- [96] P. A. Rodriguez, W. J. Geckle, J. D. Barton, J. Samsundar, T. Gao, M. Z. Brown, and S. R. Martin “An Emergency Response UAV surveillance System,” *Proceedings of AMIA 2006 Symposium*, Washington, USA, pp. 1078, 2006.
- [97] F. Sadjadi, “Theory of invariant algebra and its use in automatic target recognition,” *Physics of Automatic Target Recognition*, vol. 3. pp. 23-40, 2007.
- [98] K. Sasajima and H. Fujimoto, “6 DOF Multirate Visual Servoing for Quick Moving Objects,” *Proceedings of the 2007 American Control Conference*, New York City, USA, pp. 1538–1543, 2007.
- [99] R. Sattigeri, E. Johnson, A. Calise and J. Ha “Vision-based target tracking with adaptive target state estimator,” *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, USA, pp. 1-13, 2007.
- [100] Z. Sarris and S. Atlas, “Survey of UAV applicatins in civil markets,” *Proceedings of the 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia, pp. 1–11, 2001.
- [101] C. S. Sharp, O. Shakernia and S. S. Sastry, “A Vision System for Landing an Unmanned Aerial Vehicle,” *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, pp. 1720-1727, 2001.
- [102] S. Saripalli, J.F. Montgomery, and G.S. Sukhatme, “Visually-Guided Landing of an Unmanned Aerial Vehicle,” *IEEE Transactions on Robotics and Automation*,
- [103] F. R. Schell and E. D. Dickmanns, “Autonomus landing of airplanes by dynamic machine vision,” *Machine vision and Applications*, vol. 7, pp. 127–134, 1994.

- [104] G. Schweighofer and A. Pinz, “Robust pose estimation from a planar target,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2024–2030, 2006.
- [105] G. Schweighofer and A. Pinz, “Robust pose estimation from a planar target,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2024–2030, 2006.
- [106] S. Shafer, “Using color to separate reflection components,” *Color Research and Application*, vol. 10, pp. 210–218.
- [107] O. Shakernia, Y. Ma, T. J. Koo and S. Sastry, “Landing an Unmanned Air Vehicle: Vision Based Motion Estimation and Nonlinear Control,” *Asian Journal of Control*, vol. 1 pp. 128–204, 1999.
- [108] O. Shakernia, R. Vidal, C. S. Sharp, Y. Ma and S. Sastry, “Multiple view motion estimation and control for landing an unmanned aerial vehicle,” *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, Washington, DC, USA, pp. 2793–2798, 2002.
- [109] C. S. Sharp, O. Shakenia and S. Sastry, “A vision system for landing an unmanned aerial vehicle,” *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 1720–1727, 2001.
- [110] Y. T. Shih, “The reversibility of six geometric color space.” *Photogrammetric Engineering and Remote Sensing*, vol. 61, pp.1223–1232, 1995.
- [111] D. Shim, H. Chung, H. J. Kim and S. Sastry, “Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles,” *AIAA GN&C Conference*, San Francisco, USA, pp. 1–8, 2005.
- [112] D. H. Shim, H. J. Kim and S. Sastry, “Control system design for rotorcraft-based unmanned aerial vehicle using time-domain system identification,” *Proceedings of the*

- 2000 *IEEE Conference on Control Applications*, pp. 808-813, Anchorage, Alaska, USA, 2000.
- [113] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, pp. 3621-3626, Maui, Hawaii, USA, 2003.
- [114] A. R. Smith, "Color gamut transform pairs," *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, New York, USA, pp. 12-19, 1978.
- [115] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, pp. 56-68, 1986.
- [116] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 3rd edn., Toronto, Ontario: Thomson, 2008.
- [117] K. Sprague, V. Gavrillets, and D. Dugail, "Design and applications of an avionic system for a miniature acrobatic helicopter," *Proceedings of the 20th Digital Avionics Systems Conferences*, Daytona Beach, USA, pp. 1-10, 2001.
- [118] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Fort, Collins, pp. 246-252, 1999.
- [119] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.
- [120] T. Y. Tian, C. Tomasi and D. J. Heeger, "Comparison of Approaches to Egomotion Computation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 315-320, 1996.

- [121] H. Veeraraghavan, P. Schrater and N. Papanikolopoulos, “Robust target detection and tracking through integration of motion, color and geometry,” *Computer Vision and Image Understanding*, vol. 103, pp. 121-138, 2006.
- [122] C. D. Wagter, A. A. Proctor and E. N. Johnson, “Vision-Only aircraft flight control,” *Proceedings of the 22nd Digital Avionics Systems Conference*, Indianapolis, IN, pp. 8B2.1–8B2.11, 2003.
- [123] E. A. Wan and A. A. Bogdanov, “Model predictive neural control with applications to a 6 DOF helicopter model,” *Proceedings of the 2001 American Control Conference*, pp. 488-493, Arlington, Virginia, USA, 2001.
- [124] M. W. Weilenmann and H. P. Geering, “A test bench for the rotorcraft hover control,” *Proceedings of AIAA Guidance Navigation and Control Conference*, Monterey, CA, 1993.
- [125] M. W. Weilenmann, U. Christen and H. P. Geering, “Robust helicopter position control at hover,” *Proceedings of American Control Conference*, pp. 2491-2495, Baltimore, MD, June-July 1994.
- [126] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland “Pfinder: Real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, 1997.
- [127] S. Yoshimura, and T. Kanade, “Fast template matching based on the normalized correlation by using multiresolution eigenimages,” *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 2086-2093, 1994.
- [128] H. Yu, R. Beard and J. Byrne, “Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles,” *Control Engineering Practice*, vol. 18, pp. 824–836, 2010.

- [129] B. Yun, K. Peng, B. M. Chen, “Enhancement of GPS signals for automatic control of a UAV helicopter system,” *Proceedings of IEEE International Conference on Control and Automation*, Guangzhou, China, pp. 1185–1189, 2007.
- [130] Q. Zhu, S. Avidan and K. T. Cheng, “Flexible Camera Calibration by Viewing a Plane from Unknown Orientations,” *Proceedings of the 7th IEEE International Conference on Computer Vision*, Kerkyra, Corfu, Greece, pp. 666–673, 1999.
- [131] Q. M. Zhou, and J. K. Aggarwalb, “Object tracking in an outdoor environment using fusion of features and cameras,” *Image and Vision Computing*, vol. 24, pp. 1244–1255, 2006.
- [132] Q. Zhu, S. Avidan and K. T. Cheng, “Learning a sparse, corner-based representation for time-varying background modelling,” *Proceedings of the 10th IEEE International Conference on Computer Vision*, Beijing, China, pp. 1–8, 2005.
- [133] AutoCopter Express UAV, <http://www.neural-robotics.com/Products/Express.html>.
- [134] Camera Calibration Toolbox for Matlab,
http://www.vision.caltech.edu/bouguetj/calib_doc/
- [135] NUS UAV Research, <http://uav.ece.nus.edu.sg>
- [136] OpenCV Wiki, <http://opencv.willowgarage.com/wiki/>.
- [137] QNX Neutrio RTOS, <http://www.qnx.com/>.
- [138] Raptor 90 SE helicopter, <http://www.tiger.com.tw/product/4891.html>.
- [139] Real-time operating system, http://en.wikipedia.org/wiki/Real-time_operating_system.
- [140] RTLinux RTOS, <http://www.rtlinuxfree.com/>.
- [141] Sky Surveyor UAV helicopters, <http://me2.tm.chiba-u.jp/>.

- [142] Visual C++ developer center, *<http://msdn.microsoft.com/en-us/visualc/>*.
- [143] VxWorks RTOS, *<http://www.windriver.com/products/vxworks/>*.

Published/Submitted Papers

Refereed Journal Articles:

1. F. Lin, K. Y. Lum, B. M. Chen and T. H. Lee, “Development of a vision-based ground target detection and tracking system for a small unmanned helicopter,” *Science in China – Series F: Information Sciences*, vol. 52, pp. 2201-2215, 2009.
2. G. Cai, F. Lin, B. M. Chen and T. H. Lee, “Systematic design methodology and construction of UAV helicopters,” *Mechatronics*, Vol. 18, No. 10, pp. 545-558, December 2008.

International Conference Articles:

1. G. W. Cai, F. Lin, B. M. Chen and T. H. Lee, “Development of fully functional miniature unmanned rotorcraft systems,” *To be presented at the 29 Chinese Control Conference*, Beijing, China, 2010.
2. F. Lin, B. M. Chen, K.-Y. Lum and T. H. Lee, “A robust vision system on an unmanned helicopter for ground target seeking and following,” *Proceedings of the 8th World Congress on Intelligent Control & Automation*, Jinan, China, pp. 276-281, July 2010.

3. F. Lin, B. M. Chen and T. H. Lee, "Vision aided motion estimation for unmanned helicopters in GPS denied environments," *Proceedings of the 2010 IEEE International Conference on Cybernetics and Intelligent Systems*, Singapore, pp. 64-69, June 2010.
4. X. Dong, G. Cai, F. Lin, B. M. Chen, H. Lin and T. H. Lee, "Implementation of formation flight of multiple unmanned aerial vehicles," *Proceedings of the 8th IEEE International Conference on Control and Automation*, Xiamen, China, pp. 904-909, June 2010.
5. F. Lin, B. M. Chen and T. H. Lee, "Robust vision-based target tracking control system for an unmanned helicopter using feature fusion," *Proceedings of the 11th IAPR Conference on Machine Vision Applications*, Yokohama, Japan, pp. 398-401, May 2009.
6. F. Lin, B. M. Chen and K. Y. Lum, "Integration and implementation of a low-cost and vision UAV tracking system," *Proceedings of the 26th Chinese Control Conference*, Zhangjiajie, Hunan, China, Volume 6, pp. 731-736, July 2007.