博士論文

Towards Human-Robot Interaction in Flying Robots: A User Accompanying Model and A Sensing Interface

(飛行ロボットにおける人間・ロボットインタラクションの実現に向けて:
ユーザー同伴モデルとセンシングインターフェース)

A PHD DISSERTATION

Submitted to the Graduate School of Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

AERONAUTICS AND ASTRONAUTICS ENGINEERING

リュー ジュン フィ

Chun Fui Liew

August 2016

Candidate:	Chun Fui Liew
Chief examiner:	Takehisa Yairi
Examiner:	Koichi Hori
Examiner:	Akira Iwasaki
Examiner:	Takeshi Tsuchiya
Examiner:	Hiroshi Mizoguchi

UNIVERSITY OF TOKYO, 2016

ABSTRACT

Flying robots are one of the most important inventions in robot history. Widely known as drones or unmanned aerial vehicles (UAVs), flying robots have a wide range of applications and bring major impacts to academia, commercial sectors, government, and society.

Over the past few decades, flying robot research has focused on autonomy, which includes sensing, control, localization, mapping, and planning, while paying less attention to the sociability of the flying robots. Focusing on both autonomy and sociability, this thesis proposes the concept of *companion* flying robots, where the flying robots are expected to accompany and interact with us in our daily lives. Among all the challenging tasks involved, we believe that achieving a safe, intuitive, natural, and social flying behavior is of vital importance for a *companion* flying robot. Having this priority in mind, we focus on three topics to realize the goal of *companion* flying robots: (i) a holonomic hexacopter for human-robot interaction (HRI), (ii) a general model for human accompanying, and (iii) a sensing interface for human understanding.

First, we design a new form of hexacopter that has several merits for HRI in companion flying robots. As opposed to a conventional flying robot, the proposed holonomic hexacopter is able to maintain attitude while flying horizontally; it does not need to tilt. As a result, the holonomic hexacopter provides intuitive flight motions from the user's point of view and achieves better flight stability from the control perspective. The holonomic hexacopter also provides a stable video feed for high level tasks such as human detection without the need of an additional heavy gimbal platform. Moreover, thanks to its six degree-of-freedom motion, the holonomic hexacopter can produce 3D force naturally in the air and achieves safer physical interaction with user without a complex dynamic model.

Second, we design a general model to unify various human accompanying behaviors of a *companion* flying robot. Human accompanying, including human approaching, following, leading, and side-by-side walking are important behaviors of a *companion* flying robot. Robots to-date focus on one or two human accompanying modes; there is no existing work to unify these modes for robots to achieve natural and rich interaction with humans. In this work, we propose a two-level model to achieve this goal. At the top level, we adopt a hierarchical finite state machine (FSM) to organize the behavior flow of a *companion* flying robot. The hierarchical FSM has the merits of simplicity and expandability, where the robot, environment, and human states can be incorporated into the model to achieve a

rich HRI behavior with a person. At the bottom level, we use a relative positioning control method for robots to achieve smooth and natural accompanying motions. While the top-level hierarchical FSM alone can be viewed as a rule-based approach, together with the bottom level relative positioning controller, they form a powerful hybrid approach that is able to achieve natural and rich HRI behaviors with minimal computational load.

Finally, we design a human sensing interface for *companion* flying robots to have a better understanding of user. We study, implement, and improve several human sensing techniques, such as human detection, human body orientation estimation, hand detection, hand shape recognition, facial expression recognition, and face alignment. While we only focus on human detection with our companion flying robot in this work, we aim to integrate multiple human sensing techniques for a more advanced *companion* flying robot in the future.

Acknowledgments

I am grateful to many people who have supported my research and life since I came to Japan to start my graduate study five years ago in 2011. My experiences in Japan would not be the same without them.

First, I sincerely thank Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan for generously supporting my tuition fees at the University of Tokyo and offering a living allowance for me to stay in Tokyo for five years. During my graduate study, I am also very thankful to the Tateishi Science and Technology Foundation for offering me an one-year research grant and supporting my research generously.

I would like to express my sincere gratitude towards my supervisor Prof. Takehisa Yairi for his full support in my research and for always encouraging me to create my own research field. Along the path, I modified my research topics a few times, and I really appreciate his patience and guidance. This thesis would not be the same without his continuous inspiration, motivation, and countless advice and feedback. Throughout my research life at the University of Tokyo, he has been gradually widening my network with professors, industrial partners, and colleagues. In addition to my academic performance, he provides useful advice in helping me to extend my scholarship, in applying research grants, and in adapting to my new life in Tokyo. I could not have imagined having a better advisor and mentor for my graduate study and life experience in Tokyo.

Besides my advisor, I would like to thank my thesis committee—Prof. Akira Iwasaki, Prof. Koichi Hori, and Prof. Takeshi Tsuchiya from the University of Tokyo, and Prof. Hiroshi Mizoguchi from the Tokyo University of Science for their valuable comments and suggestions towards improving this thesis. Special thanks Prof. Iwasaki for always encouraging me to pursue my dream while I am still young. I am also grateful to Prof. Hori for always encouraging me to look for my own research world.

Throughout my research, I am also very fortunate to receive academic guidance from Prof. Yoshinobu Kawahara from Osaka University, Japan, Prof. Junsong Yuan from Nanyang Technological University, Singapore, and Prof. Chee Seng Chan from University of Malaya, Malaysia. I sincerely thank them for sharing their research insights with me and I learned a lot about research from them.

I am blessed to have been surrounded by amazing colleagues through my research life in the Space Application Laboratory, University of Tokyo. Hiroaki Shioi and Yuki Itoh were my kind tutors when I first joined the Department of Aeronautics and Astronautics, and we have been good friends since. I am also thankful for Takaaki Tagawa, who first showed me an abandoned store room in our laboratory that is full of valuable robots, sensors, and equipment. That hardware would later play an important role in my research. I also learned a lot about topics on computer vision from Naoto Yokoya, topics on robotics from Yasuyuki Takeuchi, and topics on remote sensing from Taichi Takayama. I also appreciate research discussions and Japanese cultural experience with every member in Space Application Laboratory, including Junichi Kuwabara, Riki Eto, Dai Okano, Sho Nakamura, Takeshi Arai, Ryoma Yasunaga, Shinji Nakazawa, Tetsuo Oda, Akira Tanimoto, Naoya Takeishi, Toshiharu Tashiro, Mayu Sakurada, Mari Ito, Haruki Nishimura, and Daisuke Niina.

I am very grateful to work with awesome colleagues in the Artificial Intelligence Laboratory, University of Tokyo. Having flying robotics and hardware development discussions with Ryo Nakamura, Kenya Kaneda, Motohiro Hiraoka, Petr Khrapchenkov, and Takahiro Miki always inspire me and sped-up my research progress significantly. Special thanks to Nakamura for sharing his passion for flying robots with me when I first joined the Artificial Intelligence Laboratory, thanks Kaneda for offering me a position in his robotic startup company, and thanks Hiraoka for helping me in software development generously. I am also very fortunate to meet Danielle DeLatte and Mario Rodriguez. I enjoy brainstorming new ideas and working with them on human-drone interaction research. I would like to thank DeLatte for spending her precious time to help me proofread this thesis. I also appreciate Rodriguez's generous help in many of the flying robot experiments.

I am also indebted to many members from the MultiWii, BaseFlight, CleanFlight, and iNavClight projects, for devoting tremendous effort and time to make open-source flight controllers (both hardware and software) possible. Building a new flying robot for human-robot interaction (HRI) in such a short time would not have been possible without their remarkably rapid development, constant feedback, and valuable suggestions.

I would like to extend my sincere thanks to Ms. Mami Kubota from the Space Application Laboratory, Ms. Mitsuko Kimura and Ms. Tomoe Minemura from the Department of Aeronautics and Astronautics, Ms. Yoshiko Shiraishi at the Office of International Students, staff at the International Center, and staff at the Research Center of Advanced Science and Technology for their kind advice and continuous administrative support.

I am grateful to meet with Dr. Hiroshi Yotsuyanagi and Dr. Yukiko Inoue from the University of Tokyo Hospital. I specially thank them for their kindness and continuous effort in ensuring my health since I came to Japan.

I am thankful to meet Ms. Fukuko Sekine, Ms. Akiko Inoue, and Ms. Tomisaki from the Jishu-gakusha dormitory, Ms. Noriko Shirai from the Mitsui V-net, and Ms. Chie Miyamoto from the Soshigaya International House volunteering group. My life in Tokyo would not be so fantastic without the cultural events and diverse one-day trips that they have organized.

My special thanks to Ms. Tanaka, Ms. Oura, and Ms. Shimizu from the Co-op canteen at Komaba campus for their warm words always. I also appreciate the responsive purchasing services provided by Ms. Ofusa and Mr. Imai from the Yatoro-denshi company.

I am also fortunate to know Prof. Nobuo Takeda and Ms. Takiko Hirano, who moved to the office next to my experiment room. Their warm greetings and kind words always enabled me to work a little bit harder and travel one step closer to my research goal.

I am also indebted to many diplomatic managers and officers in the Embassy of Malaysia, Paris and the High Commission of Malaysia, London. I appreciate all their kind help when I had a terrible accident during my conference trip in Europe in 2014. I would like to specially thank Mrs. Anil Fahriza Adenan, Mrs. Juliana Razid, Mrs. Nurulazunia Yusof, Mrs. Rogayah Decombredet, Mr. Arun, Mr. Rosni Ahmad Shah, and Mr. Faizal.

I am lucky to know many friends in Japan, including Cindy Wong, Jack How, Cheryl Tan, Chloe Cheow, Pearly Wong, Jeffrey Tan, Loo Mun, Jeannette Aiko, Yee Seng Teh, Soh Yeep Ooi, Serene Thong, Michael Tiong, Yau Kiong Lew, Voon Yau Chew, Seng Pei Liew, Lee Fueng Yap, Ching Li Goh, Jeng Kae Tan, Hui Zee Then, Dandy Kwong Ling, Hui Yee Yong, Meiyun Chen, Nobutada Yokouchi, Echo Zhang, and Tana Qian. Having a break with them occasionally enables me to pursue my research goal again in full power mode.

I am also blessed to be surrounded by loving friends around the world, including Mei Hwee Hoe, Ka Wuei Chong, Lin Wei Wee, Ester Ng, Chian Huey Wong, Alex Nyiam, Li Yin Nyiam, Hanny Liew, Max Lee, Wei Leng Chin, Jia Xuan Hon, Ee Ling Tan, Han Meng Teo, Jia Yun Lee, Seong Fui Chew, Jason Mak, Luther Lee, Min Lim, Ven Jyn Kok, William Hoo, Mei Kuan Lim, Chin-wen Chang, Kain Lu Low, Yinbei Li and Xian Qu.

Last but not the least, I would like to dedicate my achievements to my loving parents, my brothers, and my sister. This work would not have been possible without their continuous encouragement and spiritual support throughout my graduate study in Japan and my life.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation: Companion Flying Robots

John used to jog alone near his house after a long work day. But now things have changed. Every day, his companion flying robot, Mamoru, jogs with him. Mamoru likes to follow him during jogging, just like a pet chasing its master. John feels happier and more motivated when jogging with Mamoru. Sometimes, when the jogging path is more spacious, Mamoru will fly to his side and jog side-by-side with him. There were a few times where John ran too far away from his house and got lost. There was no worry as Mamoru could always guide him back. Today John ran along the river at sunset. Through hand gestures, he asked Mamoru to fly across the river to take a photo for him. Mamoru did a great job, and John smiled happily. Interestingly, Mamoru understood his emotion through facial expression recognition and 'jumped' around in the sky to express its happiness. John laughed and had a deeper connection with Mamoru.

Figure 1.1 illustrates a scene when John is jogging with *Mamoru*. *Mamoru* is one kind of companion robot, which can perform various human-robot interaction (HRI) tasks with us. In addition to accompanying us during jogging and taking photo for us like a friend, *Mamoru* can also accompany our children to school like a bodyguard. We have a vision that in the near future, everyone will have a companion flying robot in our daily lives, just like we have a personal computer in our home or a smartphone in our pocket.

While the concept of "companion robot" is not completely new, companion flying robots has very different features than other companion robots like humanoid robots and pet robots. Specifically, flying robots have two major advantages. First, they can hover and move freely in a 3D space. This could be a very useful feature in practice. For example, imagining a person is walking up stairs. A conventional mobile robot without sophisticated mechanics that allow it to "climb" up the stairs would have a hard time to continue following the person. Similarly, a humanoid robot without a robust walking algorithm would have difficulties to continue following the person. In contrast, a flying robot is not affected by the stairs and can continue following the person. Second, flying above the ground enables flying robots to see the surrounding environment with a better field of view and less chance to

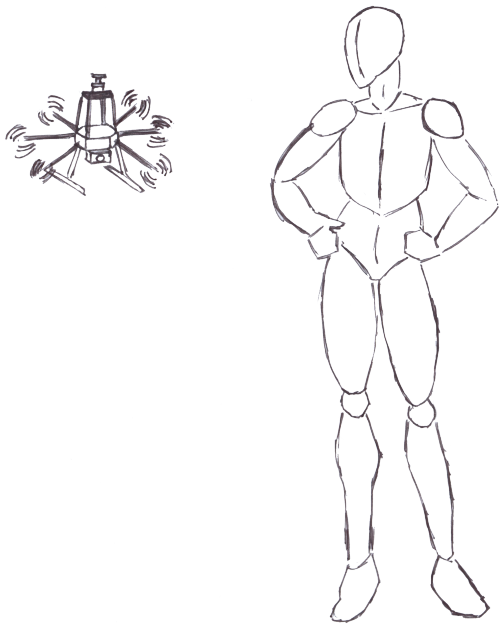

Figure 1.1: A sketch of the concept of a companion flying robot.

be blocked by obstacles. This is especially useful when companion robots perform routine tasks such as detecting a person, performing localization, or planning a path.

1.2 Position of Companion Flying Robots

In literature, researchers have been using many terms that have a similar meaning with the term "flying robot". These include unmanned aerial vehicle (UAV), micro-aerial vehicle (MAV), unmanned aircraft system (UAS), aerial robot, autonomous aircraft, robotic aircraft, vertical take-off and landing (VTOL) aircraft, multicopter, multirotor, and drone. In this dissertation, we will use the terms "flying robot", "UAV", and "drone" interchangeably.

By defining vertical axis as degree of autonomy and horizontal axis as degree of sociability, we illustrates the position of a companion flying robot in **Fig. 1.2**. Traditionally, flying robots are mostly controlled manually by human operators (low degree of autonomy) and they have low sociability. We call these flying robots "remote control UAV." Gradually, along the vertical axis of autonomy, robotic researchers have been improving the autonomy

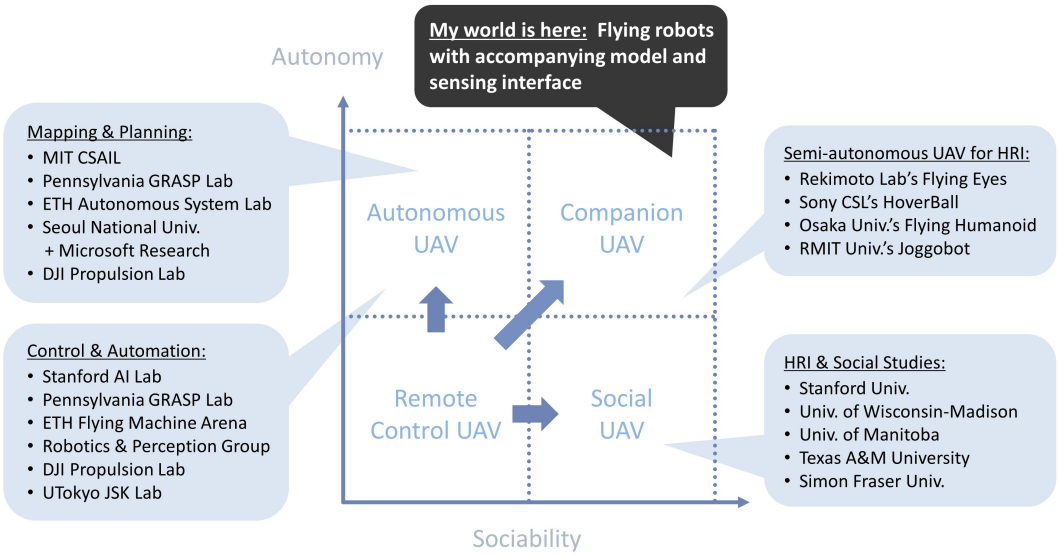

Figure 1.2: Position of a companion flying robot (autonomy + sociability = companion).

aspects of flying robots such as better modeling, more robust control, and path planning algorithms. We call these flying robots "autonomous UAV." Essentially, autonomous UAVs are less dependent on human operators and are able to perform some simple flight tasks autonomously.

On the other hand, along the horizontal axis of sociability, HRI researchers have been improving the social aspects of flying robots such as designing a flying robot for safer HRI, developing a motion planning model that is more comfortable to humans for flying robots, and building an intuitive communication interface for flying robots to understand humans. We call these flying robots "social UAV." Different from autonomous UAVs, social UAVs often have low degree of autonomy. Most HRI researchers solely focus on social aspects and manually control a flying robot during HRI experiments.

We see companion flying robots as both autonomous and social flying robots. In addition to the autonomy aspects, such as aircraft modeling, stabilization control, and motion planning, companion flying robots must also focus on the sociability aspects such as safe HRI and intuitive communication interface for HRI.

1.3 Challenges towards a Companion Flying Robot

Designing a companion flying robot is not a trivial task. First of all, the companion flying robot needs to ensure the safety of user, its nearby objects, and the robot itself during flight operation. In addition, it involves multi-disciplinary knowledge across hardware design, control engineering, machine learning, social robotics, ergonomic design, and law enforcement fields.

Safety issue. Compared to a balloon-type flying robot, a rotor-type flying robot has the merits of higher mobility and less susceptible to wind disturbance during flights. However, due to their inherent characteristic, a rotor-type flying robot usually needs to roll and pitch in order to move horizontally. While moving horizontally, since the flying robot changes its attitude, a few problems rise:

- From the control perspective, the flying robot is less stable (more likely to crash and injure people around).
- The flying robot is susceptible to wind disturbance (less precise flight and possibly hit the user) because it cannot generate horizontal thrust to counter the wind disturbance.
- Horizontal flight motions not natural and intuitive. From our experience, novice users always feel panic when they make the flying robot to move horizontally.

Hardware design. An ideal companion flying robot should have long flight time and be small to fit in the palm of our hand. However, with today's battery technology available on the market, a palm-size flying robot could hardly fly longer than ten minutes. Besides, a flying robot also has the noise vs. mobility issue. On one hand, the rotor-type flying robot has high mobility but noises produced by the propellers make users feel uncomfortable and unsafe. On the other hand, a balloon-type flying robot is quiet but has low mobility and is strongly subjected to wind disturbance. Robotic researchers are still actively developing new hardware prototypes, new system modeling, and new control algorithms in order to realize a more robust flying robot.

Control engineering. Flying robots have a few major challenges in control engineering—it needs to be actively controlled for flight stabilization. More importantly, most flying robots have coupled motion control, that is, roll/pitch orientations and horizontal motions cannot

be controlled separately. This is a huge limitation as it makes all the conventional algorithms developed for ground mobile robots not directly applicable to a flying robot. We believe that these challenges are the reasons that makes the development progress of a companion flying robot slow.

Machine learning. Researchers are still actively developing localization, mapping, path planning, and obstacle avoidance algorithms for flying robots. Current state-of-the-art methods still have much room for improvements in order to deal with moving objects, sensor or environmental noises, and flying robots' limited onboard computational resources.

Social robotics. Current state-of-the-art flying robots focus on autonomy and lack of sociability. For instance, in the case of *Mamoru*, instead of always following John at a fixed distance from behind, *Mamoru* should accompany John in a natural way like a living being. When the time is right, either stimulated by the environment context or John's direction, *Mamoru* can fly side-by-side with John, guide John at front, or carry out other accompanying modes.

Ergonomic design. As flying robot technology advances, the ergonomic design of the hardware would become more important. For example, instead of designing an optimal flying robot from the control and engineering perspective, we should also consider its safety and psychological effect onto the users. This is a very important topic in the HRI field. Without these considerations, humans and flying robots cannot carry out tasks efficiently.

Law enforcement. Many countries are imposing new laws for flying robots in order to ensure public safety. For example, in Japan, if a flying robot weights more than 200 grams, it is forbidden to fly the robot in populated area (more than 4,000 people per square kilometer). In addition, flights are only allowed in the daytime within visual line of sight. Flight pilot also needs to maintain a certain operating distance between the flying robot and the human operator or properties on the ground. In general, it is very hard to fly a robot outdoor without permission and careful consideration in many countries.

1.4 Scope and Contributions of This Dissertation

The multi-disciplinary aspects mentioned in the previous section are the issues that we need to tackle in order to realize a companion flying robot like *Mamoru*. Overcoming all of

them at once is difficult, if not impossible. Among the listed challenges, we believe that (i) safety (of the flying robot and its user), (ii) natural HRI in between the flying robot and its user, and (iii) an interface for companion flying robot to understand human are the three most important topics toward our goal. Therefore, we focus on three main topics in this dissertation—a holonomic hexacopter, a human accompanying model and a human sensing interface. In this section, we discuss the relationships between the three main topics and summarize our contributions in this dissertation.

Figure 1.3 illustrates the contributions of the three chosen topics toward the goal of a companion flying robot. The holonomic hexacopter enables intuitive flight motions and possesses enhanced flight stability; the human accompanying model has two levels—the top level acts as a global behavior controller for companion flying robot to achieve rich HRI while the bottom level acts as a local motion controller for companion flying robot to achieve natural and smooth accompanying motions; the human sensing interface focuses on human/user understanding by using onboard camera and sensors. As shown in the figure, the three topics are connected to each other. For example, the holonomic hexacopter provides the robot and environment states for the human accompanying model. Thanks to the holonomic flight, the hexacopter is also able to provide a stable video feed for the human sensing interface to perform high-level tasks without the need of a mechanical gimbal. On the other hand, the human sensing interface provides the human states information for holonomic hexacopter and human accompanying model. Finally, the human accompanying model process all the received human, robot, and environment states information and output high-level commands to holonomic hexacopter and human sensing interface to perform robot control and human accompanying.

Figure 1.4 illustrates an overview of this dissertation. In the following, we present our contributions in each chapter of this dissertation:

- **Chapter 2**

  First, we present a research survey on flying robots, focusing on papers published at four top robotic conferences and four top robotic journals from 2006 to 2016. Interestingly, by categorizing the related papers into several common robotic topics, we found a pattern that is reminiscent of the mobile robots research. We provide a big picture of the research trend in flying robots and list up state of the arts achieved by some autonomous flying robots. After, we review HRI researches on flying robots in detail.

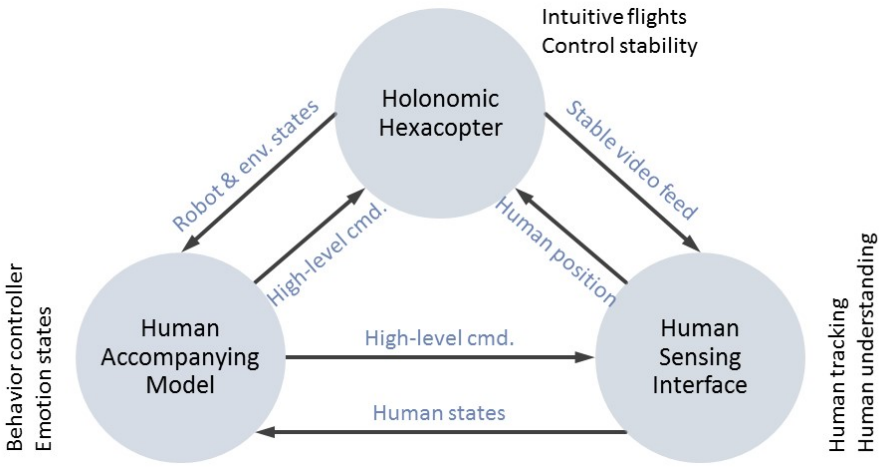

Figure 1.3: Contributions of and connections in between each chosen topics toward the goal of a companion flying robot.

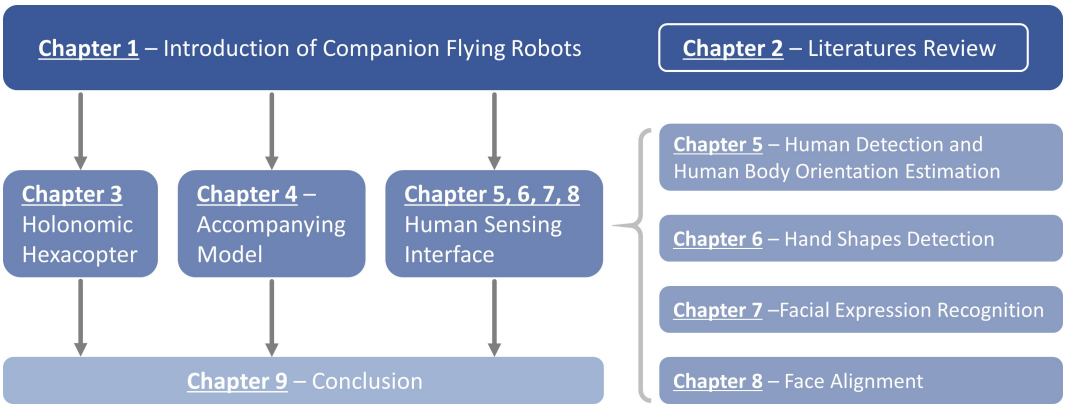

Figure 1.4: Dissertation overview.

- **Chapter 3**

  Most flying robots, such as helicopters, quadcopters, hexacopters, and octocopters, have coupled motion control; they must perform pitch/roll motion in order to move forward-backward/leftward-rightward. In other words, horizontal motions cannot be controlled independently without changing roll/pitch orientations. This is undesirable. Instead, we adopt a unique design such that a hexacopter can move horizontally while maintaining its level orientation. We derive the kinematic model and show that the new design can achieve holonomic motions. Moreover, it can be viewed as a generalization of the conventional hexacopter. The holonomic design leads to three main merits: (i) easier navigation with decoupled control, (ii) stable video feed without additional gimbal, and (iii) safer HRI.

- **Chapter 4**

  Accompanying a human is an important topic not only for companion flying robots but also for mobile robots. Different from previous works, where the robots focus on one single mode, our human accompanying model can be viewed as a generalized framework that unifies multiple modes such as approaching a person from a distance, following a person from behind, walking side-by-side with a person, leading a person at the front, and circling a person. We consider three important aspects in our model—human's various states, environmental context, and robot's states. This model is crucial for flying robots to achieve a natural accompanying behavior next to a person and excel in high level autonomy tasks.

- **Chapter 5**

  Human sensing techniques, such as human detection, body orientation estimation, hand shape detection, facial expression recognition, and facial feature tracking are indispensable skills for companion robots to understand humans. We believe that a human sensing interface integrated with multiple human sensing techniques is imperative for long term development of a robust companion robot. This is not easy as it involves huge efforts on literature survey, implementation, testing, and system integration. Developing such a comprehensive human sensing interface is our goal towards a companion flying robot. We will discuss our contributions in four human sensing techniques in **Chapter 5–8**.

In **Chapter 5**, we present our first human sensing technique—a robust technique to detect a human upper body and estimate human body orientation (HBO) in real-time. Instead of using multi-class detections or relying on filtering methods such as particle filters to improve the tracking performance, our method uses random forest to perform a continuous and full 360° HBO regression. In addition, we find that since the HBO labels (-180°<θ<180°) are not continuous in the regression space, the estimation results near to the -180° and 180° regions deteriorate significantly. To tackle this problem, we propose a xy-based random forest regression that could improve the estimation accuracy in both theory and the actual experiments. Our experiment results show that the proposed method successfully estimates HBO in real-time with regression error less than 25° for color images and less than 10° for depth images.

- **Chapter 6**

  We present our second human sensing technique—hand detection and hand shape recognition. Hand detection and hand shape recognition are of vital importance for HRI. Most previous works rely on image processing approaches such as skin detection and depth thresholding for hand detection. These techniques are restricted by strong assumptions and normally have low robustness in actual applications. In this chapter, we focus on an appearance-based approach and propose a new feature extraction method based on sparse pixel-pairwise intensity comparisons for hand detection. Our method can be viewed as a generalized BRIEF descriptor and can be easily adopted for other object detection or recognition tasks. We perform extensive experiments and prove that our method achieves comparable results with normal, noisy, and occluded hand images in term of both test accuracy and ROC. Our contribution includes: (i) a new and simple feature extraction method that is robust against image noise, cluttered backgrounds, and partial occlusion; (ii) combined with AdaBoost, we show that the new feature descriptor is effective for hand detection; (iii) the new feature descriptor has been rigorously compared with existing feature descriptors with a new hand database that has very challenging image backgrounds.

- **Chapter 7**

  We present our third human sensing technique—facial expression recognition. Facial expression recognition (FER) is a crucial technique for HRI. Previous methods have been using different feature descriptors for FER but there is a lack of comparison

studies. In this chapter, we aim to identify the best features descriptor for FER by empirically evaluating *five* feature descriptors, namely Gabor, Haar, local binary pattern (LBP), histogram of oriented gradients (HOG), and binary robust independent elementary features (BRIEF) descriptors. We examine each feature descriptor by considering *six* classification methods, such as k-nearest neighbors (k-NN), linear discriminant analysis (LDA), support vector machine (SVM), and adaptive boosting (AdaBoost) with *four* unique facial expression datasets. In addition to test accuracies, we present confusion matrices of FER. We also analyze the effect of combined features and image resolutions on FER performance. Our study indicates that the HOG descriptor works the best for FER when image resolution of a detected face is higher than 48×48 pixels.

- **Chapter 8**

  We present our fourth human sensing technique—face alignment, which is useful for flying robots to estimate human head pose and have a better understanding of human intention. Recently, it has been shown that random forest regressor can achieve accurate and fast local landmark estimation when coupled with a global face shape regularizer. In this chapter, we extend this approach and propose a new local forest classification and regression (LFCR) framework in order to handle face images with large yaw angles. We analyze each system component through detailed experiments. In addition to the selection of feature descriptors and several important tuning parameters of the random forest regressor, we examine different initialization and shape regularization processes. We compare our best outcomes to the state of the art and show that our method outperforms other parametric shape fitting approaches.

- **Chapter 9**

  We summarize our results towards realizing a companion flying robot and discuss various potential future works and applications.

1.5 List of Publications

Most contributions described in this dissertation have first appeared in the following publications:

- **Chapter 2**

  C. F. Liew and T. Yairi, "Quadrotor or blimp? Noise and appearance considerations in designing social aerial robot," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2013, pp. 183–184.

- **Chapter 3**

  C. F. Liew and T. Yairi, "Towards a compact and autonomous hexacopter for human robot interaction," in *Proc. Annual Conference of Robotics Society of Japan (RSJ)*, September 2015, pp. 1–4.

  C. F. Liew and T. Yairi, "Designing a compact hexacopter with gimballed lidar and powerful onboard Linux computer," in *Proc. IEEE International Conference on Information and Automation (ICIA)*, August 2015, pp. 2523–2528.

- **Chapter 5**

  C. F. Liew and T. Yairi, "Human body orientation estimation with xy-based random forest regression (submitted)," in *Proc. 23rd International Conference on Pattern Recognition (ICPR)*, 2016.

- **Chapter 6**

  C. F. Liew and T. Yairi, "Generalized BRIEF: A novel fast feature extraction method for robust hand detection," in *Proc. 22nd International Conference on Pattern Recognition (ICPR)*, August 2014, pp. 3014–3019.

  C. F. Liew and T. Yairi, "A new RGB-Depth hand shape database for natural gesture interaction," in *Proc. 16th Meeting on Image Recognition and Understanding (MIRU)*, June 2013, pp. 1–2.

- **Chapter 7**

  C. F. Liew and T. Yairi, "Facial expression recognition and analysis: A comparison study of feature descriptors," in *IPSJ Transactions on Computer Vision and Applications*, vol. 7, pp. 104–120, August 2015.

  C. F. Liew, "Machine learning techniques for facial expression recognition - A comparison study of feature spaces and classification methods," Master's thesis, Department of Aeronautics and Astronautics, The University of Tokyo, Japan, 2013.

C. F. Liew and T. Yairi, "A comparison study of feature spaces and classification methods in facial expression recognition," in *Proc. International Conference on Robotics and Biomimetics (ROBIO)*, December 2013, pp. 1294–1299.

- **Chapter 8**

  C. F. Liew and T. Yairi, "Robust face alignment with random forest: Analysis of initialization, landmarks regression, and shape regularization methods," in *IEICE Transactions on Information and Systems*, vol. E99-D, no. 2, pp. 496–504, February 2016.

  C. F. Liew, N. Yokoya, and T. Yairi, "Face alignment by using sparse initialization and random forest," in *Proc. IEEE International Conference on Image Processing (ICIP)*, October 2014, pp. 1–6.

Chapter 2
Related Works

2.1 Background

Looking back into the past, most of the flying robots are in fixed wing [1–3], flapping wing [4–6], and helicopter [7] forms. These flying robots used to be rare in the research laboratories due to the high cost of sensors, control difficulties, and high maintenance cost. However, flying robots started evolving at an unprecedented speed since pioneers like X-UFO [8], STARMAC [9], and MIT Drone [10] adopted the quadcopter form. This new form of flying robots makes control significantly easier and has lower maintenance cost compared to fixed wing, flapping wing, and helicopter forms. Furthermore, the cost of processors and sensors such accelerometer, gyroscope, magnetometer, and barometer dropped significantly in the late 2000s. Thanks to these reasons, the development progress of flying robots in all research, industrial, commercial, and Do-It-Yourself (DIY) sectors have been advancing rapidly over the past few years.

Among the commercial flying robots, DJI [11] and 3DR [12] are perhaps the most famous companies to date. DJI produces a wide range flying robots, including its Phantom series that focus on the aerial photography market [13], Matrice 100 platform for the research community [14], and AGRAS MG-1 platform for precision agriculture applications [15]. On the other hand, 3DR focuses on open source systems such as the Solo flying robots [16] and the Pixhawk flight controller [17]. In addition to the applications mentioned above, flying robots have huge possibilities and essentially can be viewed as a flying smartphone [18]. Potential applications includes package delivery [19], aerial mapping [20], security enhancement [21], fire monitoring [22], infrastructure inspection [23], search and rescue [24], entertainment [25], robotic education [26], and even drone racing [27].

In this chapter, we provide a big picture view of the research trends in flying robots and list state-of-the-art achievements by some autonomous flying robots. After, we review HRI research of flying robots in detail. **Table 2.1** summarizes the features comparison in between our companion flying robot and others. As mentioned in the **Sect. 1.4**, our companion flying robot focuses on three main points:

- Achieve safer HRI by designing a holonomic platform with decoupled control and

Table 2.1: Feature comparison in between our companion flying robot and others.

	Decoupled control	Stable video feed	Human detection	Human body orientation estimation	Accompanying model	Obstacle detection	Lightweight [1]
Joggobot [28]	×	×	△ [2]	×	△ [3]	×	×
FHR [29]	×	×	×	×	△ [4]	×	◯ [5]
Pestana et al. [30]	×	×	△ [6]	×	△ [7]	×	×
Higuchi et al. [31]	×	×	△ [8]	×	△ [9]	×	×
Papachristos et al. [32]	×	×	△ [10]	×	×	×	×
3DR Solo [16]	×	△ [11]	△ [12]	×	×	×	×
Lily Camera [33]	×	△ [13]	△ [12]	×	△ [9]	×	×
Hover Camera [34]	×	×	◯	×	△ [9]	×	◯
Parrot Bebop [35]	×	◯ [13]	×	×	×	×	×
DJI Phantom 4 [36]	×	◯ [11]	△ [14]	×	△ [9]	△ [15]	×
Lim & Sinha [37]	×	×	◯	×	△ [6]	△ [16]	×
Naseer et al. [38]	×	×	◯	×	△ [6]	△ [15]	×
Higuchi et al. [39]	◯	△ [17]	×	×	×	×	×
Ours	◯	◯ [18]	◯	◯	◯	△ [19]	×

[1] Less than 250 grams.
[2] Marker tracking.
[3] Leading only.
[4] Approaching only.
[5] Balloon flying robot.
[6] Manual initialization.
[7] Following only.
[8] Color tracking.
[9] Following and circling only.
[10] Moving objects only.
[11] 3D gimbal hardware.
[12] GPS tracking.
[13] Image processing techniques.
[14] Color & depth tracking.
[15] Frontal only.
[16] Subjected to available map.
[17] Hard to mount camera.
[18] Holonomic flight.
[19] 2D, 240° only.

stable video feed without additional gimbal. (**Chapter 3**).

- Achieve richer HRI by developing a human accompanying model that could unify approaching, following, side-by-side walking, circling, and other flying modes (**Chapter 4**).

- Achieve better HRI by integrating human sensing interfaces such as human detection and human body orientation estimation with flying robots (**Chapter 5**).

2.2 State-Of-The-Art Flying Robots

Figure 2.1 illustrates a pie chart of the topic distribution of flying robots, focusing on papers published on four top robotic conferences (IEEE International Conference on Robotics and Automation (ICRA), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), ACM/IEEE International Conference on Human-Robot Interaction

(HRI), IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)) and four top robotic journals (The International Journal of Robotics Research (IJRR), IEEE Transactions on Robotics (TRO), IEEE/ASME Transactions on Mechatronics (TME), Robotics and Autonomous Systems (RAS)) from 2006 to 2016. From the chart, we can observe that there are large amount of *hardware* papers in the past ten years, including papers (to name a few) on quadcopters [40–44], hexacopters [39, 45–47], octocopters [48–50], coaxial flying robots [51–56], a helicopter [57], a tandem helicopter [58], a bicopter [59], a Y4 flying robot [60], a trirotor flying robot [61], flying robots with fixed wings [62, 63], rotary wings [64, 65], and flapping wings [66–70], and balloon or blimp robots [71, 72].

As expected, *control* papers are closely related to *hardware* papers and are the second largest pie in **Fig. 2.1**. It has been shown that a simple model-free PID controller is good enough for the basic maneuvers of a flying robot [40, 41, 43, 55, 63, 71]. For aggressive maneuvers [73, 74] or more complex dynamics with onboard manipulators [75, 76], dynamic models are normally employed. With a precision indoor positioning system, current state-of-the-art methods have successfully demonstrated formation flights [77], flying inverted pendulum [78], pole acrobatics [79], ball juggling [80], cooperative operation [81–83], and failure recovery [84].

In recent years, researchers start to focus on higher level tasks such as navigation and task planning in flying robots [85–87]. In addition, researchers also pay attention to visual odometry [88, 89], localization [90–93], and mapping [94–97] applications of flying robots. More recently, researchers work on obstacle or collision avoidance [98–101], which is an important topic of flying robots. Current state-of-the-art flying robots could perform robust image-based six degree-of-freedom (DOF) localization [90], cooperate mapping [94], aggressive flight in dense indoor environment [98], and flying through a forest autonomously [99]. From the next section, we review the HRI research that is closely related to the companion flying robot concept that we propose in this dissertation.

2.3 Human-Robot Interaction in Flying Robots (Platform)

HRI in flying robots is still in its infancy. In 2012, Graether & Mueller developed Joggobot to support exertion activity such as jogging [28]. Based on a commercial platform called AR.Drone [25], Joggobot used the built-in onboard camera to track a visual marker on the jogger's T-shirt and to maintain a fixed distance in front of the jogger. Upon analyzing

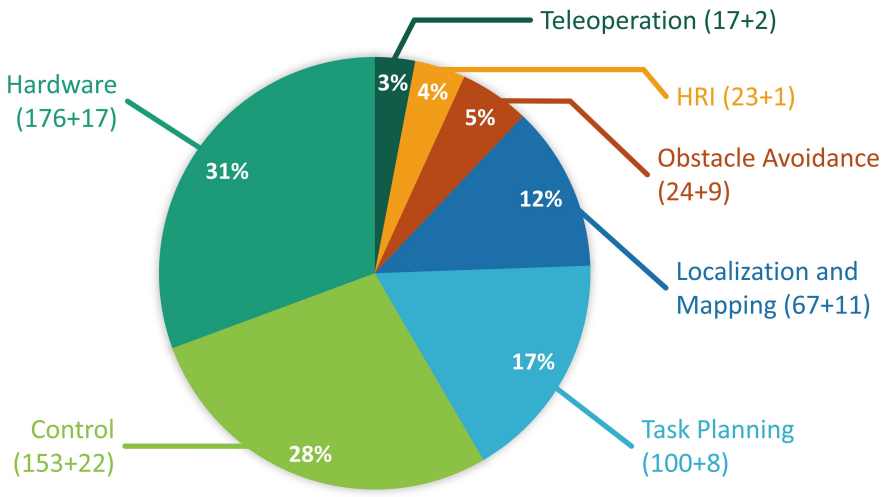

Figure 2.1: Topic distribution of flying robots from 2006 to 2016. Note: the numbers in each category's parentheses are the total number of conference and journal papers respectively.

HRI experiments with Joggobot, Graether & Mueller presented four important preliminary insights—embodiment, communication, personality, and control. The first two themes are closely related to our motivation of realizing a companion flying robot. In the embodiment theme, participants were positive about the idea of having a flying robot accompanying them while jogging. Compared to other jogging support systems such as smartphones and sport watches, participants thought that an embodied agent such as Joggobot could be easier to comprehend. Besides, participants felt that Joggobot can distract them from their exhaustion and also challenge them to increase their jogging effort. In the communication theme, participants expressed their need to direct Joggobot using hand gestures, for example, to tell Joggobot which way to go or change jogging speed. This suggests that a human sensing interface is crucial for a companion flying robot to have a better HRI with user.

In 2015, Mueller & Muirhead extend the HRI jogging experiment with a new flying robot and deep analysis [102]. Different from Joggobot, which follows the jogger in an indoor environment, the new flying robot flies around a preset jogging path at a fixed speed in an outdoor environment. Mueller & Muirhead conducted a HRI study and analyzed the jogging experience of thirteen casual joggers via interviews. Most joggers described that the experience was very interesting and expressed their desire to jog with the flying robot again in the future. In particular, joggers are able to sense a companionship with the flying robot

during the experiment; joggers experienced peer pressure from the flying robot and were motivated to keep running. And interestingly, some technical shortcomings (from robotic or engineering point of view) of the flying robot are perceived positively by the joggers. For example, first, some joggers feel that the flying robot is communicating with them (in fact, it is not) when the flying robot has some small and unstable movements caused by the wind (hovering instability from robotic point of view). Second, Mueller & Muirhead notice that the motor noise attracts the joggers' attention during the experiments. Some joggers also appreciated that the motor noise help them to distract from their running fatigue. Third, due to the hardware limitations, the battery needs to be changed from time to time (inconvenient during experiment). However, joggers feel that the flying robot is working hard too and share a mutual jogging experience the flying robot. Despite the positive outcomes of the experiments, some joggers also raised their concerns over the lack of autonomy of the flying robot; they are not able to control the running path and speed. Moreover, the flying robot could not sense the jogger and could not have more intimate HRI with the joggers.

The flying humanoid robot (FHR) proposed by Cooney et al. [29] is another example of companion flying robot developed for HRI applications. For safety considerations, Cooney et al. adopt a lighter-than-air platform and design the FHR using three saucer-shaped balloon modules. In addition to the discussion of design consideration and implementation details, Cooney et al. describe some interesting scenarios or potential applications of the FHR such as accompanying children in home, walking with elderly, and taking photos for users. Nevertheless, FHR focuses on the theoretic model but lacks platform developments and actual experiment results.

The HoverBall designed by Nitta et al. [103] is closely related to the concept of companion flying robots. HoverBall is based on a hand-size quadrotor and has a ball shape. It utilizes several pre-designed motion vocabularies for users to have unique sport experience . For example, a user can lift the HoverBall slowly upward with his/her hand and the HoverBall will start hovering in the air, simulating a real ball floating in the air. The user can also gently push the HoverBall forward and then lower down his/her hand, the HoverBall will then return to his/her hand, simulating a boomerang motion. The current system relies on an indoor positioning system to track the position of the HoverBall and user.

The flying lampshades co-developed by Cirque du Soleil, ETH Zurich, and Verity Studios feature a enchanting choreography where flying robots could interact with humans, such as react to human gestures and perform flying dances [104, 105]. The flying lampshades use an

indoor positioning system for position tracking and flying movements are preprogrammed for the choreography. Nevertheless, the flying lampshades demonstrate the potential of flying robots in HRI. With the improvements of autonomous and sociable capabilities, we believe that flying robots like flying lampshades can form a deep companionship with humans.

2.4 Human-Robot Interaction in Flying Robots (Human Tracking)

Similar to the Joggobot, Pestana et al. used flying robot's onboard camera to realize a human following application [30] by relying on OpenTLD [106] to track a manually initialized user. Higuchi et al. also performed human following with flying robot's onboard camera by using a color-based particle filter to track a user [31]. On the other hand, Papachristos et al. demonstrated a human tracking application with flying robot's onboard stereo camera [32]. All these systems need special requirements, i.e. manual initialization of the user (person to be tracked) [30], the user have to wear a shirt with a specific color [31], or the user has to move [32].

Lately, commercial flying robots such as 3DR Solo [16] and Lily Camera [33] claim to have human following functions. Strictly speaking, they are not following a person but a GPS device carried by the person. As a result, they have the weakness of a GPS sensor; they do not work at indoor environment or whenever GPS signal is not available. Commercial flying robots Hover Camera [34] and DJI Phantom 4 [36] demonstrate human following applications with computer vision and machine learning techniques.[1] While the Phantom 4 has a 3D gimbal to obtain stabilized video feeds for human tracking, the gimbal increases weight unnecessarily. They also do not have a human accompanying model to perform better HRI. We believe that they could only approach or follow a human but would not be able to walk side-by-side or switch operating mode based on environment or robot states.

By integrating a visual SLAM technique and a vision-based human tracking algorithm, Lim & Sinha presented a flying robot that can map the human walking path in real time [37]. On the other hand, Nasser et al. proposed a flying robot that can perform human following and gesture recognition with an onboard Xtion depth camera [38]. Nevertheless, both flying robots do not have stable video feeds, could not estimate human body orientation, and still lack a human accompanying model for HRI applications.

[1] Since there is no scientific publication, we derive our conjectures from product reviews and online videos.

2.5 Human-Robot Interaction in Flying Robots (Interface)

Human-robot communication or control interface is another active area of HRI research. In 2013, Monajjemi et al. presented a method to create, modify, and command a team of flying robots by using face detection and hand gestures [107]. More specifically, a user can add a flying robot into the team by only looking at the flying robot and waving his left or right hand. To command the team of flying robots, the user has to wave both hands together. Monajjemi et al. do not claim that their system is intuitive for a naïve participant, but "selecting a robot by looking at it is really fun, and even in our proof-of-concept implementation it is responsive and feels easy and natural" [107]. This insight gained during the HRI experiment also shows that HRI with flying robots could also be a very good entertainment application. Similar to their works, Lichtenstern et al. also demonstrate a system whether a single user can control multiple flying robots using hand gestures and a RGB-D sensor [108].

Later in 2014, Monajjemi et al. extended their work by commanding a team of two flying robots using not only face engagement and hand gestures but also voice and touch interfaces [109]. The multimodal interfaces are useful when one particular interface fails to work. For example, before taking off, the flying robot is normally on the ground and cannot detect human face and hand gestures properly due to the limited field of view. In their demonstration, the user first select the desired flying robot by using voice commands. By monitoring the accelerometer readings, the selected flying robot can confirm this selection when the user gently touches the flying robot.

More recently in 2015, Monajjemi et al. use hand gestures to establish mutual attention between a user and an outdoor flying robot [110]. This task is challenging compared to an indoor flying robot. An outdoor flying robot is up in the sky, and the user on the ground is represented by only a few pixels in the image. In this work. Monajjemi et al. use a periodic waving gesture of both arms as a signal to attract the flying robot's attention. When the flying robot detects this salient gesture, it performs a distinctive wobble movement for the user and hence further interaction can be carried out.

On the other hand, Constante et al. aim to improve the hand gesture interface of flying robots [111]. In general, collecting more training data could improve a hand gesture recognition system but the data collection process is time-consuming. On the other hand, using only user-specific hand gesture data could lead to bad generalization performance. To solve this

problem, they propose a transfer learning algorithm that can exploit both online generic and user-specific hand gestures data, where they first select useful data from the generic dataset and obtain a trained system with better performance than a system that is either trained with all generic dataset or with only user-specified dataset.

Burke & Lasenby propose to control a flying robot with pantomimic gestures [112]. The main idea of pantomimic gesture is to use a gesture that is similar to the desired action of flying robot as a gesture command. For example, to direct the flying robot to take-off, the corresponding pantomimic gesture would be raising your hand; to direct the flying robot to circle around an object, the corresponding pantomimic gesture would be circling your hand in the air. Burke & Lasenby argue that pantomimic gestures are more intuitive and presented a very fast and simple classification method based on principal component analysis.

Different from works mentioned above, Huang et al. aim to direct a flying robot in a previously mapped and labeled environment via natural language commands [113]. Operating a flying robot in a constrained indoor environment is challenging. Instead of using a remote controller, where a joystick or buttons are used to control the flying robot, users can issue a long sequence of natural language commands for the flying robot to operate in a 3D indoor environment. For instance, if we want the flying robot to go up and fly to Room A, we only need send the command of "go up and fly to Room A" to the flying robot. While their system currently only works in a previously mapped and labeled environment, it could be useful for a companion flying robot in the future.

2.6 Human-Robot Interaction in Flying Robots (Social Study)

Inspired by falconeering, that is, human interaction with birds, Shan & Sharlin studied the effectiveness of a few hand gestures in commanding a flying robot [114]. In line with the embodiment idea of Joggobot [28], Shan & Sharlin observed that participants of the HRI experiments were very engaged in having gesture interaction with the flying robot and spoke to the flying robot like a pet. Besides, they also found that participants like the "stop" and "come" gestures the most, as these gestures are common and often used.

More recently, in 2015, Cauchard et al. coined the term human-drone interaction (HDI) and performed a similar HRI experiment; the outcomes suggest that users are very positive about the idea of a companion flying robot [115]. Specifically, they carried out HRI Wizard-

of-Oz experiments[2] with nineteen participants, where sixteen participants mentioned that they interact with the flying robot as if it were a pet. Interestingly, participants even named the flying robot or assigned an ID number to the flying robot during the experiments. Besides, several participants mentioned feeling attached to the flying robots. Cauchard et al. also observed that out of the 216 found unique interactions, 96 interactions are related to body or hand gestures, 59 interactions are related to sound commands, 53 interactions include both gesture and sound commands, and 8 interactions are related to touch. Among these interactions, the most commonly used gestures and sound commands are *fly closer*, *stop by me*, *follow me*, *fly sideways*, *fly higher/lower*, *fly to a precise location*, *get attention*, and *take a picture of an object*. After the experiments, several participants expressed their concerns that they would like an interface for emergency landing in case anything goes wrong. In the end of the paper, Cauchard et al. also share their belief that the idea of a companion flying robot will be realized as flying robots become smaller and quieter.

Dancers use various kinds of motion to express their emotions. Based on this idea, Sharma et al. use Laban Effort System, a common method used by artists to express their emotions, for flying robots to express their affective states [116]. In their work, an artist was hired to move a quadcopter by hand, imagining that the quadcopter wants to express certain emotions. The quadcopter motions were captured by a 3D tracking system and later played back to the participants of the HRI experiments. All eighteen participants like the idea of robots expressing their emotions through affective motion and perceive most of the expressed affect correctly. In the end of their paper, they summarize a set of guidelines for flying robots to communicate their affect via affective locomotion paths.

Closely related to work of Sharma et al. [116], Szafir et al. use the flying robot's motion to express the robot's intention [117]. Specifically, they design a few motion primitives for flying robots such that user may perceive the robot's intention better, build up human-robot rapport, and hence increase task efficiency. One of the motion primitives is to fly in a more natural arc shape rather than a robotic straight line. Another motion primitive is to start and stop moving smoothly rather than always move at constant speed. Eighty-five participants ranging from eighteen to eighty years old are hired and three important hypothesis are confirmed. First, participants significantly preferred the flying robots with the designed primitive motions. Second, participants agreed that the flight motions with the

[2] A common experiment setting used by researchers, where participants interact with a robot that participants believe to be autonomous, but in fact it is being manually controlled by a human behind the scene.

designed motion primitives are more natural, smoother, and intuitive. More importantly, third, participants perceived that the flying robot with motion primitives are safer and in control.

In 2016, Cauchard et al. presented a new model for flying robots to express emotions via movements [118]. Specifically, they proposed a set of eight personalities (emotional states), namely *brave*, *dopey*, *sleepy*, *grump*, *happy*, *sad*, *scared*, and *shy*. Then, they designed the corresponding flying robot's movements for each personality. For example, a *brave* flying robot would fly quickly, look at a person directly, and never go backwards. A *sleepy* flying robot would fly slower, stay low, and respond to commands with delays. Cauchard et al. believe that encoding these characteristics into movements could help untrained users to comprehend the internal states of the flying robot. One good example is that, without looking into controller screen, users could identify a flying robot's state of tiredness (*low battery* or in *sleepy* emotional state) intuitively when the flying robot starts to move sluggishly and fly low.

In addition to flight motions, LED light is another good option for flying robots to express their emotion and intent. Arroyo et al. described their early prototype of a social flying robot called Daedalus, where it can perform four different expressions with head movement and two color LED eyes [119]. On other hand, Szafir et al. used a ring of sixty-four color LEDs as a reliable cue for user to comprehend the goal of the flying robot [120]. In particular, they found success in a *gaze* signaling pattern of the ring LEDs, where two regions of LEDs (reminiscent of human eyes, same size and same interocular distance) toward a certain direction (where the robot intends to fly to) are lit up. The second useful signaling pattern is *blinker* (region of the LEDs that light up in coordination with the robot's flying direction), where participants found it easier to tell when the flying robot was off course. As a future work, Szafir et al. intend to use the LED color space to communicate affect and interruptibility of the flying robot.

2.7 Human-Robot Interaction in Flying Robots (Operation)

HRI focusing on safety and operational effectiveness is another research area, especially for flying robots in search and rescue mission, and disaster management applications. During the following reviews, we focus on the elements that are related to companion flying robots.

Drury et al. presented a decomposition of situation awareness of flying robot from

the operator's point of view [121]. In particular, they described the needs of the operator (the user in our companion flying robot case) to know the health and status of the flying robot. For example, the flying robot's battery level and motors condition are important health information; flying speed and flying mode are the status information. As we will discover soon in the next section, companion flying robots could utilize motions and lights to signal this information to the user.

Murphy et al. analyzed thirty-eight flights of a flying robot during disaster management and concluded that three distinct roles are necessary to ensure the safety of people and objects nearby the flying robot and the flying robot itself [122]. Murphy also proposed some experience gained during rescue robotics missions that suggests the importance of obstacle avoidance and response to failures [123]. On the other hand, Morse et al. presented a method to estimate and visualize the coverage quality maps in a video taken from a flying robot's onboard camera during a search and rescue mission [124].

2.8 Human-Robot Interaction in Flying Robots (Design)

It is interesting to observe that the design of the flying robots is becoming more sociable. For example, AR.Drone developed by Parrot Company in 2010 [125] and Aibot X6 developed by Aibotix in 2013 [126] have protective hulls made of expanded polypropylene and carbon fiber, respectively, in order to cover the propellers from users. However, the top and bottom sides of the propellers of AR.Drone and Aibot X6 are still exposed to the users. Both Snap [127] and Fleye [128] flying robots developed in 2015 took a step further and have a case that fully prevents propeller injuries. While these safety designs are not the main focus of HRI research, it is an important aspect towards realizing a companion flying robot.

Over the years, various flying robots have been proposed but the noise issue has never been a significant factor in the design process. Existing rotor-type flying robots such as helicopters and quadcopters rely on motors to fly and unwanted noise is always generated during the flight. This noise is measured as high as 82 dB (one meter away from a commercial quadcopter [25]) and is very close to a hazardous level of 85 dB as legislated by most countries [129]. Physiology and psychology studies also suggest that noise has a strong association with health issues such as high blood pressure [130], work stress, and increased risk of accidents [131]. These findings suggest that the noise issue must be seriously

considered when designing a companion flying robot for long term interaction. Compared to rotor-type flying robots, we believe that blimp or balloon-type robots are a better platform for companion flying robots because of their lower noise level (measured as lower than 40 dB at one meter away from a blimp [132], comparable to the noise level in a library) during flight. Nevertheless, in this thesis, we focus on using rotor-type flying robots in order to make the companion flying robot more compact, have better flight responsiveness, and be less susceptible to wind disturbance. Note that a blimp requires a minimum size of 84 cm $\times$ 84 cm $\times$ 84 cm in order to handle payload of 500 grams, in which the blimp is very susceptible to wind disturbance but barely supports the overall payloads of the motors, sensors, and onboard computer.

Appearance design is also another important topic in designing a companion flying robot. In our preliminary study [132], we performed an online survey on SurveyMonkey [133] with forty-two participants that include robotic researchers, architects, business administrators, pharmacists, housewives, and students. Ranging from nineteen to forty-nine years old, all participants are from Japan, Malaysia, Singapore, and Australia. We showed three different flying robots to the participants: a penguin-shaped blimp (nature-inspired design), a ball-shaped rotorcraft (functional design), and an electro-hydro-dynamic (EHD) flying robot that requires no rotating parts to fly (scientific design). All flying robots are described briefly to the participants. Surprisingly, the penguin-shaped blimp has the top ranking, followed by the ball-shaped rotorcraft, and then the EHD flying robot. We expected the ball-shaped rotorcraft to be at the top rank as it is very useful and possess unique characteristic for it to excel in disaster situation (which was also explained to participants). In contrast, the penguin-shaped blimp has slower motion but has a charming outlook and is able to "swim" naturally in the air with very little noise. This finding is coherent with design concept proposed by HCI researchers [134], suggesting that appearance is an important factor in designing companion flying robots. Nevertheless, in this dissertation, we focus on using rotor-type flying robots in order to make the companion flying robot more compact, have better flight responsiveness, and be less susceptible to wind disturbance.

Chapter 3
Towards a Better Form of Flying Robot for Human-Robot Interaction

3.1 Introduction

In this chapter, we design a new type of companion flying robot in order to achieve safer and more effective human-robot interaction (HRI). We aim to design a flying robot with vertical take-off and landing (VTOL) and hovering capabilities. Hereafter, unless specified otherwise, we refer to VTOL when we use the term *flying robot* or *UAV*.

Conventional flying robots, such as quadcopters, hexacopters, and octocopters, have a limitation that hinders them from achieving safe and effective HRI. The problem originates from one of the common characteristics of conventional flying robots—they all have coupled motion control. Due to their simple mechanical design, in which all the rotors point up to the top, conventional flying robots cannot produce horizontal thrust. As a result, conventional flying robots cannot perform horizontal motions independently without changing their roll/pitch orientations. During horizontal motion, flying robots are dynamically more unstable since they are away from the equilibrium state, i.e., the horizontal position. Due to the change of roll/pitch orientation, conventional flying robots are also more susceptible to wind disturbance. Theoretically, the larger the pitch/roll angle (the higher the horizontal speed), the more unstable and dangerous the flying robot.

In a HRI application, it is also crucial for the companion flying robot to obtain a stable video feed. Obtaining a stable video feed enhances the flying robot's performance in understanding the accompanied person and surrounding environment via computer vision or pattern recognition techniques. Conventional flying robots use mechanical gimbals to obtain stable video feeds. However, the mechanical gimbal increases the payload weight and worsens the flying robot's weight distribution. Normally the mechanical gimbal is mounted in front of a flying robot for a better field of view. To counter-balance it, the battery is mounted at the back of the flying robot. As a result, more weight is spread away from the center of gravity. This makes it harder for the flying robot to stabilize due to the increased momentum, especially when the flying robot has large pitch/roll angles.

Aiming for a safe and effective HRI, we adopt a unique design such that a hexacopter can have decoupled motion control—it is able to move horizontally while maintaining its

Table 3.1: Comparison of the features of conventional and holonomic hexacopters.

	Holonomic flight	Intuitive flight	Decoupled model	Control stability	Robust to wind	Stable video w/o gimbal	Physcial HRI	Power efficiency
Conventional hexacopter [45]	×	×	×	×	×	×	×	○
Holonomic hexacopter	○	○	○	○	○	○	○	△

flying attitude. We call the new flying robot a holonomic hexacopter. As opposed to the flat structure of a conventional hexacopter, where all the motors are facing to the top of the aircraft, our holonomic hexacopter has tilted motors; the six motors can generate thrust in six different directions and achieve three translational and three rotational forces. Since the modification this small (compared to the conventional hexacopter), the proposed holonomic hexacopter remains mechanically simple and easy to build.

In a nut shell, we summarize the merits of the proposed hexacopter in **Table 3.1**. In contrast to a conventional hexacopter, the proposed holonomic design has the following characteristics:

- Has six degree-of-freedom (DOF), able to move horizontally without roll/pitch motion;
- Provides more intuitive flight (move horizontally without tilt motion) for users;
- Has decoupled control, horizontal motion can be controlled independently;
- More stable and less susceptible to wind disturbance from the control perspective;
- Provide a stable video feed for high level tasks without an additional gimbal;
- Able to achieve safe physical HRI with generated horizontal forces naturally;
- Compared to a conventional hexacopter, it has power efficiency of about 70–90%, depending on the tilt angle of the motors.

In the following, we first review some related flying robots that could perform holonomic flights. After, we derive the kinematic models of a conventional hexacopter and the proposed holonomic hexacopter. In our analysis, we will also show that the holonomic hexacopter is a generalization of the conventional hexacopter. Then we describe our hardware implementation and the controller designs. Last, we present the experiment results and conclude this chapter by discussing potential future work.

Table 3.2: Comparison of the features of state-of-the-art flying robots that could perform holonomic flights. (See text for details.)

	No. of motors & servos	Simple design	Decoupled motion control	Stable video feed w/o gimbal	Human accompanying model	Human sensing interface
Our holonomic hexacoper	**6 + 0**	○	○	○	○	○
Jiang & Voyles [135–137]	6 + 0	○	○	△ [1]	×	×
Crowther et al. [138]	6 + 0	○	○	△ [2]	×	×
CyPhy LVL 1 [139]	6 + 0	○	○	△ [1]	×	×
Tetrahedron hexacopter [39]	6 + 0	×	○	△ [2]	×	×
Rajappa et al. [140]	6 + 0	×	○	△ [3]	×	×
Eight-rotor aircraft [49,50]	8 + 0	×	△ [4]	△ [5]	×	×
Omnicopter [141]	8 + 0	×	△ [4]	△ [2]	×	×
Holonomic tricopter [142]	3 + 6	×	△ [6]	△ [2]	×	×
Omni-tricopter [143,144]	5 + 3	×	△ [6]	△ [2]	×	×
Bi^{2}copter [59]	4 + 2	×	△ [6,7]	△ [8]	×	×
Oosedo et al. [145]	4 + 4	×	△ [6]	△ [2]	×	×
Ryll et al. [146]	4 + 4	×	△ [6]	△ [2]	×	×
Omnidirectional blimp [71]	3 + 3	×	△ [6]	○	×	×
Holonomic airship [72]	6 + 0	×	○	△ [2]	×	×
Festo flying sphere [147]	8 + 0	×	○	△ [9]	×	×

[1] Camera view is blocked by the propellers.
[2] No onboard camera was used.
[3] No actual platform was built.
[4] Involve complex airflow dynamics for precise control.
[5] Only have downward camera for positioning control.
[6] Kinematic model changes whenever servo angles change.
[7] Could only achieve holonomic flight in one direction.
[8] Only have upward camera for infrastructure inspection.
[9] Only have cameras near to the gripper for manipulation control.

3.2 Related Works

Holonomic flying robot, that is, an aircraft that could fly and move horizontally without tilt motion, is not a totally new concept. In**Table 3.2**, we summarize state-of-the-art flying robots that could perform holonomic flights. Compared to other holonomic flying robots, which mainly focus on control and autonomy, the novelty of this work is to utilize the proposed holonomic hexacopter to achieve an intuitive flight (in addition to other merits mentioned in the previous section) for users. In the following, for review simplicity, we divide the flying robots in **Table 3.2** into five categories: six-rotor[1], eight-rotor[2], tricopter, quadcopter, and blimp types.

Six-rotor type. Jiang & Voyles [135–137], Crowther et al. [138], and CyPhy company [139] proposed flying robots that could perform holonomic flight. Similar to our holonomic

[1] Not necessary a (conventional) hexacopter, could be a tricopter with three lateral motors.
[2] Not necessary a (conventional) octocopter, could be a quadcopter with four lateral motors.

hexacopter, their flying robots have six rotors, simple design, and could achieve decoupled motion control. However, their flying robots are not designed for HRI and have a few limitations. For example, the onboard cameras on the flying robots designed by Jiang & Voyles and CyPhy company are blocked by the propellers (when the cameras point forward). The flying robot designed by CyPhy company also does not have a protective guard to ensure safe HRI. On the other hand, Crowther et al. only focus on control experiments and no onboard camera was used. In addition, all three flying robots have neither a human accompanying model nor a human sensing interface to interact with a human.

Higuchi et al. proposed a six-rotor flying robot in a double tetrahedron shape that could perform holonomic flight [39]. Different from ours, their flying robot has a complex 3D shape and is presumably hard to build. In contrast, one could achieve our holonomic structure by simply rotating the six propeller arms of a conventional hexacopter. Moreover, they only focus on control experiments and no onboard camera was used. Recently, Rajappa et al. derived a generalization model of a holonomic hexacopter [140]. However, the actual design becomes more complex and they only validated their design in a simulation environment.

Eight-rotor type. Salazar et al. [49] and Romero et al. [50] have developed flying robots that could perform holonomic flights by simply attaching four additional lateral rotors to their quadcopters. While the four lateral rotors could provide decoupled lateral motion control, the control of the flying robots is challenging due to the increased complexity of airflow dynamics in between the vertical and lateral rotors. In practice, it is also not easy to mount the lateral rotors. The resulting platforms are also hazardous for HRI, since the lateral propellers face toward the accompanied person directly. Brescianini & D'Andrea proposed an omnicopter with eight rotors that could perform holonomic flight at any flying attitude [141]. Similar to the flying robots developed by Salazar et al. and Romero et al., the control of the flying robots is challenging due to the increased complexity of airflow dynamics, which all the eight rotors face toward the center of the flying robot. Furthermore, the omnicopter has low payload capacity and is less power-efficient, since a large amount of rotor thrusts cancel with each other in the proposed 3D-symmetry design.

Tricopter type. To achieve holonomic flight, Ramp & Papadopoulos proposed to control the orientations of the three main rotors of a tricopter [142]. Their design is over-actuated, and the control of the flying robot is more complex than ours since the additional servo motors change the thrust orientation of the main rotors. Long & Cappelleri proposed a

flying robot that consists of two coaxial main rotors for main thrust generation, three smaller ducted fans for level control, and three servo motors to rotate the ducted fans for horizontal motion control [143, 144]. Since the flying robot is over-actuated, it is big and has limited payload. Moreover, the rotations of high speed ducted fans generate unwanted gyroscopic torques and make flight control harder.

Quadcopter type. Kawasaki et al. proposed a flying quadcopter with two connected bi-copters, where the orientation of the bi-copters could be adjusted with two additional servos. While their flying robot could change flying attitude along either pitch or roll direction while still remaining stable, their flying robot could only perform holonomic flight while maintaining either pitch or roll angle of the aircraft. Aiming for a fully-actuated flying robot, Oosedo et al. and Ryll et al. added four servos to tilt the four rotors in their quadcopters. Similar to the tricopter type flying robots, the dual bi-copters and quadcopters with tiltable rotors have a constantly changing kinematic model and a more sophisticated controller is necessary for stabilization control.

Blimp type. Burri et al. designed a blimp flying robot that has six degree of freedom (DOF) motions by using three main rotors and three servos, which it could move and rotate freely in 3D space [71]. Yang et al. also developed a similar blimp flying robot capable of holonomic flight with six rotors [72] while the Festo company demonstrated a blimp flying robot capable of holonomic flight with eight rotors [147]. However, all three blimp flying robots are very large (with diameters of 2.7, 1.8, and 1.8 meters respectively), which makes them more susceptible to wind disturbance and not suitable for outdoor flight. In the following, we aim for a more compact flying platform that could handle higher payload and is less susceptible to wind disturbance.

3.3 Conventional Hexacopter Model

We review the kinematic model of a conventional hexacopter in this section. Formally, a hexacopter has six identical rotors located at the vertices of a hexagon. **Figure 3.1** illustrates the motor layout of a conventional hexacopter when viewed from the top. We can observe that all motors face toward the top in a conventional design.

During flights, three rotors—M1, M2, & M6—always rotate counter-clockwise, and the other three rotors—M3, M4, & M5—always rotate clockwise during flight. This con-

figuration enables the conventional hexacopter to achieve one translational force and three rotational moments during flight. For example, assuming that the hexacopter is hovering above the ground, all motors rotate at equal speed and generate just enough thrust to lift the hexacopter. Since three motors rotate counter-clockwise and three motors rotate clockwise, there would be no rotational moment, and the hexacopter can hover stably. To fly the hexacopter up/down, one could increase or decrease the thrusts from all motors at the same time. To rotate the hexacopter forward, i.e. pitch motion along the y-axis, one could increase M1 & M3 thrusts and decrease M2 & M4 thrusts at the same time. Since the hexacopter now has higher thrust on the bottom side (viewed from the top as in **Fig. 3.1**), it would pitch forward (nose down) and fly forward.

Since all motors face toward the top direction, the relationship between every motor's force and the hexacopter's total translational force F and rotational moment τ can be easily written in matrix form as

$$\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -r\cdot\cos(\alpha) & -r\cdot\cos(\alpha) & r\cdot\cos(\alpha) & r\cdot\cos(\alpha) & -r & r \\ r\cdot\sin(\alpha) & -r\cdot\sin(\alpha) & r\cdot\sin(\alpha) & -r\cdot\sin(\alpha) & 0 & 0 \\ -k & -k & k & k & k & -k \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \tag{3.1}$$

where F_z, τ_x, τ_y and τ_z represent translational force along the z-axis and rotation moments along x-, y-, and z-axis, respectively. On the right hand side of **Eq.** (3.1), r is the distance from the center of the hexacopter (assuming center of gravity is at the center of the hexacopter) to the center of each motor, α is the angle in between two adjacent motors from the center of the hexacopter, and k is a constant value that maps the motors' rotational moments to the hexacopter's yaw moment along the z-axis. Note that F_x and F_y are omitted in **Eq.** (3.1) because all motors cannot generate force on x- and y-axes. In addition, since r and k are non-zero and $\alpha = 60°$, the matrix inside **Eq.** (3.1) is invertible. In other words, desired inputs of F_z, τ_x, τ_y and τ_z are controllable by adjusting each motor force. Mathematically,

this inverse relationship can be written as

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{1}{6r} & \frac{1}{2r\sqrt{3}} & -\frac{1}{6k} \\ \frac{1}{6} & -\frac{1}{6r} & -\frac{1}{2r\sqrt{3}} & -\frac{1}{6k} \\ \frac{1}{6} & \frac{1}{6r} & \frac{1}{2r\sqrt{3}} & \frac{1}{6k} \\ \frac{1}{6} & \frac{1}{6r} & -\frac{1}{2r\sqrt{3}} & \frac{1}{6k} \\ \frac{1}{6} & -\frac{1}{3r} & 0 & \frac{1}{6k} \\ \frac{1}{6} & \frac{1}{3r} & 0 & -\frac{1}{6k} \end{bmatrix} \begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \tag{3.2}$$

Assuming that each force and moment has a separate controller, each column of the inverse matrix can be normalized independently and **Eq.** (3.2) can be simplified and written as

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & 1 & -1 \\ 1 & -0.5 & -1 & -1 \\ 1 & 0.5 & 1 & 1 \\ 1 & 0.5 & -1 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \tag{3.3}$$

Equation 3.3 can be used to verify our earlier example to control a conventional hexacopter. For example, given a desired non-zero force F_z and zero moments in along three axes, **Eq.** (3.3) becomes

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} F_z \\ F_z \\ F_z \\ F_z \\ F_z \\ F_z \end{bmatrix}, \tag{3.4}$$

which essentially means that all motors increase or decrease thrust together. Note that the thrust ratio of the motors is more important, and the overall magnitude of the thrust is taken care by the thrust controller's gain. For a second example, given a desired non-zero pitch moment along the y-axis (positive τ_y, in order for the hexacopter to fly forward), **Eq.** (3.4)

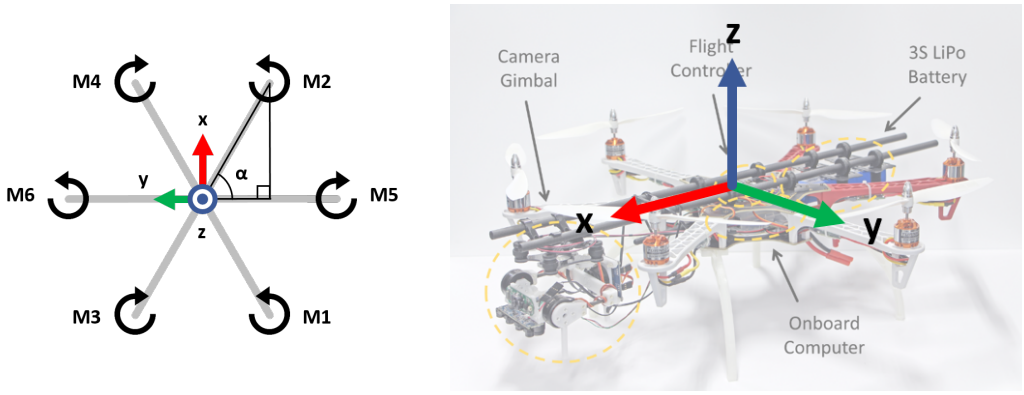

Figure 3.1: Kinematic model of a conventional hexacopter. **Left:** Motor layout, viewed from the top of the hexacopter. All six motors are located at the vertices of a hexagon. From the center, each adjacent pair of motors has an angle of 60°. During flights, M1, M2, & M6 rotate counter-clockwise, while M3, M4, & M5 rotate clockwise. **Right:** An actual hexacopter platform shows that all motors face vertically to the top.

becomes:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} \tau_y \\ -\tau_y \\ \tau_y \\ -\tau_y \\ 0 \\ 0 \end{bmatrix}, \tag{3.5}$$

which essentially means that M1 & M3 thrusts are increased and M2 & M4 thrusts are decreased, leading to a pitch forward motion. Again, the thrust ratio of the motors is more important, and the overall magnitude of the thrust is taken care by the pitch controller's gain. The same procedures can be applied to **Eq.** (3.5) given desired roll or yaw motions. When multiple desired inputs exist, the same procedures are still applicable since **Eq.** (3.5) has the superposition property.

3.4 Holonomic Hexacopter Model

Figure 3.2 illustrates the motor layout of a holonomic hexacopter viewed from the top. All six motors locate at the vertices of a hexagon. Similar to the conventional configuration, from the center, each adjacent pair of motors has an angle of 60°. During flights,

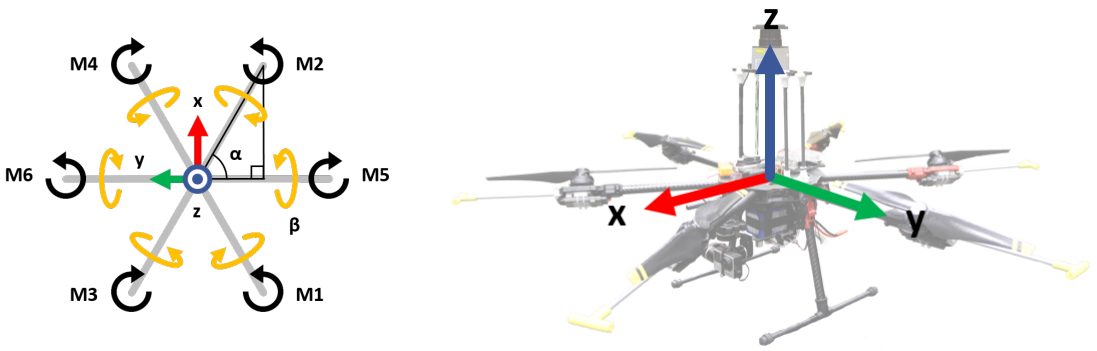

Figure 3.2: Kinematic model of a holonomic hexacopter. **Left:** Motor layout, viewed from the top of the hexacopter. All six motors are located at the vertices of a hexagon. Similar to the conventional configuration, from the center, each adjacent pair of motors has an angle of 60°. During flights, M1, M2, & M6 also rotate counter-clockwise, while M3, M4, & M5 also rotate clockwise. **Right:** However, different from the conventional hexacopter, all motors face up at an angle rotated about the propeller arms.

M1, M2, & M6 always rotate counter-clockwise, and M3, M4, & M5 always rotate clockwise. However, different from the conventional hexacopter, all motors face up at an angle rotated around the propeller arms. For example, considering motor M1, if we use our right hand to grasp its propeller arm and if our thumb is facing outward, the golden arrows in **Fig. 3.2** imply that we need to rotate to the direction of other four fingers.

Rotating the propeller arms this way has at least two benefits. First, the new configuration has each motor generating thrust at different orientations. As we will discover soon in the following analysis, this small modification eventually enables the hexacopter to move horizontally without tilting itself. Second, compared to the approach of Rajappa et al., our approach is simpler in both model analysis and hardware implementation. In their work, in addition to the rotation of motor around the propeller's arm, they also consider rotation along the axis perpendicular to the propeller's arm. While their approach might lead to a more efficient configuration, it makes the hardware implementation very difficult.

With the rotated propeller arms, now it is possible to generate horizontal forces on the

hexacopter and the Equation can be extended into

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} -\cos(\alpha)\sin(\beta) & -\cos(\alpha)\sin(\beta) & -\cos(\alpha)\sin(\beta) & -\cos(\alpha)\sin(\beta) & \sin(\beta) & \sin(\beta) \\ \sin(\alpha)\sin(\beta) & -\sin(\alpha)\sin(\beta) & -\sin(\alpha)\sin(\beta) & \sin(\alpha)\sin(\beta) & 0 & 0 \\ \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) & \cos(\beta) \\ -r\cdot\cos(\alpha)\cos(\beta) & -r\cdot\cos(\alpha)\cos(\beta) & r\cdot\cos(\alpha)\cos(\beta) & r\cdot\cos(\alpha)\cos(\beta) & -r\cdot\cos(\beta) & r\cdot\cos(\beta) \\ r\cdot\sin(\alpha)\cos(\beta) & -r\cdot\sin(\alpha)\cos(\beta) & r\cdot\sin(\alpha)\cos(\beta) & -r\cdot\sin(\alpha)\cos(\beta) & 0 & 0 \\ -(k+r\cdot\sin(\beta)) & -(k+r\cdot\sin(\beta)) & (k+r\cdot\sin(\beta)) & (k+r\cdot\sin(\beta)) & (k+r\cdot\sin(\beta)) & -(k+r\cdot\sin(\beta)) \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}. \tag{3.6}$$

Note that when $\beta = 0$, **Eq.** (3.6) reduces to **Eq.** (3.1), where the elements of the first two rows were zeros and omitted. Therefore, our proposed hexacopter can be viewed as a generalized configuration of a conventional hexacopter. In contrast to the conventional hexacopter, our proposed hexacopter is able to move horizontally without tilting itself. In this regard, it can be considered as holonomic, i.e., it can perform six DOF motions in a 3D space (three translational and three rotational motions) and has a few advantages, such as decoupled control and smoother onboard video processing without a gimbal.

For simplicity, we substitute real design values of $\alpha = 60°$ and $\beta = 36°$ into **Eq.** (3.6) and find the inverse matrix in order to map desired control inputs into motor thrusts. Note that the larger the angle β, the higher the power loss of the hexacopter during hovering, since the generated lateral forces cancel out on the horizontal plane. On the other hand, the smaller the angle β, the more difficult for the hexacopter to move horizontally. The selection of angle β is tuned manually during the flight experiments and could be optimized according to specific tasks. Similar to the derivation in **Sect. 3.3**, if we assume that each force and moment has a separate controller, then each column of the inverse matrix can be normalized individually and **Eq.** (3.6) can be simplified and written as

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} -0.5 & 1 & 1 & -0.5 & 1 & -1 \\ -0.5 & -1 & 1 & -0.5 & -1 & -1 \\ -0.5 & -1 & 1 & 0.5 & 1 & 1 \\ -0.5 & 1 & 1 & 0.5 & -1 & 1 \\ 1 & 0 & 1 & -1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}. \tag{3.7}$$

Note that the last four columns of the matrix in **Eq.** (3.7) are exactly the same as the matrix in **Eq.** (3.3). When F_x and F_y are zeros, **Eq.** (3.7) reduces to **Eq.** (3.1). On the other

hand, given a desired non-zero force F_x, **Eq.** (3.7) becomes

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} -0.5 \cdot F_x \\ -0.5 \cdot F_x \\ -0.5 \cdot F_x \\ -0.5 \cdot F_x \\ F_x \\ F_x \end{bmatrix}. \tag{3.8}$$

Equation (3.8) implies that in order to move forward, one has to increase M5 & M6 thrusts by 1 unit and decrease M1, M2, M3, & M4 thrusts by 0.5 units. From the holonomic configuration, we can observe that when the M5 & M6 thrusts increase, there is a surplus force applied on the x-axis positive direction. Besides, when the M1, M2, M3 & M4 thrusts decrease, there is a lack of force applied on the x-axis in the opposite direction. Overall, all motors would generate 4 units of translational forces along the x-axis and move hexacopter forward. In addition, thanks to the symmetry of design, the total forces applied on the y-axis remains balanced. The total force applied on the z-axis also remains balanced since M5 & M6 thrusts increase by 2 units while the other motor thrusts decrease by 2 units. By substituting **Eq.** (3.8) into **Eq.** (3.7), we can verify that except F_x, all other resulting forces (F_y, F_z) and moments (τ_x, τ_y, τ_z) are equal to zero. Similarly, given a desired non-zero τ_y, **Eq.** (3.7) becomes

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} F_x \\ -F_x \\ -F_x \\ F_x \\ 0 \\ 0 \end{bmatrix}. \tag{3.9}$$

Equation (3.9) implies that in order to move left, one has to increase M1 & M4 thrusts by 1 unit and decrease M2 & M3 thrusts by 1 unit. From the holonomic configuration, we can observe that when the M1 & M4 thrusts increase, there is a surplus force applied on the y-axis positive direction. Furthermore, when the M2 & M3 thrusts decrease, there is a lack of force applied on the y-axis in the opposite direction. Note that M5 & M6 thrusts have no effect on the y-axis. Overall, all motors would generate 4 units of translational forces along the y-axis and move the hexacopter left. In addition, thanks to the symmetry of the design, the total force applied on the x-axis remains balanced. The total force applied on the

z-axis also remains balanced since M1 & M4 thrusts increase by 2 units while the M2 & M3 thrusts decrease by 2 units. By substituting **Eq.** (3.9) into **Eq.** (3.7), we can easily verify that except F_y, all other resulting forces (F_x, F_z) and moments (τ_x, τ_y, τ_z) are equal to zero.

3.5 Hardware Implementation

We first describe the basic hardware in our design. Then, we explain the use of simple proportional–integral–derivative (PID) controllers for hovering control in the next section. **Figure 3.3** illustrates the holonomic hexacopter developed in-house using the Tarot 680PRO carbon fiber frame. We rotate each propeller arm according to the kinetic model discussed in **Sect. 3.4** such that $\beta = 36°$. The developed hexacopter is equipped with a Turnigy 5000 mAh (6S, 20C) Lithium-polymer battery. Switching power regulators are employed in order to convert the input 24 V into 12 V and 5 V that are required by the onboard flight computer and controller. The overall weight of the holonomic hexacopter is 3.5 kg, including all the systems shown in **Fig. 3.3**. Note that we use a gimbal to mount a GoPro camera in order to take onboard flight videos for qualitative evaluation purposes.

The hexacopter also has six identical DJI E800 propulsion units. Each propulsion unit consists of one 13×4.5 inch propeller, one brushless direct-current (BLDC) motor with stator size of $35 \times 10mm$, and one electronic speed controller (ESC). **Figure 3.4** summarizes the relationship of motor input signal (1000–2000 range), current consumption, and generated thrust of each propulsion unit, where the blue cross points are our collected data, the red line is a 4^{th} order polynomial fitting of the collected data points, and the purple circles are data taken from the online community [148]. During hovering operations, the motor input commands usually lie within the 1600–1800 region in our design.

The flight controller (FC) is a mid-level controller in our hexacopter, where it serves as a hub to collect information from all low-level sensors such as the accelerometer, gyrometer, magnetometer, barometer, and ultrasonic sensor. The flight controller also performs attitude estimation and sends control signals to each propulsion unit. There are many high-quality FCs on the market such as Naze32, CC3D, Crius, MultiWii, KK2, Naza, Pixhawk, and APM 2.6. We select our FC based on a few requirements. In addition to being lightweight and powerful, the FC should also be open-source since we might need to include additional sensors in the future. Among the possible candidates, we chose Naze32 in our design since it is open-source, has a 32-bit 72MHz processor, has all the sensors above (except ultrasonic

Figure 3.3: Our holonomic hexacopter developed using the Tarot 680PRO carbon fiber frame.

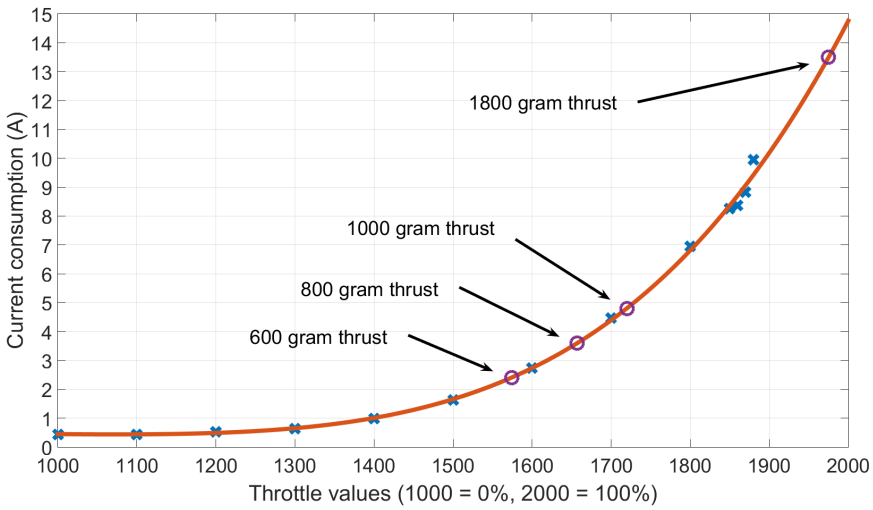

Figure 3.4: The relationship of motor input signal (1000–2000 range), current consumption, and generated thrust of each propulsion unit, where the blue cross points are our collected data, the red line is a 4^{th} order polynomial fitting of the collected data points, and the purple circles are taken from the online community [148]. While hovering, the motor input commands usually lie within the 1600–1800 region in our design.

sensor), is lightweight (only 8 grams), and is low-cost ($25). Since Naze32 does not come with an integrated distance sensor, we connect an external ultrasonic sensor to it and mount the ultrasonic sensor on the bottom of the hexacopter (facing vertically downward).

In our design, Naze32 is flashed with iNavFlight 1.1 firmware [149], where various parameters can be configured easily via a configuration app [150]. In general, iNavFlight uses a complementary filter to fuse accelerometer and gyrometer data (function `imuMahonyAHRSupdate` inside `imu.c`) in order to compute the hexacopter's attitude. Based on the computed attitude, iNavFlight employs PID controllers (function `navPidApply2` inside `navigation_rewrite.c`) to control the hexacopter's yaw, pitch, and roll attitudes. We will present the controller structure in the next section.

Note that we do not use the integrated magnetometer and barometer in the FC due to the high measurement noise. The altitude of the flying hexacopter is measured by using the external ultrasonic sensor. This altitude information is also used to estimate the velocity along the z-axis (function `updateSonarTopic` inside `navigation_rewrite_pos_estimator.c`). iNavFlight also uses a simple P- and PID controller to maintain the flying height of the aircraft. The first P- controller is used to convert the position error to a desired velocity, and the second PID controller is used to maintain the velocity of the aircraft. Therefore, in steady state, both position and velocity errors are equal to zero.

In addition, our hexacopter is equipped with an onboard minicomputer called Odroid-XU4. Odroid-XU4 has a processor with A15 quad-core 2.0 GHz and A7 quad-core 1.5 GHz and is the smallest onboard computer that we have surveyed (slightly larger than a credit card and only weights 72 grams) at a surprisingly low cost of $180. Odroid-XU4 acts as a high level controller for the hexacopter and runs a multi-threading `C++` program in the Linux Ubuntu 14.04 environment to interface with the Naze32 FC via a serial connection, an Xtion sensor (color and depth cameras) via a USB connection, a Hokuyo URG-04LX 2D lidar sensor via a USB connection, and an ADNS3080 optical flow sensor via SPI connection. We use the Xtion sensor for human detection (**Chapter 5**). The optical flow sensor will be used for position holding in the future work. The 2D lidar sensor is used as a failsafe for collision avoidance. During flights, the hexacopter tries to maintain a two-meter distance from any detected nearby objects. In case the hexacopter cannot maintain this condition, it will trigger the auto-landing function automatically for safety purposes.

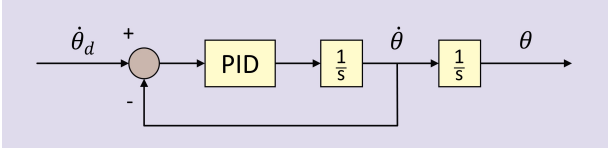

Figure 3.5: PID controller structure of the yaw controller in our holonomic hexacopter.

3.6 Controller Design

In this section, we summarize the PID controllers that we are using in our holonomic hexacopter, including (i) a yaw controller, (ii) pitch and roll controllers, (iii) a height controller (in velocity and position modes), and (iv) horizontal position controllers. Note that the first three PID controller structures in this section are derived from the iNavFlight source codes [149] and have I-term anti-windup implementation based on a formal controller design [151].

3.6.1 Yaw Controller

Figure 3.5 illustrates the PID controller structure of the yaw controller in our holonomic hexacopter. The yaw controller is the simplest control used in our holonomic hexacopter, where only the gyro rate along the Z-axis is used as feedback in this controller. Upon tuning, a non-zero input of $\dot{\theta}_d$ can change the yaw motion of the hexacopter smoothly.

3.6.2 Pitch and Roll Controllers

Figure 3.6 illustrates the PID controller structure of the pitch and roll controllers in our holonomic hexacopter. The pitch and roll controllers have two inner loop, where both gyro rate and attitude information (by fusing gyro rate and accelerometer measurements with a complementary filter) are used for stabilization control. The inner loop has the same structure with the yaw controller. However, the pitch/roll rate is not directly controllable by the user. The outer loop is a simple P-controller that accept desired roll/pitch angle input from user. Upon tuning, a non-zero input of θ_d can change the roll/pitch angle of the hexacopter smoothly.

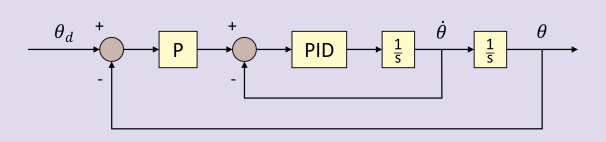

Figure 3.6: PID controller structure of the roll and pitch controllers in our holonomic hexacopter.

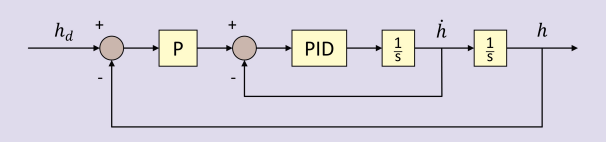

Figure 3.7: PID controller structure of the altitude controller during position mode in our holonomic hexacopter.

3.6.3 Altitude Controller

Altitude controller has two modes. When there is no input from the user, altitude controller will be in the position mode; when there is a non-zero input from the user, altitude controller will be in the velocity mode.

Figure 3.7 illustrates the PID controller structures of the altitude controller in position mode in our holonomic hexacopter. Note that the altitude controller in position mode has the same structure with the roll/pitch controller. The only difference here is that altitude controller uses altitude information obtained from the sonar sensor as the control feedback.

Figure 3.8 illustrates the PID controller structures of the altitude controller in velocity mode in our holonomic hexacopter. Note that the altitude controller in velocity mode has the same structure with the yaw controller. The only difference here is that altitude controller uses velocity information as the control feedback.

3.6.4 Horizontal Position Controllers

Figure 3.9 illustrates the PID controller structure of the horizontal position controller in our holonomic hexacopter. The horizontal position controller uses the depth information from the Xtion camera as feedback in this controller. Upon tuning, a non-zero input of $\dot{\theta}_d$ can change the horizontal position of the hexacopter smoothly.

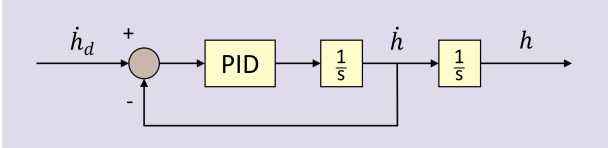

Figure 3.8: PID controller structure of the altitude controller during velocity mode in our holonomic hexacopter.

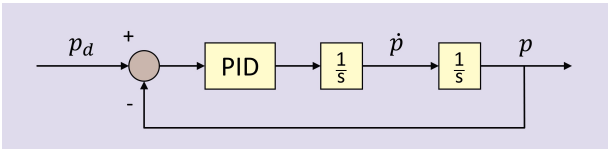

Figure 3.9: PID controller structure of the horizontal position controller in our holonomic hexacopter.

3.7 Experiment Results

3.7.1 Hovering Experiment

We perform an experiment to analyze the hovering stability of the proposed holonomic hexacopter. For simplicity, we fly the hexacopter with active altitude-holding control (at height of 80 cm) and manually maintain its horizontal position at the take-off position. Due to the wind disturbance, sensor noise, and imperfect hardwares, occasionally we need to move the hexacopter horizontally during this task. We collect the onboard flight information a analyze the variance of roll-pitch angles after the hovering experiment.

Figure 3.10 illustrates the plots of roll-pitch angles during 1-minute non-holonomic (left) and holonomic (right) flights with the developed hexacopter shown in **Fig. 3.3**. We can observe that the variance of roll-pitch angles on the left figure during the position-holding task is relatively large (around ±2°) compared to the holonomic flight on the right figure (within ±0.5°). With an active and robust position-holding controller, the variance of roll-pitch angles in a non-holonomic flight could be improved. Nevertheless, this experiment results show that the developed holonomic hexacopter can achieve stable hovering flight.

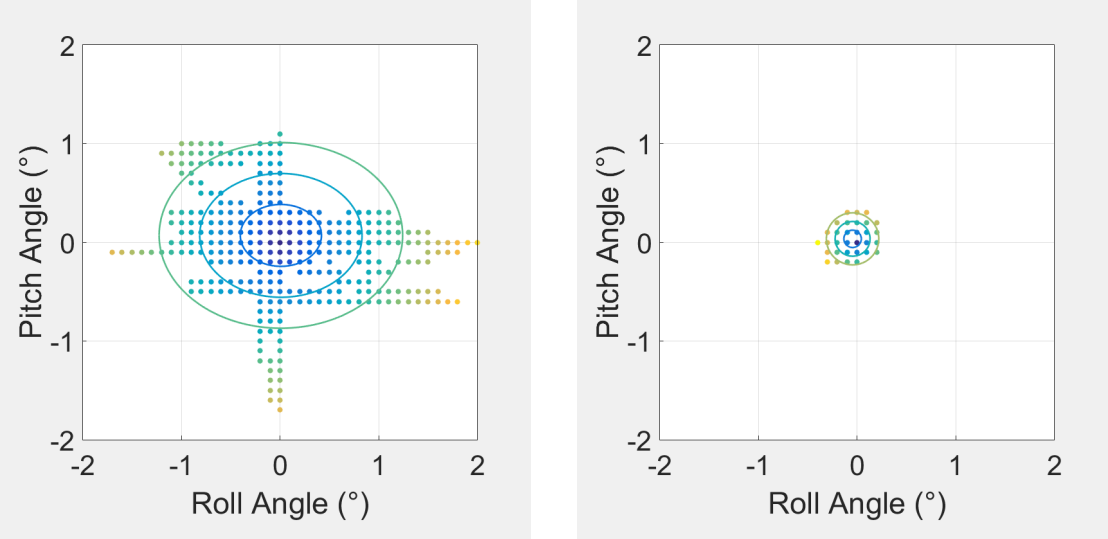

Figure 3.10: Plots of roll-pitch angles during 1-minute non-holonomic (left) and holonomic (right) flights with the developed hexacopter. The three circles in the plot correspond to standard deviation of one, two, and three.

3.7.2 Altitude-Holding Experiment

We also perform an experiment to analyze the altitude-holding stability of the proposed holonomic hexacopter. We fly the hexacopter with active altitude-holding control at 100 cm and record the altitude data during the experiment.

Figure 3.11 shows the flight heights over about 60 seconds. During the 1-minute flight, the holonomic hexacopter achieves altitude-holding performance with maximum error of ±2 cm and standard deviation of 0.8159. Note that the developed hexacopter performs altitude-holding with a simple PID controller with noisy velocity estimates (due to low-cost sensors) and without dynamic modeling. While the altitude-holding performance might not be satisfactory for a precision flight application, it is stable enough to perform an HRI experiment. In fact, a precise flight (e.g., within altitude error of ±0.5 cm) might not be necessary for a natural or social HRI. Analogous to human activities, we indeed do not have "precise" standing or walking performance while having interaction with others.

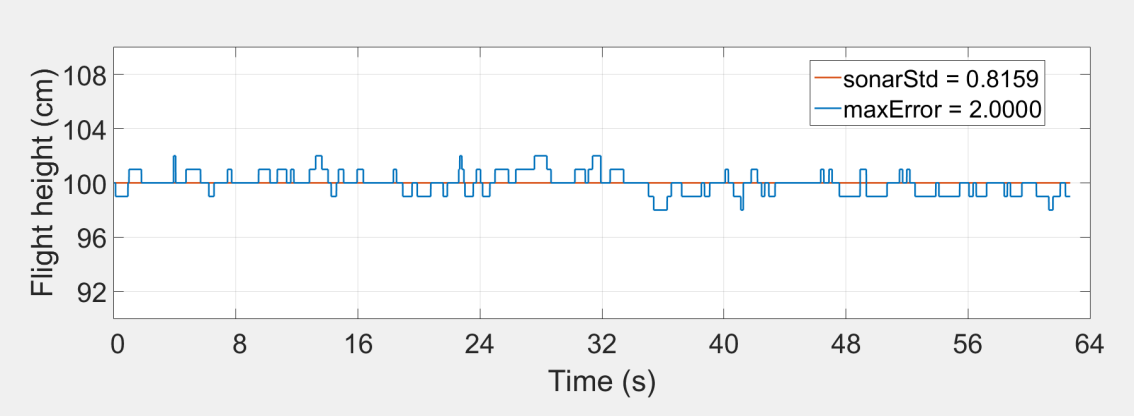

Figure 3.11: Plot of the altitudes in a 1-minute flight with the developed hexacopter.

3.7.3 Holonomic Flight Qualitative Results

We illustrate the onboard time-lapse images during roll motions in **Fig. 3.12**. In the first row of images, the hexacopter rolls counter-clockwise (viewed from the back) and starts moving to the left. Then, in the second row of images, the hexacopter rolls clockwise (viewed from the back) and starts moving to the right. Finally, in the third row of images, the hexacopter rolls counter-clockwise (viewed from the back) again and moves back to the starting point. From the time-lapse images, we can find that without a gimbal platform (which normally increases the hexacopter payload and deteriorate the hexacopter's weight distribution), the video feed is rotated and further image processing is required for high-end computer vision tasks.

We illustrate the onboard time-lapse images during pitch motions in **Fig. 3.13**. In the first row of images, the hexacopter pitches clockwise (viewed from the right) and starts moving forward. In the second row of images, the hexacopter is still moving forward and then pitches counter-clockwise (viewed from the right) and starts moving backward. Finally, in the third row of images, the hexacopter pitches clockwise (viewed from the back) slightly and tries to stabilize in the air. From the time-lapse images, we can find that without a gimbal platform, the video feed is tilted and further image processing might be required for high-end computer vision tasks.

In contrast, we illustrate the onboard time-lapse images during holonomic flight in **Fig. 3.14** (roll-less leftward/rightward motion) and **Fig. 3.15** (pitch-less forward/backward

Figure 3.12: Onboard time-lapse images during the roll motion, from left to right and then top to bottom. (See text for details.)

Figure 3.13: Onboard time-lapse images during the pitch motion, from left to right and then top to bottom. (See text for details.)

Figure 3.14: Onboard time-lapse images during the roll-less leftward/rightward motion, from left to right and then top to bottom. (See text for details.)

motion). We can observe that even without a gimbal platform, the video feed is always stable (neither rotated nor tilted) regardless of the hexacopter's motions.

Moreover, according to the user's feedback, when flying a conventional hexacopter, the user feels nervous when the hexacopter roll or pitch in order to move horizontally. While it might be normal for a professional user, it is important to note that novice user expects flying robot to move leftward/rightward naturally (not "roll" forward/backward) or to move leftward/rightward naturally (not "pitch" forward/backward). Therefore, the holonomic hexacopter is not only safer and more stable from the control's point of view (because it always stay in or close to the equilibrium point), but also makes user feel more comfortable and understand the flight motions *intuitively*.

3.8 Discussion

While the hovering and altitude-holding performance is stable to perform an HRI experiment, there are several ways to further improve the the developed holonomic hexacopter. First, the hexacopter body frame used at the meantime is not rigid and the motor thrusts are not transfered to the body frame effectively. Second, we are using low-cost sensors and simple low-pass filters to obtain the necessary information for flight control. With

Figure 3.15: Onboard time-lapse images during the pitch-less forward/backward motion, from left to right and then top to bottom. (See text for details.)

better sensors (and better hardware damping), we could increase the cut-off frequency of the low-pass filters and obtain a more responsive and precise flight. Third, we are also using slower main flight and motor controllers. We believe that the flight performance could be boosted with faster controllers.

In addition to hardwares, we can also improve the flight performance via controller softwares. Currently, we are using basic PID controllers in the developed hexacopter. As one of our future work, we aim to develop a dynamic model for the holonomic hexacopter with system identification techniques. With the system identification techniques, it is possible to optimize and even personalize the flight performance more systematically and analytically.

It is also interesting to note that the proposed holonomic hexacopter could become fully-actuated with an updated kinematic model. By using reversible propeller units and setting the tilt angles of all motors (β in **Sect. 3.4**) to 45^o, the proposed holonomic hexacopter can stabilize itself at different attitude like the omnicopter [141] and potentially achieve a new way of interaction with users.

Currently, the only drawback of the holonomic hexacopter is its power efficiency, which is about 70–90% when compared to a conventional hexacopter (depending on the tilt angle, β). However, since a companion flying robot is likely to be near to its user, it is reasonable (and technically possible) to supply power to the flying robot via a tether. In addition to

the merit of enhanced safety (the flying robot's activity space is limited and could not fly far away from the user), the tethered companion flying robot is analogous to a real pet. Moreover, the use of tether in companion flying robots could also potentially help other nearby people to feel more secure. By further optimizing the hardware design and using a smaller flying robot in the future, we believe that the concept of tethered flying robot will even bring more merits to the field of companion flying robots.

3.9 Conclusion

In this chapter, we design a new type of companion flying robot in order to achieve more effective HRI. We adopt a unique design such that a hexacopter can have decoupled motion control—it is able to move horizontally while maintaining its flying attitude. Compared to a conventional hexacopter, the proposed holonomic hexacopter provides safer operation from the control perspective and offers intuitive flight from the user's point of view. It also provides a stable video feed for high level tasks without an additional gimbal. Moreover, since it always maintains horizontal attitude, it is less susceptible to wind disturbance. While it is not demonstrated in this work, the proposed hexacopter is also expected to achieve safer physical interaction with the user by exerting horizontal forces without changing attitude.

Chapter 4
Human Accompanying Model for Flying Robots

4.1 Introduction

Autonomy has been the main focus in flying robot research in the past, where "autonomy" includes sensing, control, localization, mapping, planning, and obstacle avoidance. In this dissertation, our goal is to design a more sociable flying robot. For example, in addition to the basic autonomy capabilities like attitude sensing and stabilization control, we are interested in building a companion flying robot that could accompanying a person, where "accompanying" includes *approaching*, *following*, *side-by-side walking*, *leading*, and *flying above*. We note that robots—including mobile robots—to-date could not achieve natural and rich human accompanying behaviors.

In this chapter, our aim is to design a general model to unify various human accompanying behaviors of companion robots. Human accompanying, including human approaching (from a distance away), human following (at the back), side-by-side walking (along with the user), human leading (at the front), and flying above the person, are important behaviors of companion robots. To the best of our knowledge, robots to-date focus on one or two human accompanying modes; there is no existing work to unify these human accompanying modes in order for robots to achieve natural and rich interaction with humans.

We propose a two-level model to achieve this goal. At the top level, we adopt a hierarchical finite state machine (FSM) to organize the behavior flows of a companion flying robot. Hierarchical FSM has the merits of simplicity and expandability. These merits are useful for companion flying robots to consider the robot, environment, and human states and achieve a rich interaction behavior with a person. Moreover, hierarchical FSM is lightweight and computational friendly to the flying robot's onboard processor.

At the bottom level, we use a relative positioning control method for robots to achieve smooth and natural accompanying motions. We adopt the idea of steering behaviors proposed by Reynolds [152] in the computer animation and interactive media fields. Steering behaviors were developed for autonomous characters to move in a realistic manner by using only local information. In contrast to global planning and optimization approaches, steering behaviors are simple, fast, and require no pre-built map to work. In the followings, by using a few

basic controllers, we show that steering behaviors could help flying robots achieve smooth and natural accompanying motion.

While the top-level hierarchical FSM alone can be viewed as a rule-based approach, together with the bottom level relative positioning controller, they form a powerful hybrid approach that is able to achieve natural and rich HRI behaviors with minimal computational load. In the following, we first review some previous works related to human accompanying model and navigation. Then, we describe the hierarchical FSM that we have designed for our flying robots. After that, we explain the steering behaviors in our companion flying robot, along with our simulation and experiment results. Last, we summarize this chapter with some discussion and potential future works.

4.2 Related Works

Human accompanying in flying robots is relatively new; there are only a few flying robots that could achieve human following and leading at the time of writing (May 2016). To present a more complete review, we also include related works in mobile robots.

4.2.1 Human Accompanying (Flying Robots)

Human following. As mentioned in **Chapter 2**, Pestana et al. realized human following with a flying robot by using a vision-based tracking method [30]. Higuchi et al. also demonstrated human following with a flying robot by using a color-based particle filter to track a user [31]. Lim & Sinha presented a flying robot capable of human following by using a vision-based human tracking algorithm as well [37]. Using a depth camera, Naseer et al. developed a flying robot that could perform human following and gesture recognition. Recently, Skydio company demonstrated a flying robot that could avoid obstacles while following a person riding a bicycle [153]. All the five flying robots focus on human following mode, and could not perform other human accompanying modes.

Human leading. In **Chapter 2**, we also mentioned that Graether & Mueller demonstrate a flying robot that could jog together with a human, where they program a commercial flying robot to maintain a fixed distance in front of the jogger. Their flying robot also only focuses

on human leading[1] and could not perform other human accompanying modes.

4.2.2 Human Accompanying (Mobile Robots)

Human approaching. Dautenhahn et al. developed a mobile robot that could approach a person who is sitting on an armchair naturally [154]. Satake et al. also presented a mobile robot that could approach humans in a natural way in shopping mall such that the robots can initiate conversations with people better [155, 156]. Using the same mobile robot (Robovie), similar human approaching behavior has also been demonstrated by Kanda et al. [157], Satake et al. [158], Kato et al. [159], and Ratsamee et al. [160] in their HRI experiments.

Human following. Cosgun et al. proposed a path planning algorithm for a telepresence robot to follow a person semi-autonomously [161]. Jung et al. also designed a high speed mobile robot that could follow a marathoner [162]. On the other hand, Tani et al. developed a mobile platform to follow oxygen therapy patients [163]. Gockley et al. [164] and Granata & Bidaud [165] also presented human following behaviors in their HRI experiments with mobile robots. In addition, human following behaviors have been demonstrated with wheelchair robots developed by Leight et al. [166], Hemachandra et al. [167], and Tsuda et al. [168].

Side-by-side walking. Park & Kuipers designed a mobile robot that could walk side-by-side with person by using online local trajectory planning, where their method takes the robot, obstacle, and human states into consideration [169]. Morales et al. also developed a mobile robot that could walk side-by-side with a person by maximizing a likelihood function that includes robot, obstacle, and human states [170, 171]. A similar likelihood function model has also been used by Murakami et al. to demonstrate walking side-by-side with a mobile robot [172]. On the other hand, Pucci et al. proposed a new nonlinear controller that allow a mobile robot to walk side-by-side with a person effectively [173]. Kobayashi et al. also demonstrated a wheelchair robot that could walk side-by-side with a caregiver by using a laser range sensor [174].

Human leading.[2] Jung et al. demonstrated a mobile robot that could perform a human leading operation [175]. A human leading scenario has also been demonstrated by Garrell & Sanfeliu with a mobile robot in simulation environment [176]. Specifically, they used a

[1] Strictly speaking, the flying robot is not leading the jogger, but rather "following" the jogger at the front.
[2] A situation where the robot is in front of the user. Technically, a robot could "follow" a user at the front.

particle filter to track the position of a group of people, and program a few mobile robots to help people to stay together while moving in group. We note that Garrell & Sanfeliu also presented a few human accompanying scenarios with an actual robot but the robot was controlled by teleoperation in each scenario.

4.2.3 Navigation Model

Researchers have been proposing a large variety of navigation models for ground robot systems in the past. In general, most navigation models in robotic systems can be categorized into four main approaches: (i) rule-based, (ii) social force model, (iii) utility functions, and (iv) dynamic path planning approaches.

Many ground robot systems, including humanoid robots, mobile robots, and wheelchair robots, rely on simple rules to follow a human. For example, the wheelchair robot in [168] follows a person by adjusting its speed and direction based on the estimated position and orientation errors. The wheelchair robot in [174] also uses a similar rule-based controller. On the other hand, mobile robots in [162], [163], and [166] use a basic P-controller to adjust their speed and maintain a fixed distance with a human. One can also extend these simple methods to avoid obstacles like the wheelchair robot in [165].

By representing the influences of the human, destination goal, and obstacles as virtual forces, Ferrer et al. proposed a social force model for the Tibi mobile robot to successfully accompany a human and avoid obstacles in an urban area [177, 178]. Yoshimi et al. also presented a similar force-based approach for their humanoid robot to follow a human and avoid obstacles simultaneously [179]. These models are intuitive but have coupled translational and (yaw) rotational velocity controllers. Compared to social force approaches, the steering behaviors approach that we are using is easier to implement.

Different from the social force model, Morales et al. proposed a statistical approach for their Robovie humanoid robot to walk side-by-side with a human and avoid obstacle simultaneously [170, 171]. Specifically, after analyzing recorded videos of two humans walking side-by-side, they defined eight utility functions such as relative distance and relative velocity to characterize the human-human walking in a statistical way. With the defined utility functions, they can compute a 2D likelihood map of utility values around the predicted next robot's position. Therefore, Robovie can approach to the best position by looking for the maximized utility value. The wheelchair robot in [167] also takes a

similar approach but only computes the 1D scores at each heading based on the two distance parameters. While effective, the calibration and tuning processes of the utility function approaches are time-consuming.

We note that both social force model and utility maximization approaches are reminiscent of the potential field method [180]. However, since we focus on the human accompanying application and assume that users can avoid obstacles naturally, both approaches are not subjected to the well-known local minima problem. Alternatively, one may pursue dynamic path planning approaches like the telepresence robot in [161] and wheelchair robot in [169] to accompany a human while avoiding obstacles. While these online path planning methods have been successfully implemented on mobile robots, it is a challenging task for a flying robot. A flying robot normally has payload limitations and its onboard processors are less powerful to perform intensive computations.

4.3 Finite State Machine Framework

In this section, we describe the top-level model—a hierarchical finite state machine (FSM) that we propose to unify several human accompanying modes in our flying robot. The hierarchical FSM has the main advantage of combining several simple rules to achieve complex human accompanying behaviors. Moreover, hierarchical FSM can be extended easily to consider robot state, environment, and human robot state, with minimal increases in model complexity and computational loads.

Figure 4.1 shows a hierarchical FSM that we have designed for our companion flying robot. The hierarchical FSM is designed to consider the user, environment, and robot states. Note that the list of states in the figure is not exhaustive. In general, user state includes human position and body orientation, hand gestures, voice commands, movements, emotion, etc.; environment state includes obstacles, lighting conditions, wind speed, temperature, humidity, air quality, etc.; and robot state includes battery level, attitude, altitude, position, flying speed, emotion, etc. In the flight experiments, we focus on using human position to realize a few human accompanying behaviors.

From the starting point in **Fig. 4.1**, we expect the flying robot to take-off and start wandering around. When a nearby user is detected (more details in **Chapter 5**), the flying robot would approach the user and have a face-to-face interaction with the user. At this point, in order to have a better understanding of the user, the flying robot could have various

HRI with the user, such as hand detection and hand shape recognition (**Chapter** 6), facial expression recognition (**Chapter** 7), and face alignment (**Chapter** 8). Moreover, while it is not our main focus, it is also possible for a flying robot to exhibit intelligence and emotion in the future. For example, as mentioned in **Section 2.6**, Cauchard et al. presented a HRI experiments with flying robots expressing emotions via movements [118]. Cauchard et al. believe that encoding these characteristics into movements could help users to comprehend the internal states or understand the intention of the flying robot better.

The star-liked structure in the center of **Fig. 4.1** shows the inter-connected human accompanying behaviors that we aim to realize in our flying robot, which include human following, human leading, side-by-side walking, flying high, and human circling. Each flying mode can be switched by either a user, environment, or robot state. For example, if user signals a "start human circling" gesture (hand shape), the flying robot would start to circle around the user. If a potential collision is detected during the human circling mode, the flying robot can automatically go back to the human following mode (inherent collision avoidance, assuming that the user avoids obstacles by nature). When the flying robot intends to save battery, it could land on the shoulder of the user while still maintaining an appropriate altitude to monitor the surroundings[3].

With the top-level hierarchical FSM in mind, in the following sections, we focus on realizing four accompanying behaviors in simulations and experiments: (i) human approaching, (ii) human following, (iii) human circling, and (iv) side-by-side walking. After describing the control mechanism of each accompanying behavior in the next section, we will present the simulation and experiment results.

4.4 Relative Positioning Control

We describe the relative positioning control method used by our companion flying robot in this section. Specifically, we are using the idea of steering behaviors proposed by Reynolds [152] in the computer animation and interactive media fields. While simple, these steering behaviors allow the companion flying robot to achieve smooth navigation in a realistic manner by using simple forces around the flying robot's environment. (See [181] for a great tutorial.) Note that for simplicity, we describe the idea of steering behaviors in the following in 2D space, but it can be generalized to 3D space.

[3] We believe that this is possible when flying robots become smaller, safer, more intelligent in the future.

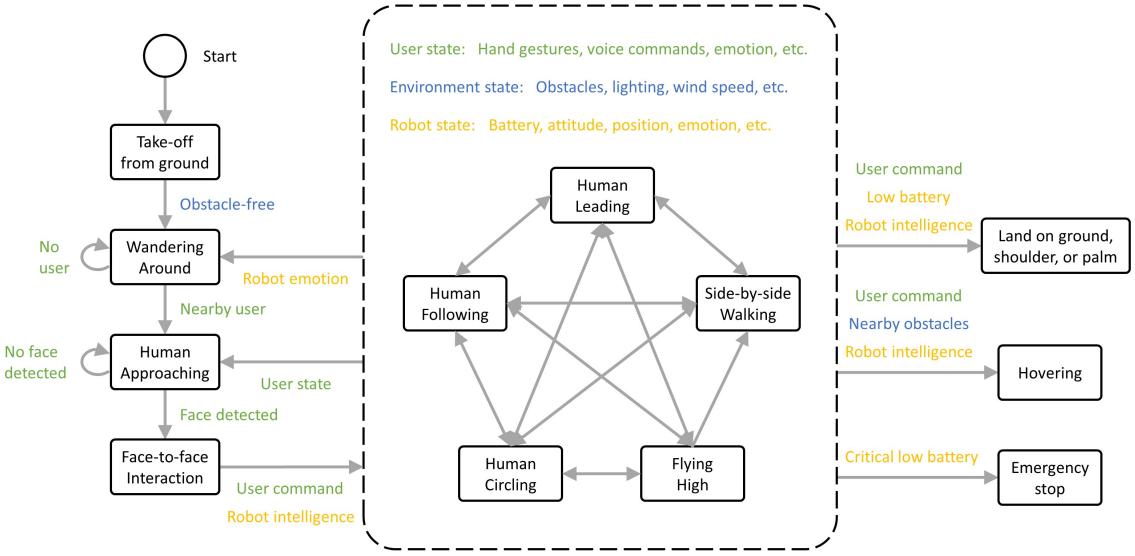

Figure 4.1: An overview of the hierarchical finite state machine designed for the companion flying robot. (See text for details.)

Our main idea is to use simple elementary motion controllers (forward speed controller, yaw orientation controller, lateral speed controller) to achieve natural and smooth accompanying behaviors. Different from the steering behaviors proposed by Reynolds, we have an additional lateral speed controller in flying robot, which allows us to realize a human circling behavior that is not discussed in the original steering behaviors paper.

In the following, we aim to realize four accompanying behaviors with our flying robot, namely *human approaching*, *human following*, *human circling*, and *side-by-side walking*. We first describe the working processes of the four controllers. Then we discuss the simulation and experiment results. For safety reason, instead of performing the experiments with a real person, we apply the human detection and body orientation estimation methods described in **Chapter 5** for a mannequin. While the trained mannequin upper-body detector works as expected, the mannequin body orientation estimator does not have enough accuracy and reliability to start the side-by-side walking experiment (the mannequin is put on a mobile robot and has significant shaking motion noise during movements). Therefore, while we have four simulation results of different human accompanying behaviors, we only focus on the approaching, following, and circling experiments.

4.4.1 Approaching Controller

Figure 4.2 illustrates the approaching behavior of our flying robot, which is analogous to the *seek* behavior proposed by Reynolds. In approaching mode, the steering force is computed from the current velocity vector and the desired velocity vector that is pointing toward the target. In simulation, we assume the flying robot knows the position of the user. In the flight experiments, we use a depth camera and a vision-based method to find the position of the user. The flying robot's speed and yaw orientation are controlled independently by two PID controllers of the holonomic hexacopter (**Section 3.6**). The speed linearly increases with the distance in between the robot and the target (user), and the yaw rate linearly increases with the angle difference in between the current velocity and desired velocity vectors.

Similar to the steering behaviors proposed by Reynolds, our flying robot also has a limited maximum speed and yaw rate. If the desired speed or yaw rate is higher than the maximum setting, the maximum value will be used. Inspired by the *arrival* behavior proposed by Reynolds, we also adopt smooth start and smooth stop behaviors in the approaching controller. Simply put, the flying robot will gradually increase its speed towards the target speed during the start stage. Using smooth start helps to achieve a smooth and natural motion. Similarly, the flying robot will gradually decrease its speed after reaching a circumference around the user.

While it is not implemented in our experiments, the approaching controller can achieve some interesting behaviors with some simple tweaks. For example, if the target position is reset randomly right before the flying robot reach the target, the flying robot would exhibit a natural wandering behavior. On the other hand, if direction of the desired velocity vector is reversed, in contrast to human approaching, the flying robot would exhibits a human avoiding behavior (analogous to *flee* behavior proposed by Reynolds).

4.4.2 Following Controller

The following behavior showed in **Fig. 4.3** is an extension of the approaching behavior—the target starts to move. While the velocity vectors seem to be more dynamic and complex, our simulation and experiment results below show that the flying robot is able to achieve a smooth and natural human following behavior with the same forward speed and yaw rate using elementary motion controllers.

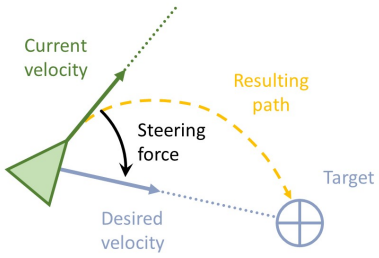

Figure 4.2: Controller model of a flying robot in human approaching mode, where the steering force is computed from the current velocity and the desired velocity that is pointing toward the target.

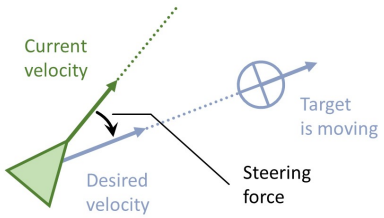

Figure 4.3: Controller model of a flying robot in human following mode, where the steering force is computed from the current velocity and the desired velocity that is pointing toward the target.

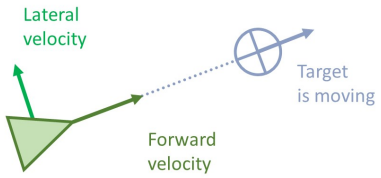

Figure 4.4: Controller model of a flying robot in human circling mode. Essentially, the circling controller is the same with the following controller, but with an additional lateral force applied to the flying robot.

4.4.3 Circling Controller

The circling behavior showed in **Fig. 4.4** is an extension of the following controller—a constant lateral speed is applied to the flying robot. Together with the same forward speed and yaw rate elementary motion controllers, our simulation and experiment results below show that the trio is able to achieve a smooth and natural human circling behavior when the target is not moving. In addition, when the target starts to move, the trio is able to exhibit both following and circling behaviors at the same time.

4.4.4 Side-By-Side Walking Controller

The side-by-side walking behavior is a circling behavior plus a constraint—the flying robot will try to go to the left/right hand side of the user, depending on which side is closer. Once again, together with the same forward speed and yaw rate elementary motion controllers, our simulation results below show that the trio plus a simple constraint is able to achieve a smooth and natural human walking behavior when the target is moving. This side-by-side walking controller shows that by including a simple constraint, elementary motion controllers are able to achieve natural and rich accompanying behaviors.

4.5 Simulation Results

We performed all human accompanying behavior simulations in 2D with a real-time visualization software—Processing 3 [182]. We wrote the code for the four discussed human accompanying behaviors and summarize their results in this section.

4.5.1 Approaching Simulation

In the simulation, we assume the flying robot knows the position of the user. The flying speed linearly increases with the distance between the robot and the target (user). Flying speeds that are higher than a limit will be truncated to a maximum speed value of three pixels per simulation cycle. The yaw rate linearly increases with the angle difference in between the current and desired velocity vectors. Similarly, yaw rates higher than the maximum yaw rate of 18° per simulation cycle will be truncated.

Figure 4.5 shows the simulation result of a flying robot's approaching behavior with eighteen frames. The triplet on the top left corner of every frame represents the horizontal position, vertical position, and yaw orientation of the flying robot. In the 2D simulation, the target user has a white cross shape and is set at the center of a square room. The flying robot, represented as a white circle with an arrow pointing to its forward direction, gradually approaches the target user at the center.

Figure 4.6 and **Fig. 4.7** show two other simulation results of a flying robot's approaching behavior. In both cases, the flying robots have starting yaw orientations that are not pointing towards the target user. The independent yaw rate controller actively turns the flying robot towards the target user in the center at the beginning of both simulations. While this approach does not take the shortest path, our main objective is to achieve a smooth and natural accompanying behavior.

4.5.2 Following Simulation

Figure 4.8 shows the simulation result of a flying robot's following behavior with thirty-six frames. The two triplets on the top left corner of every frame represent the horizontal positions, vertical positions, and yaw orientations of the flying robot and the moving user. In the 2D simulation, the target user is positioned at the starting point of a white arrow (with the arrow pointing to his/her walking direction) and is not allowed to move laterally. The flying robot, represented as a white circle with an arrow pointing to its forward direction, follows the moving target user. Note that the flying robot is not following the user path exactly. It has been pointed out by Gockley et al. [164] that following the user's path is rated unnatural by participants in their HRI experiments. Our controller uses the latest measurements for direct human following and does not follow the user path exactly.

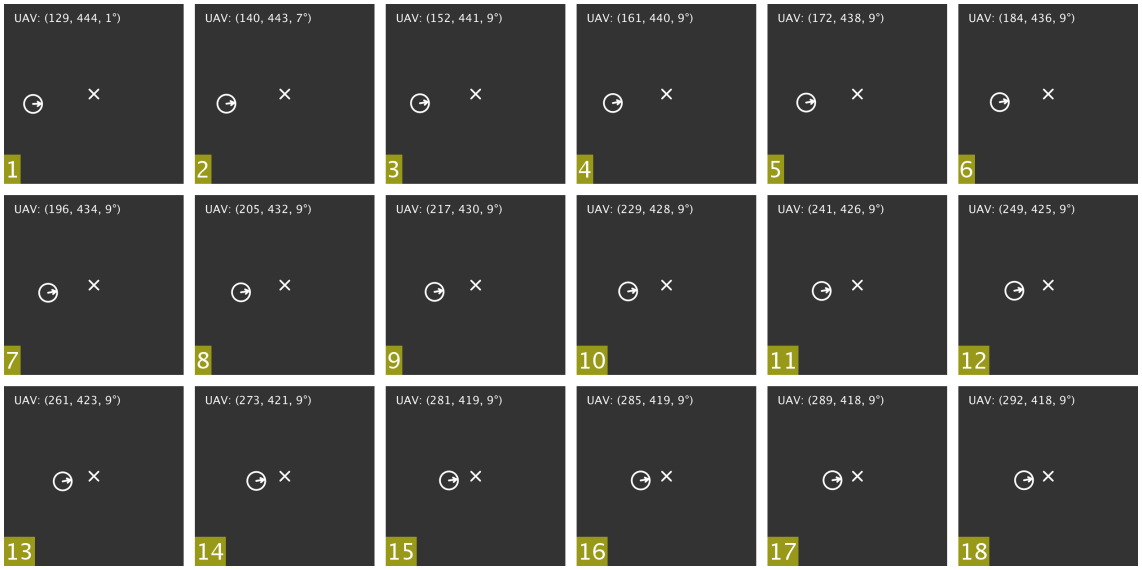

Figure 4.5: 2D simulation of approaching behavior of a flying robot in Processing environment. (See text for details.)

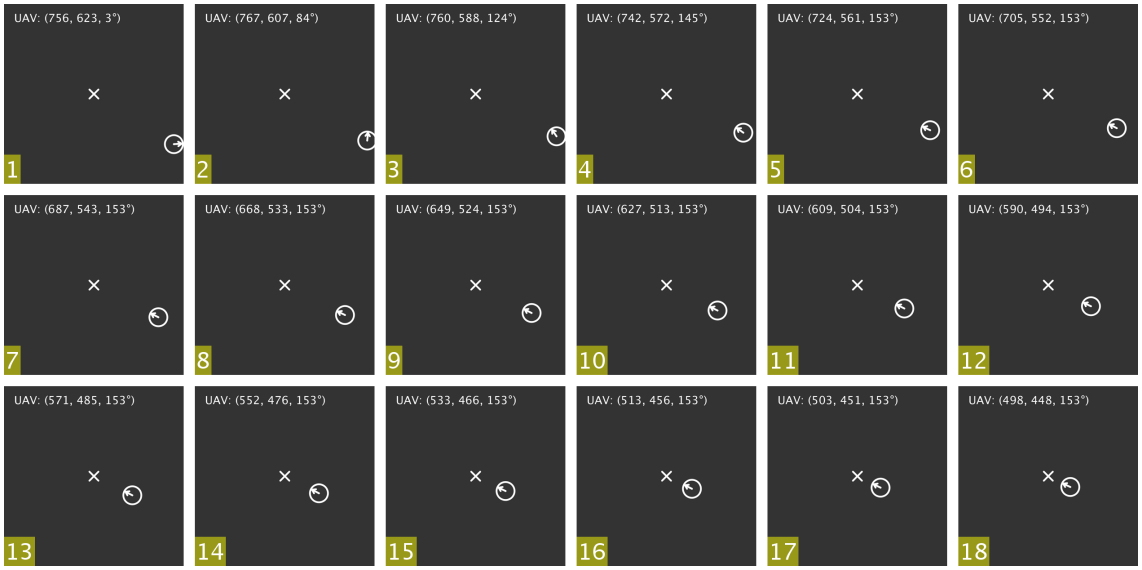

Figure 4.6: 2D simulation of approaching behavior of a flying robot in Processing environment. (See text for details.)

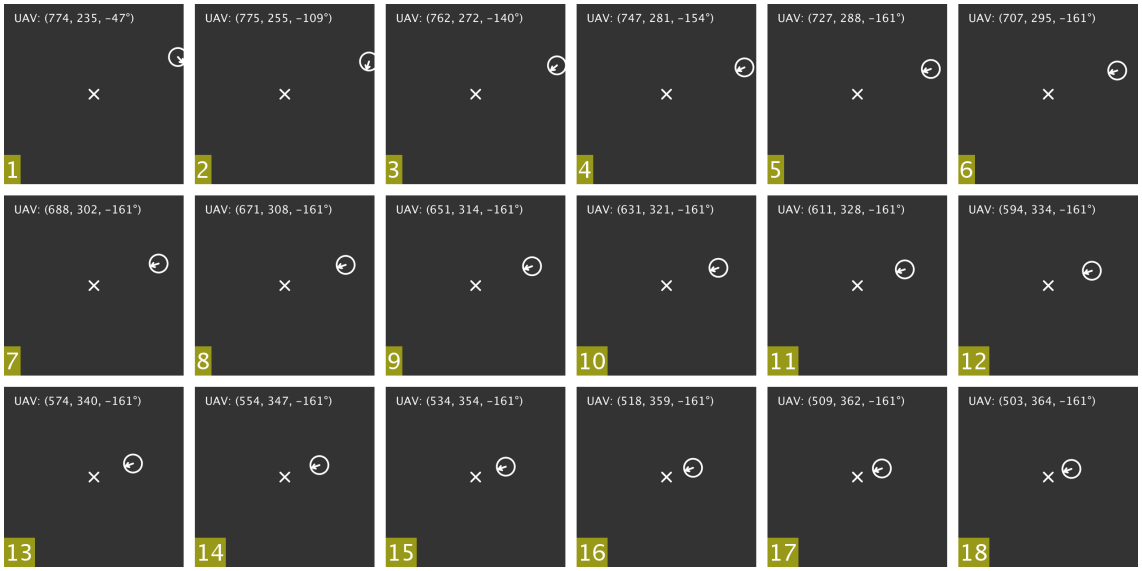

Figure 4.7: 2D simulation of approaching behavior of a flying robot in Processing environment. (See text for details.)

Note that in **Figure 4.8**, when the user moves at a slower speed, the flying robot can still maintain a preset distance with the moving user. On the other hand, when the user moves at a speed faster than the maximum speed of the flying robot, the flying robot can still follow the user smoothly and naturally, even though it cannot catch up the user in the simulation.

4.5.3 Circling Simulation

Figure 4.9 shows the simulation result of a flying robot's circling behavior with thirty-six frames. The two triplets on the top left corner of every frame represent the horizontal positions, vertical positions, and yaw orientations of the flying robot and the moving user. In the 2D simulation, the target user is positioned at the starting point of a white arrow (with the arrow pointing to his/her forward direction) and is not allowed to move in this example. The flying robot, represented as a white circle with an arrow pointing to its forward direction, circles around the non-moving target user at the center of the room. Note that the flying robot successfully maintains a preset distance with the user.

Figure 4.10 shows another simulation result of a flying robot's circling behavior with the user moving. In this example, the flying robot successfully maintains a preset distance with

Figure 4.8: 2D simulation of following behavior of a flying robot in Processing environment. (See text for details.)

the user while able to exhibit circling behavior at the same time. For safety purposes, the user is not allowed to move faster than the flying robot; otherwise, collision would happen when the flying robot is in front of the user, since the flying robot is not able to avoid the user with a limited backward and lateral speed. When the user is moving slower than the flying robot, we confirmed that the flying robot is able to avoid the collision, provided that a provisional backward motion is allowed.

4.5.4 Side-By-Side Walking Simulation

Figure 4.11 shows the simulation result of a flying robot's side-by-side walking behavior with thirty-six frames. The two triplets on the top left corner of every frame represent the horizontal positions, vertical positions, and yaw orientations of the flying robot and the moving user. In the 2D simulation, the target user is positioned at the starting point of a white arrow (with the arrow pointing to his/her forward direction) and does not move in the first six frames. The flying robot, represented as a white circle with an arrow pointing to its forward direction, starts at a position behind the user. The flying robot then circles around the user (who is not moving) and its lateral speed decreases gradually when it almost reach to the side of the user. From the seventh frame, the user starts to move and the flying robot is able to fly side-by-side with the user.

From the thirteenth frame onwards in **Fig. 4.11**, the user starts to change his/her body orientation while moving. Note that a small change of the user body orientation means a relatively large change of the side position of the user; the larger the distance between the user and the flying robot, the larger the position change. Since the flying robot has a small lateral speed, it will assume the circling behavior in order to reach the side of the user. At the twenty-first frame, the flying robot notes that it is easier to go to the left side of the user and changes its lateral speed from the right to the left hand side. After that, the user maintains his/her body orientation, and the flying robot successfully flies along the side of the user.

4.6 Experiment Results

We summarize the experiment results of human approaching, following, and circling in this section. As mentioned earlier, we use a mannequin (instead of a real person) in our experiment for safety reason. The mannequin has a height of 150 cm and is put on a Pioneer

Figure 4.9: 2D simulation of circling behavior of a flying robot in Processing environment when the target is not moving. (See text for details.)

Figure 4.10: 2D simulation of circling behavior of a flying robot in Processing environment while the target is moving. (See text for details.)

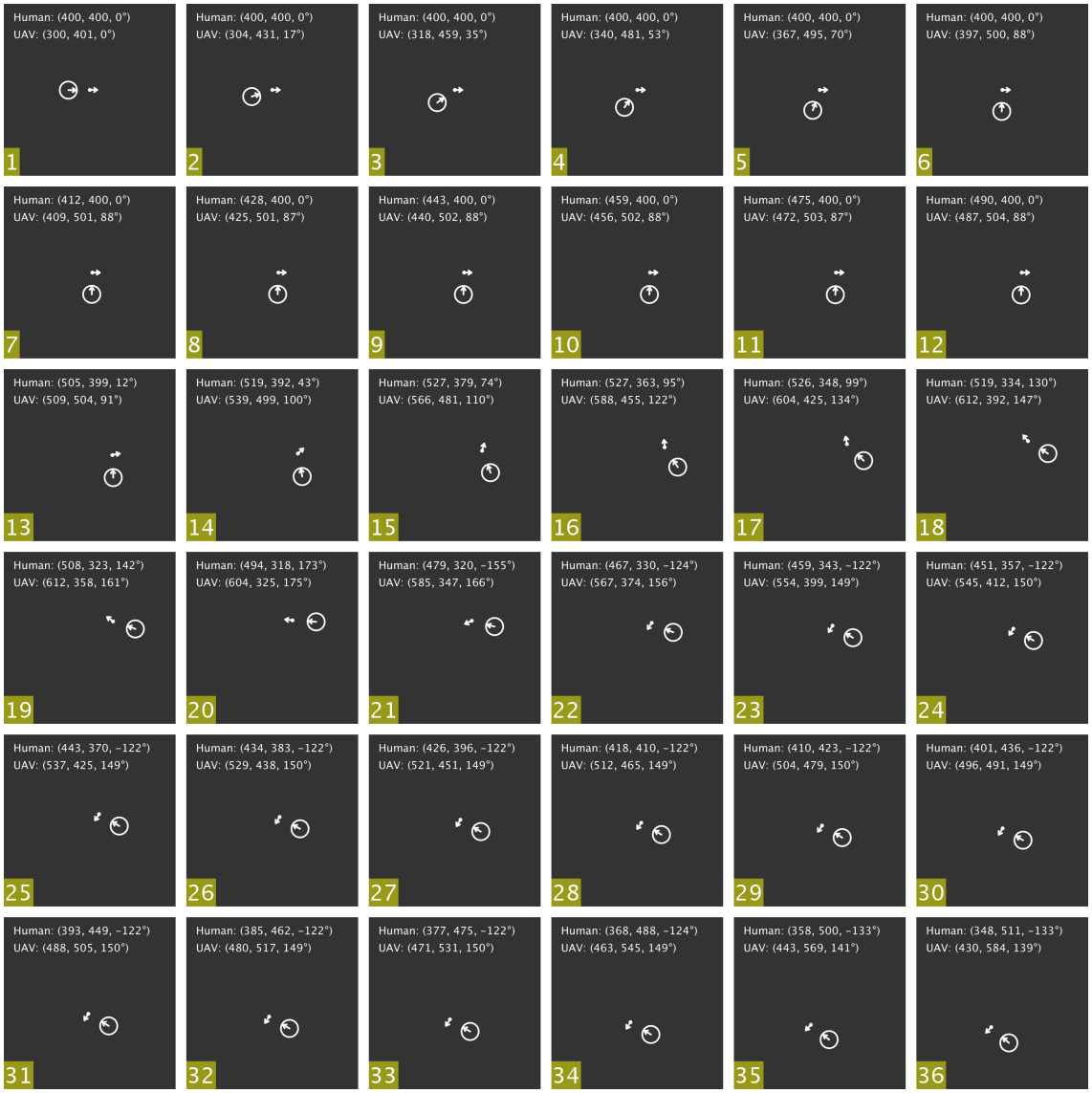

Figure 4.11: 2D simulation of side-by-side walking behavior of a flying robot in Processing environment. (See text for details.)

3-DX mobile robot. The mobile robot has an integrated Raspberry Pi computer and can be controlled wireless via Bluetooth. In addition, we use a tether-powered hexacopter in our experiment for long time experiments. The working principle of this tether-powered holonomic hexacopter is same with the battery-powered holonomic hexacopter described in **Chapter 3**. The holonomic hexacopter has a Xtion depth camera for mannequin detection.

4.6.1 Approaching Experiment

Figure 4.12 illustrates the first experiment results of the approaching behavior with the tethered holonomic hexacopter over a 25-second flight. The top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. The top plot has a spike at the third second due to the mannequin detection failure. However, it does not affect the accompanying performance significantly. We plan to use a simple low pass filter to solve this issue in our future work. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The vertical white lines in the sub-figures in the position where the hexacopter is 200 cm away from the mannequin.

From the plots and sub-figures, we can observe that when the approaching controller is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points. We set the social distance to 200 cm in our experiments. To ensure that the flying robot approaches the mannequin responsively, the hexacopter is set to have a higher horizontal thrust. As a result, we also observe a small overshot (about 20 cm) in the plot and sub-figure (b). As shown in the sub-figure (b), we find that this small overshot is acceptable, as long as the hexacopter does not harm the mannequin or an accompanied person (stay 100 cm away from the mannequin). After, the hexacopter maintains its position and facing direction towards the mannequin well (sub-figures (c), (d), (e), and (f)).

Figure 4.13 illustrates the second experiment results of the approaching behavior with the tethered holonomic hexacopter over a 30-second flight. Similarly, the top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. Different from the first approaching experiment, the top plot has no mannequin detection failure and has a nicer plot. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The vertical white lines in the

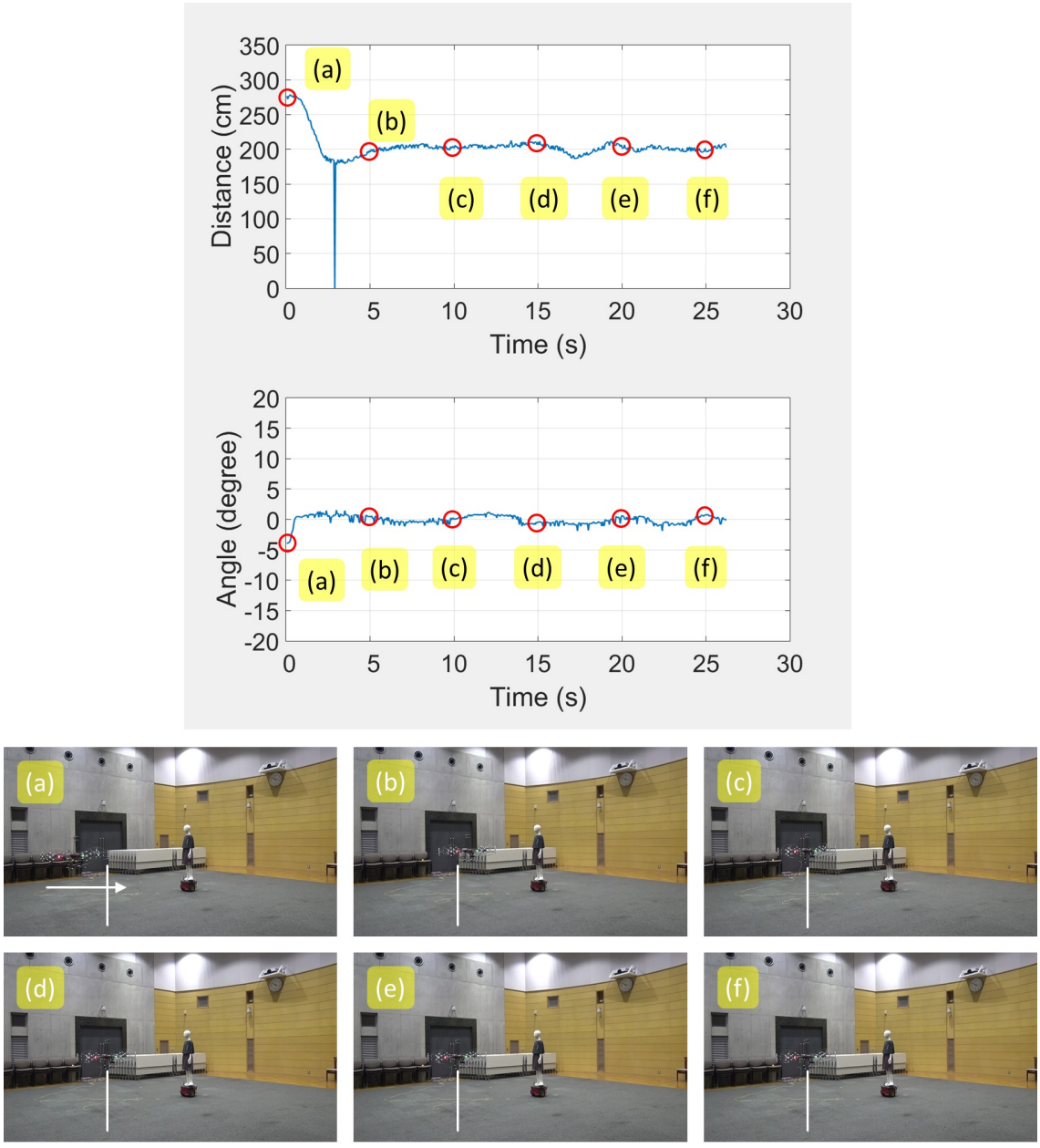

Figure 4.12: First experiment of "human" approaching with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 30-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

sub-figures in the position where the hexacopter is 200 cm away from the mannequin.

From the plots and sub-figures, we can observe that when the approaching controller is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points (the social distance is set to 200 cm in our experiments). Once again, to ensure that the flying robot approaches the mannequin responsively, the hexacopter is set to have a higher horizontal thrust. As a result, we also observe a small overshot (about 20 cm) in the plot and sub-figure (b). As shown in the sub-figure (b), we find that this small overshot is acceptable, as long as the hexacopter does not harm the mannequin or an accompanied person (stay 100 cm away from the mannequin). After, the hexacopter maintains its position and facing direction towards the mannequin well (sub-figures (c)–(f)).

In both experiments, the distance and facing direction plots have noisy measurements. This is inevitable because the Xtion depth camera used in our experiments relies on structured infrared lights for depth measurements. As mentioned in the first experiment, we plan to use a simple low pass filter to smoothen the measurement noise in our future work.

4.6.2 Following Experiment

Figure 4.14 illustrates the first experiment results of the following behavior with the tethered holonomic hexacopter over a 60-second flight. The top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. The top plot has two spikes due to the mannequin detection failure. However, they do not affect the accompanying performance significantly. We plan to use a simple low pass filter to solve this issue in our future work. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The white and green lines in the sub-figures represent the moving direction of the hexacopter and the mannequin.

From the plots and sub-figures, we can observe that when the following controller (same with the approaching controller, except that the mannequin is moving in this case) is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points (with a small overshot). We set the social distance to 200 cm in our experiments. From the sub-figure (b), we start to move mannequin around. At this moment, the hexacopter only moves slowly since the distance between itself and the mannequin is small. From the sub-figure (c), the hexacopter moves faster when the distance between itself

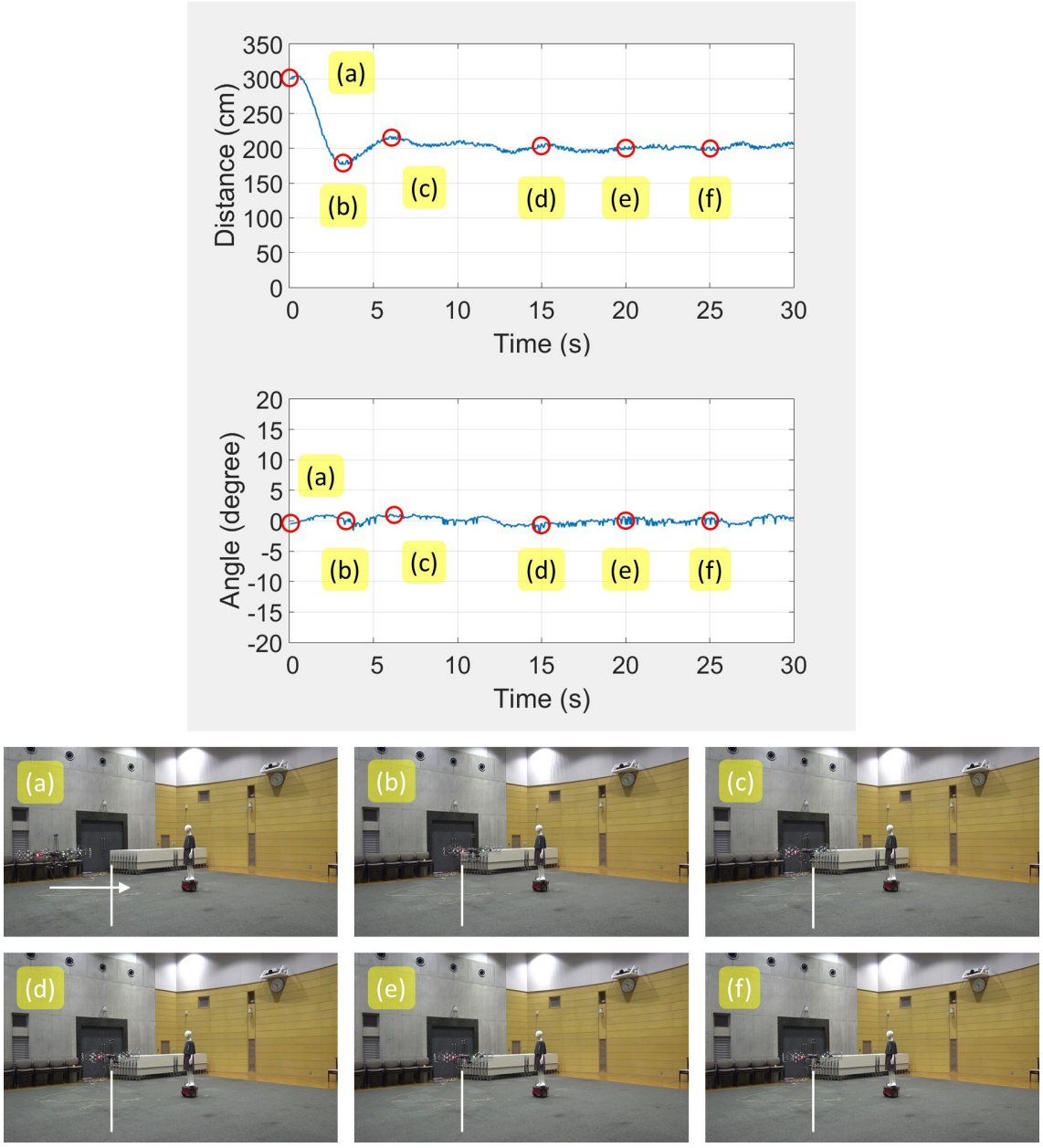

Figure 4.13: Second experiment of "human" approaching with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 30-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

and the mannequin becomes larger. From the sub-figure (d), we move the mannequin in the reverse direction and the hexacopter is able to maintain the preset following distance of 200 cm within error of 50 cm. While this error seems to be large, the following behavior in the experiment has less robotic feel (more natural). At sub-figures (e) and (f), we stop moving and start to rotate the mannequin, the hexacopter is able to maintain its position and facing direction towards the mannequin well.

Figure 4.15 illustrates the second experiment results of the following behavior with the tethered holonomic hexacopter over a 60-second flight. Similarly, the top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. Different from the first following experiment, the top plot has no mannequin detection failure and has a nicer plot. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The white and green lines in the sub-figures represent the moving direction of the hexacopter and the mannequin.

From the plots and sub-figures, we can observe that when the following controller is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points (with a small overshot). We set the social distance to 200 cm in our experiments. From the sub-figure (b) to (c), we start to move mannequin around. At these moments, the hexacopter only moves slowly since the distance between itself and the mannequin is small. From the sub-figure (d), the hexacopter moves faster when the distance between itself and the mannequin becomes larger. At this point, we also move the mannequin in the reverse direction and the hexacopter is able to maintain the preset following distance of 200 cm within error of 50 cm. Similar to the first following experiment, while this error seems to be large, the following behavior in the second experiment has less robotic feel (more natural). From the sub-figure (e), the hexacopter is able to maintain its position and facing direction towards the mannequin well while we continue to move the mannequin. The maximum error of the hexacopter's facing direction towards the mannequin is less than 10°. Considering that we are using a low resolution depth camera and a simple motion controller in our experiment, we find that this error is acceptable.

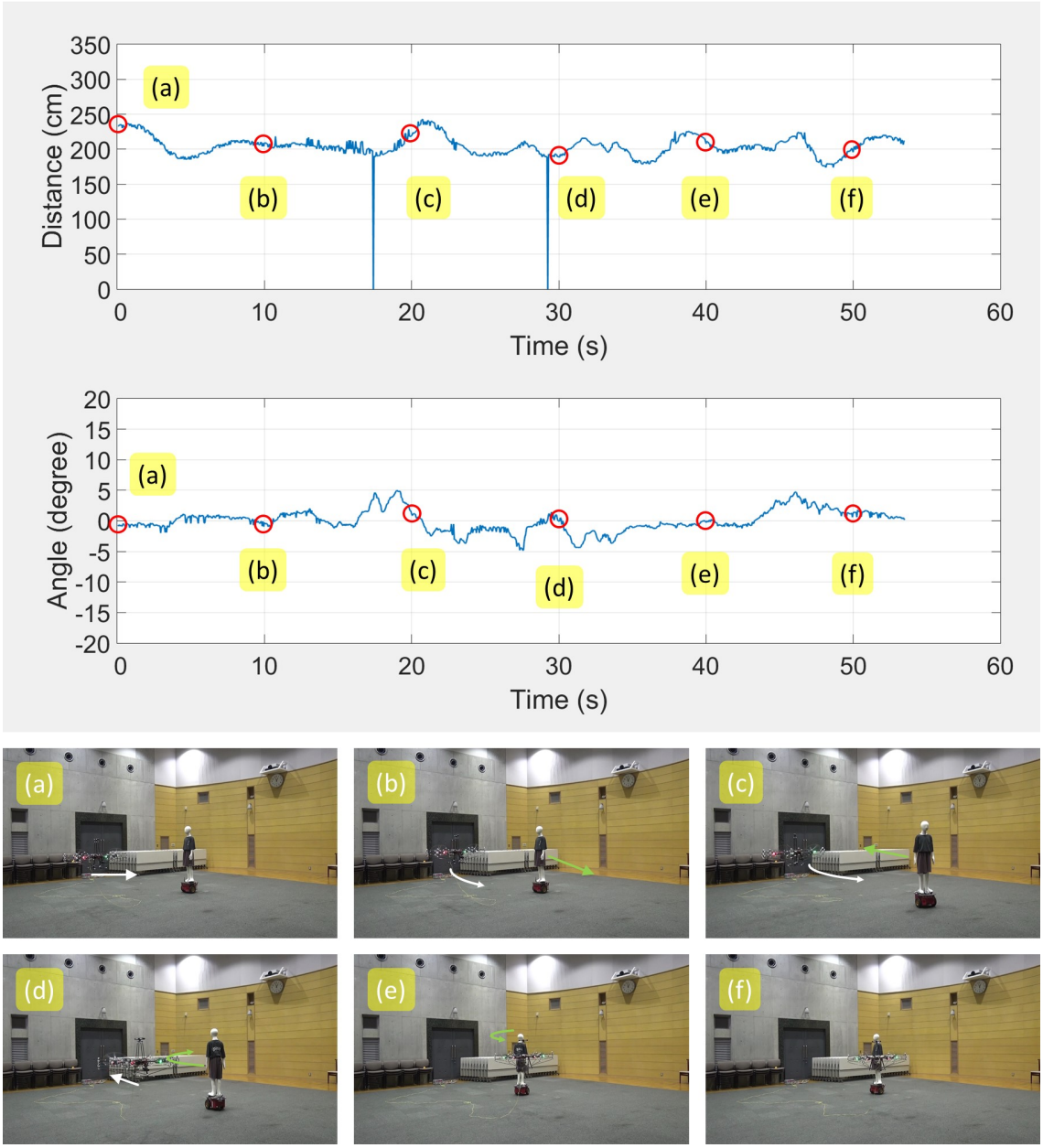

Figure 4.14: First experiment of "human" following with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 60-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

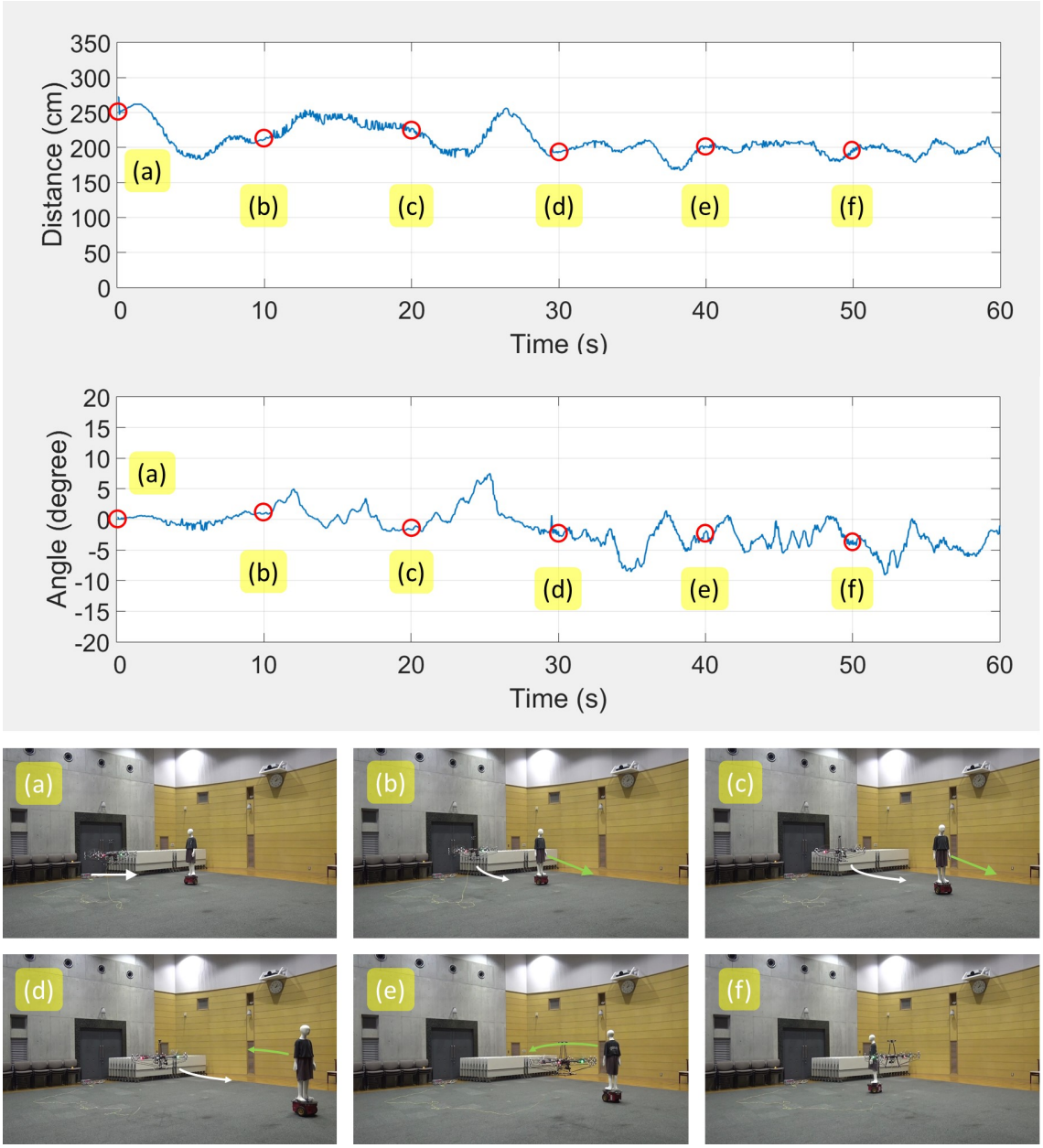

Figure 4.15: Second experiment of "human" following with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 60-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

4.6.3 Circling Experiment

Figure 4.16 illustrates the first experiment results of the circling behavior with the tethered holonomic hexacopter over a 60-second flight. The top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The white lines in the sub-figures represent the moving direction of the hexacopter.

From the plots and sub-figures, we can observe that when the circling controller (same with the approaching controller, except that a small lateral force is applied to the hexacopter when it is near to the social distance circumference of the mannequin) is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points (with a small overshot). We set the social distance to 200 cm in our experiments. At sub-figure (b), the hexacopter maintains its position and facing direction towards the mannequin. From the sub-figure (c) to (f), a small lateral force is applied and the hexacopter starts to circle around the mannequin slowly. We apply a small lateral force in the experiment in order to ensure that the mannequin detectors work reliably. The depth camera we are using now has a limited field of view (about 30°on one hand side) and the onboard computer has detection speed of around 10Hz. Increasing the hexacopter's circling speed and onboard computer's detection speed will be considered in our future works.

Figure 4.17 illustrates the second experiment results of the circling behavior with the tethered holonomic hexacopter over a 60-second flight. Similarly, the top plot represents the distance between the hexacopter (specifically the onboard Xtion depth camera at the center) and the mannequin. The bottom plot represents the facing direction of the hexacopter towards the mannequin. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. The white lines in the sub-figures represent the moving direction of the hexacopter.

From the plots and sub-figures, we can observe that when the circling controller is activated at the beginning (sub-figure (a)), both the distance and facing direction quickly converge to their equilibrium points (with a small overshot). We set the social distance to 200 cm in our experiments. From the sub-figure (b) to (c), a small lateral force to the left is applied and the hexacopter starts to circle around the mannequin slowly. As explained in

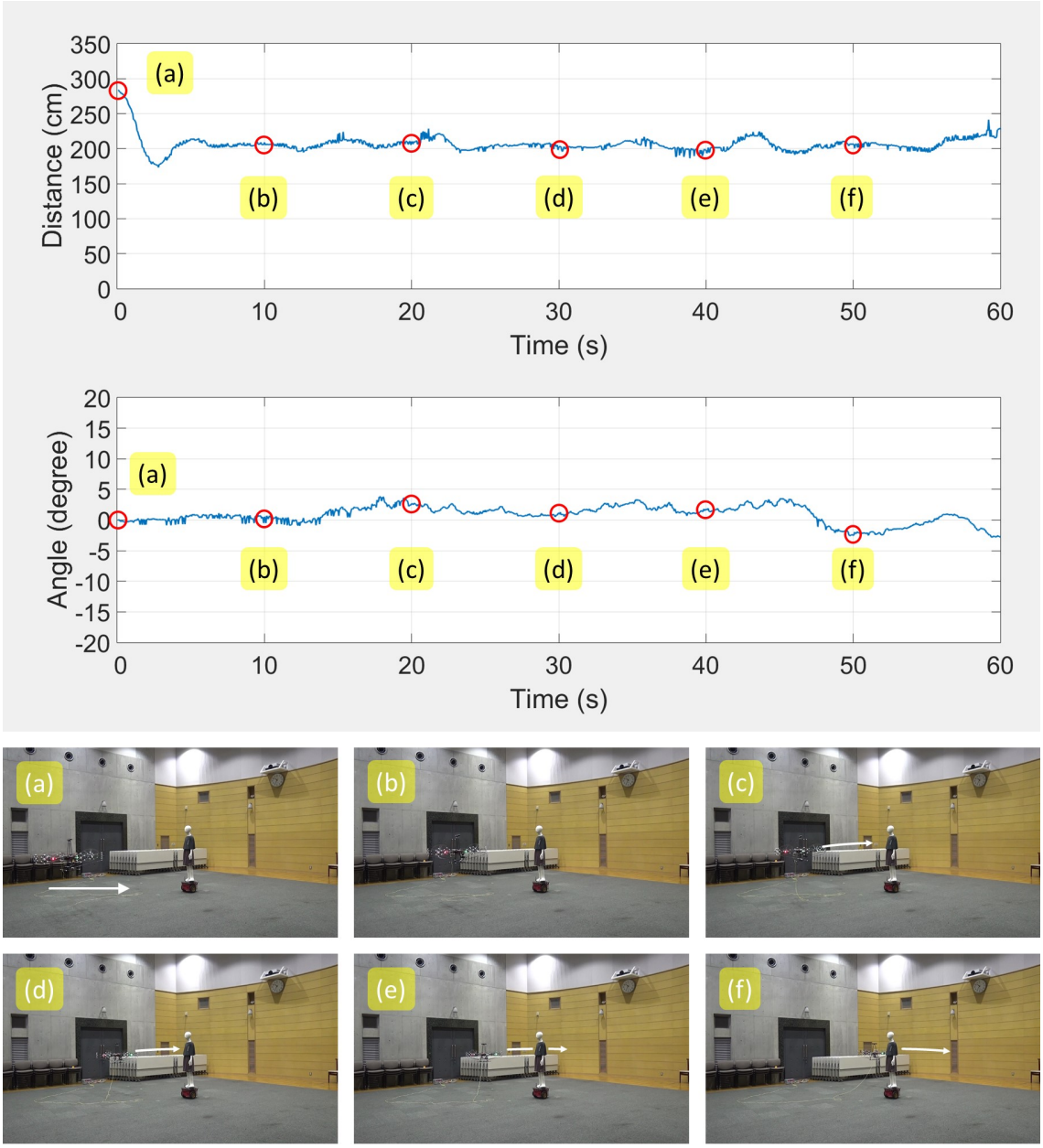

Figure 4.16: First experiment of "human" circling with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 60-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

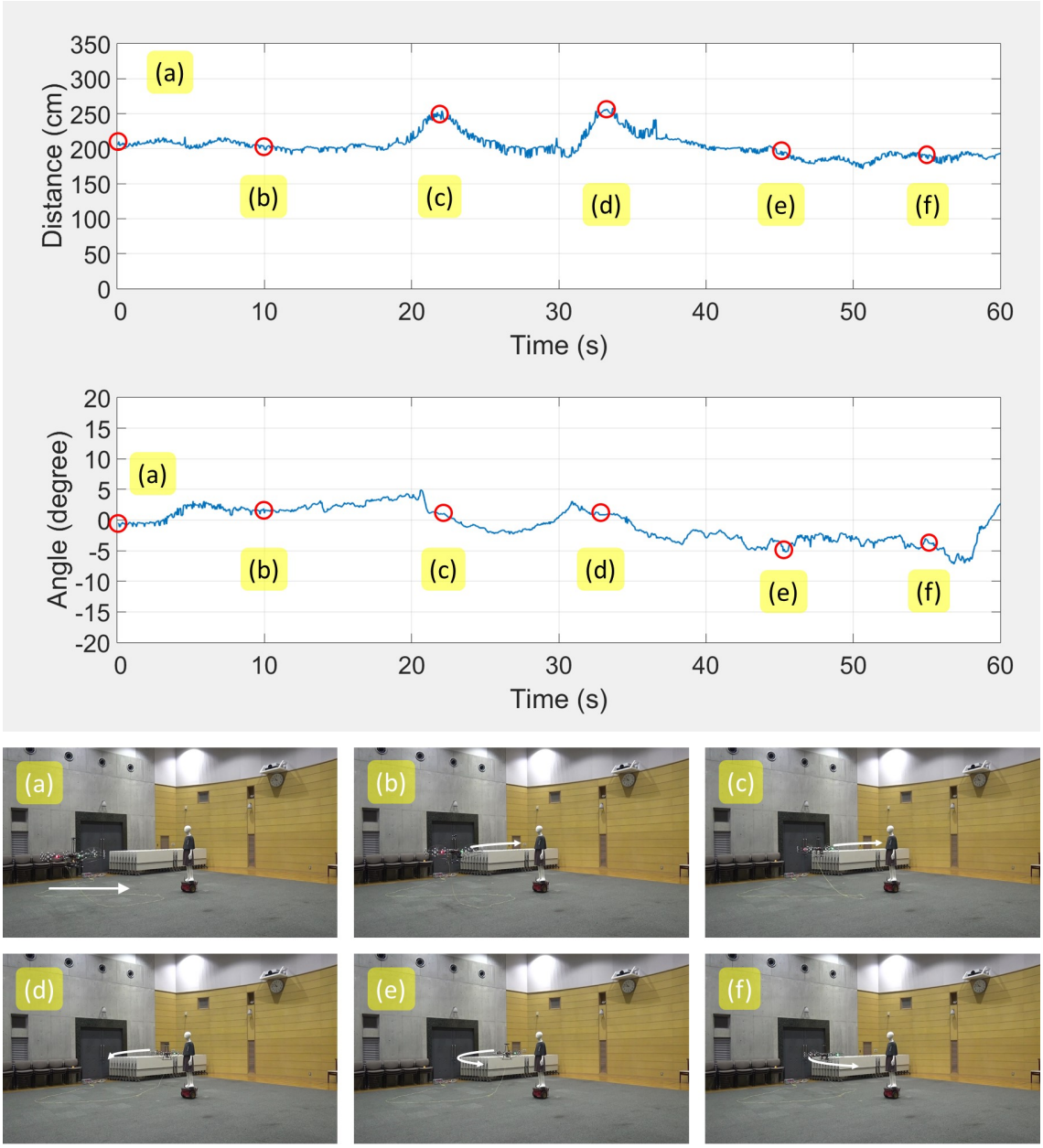

Figure 4.17: Second experiment of "human" circling with a tethered holonomic hexacopter. The top plots represent the distance and facing direction between the hexacopter and the mannequin over a 60-second flight. The sub-figures at the bottom are the screenshots at the moments of the six red circles in the plot. (Best viewed in color. See text for details.)

the first experiment, we apply a small lateral force in order to ensure that the mannequin detectors work reliably. From the sub-figure (d) to (f), a small lateral force to the right is applied and the hexacopter starts to circle around the mannequin slowly in the reverse direction. In two occasions (at the 22^{nd} and 33^{rd} seconds), the error of the distance almost exceeds an error of 50 cm. We believe that this is caused by the slow response of the simple controller used in our experiment. We aim to incorporate a PID controller into the bottom-level motion controller and use a downward-looking optical flow sensor for horizontal velocity control in our future works.

Chapter 5
Human Sensing Interface I: Human Upper-Body Detection and Orientation Estimation

5.1 Introduction

Human body orientation (HBO) estimation is an essential method for many human-robot interaction (HRI) and ubiquitous computing applications. To-date, many existing robots only detect the locations of humans and perform simple human following applications [183, 184]. With HBO estimation, we believe that robots and computer agents could achieve more natural and richer interactions with humans. For example, a flying robot could reasons the intention of an accompanied person if it knows the body orientation the person.

In this chapter, we present a robust technique to estimate HBO in real-time with either a color or depth image. To the best of our knowledge, most previous approaches focus on multi-class classification [185–187] or rely on filtering methods such as Kalman filters or particle filters to improve the tracking performance [186–188]. Instead, our method uses a random forest regressor to perform a continuous and full 360° HBO estimation. Compared to previous approaches, our approach is not only faster but also more efficient.

Secondly, since the HBO (-180°<θ<180°) is not continuous in the regression space, we find that the estimation results near to the -180° and 180° regions deteriorate significantly. Instead of estimating the HBO directly, we propose a xy-based random forest regression (such that -180° would equal to 180° in the regression space) that could improve the estimation accuracy in both theory and the actual experiments.

Last but not least, we present a full human upper-body detection and body orientation estimation framework. This full framework has two main merits: (1) we could use the same extracted features for both human body detection and orientation estimation and therefore be able to achieve better efficiency; (2) instead of applying HBO estimation onto only one detected human upper-body image patch after applying non-maxima suppression (a common method used to suppress multiple detections near to the target), we apply HBO estimation onto all detected human upper-body image patches right before the non-maxima suppression step and take their mean as the final estimation result. As we will discover soon in the experiments section, this technique could improve the performance significantly.

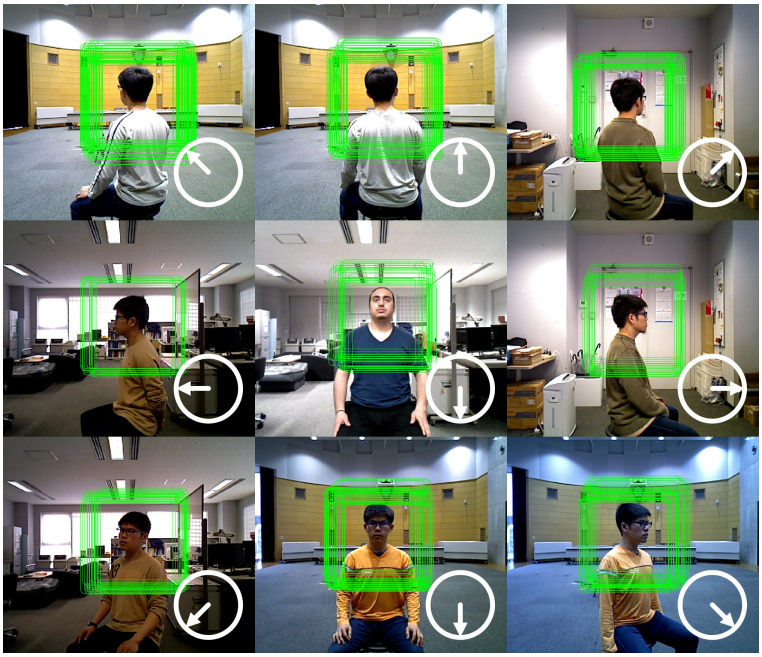

Figure 5.1: Our human body orientation (HBO) estimation method produces continuous and full 360° estimation results.

The rest of this chapter is organized as follows: In **Sect. 5.2**, we summarize general approaches that are related to HBO. In **Sect. 5.3**, we describe the human upper-body detection method used in our experiment, followed by our HBO estimation method in **Sect. 5.4**. In **Sect. 5.5**, we describe our data collection process and provide experiment results in **Sect. 5.6**. Last, we discuss some insights of our method in **Sect. 5.8** and conclude our work in **Sect. 5.9**.

5.2 Related Works

There are many previous works that are related to HBO detection and tracking. For the ease of review purposes, we divide related works into four categories: color-based, depth-based, fusion-based, and lidar-based approaches.

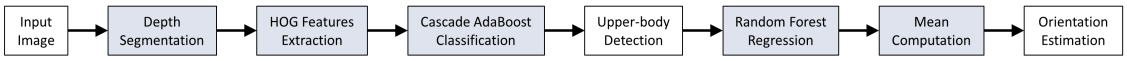

Figure 5.2: An overview of our human upper-body detection and orientation estimation pipeline. Depth segmentation is used for performance speedup in the sliding window search process. Histogram of oriented gradients (HOG) features are used together with a cascade AdaBoost classifier for human upper-body detection. Note that we do not use non-maxima suppression. Random forest regression is applied to all detected human upper-body image patches (same HOG features are used). Regression results of all image patches and all regression trees in the forest are averaged to obtain human body orientation eventually.

5.2.1 Color-Based Approaches

Color-based multi-class classification using a normal color camera is common. For example, Weinrich et al. [185] designed multiple support vector machines (SVMs) as binary decision makers inside a decision tree to perform a 8-class (i.e., 0°, 45°, ..., 315°) upper-body orientation classification. Similarly, Ardiyanto & Miura [188] used random forest to perform a 8-class upper-body classification and employed an Unscented Kalman Filter (UKF) to improve the HBO tracking performance.

Instead of using a multi-class classifier, Flohr et al. trained eight detectors to evaluate how well an image patch corresponds to a specific body orientation and then used the detector responses for particle filter-based tracking. On the other hand, Chen et al. assumed that a human upper-body's histogram of oriented gradients (HOG) descriptor can be approximated by a small number of training data and used a sparse coding method to perform 8-class classification. In their following work, Chen et al. [189] proposed a coupled learning method for head and body orientation estimation but essentially still perform a 8-class classification to estimate HBO.

Different from previous approaches, Rybok et al. [190] employed a background subtraction method to extract silhouette information from multiple cameras for a 12-class HBO classification. While they have finer quantization in their designed multi-class classifier, they still do not take full advantage of the available information inside an image for a continuous HBO estimation with regression. Moreover, their method is not directly applicable for most robot applications. Normally, robots are expected to move and background subtraction methods could not work reliably.

5.2.2 Depth-Based Approaches

Using a depth camera, Yang et al. [191] proposed a HBO estimation system that is similar to our concept in this work. Specifically, they extracted hand-tuned features from a fixed region inside the detected human upper-body depth image and used a support vector regressor (SVR) to estimate the continuous HBO. However, their method focuses on frontal images (-90°<θ<90°). In contrast to their method, our method produces a continuous and full 360° HBO estimation.

Instead of estimating HBO directly, Shotton et al. [192] tackled a more general problem, namely human pose estimation, where they estimate the locations of body joints from a single depth image. Note that with the locations of human body shoulder joints, it is possible to estimate HBO. Based on the same settings, Ye et al. [193], Jung et al. [194], and Zhang et al. [195] improved the human pose estimation in term of accuracy, speed, and robustness to partial occlusions, respectively. Compared to their approaches, our HBO estimation method is more direct and faster.

5.2.3 Fusion-Based Approaches

To reduce the color-based approaches' sensitivity to cluttered environments or illumination changes and to reduce the depth-based approaches' sensitivity to noisy measurement, Liu et al. [196] proposed a method to take full advantage of the RGB-Depth information for HBO. However, similar to most previous works (except [191]), they only focus on eight non-overlapping body orientation classes. Furthermore, their method requires superpixel extraction and could not achieve real-time performance. Similarly, Shinmura et al. [197] took advantage of the RGB-Depth information and used a 8-class SVM classifier to estimate the eight discrete HBO.

5.2.4 Lidar-Based Approaches

In addition to color and depth cameras, 2D and 3D lidar sensors have also been used for HBO estimation. Based on our survey, all lidar-based approaches rely on filtering methods to track the HBO. For example, Glas et al. [198] presented a HBO tracking method based on a 2D lidar sensor by using a particle filter with an adaptive shape modeling algorithm. Kobayashi et al. [199] also employed a particle filter to track the HBO by

approximating human body shape with a simple ellipse model. Compared to their approaches, our method is based on pure regression and does not require the design of likelihood functions of the filters. Designing likelihood functions can be quite tedious and time-consuming in some cases.

On the other hand, by assuming that the human always face toward his/her walking direction, Shackleton et al. [200] and Shao et al. [201] proposed to track HBO with a Kalman filter and particle filter respectively. In contrast to their method, our method does not have this assumption and could estimate HBO and track human walking direction independently.

Lastly, it is also worth noting that Ziegler et al. [202] combined lidar and wearable sensors information with a particle filter to estimate full human body pose. However, in contrast to their method, our method requires no wearable sensor. The wearable sensor can sometimes be an inconvenience.

5.3 Human Upper Body Detection

In this work, we present a full human detection and body orientation estimation framework (**Fig. 5.2**). Note that we use the same extracted features from the image for both human upper-body detection and orientation estimation. In this way, we could achieve both human upper-body detection and orientation estimation at minimal processing time. This section gives an overview of our human upper-body detection method (the first half of **Fig. 5.2**). While it is more common for human full-body detection in the literature, we focus on upper-body detection because of its robustness to occlusion. In addition, human upper-body detection can be applied to a wide range of scenarios such as standing, sitting, and even riding a bicycle.

Similar to the human full-body detection method proposed by Dalal & Triggs [203], we use HOG features for human upper-body detection. However, we do not use the SVM for classification since it is time-consuming. To achieve efficient performance, we train a 4-stage cascade AdaBoost classifier [204]. From the literature, we find that our human upper-body detection approach is similar to the work by Zhu et al. [205].In the following, we briefly review the HOG features extraction process and AdaBoost classification process.

5.3.1 HOG Features Extraction

Histogram of oriented gradients (HOG) features were first proposed by Dalal & Triggs [203] for human detection. While being similar to scale-invariant feature transform (SIFT) feature, HOG represents dense coding of image and possesses a few different implementation details such as number of histogram bins and local contrast normalization. In a nut shell, HOG feature can be computed in the following steps:

1. Image is first convoluted with 1-D Sobel filters, i.e. [-1,0,1] and $[-1,0,1]^T$, in order to compute the horizontal gradient dx and vertical gradient dy at each pixel.

2. With horizontal and vertical gradients, magnitude and orientation at each pixel can be computed as $\sqrt{(dx)^2 + (dy)^2}$ and $tan^{-1}(\frac{dy}{dx})$ respectively.

3. Image is divided into 8×8 cells. Orientation values inside each cell are quantized into a 9-bin histogram, where the magnitude value of each pixel represents the binning weights, and bilinear interpolate is used during the binning process.

4. Four adjacent cell histograms are locally normalized. Each block has 50% overlapping with adjacent block and all normalized block histograms are concatenated into a 1-D feature vector.

5.3.2 AdaBoost Classification

AdaBoost is a useful machine learning algorithm that was proposed by Freund & Schapire [204]. In general, AdaBoost iteratively looks for the best training feature to build a weak classifier in each training cycle. After each feature selection, sample weights are re-adjusted according to the local classification error. The same process repeats until a certain number of features are selected. Weak classifiers, normally linear decision stumps with only threshold and polarity parameters, are then combined together to form a strong classifier. Mathematically, the linear decision stump can be written as

$$h_t(x, p, \theta) = \begin{cases} +1, & \text{if } p \cdot f(x) < p \cdot \theta \,, \\ -1, & \text{otherwise} \,. \end{cases} \tag{5.1}$$

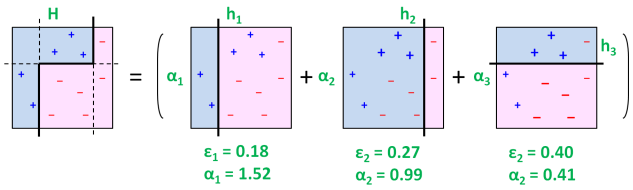

Figure 5.3: AdaBoost classifier illustration. AdaBoost select the best feature for one weak classifier at one time, and then update sample weights based on the local classification error, and repeat the selection process iteratively. The final strong classifier combines several weak classifiers in an additive manner, where voting weights of weak classifiers (α_i) are determined by the classification error of weak classifier.

where $f(x)$ is a feature descriptor (HOG features in our case), θ represents a threshold value, and p represents the direction of the inequality. We illustrate AdaBoost algorithm graphically in **Fig. 5.3** and summarize the essential steps in **Algorithm 1**.

Algorithm 1 AdaBoost Algorithm

Step 1: Denoted as (x_i, y_i), training samples consist of a features vector x_i and training label $y_i = \{+1, -1\}$.

Step 2: Given P positive and N negative training data, set the weights of positive and negative samples to $w_i = \frac{1}{P}$ and $w_i = \frac{1}{N}$ respectively.

For $t = 1, \ldots, T$:

Step 3: Normalize the weights of all training samples, $w_i \leftarrow \frac{w_i}{\sum_{i=1}^{k} w_i}$.

Step 4: By using **Eq.** (5.1) with $p = 1$ or $p = -1$ and different θ, compute the error rate ϵ_t of each decision stump $h_t(x)$.

Step 5: Select the decision stump $h_t(x)$ that has lowest error rate.

Step 6: Compute weight of selected decision stump, $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.

Step 7: Update training sample weights, $w_i \leftarrow w_i \cdot \exp(-\alpha_t y_i h_t(x_i))$.

Step 8: The final strong classifier is

$$H_t(x) = \begin{cases} +1, & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \text{ ,} \\ -1, & \text{otherwise .} \end{cases}$$

To further speedup the human detection process, we use a cascade of AdaBoost classifiers that was first introduced by Viola & Jones [206]. The main idea of the cascade AdaBoost classifier is to design a series of AdaBoost classifiers, where the initial AdaBoost classifiers are designed to be small and aim to eliminate a large number of negative examples with very

Algorithm 2 Cascade Algorithm

Step 1: Select a minimum acceptable detection rate, d_{target} and maximum acceptable false positive rate, f_{target} for each layer of AdaBoost classifier.

Step 2: Select an overall target of false positive rate, F_{target} and compute the minimum layers of AdaBoost classifiers, L, such that $f_{target}^{L} \leq F_{target}$) .

Step 3: Prepare a set of positive samples, P and negative samples, N.

For $l = 1, \ldots, L$:

Step 4: Set the number of weak learners, detection rate, and false positive rate in the current AdaBoost classifier to $n = 1$, $d = 0$, and $f = 1$.

while $f > f_{target}$

Step 5: Set $n = n + 1$ and use P and N to train a AdaBoost classifier with n weak learners with **Algorithm 1**.

Step 6: Evaluate the trained AdaBoost classifier with the validation set to update detection rate, d and false positive rate, f.

Step 7: Decrease the AdaBoost threshold until $d \geq d_{target}$.

Step 8: Evaluate the current cascaded detector on images without positive target and reset the negative samples set, N with all false detection.

little processing time. Essentially, the initial AdaBoost classifiers eliminate "easy" examples at the beginning stage and the AdaBoost classifiers at the later stage are only applied to the "hard" examples that pass through all the previous stage.

Our algorithm is summarized in **Algorithm 2**. We consider our training algorithm simpler to the original algorithm summarized by Viola & Jones [206]. The main difference is that we evaluate the AdaBoost classifier at the current layer. Using the whole cascaded classifier to evaluate the validation set is not compulsory since we use the same positive samples set and the same negative images to scan false positive samples. In details, we select a minimum acceptable detection rate, (d_{target}) and maximum acceptable false positive rate, (f_{target}) for each layer of AdaBoost classifier. Then we select an overall target of false positive rate, F_target and compute the minimum layers of AdaBoost classifiers such that the multiplication of false positive rates f of all layers is less than or equal to the overall target of false positive rate, F_target. With a set of positive samples and negative samples, we start to train one AdaBoost classifier with only one weak learner with **Algorithm 1**. We then evaluate the trained AdaBoost classifier at the current layer with the validation set and decrease the threshold until we have a detection rate that is higher than the minimum

acceptable detection rate. With a lower threshold, the false positive rate is likely to be higher. If the false positive rate is higher than the maximum acceptable false positive rate, we will continue to increase the number of weak learners in the AdaBoost classifier at the current layer. On the other hand, if conditions of both the local detection rate and false positive rate are met, we use the while current cascaded detector to scan negative image set. Since there is no positive samples in the negative image set, all the detections will considered as false positives and will be used the new negative samples to train the subsequent AdaBoost classifier.

5.4 Human Orientation Estimation

This section gives an overview of our human body orientation (HBO) estimation method, which is summarized in the second half of **Fig. 5.2**. Our HBO estimation method has two points that make it work extremely fast. First, it uses the same HOG features that were used for human upper-body detection. This enable us to avoid additional features extraction process and perform HBO estimation at minimal computational cost. Second, we use random forest regression for HBO estimation. Since random forest regression only requires simple comparison operations at each branch node, our HBO estimation method works extremely fast.

In addition to the speed performance, our HBO estimation method also produces accurate results. Instead of applying HBO estimation onto only one detected human upper-body image patch (which is common in many detection framework, where non-maxima suppression method is used to suppress multiple detections near to the target), we apply HBO estimation onto all detected human upper-body image patches right before the non-maxima suppression step and take their mean as the final result. It is interesting to note that this process is reminiscent of a particle filtering process, but our method requires no likelihood computation, in which the likelihood function could be hard to design in certain case. While the covariance of the training samples inside the leaf node of a regression tree could be used as a weighting parameter like the likelihood value in a particle filter, we aim for simplicity and opt not to do so. Instead, we take the mean of all regression results produced by all human upper-body image patches and regression trees. In the following, we describe our design process of the random forest regression in detail.

5.4.1 Random Forest Regression

Random forest (shown in **Fig. 5.4**) is an ensemble of several regression trees that was proposed by Breiman [207]. We trained our random forest in a way that is slightly different from the original approach. During the training stage, we did not use the bagging process, where a subset is selected from the whole training dataset with replacement to train each regression tree. For the ease of program development, we trained a new regression tree with new data that we collected in a data collection session.

When training each regression tree, we use 10-fold cross validations in order to avoid overfitting. Specifically, we randomly divide our data into ten subsets and then use the first nine subsets to grow the tree branches. At each branch node, we examine all the HOG features and look for a decision stump rule that minimize the regression error, i.e., the mean squared error (MSE) at two child nodes. The growing of regression tree will stop if a branch node has less than ten samples or MSE could not be improved further.

In general, this training process will make the regression tree big and overfit the training data. To avoid this, we use the remaining subset to prune the tree. At every pair of child nodes with a common parent node, if the MSE of the parent node is smaller than the two combined MSE of the child nodes, we remove the two child nodes and the same process continues. After that, we use another nine subsets to re-grow the tree and use the remaining subset to re-prune the tree. This cross validation process would continue until the MSE of the tree does not improve. With this technique, the regression tree would have a higher training error but tend to be less susceptible to the overfitting issue.

During the prediction stage, a HOG vector enters the root node of each regression tree and continues to go down until it arrives at a leaf node (green circle in **Fig. 5.4**) by following the learned decision rules at branch nodes (blue circle in **Fig. 5.4**). We use the mean value of all the training samples inside the arrived leaf node as the regression tree result (HBO_1, ..., HBO_n). After that, we use the mean value of all regression trees as the random forest result.

[1] The regression trees figure is taken and modified from the tutorial notes of International Conference on Computer Vision (ICCV) 2009 by Jamie Shotton.

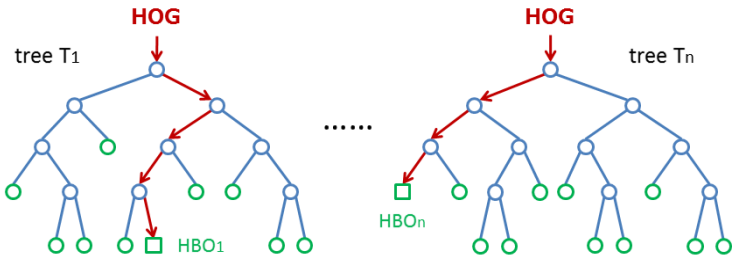

Figure 5.4: Random forest illustration.[1] Random forest is an ensemble of several regression trees ($T_1, \ldots, T_n$), where each tree is trained with a different dataset and has different branch nodes. During the prediction stage, a HOG vector enters the root node of each regression tree and continue to go down until it arrives a leaf node (green circle) by following the learned decision rules at each branch node (blue circle).

5.4.2 XY-Based Orientation Regression

While designing a regressor for HBO estimation seems straightforward, we find that the difficult point is that the HBO labels (-180°<θ<180°) are not continuous in the regression space while the HBO is obviously continuous in the physical space. It is not wise to train the random forest regressor directly with the original labels as it would lead to serious errors at the prediction stage. For example, imagine that we have a test data that is close to $+/-180°$ region. Since the images close to this region are visually similar, it is very likely that the regression trees would produce estimation results that are close to $+/-180°$ region. If we take the mean of the regression trees results, we end up in a value close 0° due to the plus and minus signs produced by different regression trees. One possible solution is to first transform the regression results from polar space into Cartesian space, compute the means in the Cartesian space, and then re-transform it back to the polar space. While we find that this process improves the HBO estimation result, in theory, we should regress human upper-body images that are close to the $+/-180°$ region to the same region in the regression space.

To this end, instead of estimating the body orientation using the original HBO labels (-180°<θ<180°) directly, we propose a xy-based random forest regression that could improve the estimation accuracy in both theory and the actual experiments. Specifically, we first transform the original HBO labels from polar space into Cartesian space, i.e., $\theta \rightarrow (x, y)$, and then train two random forest regressors—the first one for X dimension and the second one for Y dimension (hence the name xy-based). Since the xy-based labels are continuous in the

regression space, we hypothesize that this could improve the HBO estimation results significantly, especially for test images with HBO labels that are close to the 180° region. During the prediction stage, we perform two parallel regressions on the X and Y dimensions, compute their means, and then transform the mean result from the Cartesian space to the polar space. In the experiment section, we can observe that this method improves the HBO estimation significantly.

5.5 Data Collection Process

We have designed a system to collect high quality and large amounts of HBO datasets in order to train the random forest. A C++ program was developed to interface with one Xtion camera via USB and one IMU sensor wirelessly via Bluetooth. The Xtion camera provides both color and depth images in 640×480 resolutions at 30Hz. The IMU sensor has an internal gradient descent algorithm that combines accelerometer and gyroscope measurements and outputs highly accurate (up to 0.1° accuracy) 3D orientation information [208]. During the data collection session, we fixed the IMU sensor onto a rotatable chair that is placed in front of the Xtion camera at about one meter. Participants are asked to sit on the rotatable chair and face toward the Xtion camera initially (HBO is labeled as 0° when a participant is facing toward the Xtion). Then, the participant starts to continuously turn their body clockwise (or counter-clockwise) slowly while the program collects both color & depth images and IMU orientation outputs once per second. During the data collection process, participants can move their hands and turn their head slightly but generally did not have large hand and head movements. On average, participants would finish one full rotation within about three minutes. Overall, sixty datasets (roughly 200 samples on average) of two participants with very different body shapes, hair styles, and clothing have been collected at two different indoor environments with different backgrounds and illumination conditions. We visualize some of the example data in **Fig. 5.5**, where color and depth cropped images of two participants spanning from 180° to -180° are shown.

5.6 Experiment Results

In this section, we summarize our experiment results by using root-mean-square error (RMSE) between the ground truth HBO labels and the regression results as our evaluation

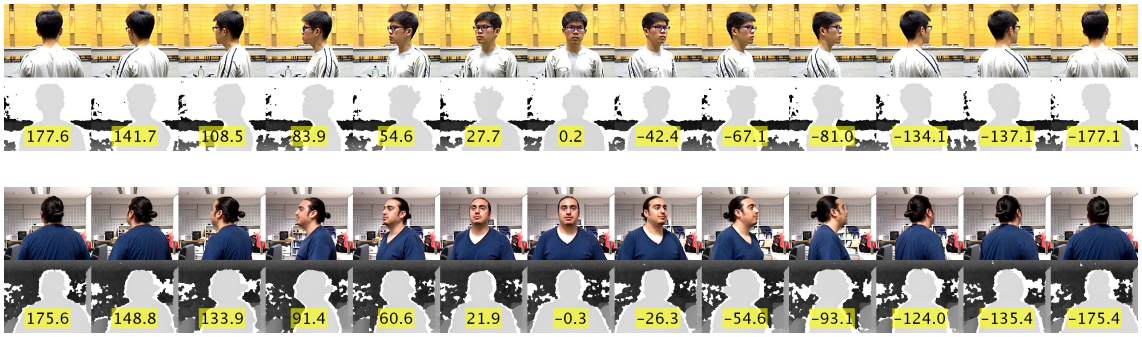

Figure 5.5: Human body orientation datasets of two people with color and depth images, where the human body orientations span from 180° to -180°. Note that the two participants have very different body shapes, hair styles, and clothing. In addition, the backgrounds and illumination conditions are different.

metric. Due to the space limitations, we focus on five comparison results. We first compare the (i) results of direct regression and xy-based regression by using depth images and analyze the effect of the number to regression trees. We then analyze (ii) the effect of non-maxima suppression and (iii) the effect of k-means filtering in the HBO estimation. After that, we compare the (iv) results of depth-based and color-based HBO estimation and present the (v) generalization performance of our method.

We train our depth-based human detector with data collected from the first person. In the first four experiments, we use forty datasets of the first person for training and use the remaining eight datasets of the first person for test. Note that in all datasets, the person turns his/her body orientation in one full cycle and hence all datasets contain sparse samples facing to all direction. While all forty-eight datasets have the same person, the person has different clothing, hand postures, and small head movements in each dataset. Furthermore, each dataset has different background and illumination conditions.

5.6.1 Direct vs. XY-Based HBO Estimation

Fig. 5.6 summarizes the RMSE results of eight test datasets of the first person, with the x-axis being the number of regression trees and y-axis being the RMSE values. From the figure, we can easily observe that the xy-based HBO estimation method performs better than the direct HBO estimation method by a large margin. Note that for the direct HBO estimation method, we have first transformed the regression results into Cartesian space

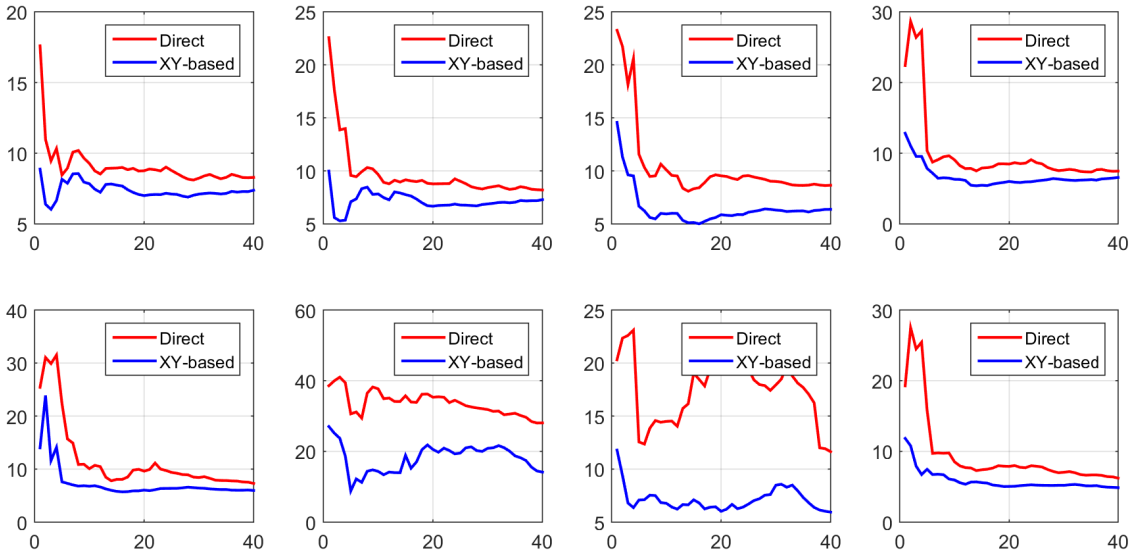

Figure 5.6: Direct vs. xy-based HBO estimation results of eight test datasets, where the x-axis represents the number of regression trees and y-axis represents the RMSE.

before computing their means (refer to **Sect. 5.4** for more details). If no transformation is used at all, we expect that direct HBO regression method would perform more poorly.

We can also observe that most regression results become stable when the number of regression trees is larger than twenty, except in the 6^{th} test dataset. Upon investigation, it is found that three images with orientation label close to 180° produce large regression errors of 100° (the three red arrows pointing to these test samples in **Fig. 5.7**). We believe that training the regression trees with more data and finer HBO labels could resolve this problem.

5.6.2 With vs. Without Non-Maxima Suppression

We also investigated the results with and without non-maxima suppression before applying the regression trees for HBO estimation. Without non-maxima suppression, we have more human upper-body samples for HBO estimation and we use the mean result as the final result. From **Fig. 5.8**, we can observe that without non-maxima suppression, the HBO estimation improves by a large margin. As mentioned before in **Sect. 5.4**, the process without the non-maxima suppression is reminiscent of a particle filtering process. However,

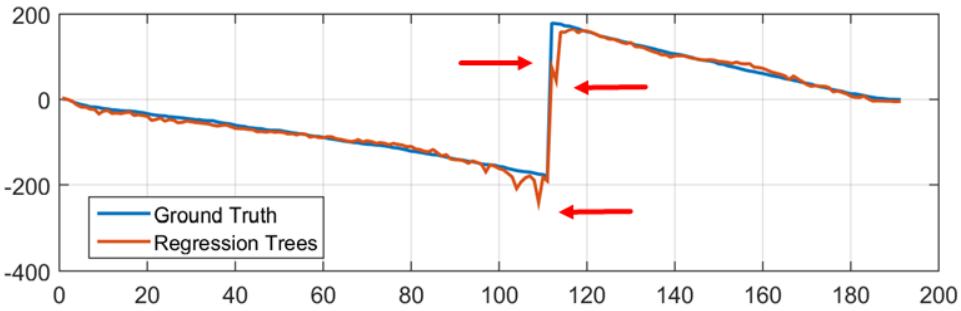

Figure 5.7: XY-based HBO estimation results of the 6th test dataset, where the x-axis represents the sample indices and the y-axis represents the HBO labels. The three red arrows point out the three images that have orientation labels close to 180° and produce regression error larger than 100°.

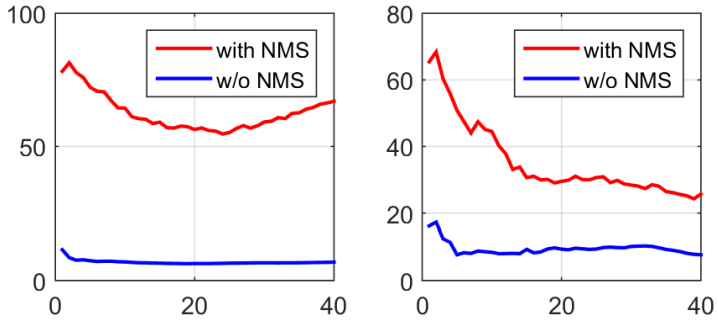

Figure 5.8: HBO estimation results with and without non-maxima suppression, where the x-axis represents the number of regression trees and the y-axis represents the RMSE. Note that the left/right figure summarizes the mean results of the first/last four test datasets.

our method requires no likelihood computation, in which the likelihood function could be hard to design in certain case.

5.6.3 Mean Computation vs. K-Means Clustering

During the early experiments, we found a few spiky HBO estimation results produced by some image patches and regression trees. While apply k-means clustering to the regression results would improve the result, overall we find that the effectiveness of k-means clustering starts to decrease when the number of regression trees increase. From **Fig. 5.9**, we can observe that both the simple mean computation and k-means clustering produce similar

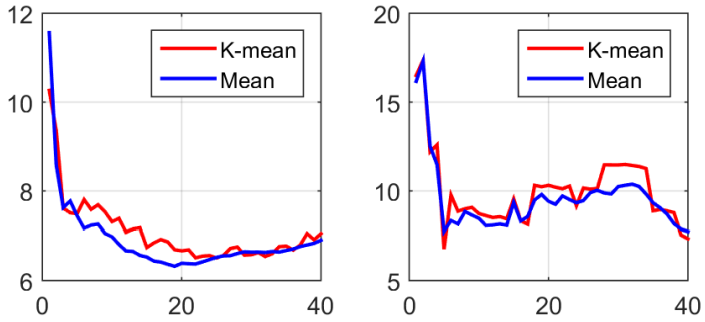

Figure 5.9: HBO estimation results with simple mean and k-means clustering, where the x-axis represents the number of regression trees and the y-axis represents the RMSE. Note that the left/right figure summarizes the mean results of the first/last four test datasets.

results. Since computing the mean is simpler and faster, we use the mean of all regression values as the final HBO estimation result.

5.6.4 Depth- vs. Color-Based HBO Estimation

In the following experiment, we study the performance of color-based HBO estimation. For comparison purposes, we use the same depth-based human upper-body detector. In **Fig. 5.10**, we can see that color-based HBO estimation does not work as well as the depth-based HBO estimation. This is reasonable as the collected (color-based) datasets are relatively small, and we have different backgrounds and illumination conditions. We believe that with more datasets, random forest could be trained to perform better.

5.6.5 Generalization Performance

We performed a simple experiment to test the generalization performance of our method and summarized the results in **Fig. 5.11**. We first applied the random forest model trained with datasets of the first person onto the test datasets of both people. As shown in the left figure of **Fig. 5.11**, the HBO estimation performance of the second person is not as good as the first person, but this is reasonable, considering that the two people have different body shapes, hairstyles, and clothing (**Fig. 5.5**).

We then train a different random forest model with datasets of the second person and test the performance with datasets of both people. Note that we only have eight training datasets

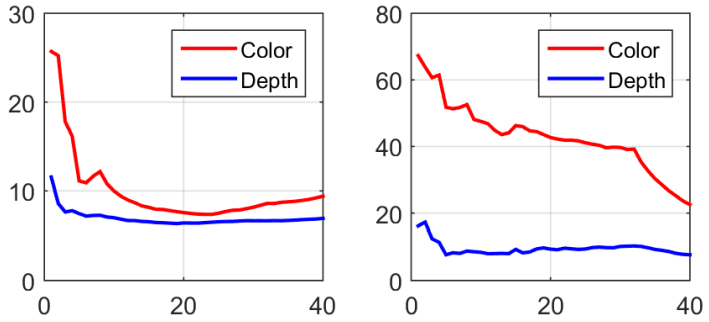

Figure 5.10: Depth- vs. color-based HBO estimation results, where the x-axis represents the number of regression trees and the y-axis represents the RMSE. Note that the left/right figure summarizes the mean results of the first/last four test datasets.

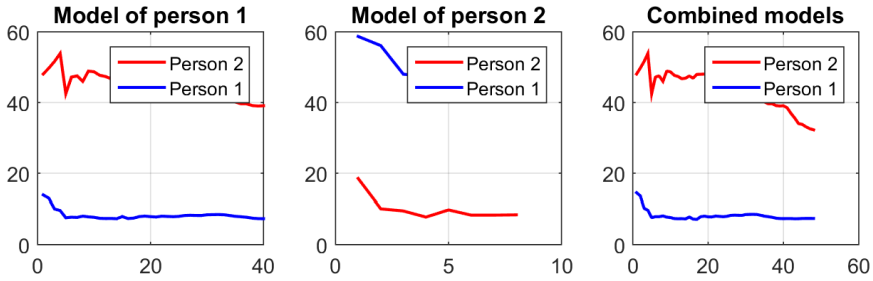

Figure 5.11: HBO estimation generalization results, where the x-axis represents the number of regression trees and the y-axis represents the RMSE. **Left:** Model (forty regression trees) was trained with the forty datasets of the first person. **Middle:** Model (eight regression trees) was trained with the eight datasets of the second person. **Right:** The first model and the second model are combined.

of the second person and we trained eight new regression trees with each training dataset. As shown in the middle figure of **Fig. 5.11**, the model works well on the second person but not as well on the first person due to same reason explained in the previous paragraph.

In the last experiment, we combined the forty regression trees trained with the forty datasets of the first person and the eight regression trees trained with the eight datasets of the second person. As shown in the right figure of **Fig. 5.11**, the test performance of the first person remains good while the test performance of the second person improves by 30% with the combined models. With more training datasets of the second person, we believe that the test performance of the second person could be further improved.

Table 5.1: Time performance in ms unit by using Xtion camera at 320 × 240 resolution with three different platforms. Note that both AdaBoost detector and random forest regressor are able to work in real-time (<5 ms).

	Depth Segmentation	Image Normalization	HOG Feature Extraction	AdaBoost Human Detection	Direct HBO Regression	XY-Based HBO Regression	Total
Platform [1]	0.372	0.014	0.098	0.001	0.229	0.505	1.219
Platform [2]	0.469	0.017	0.140	0.002	0.521	1.016	2.165
Platform [3]	1.059	0.027	0.223	0.002	0.711	1.311	3.333

[1] Intel Core i5-2540M @ 2.60GHz × 4 (64-bit), Ubuntu 14.04.4 LTS.
[2] Intel Core 2 Duo @ 2.66GHz × 2 (64-bit), Ubuntu 14.04.4 LTS.
[3] Samsung Exynos5422 Cortex-A15 2Ghz, Ubuntu 14.04.4 LTS.

5.7 Real-Time Performance

We also analyzed the time performance by using Xtion camera with three different platforms and summarized the results in **Table 5.1**. From the table, we can observe that both AdaBoost detector and random forest regressor work very fast in every depth image (<5 ms). Combined with depth segmentation technique, we compute the scanning window size based on the real depth values; we could achieve human upper-body detection and HBO estimation in real-time in an efficient manner.

5.8 Discussion

While the depth-based HBO estimation works well, combining face detection or head pose estimation with HBO estimation could potentially improve the results further. Color-based HBO estimation is also challenging and is an important future work. On the other hand, we believe that training the regression trees with the whole datasets and bagging process could improve the accuracy and generalization performance.

5.9 Conclusion

In this chapter, we present a full human upper-body detection and body orientation estimation framework. The proposed framework works efficiently since the same extracted features for both human body detection and orientation estimation. We also present a novel xy-based random forest regressor to perform a continuous and full 360° HBO esti-

mation. Compared to previous approaches, our method could produce accurate and smooth HBO estimation without additional tracking process. Moreover, we apply the xy-based random forest regressor onto all detected human upper-body image patches right before the non-maxima suppression step and take their mean as the final result. This process serves as a robust filter (like a particle filter) and our experiment results show that using this technique with the xy-based regression further improves the HBO estimation results.

Chapter 6
Human Sensing Interface II: Hand Shape Detection

6.1 Introduction

Hand shape detection is an interesting and challenging application [209] (see **Fig. 6.1** and **Fig. 6.2**). In addition to hand tracking, hand pose estimation, and hand shape recognition, hand detection has also been utilized for human-robot interaction (HRI) and ubiquitous computing applications. Previous works rely on skin detection [210–215] and depth technique [216–218] for hand detection. These systems do not work reliably under some circumstances. For instance, a hand detection system with Caucasian skin model would have difficulty to detect an African hand due to the difference of skin colors. In addition, a hand detection system that relies on depth thresholding technique would fail when the hand is not the closest object in front of the camera.

In this chapter, we focus on an appearance-based approach [219–221], where we first extract features from test images and perform classification with AdaBoost. To the best of our knowledge, appearance-based approaches are not the mainstream research for hand detection. We speculate that this is caused by the usefulness of image processing approaches (more details in **Sect. 6.2**) in many human-computer interaction (HCI) applications. In most HCI applications, image processing approaches such as skin detection and depth thresholding work well since the background and illumination conditions are normally fixed. The robustness of these approaches would decrease significantly in many HRI applications, where the robots are expected to move and subjected to very challenging backgrounds.

The rest of this chapter is organized as follows: In **Sect. 6.2**, we summarize general approaches that have been adopted for hand detection. In **Sect. 6.3**, we review the binary robust independent elementary features (BRIEF) extraction method, followed by our generalized extraction method (G-BRIEF) in **Sect. 6.4**. In **Sect. 6.6**, we provide extensive experiment results. We discuss some insights of our feature extraction method in **Sect. 6.7** and conclude our work with suggested future work in **Sect. 6.8**.

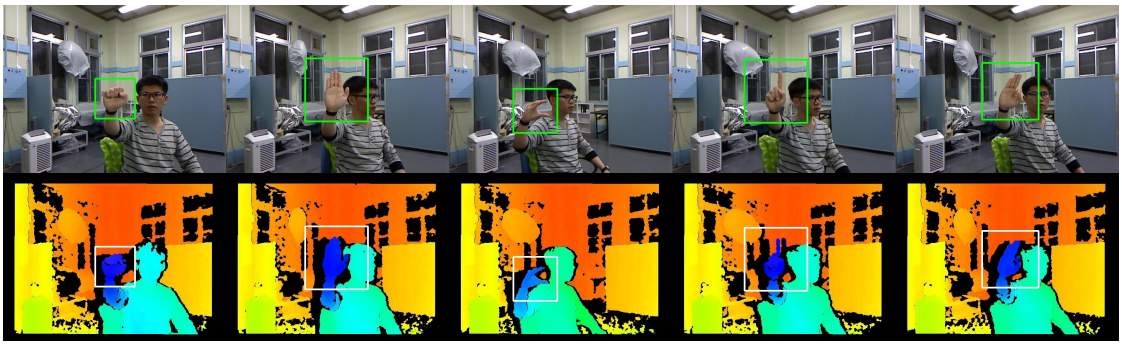

Figure 6.1: Five out of fourteen collected RGB-D hand shape images. Top and bottom rows show the full color and depth images with boxes overlaid on the hand shape regions respectively.

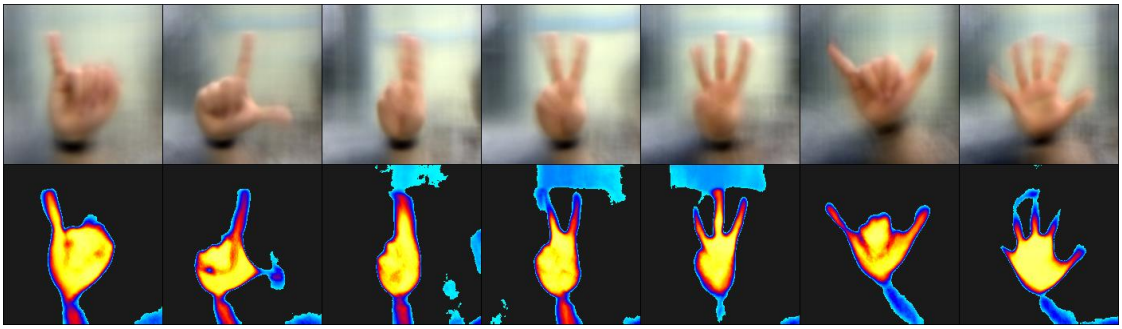

Figure 6.2: Seven out of fourteen collected RGB-D hand shape images. Top and bottom rows show the mean color and depth images of seven cropped hand shapes respectively.

6.2 Related Works

There are many previous works that are related to hand shape detection. For the ease of review purposes, we divide related works into three categories: image processing, appearance-based and filtering-based[1] approaches.

6.2.1 Image Processing Approaches

Skin detection methods have been widely adopted for hand detection in the past few years [210–215]. In a nutshell, skin and non-skin color pixels are represented with a Gaussian mixture model [210] or a Bayesian model [211–214], followed by skin detection with a nearest neighbor classifier or a Bayesian classifier. Using skin detection for hand detection is simple and fast but is susceptible to many practical issues, i.e. face or objects with skin-liked color would trigger false positive; different color models are needed for different skin colors; and etc.

On the other hand, recent advances in 3D sensor technology have promoted depth thresholding in hand detection [216–218]. A hand is normally assumed to be the closest object to the sensor in this method. With the detected hand tip point, the hand shape is segmented by thresholding nearby pixels that are within 15~20 cm. Similar to the skin detection methods, depth thresholding method is simple but has limitation, i.e. more than one detection is difficult; placing a hand behind some objects would cause a false negative result; and etc.

6.2.2 Appearance-based Approaches

Mittal et al. proposed to use histogram of oriented gradients (HOG) features for hand detection. After that, they combined skin detection and super-pixel segmentation to boost the hand detection performance. Their method achieves good detection rate but requires long computational time, which makes it unsuitable for real-time applications. In contrast, our method is faster and more efficient.

Ong & Bowden and Kölsch & Turk used Haar features for hand detection and achieved

[1] Strictly speaking, filtering-based approaches focus on hand tracking rather than hand detection. They normally start with hand detection methods and rely on temporal information to perform tracking. They are included here for comprehensiveness of the review.

good results [220, 221]. However, almost all train/test data used by Ong & Bowden in their experiments were under simple/plain backgrounds [220]. A hand detection system trained with this dataset is most likely to fail to detect hands on cluttered backgrounds. The experiment dataset used by Ong & Bowden has more varied light condition but the number of images is very small (with only six hand shapes, roughly 400 images/shape) [221]. Furthermore, their final hand detector focuses on one particular hand shape instead of general hand shapes.

Shotton et al. combined a background subtraction method, a simple depth-based feature, and a deep random forest classifier to determine locations of body parts [192]. Compared to their approach, our method for hand detection is more specific and is applicable for both color and depth images. In this chapter, we focus on hand detection with color images in the beginning, and then extend it to depth images.

6.2.3 Filtering Approaches

Oikonomidis et al. have proposed an impressive hand shape model for continuous hand tracking [211, 212]. Various filtering methods (such as Kalman filters, particle filters, and mean-shift filters) have also been proposed for hand tracking [210, 213, 218, 222]. Nevertheless, all these methods rely on image processing approaches (either skin detection or depth thresholding) mentioned in **Sect. 6.2** for initial hand detection. Therefore, these methods would not work reliably when certain assumptions are violated. In this work, we propose to use the appearance-based approach in order to overcome these limitations.

6.3 Original BRIEF Extraction Method

BRIEF extraction method was first proposed for image matching Calonder et al. [223]. Given an image, it randomly selects a few pixel pairs for binary test and then summarizes the results in binary coding fashion. These processes are straightforward and can be mathematically written as:

$$f_n = \begin{cases} 1, & \text{if } (I_a - I_b) \geq 0. \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

where n represents index of pixel pairs, I_a and I_b are the intensity values of red cross and blue circle pixels in **Fig. 6.3** respectively. After binary test, the outputs of *1*s and *0*s are

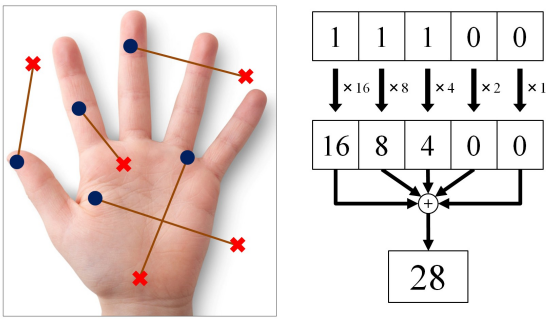

Figure 6.3: BRIEF randomly select five pixel pairs, perform binary test, and output a single value.

summarized in binary code and subsequently converted into a decimal value, which is mathematically written as:

$$F = \sum_{n=1}^{N} f_n \cdot 2^{n-1} \tag{6.2}$$

where N represents number of pixel pairs ($N = 5$ in **Fig. 6.3**).

BRIEF has been achieving remarkable results despite its simplicity and randomness during the pixel sampling process. However, we observe that the binary test of the pixel pairs in **Eq.** 6.1 may lose some important information during the extraction process, i.e. it only considers *which pixel has a larger value* but ignores *the value difference between pixel pairs*. This is especially true when the sampled pixel pair has large difference. In this work, we propose to make use of this information to improve the detection performance.

6.4 Generalized BRIEF Extraction Method

As mentioned in **Sect. 6.3**, we aim to include the value difference between pixel pairs into the BRIEF descriptors. On the other hand, only considering the pixels' difference might reduce BRIEF's robustness against image noise. In order to maintain BRIEF's simplicity and consider pixels' difference at the same time, we randomly assign Gaussian weights for pixels' difference. Mathematically, we propose to extract feature with the following equation:

$$f_n = w_a I_a + w_b I_b \tag{6.3}$$

Table 6.1: Differences between O-BRIEF and G-BRIEFs feature extraction methods.

	Gaussian Weights	Pixels' Difference	Binary Coding
O-BRIEF	No	Binary (1,0)	Yes
G-BRIEF1	No	Variable	Yes
G-BRIEF2	Yes	Variable	Yes
G-BRIEF3	No	Variable	No
G-BRIEF4	Yes	Variable	No

where I_a and I_b are the intensity values of red circle and blue circle pixels in **Fig. 6.3** (similar to BRIEF, red and blue circles are randomly sampled from the image), while w_a and w_b are the random weights taken from normal distribution with the zero mean and the standard deviation of one. After the sampling process, we produce the feature value either in a binary coding fashion, i.e. **Eq.** 6.2 or in a basic summation fashion:

$$F = \sum_{n=1}^{N} f_n \tag{6.4}$$

where N represents the number of pixel pairs.

We view our extraction method as a generalized BRIEF and hence name it as G-BRIEF (to make the contrast, we use the term O-BRIEF for original BRIEF). G-BRIEF maintains the simplicity of O-BRIEF and has minimal increase of computational cost but performs better. We summarize variants of G-BRIEF in **Table 6.1**. Note that Gaussian weights, pixels' difference, and binary coding have certain specific purposes. Firstly, Gaussian weights ensure robustness against image noise and enlarge the feature space. Secondly, variable pixels' difference captures more information than conventional binary pixels' difference. Thirdly, binary coding carries the sequence information of pixel pairs. It is also interesting to note that G-BRIEF4 is a sparse random hand feature, similar to full random face feature in [224]. Among all BRIEF features, we hypothesize that G-BRIEF2 will be the best feature in the ideal case since it models more useful information. As we will soon discover from the experiment result, G-BRIEF2 outperforms other BRIEFs significantly and perform comparably well with the Haar and HOG descriptors.

6.5 Experiment Settings

In this section, we summarize the experiment details, including the hand shape dataset and AdaBoost classifier used in our experiments. We carried out repeated random sub-sampling validation in all our experiments. Specifically, we randomly selected 44% of all images for training and used the remaining 56% images for testing. These percentages are the results of three considerations: (1) making the numbers of positive and negative training samples the same, (2) making the numbers of different hand shape images used in the training process be the same, and (3) making the sizes of training and test datasets as close as possible. We repeated this process for five times and report the average test accuracies and ROC curves in the following.

6.5.1 RGB-D Hand Shape Dataset

Instead of using small hand shape datasets in existing works, we built a high-quality RGB-D hand shape database [209] to test our algorithm. The hand images were collected from a calibrated Kinect sensor. In total, we have collected 70,000 RGB-D images for fourteen specific hand shapes (adapted from American Sign Language) from five participants. Participants were requested to move their right hand about 1.5 meters in front of Kinect cameras. In order to increase variability of the dataset, we encouraged participants to include different motions such as translation and orientation of hand in 3D space, as well as body motions during the image collection process.

In the beginning, we use color images in our experiments. To achieve robust classification, we purposely extract non-hand image patches near to the hand image patches (hard negative samples) from the original image (**Fig. 6.1**). In order to save computational cost, we use 14,840 color hand images and 7,840 color non-hand images for our experiment. All images are normalized to size of 64×64 pixels. Note that the number of images used in our experiment is at least four times larger than previous appearance-based works [219–221]. Furthermore, our dataset have very challenging backgrounds and can be observed in **Fig. 6.4**.

Figure 6.4: Examples of positive (top two rows) and negative (bottom two rows) training/test images used in our experiments. Note that this dataset has challenging background behind different hand shapes.

6.5.2 AdaBoost Classifier

AdaBoost combines several weak learners to form a strong classifier [204]. In our case, we use decision stump as our weak learners $h(f, p, \theta)$:

$$h(f, p, \theta) = \begin{cases} 1, & \text{if } p \cdot f < p \cdot \theta. \\ -1, & \text{otherwise.} \end{cases} \tag{6.5}$$

The decision stump consists of the extracted features f, i.e. O-BRIEF and G-BRIEFs in our case, a threshold θ, and a polarity value p representing the direction of the inequality. During the learning step, AdaBoost iteratively select single best training features for each decision stump. After feature selection, sample weights are re-adjusted according to local classification errors. The same process repeats until a certain number of features is successfully selected (empirically set to 100 in all our experiments).

6.6 Experiment Results

We compare test accuracies and ROC curves of G-BRIEFs with existing feature descriptors in this section. We also examine the design parameters of G-BRIEFs and analyze the

effect of Gaussian noise or occlusion to various feature descriptors. In addition to the color datasets, we also report the experiment results with depth images.

6.6.1 Comparison to the Existing Descriptors

We compare variants of G-BRIEFs with the O-BRIEF, Haar, and HOG descriptors. In this experiment, we set all BRIEF to have 5 pixel pairs and 10,000 features. For Haar, we use the basic five types of filters in [206]. We also set the filter's scaling ratio to 4 and shifting pixels to 8. Eventually, we obtain 10,368 Haar features and hence having a fairer comparison test. For HOG, we consider two cases with cell size of 8×8 and 4×4 pixels. Eventually, we obtain 9,864 features and hence having a fairer comparison test.

G-BRIEFs and O-BRIEF. From test accuracies and ROC curves in **Fig. 6.5** and **Fig. 6.6** respectively, we can observe that G-BRIEFs with Gaussian weights perform significantly better than counter alternatives (G-BRIEF2>G-BRIEF1 and G-BRIEF4>G-BRIEF3). Furthermore, we can observe that G-BRIEFs with binary coding produces better test accuracies than their counter alternatives (G-BRIEF1>G-BRIEF3 and G-BRIEF2>G-BRIEF4). Overall, G-BRIEF2 that considers Gaussian weight, variable pixels' difference, and binary coding perform significantly better than O-BRIEF.

G-BRIEFs and Haar and HOG. Under idea case, we find that the Haar and HOG descriptors perform slightly better than G-BRIEF2 (**Fig. 6.5** and **Fig. 6.6**). However, G-BRIEF has much lower computational cost and very simple design procedures. During real-time implementation, neither image normalization nor computation of integral image is required.

6.6.2 Design Parameters Analysis

We examine the design parameters of G-BRIEFs by focusing on G-BRIEF2 here. **Figure 6.7** shows the test accuracies of AdaBoost with different number of weak learners (decision stumps) in the AdaBoost classifier. We observe that the effect of number of pixel pairs is insignificant. In practice, 1 pixel pair is the best option since it has the lowest computational cost.

Figure 6.8 shows the test accuracies of AdaBoost with different number of G-BRIEF2 features, which indicates that using more features could lead to better detection accuracy. In all other experiments, we use 10,000 BRIEF features for a good balance of detection

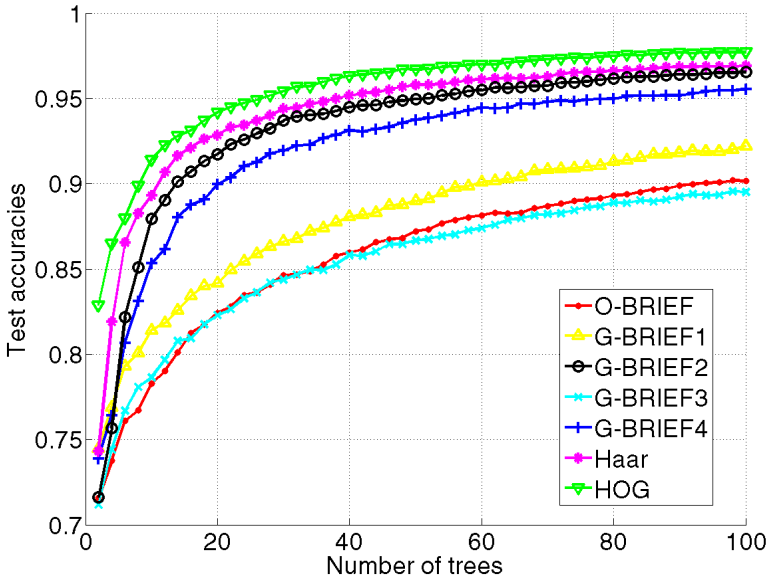

Figure 6.5: Test accuracies of O-BRIEF and G-BRIEFs with varied number of weak learners in AdaBoost. (Best viewed in color.)

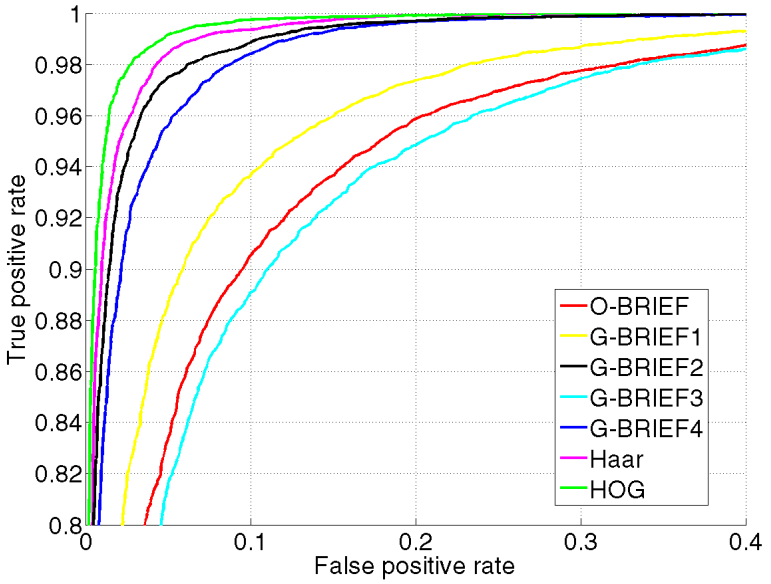

Figure 6.6: ROC curves of O-BRIEF, G-BRIEFs, Haar, and HOG under original test dataset. (Best viewed in color.)

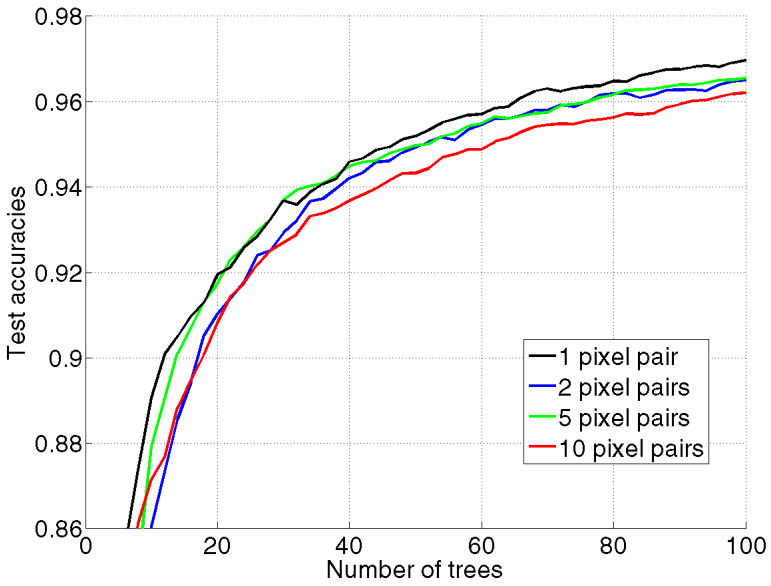

Figure 6.7: Test accuracies G-BRIEF2 with varied number of pixel pairs. (Best viewed in color.)

accuracy and computational cost.

6.6.3 Noise and Occlusion Considerations

Figure 6.9 shows ROC curves of the G-BRIEF4, Haar, and HOG descriptors with test dataset under synthetic Gaussian noise. We set the signal-to-noise ratio to 5 dB. We observe that despite its simplicity, G-BRIEF4 (without binary coding) can perform comparably well with the Haar and HOG descriptors. On the other hand, **Fig. 6.10** shows ROC curves of the G-BRIEF4, Haar, and HOG descriptors with test dataset under 25% synthetic occlusion, that is, a 32×32 random matrix is randomly placed in the test images. We observe that G-BRIEF4 performs better than the Haar descriptor, indicating that G-BRIEF4 is more robust to occlusion in the test images.

Despite of the best test accuracies, HOG has much higher computational cost than G-BRIEFs. Moreover, G-BRIEFs have speed advantage (low computational cost) over HOG and Haar since G-BRIEFs do not require image normalization and integral map during real-time implementation. This advantage is especially useful for devices with low computational

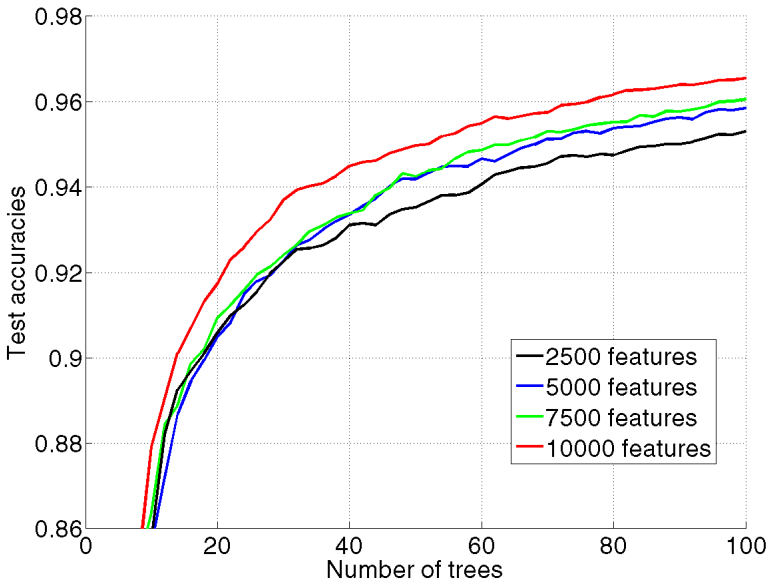

Figure 6.8: Test accuracies of G-BRIEF2 with varied number of features. (Best viewed in color.)

capability such as tablets or smart-phones.

6.6.4 Color and Depth Modalities

G-BRIEF can be extended directly to the depth images for hand shape detection. **Figure 6.11** shows the ROC curves of G-BRIEF2 with color and depth images for hand shape detection, which reveals that depth modality produces better test accuracies than color modality in hand shape detection task. This experiment outcome is reasonable; different from color images, depth images are not susceptible to color and illumination noises. Therefore, with the same number of training samples, using depth images for hand shape detection task can lead to a better test accuracy.

6.7 Discussion

In this section, we discuss the effect of the three terms in G-BRIEF. First, we show that capturing the variable pixels' difference is better than capturing the binary pixel (either 1 or

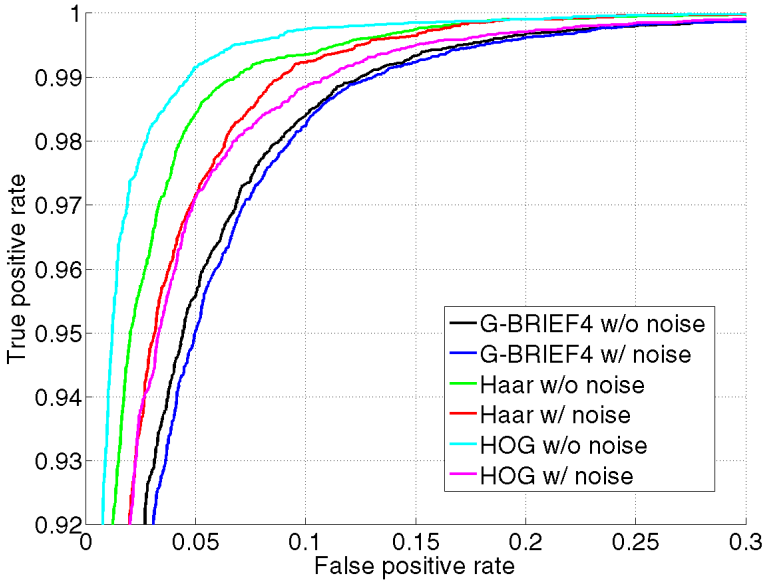

Figure 6.9: ROC curves of G-BRIEF4, Haar, and HOG with test dataset under synthetic Gaussian noise. (Best viewed in color.)

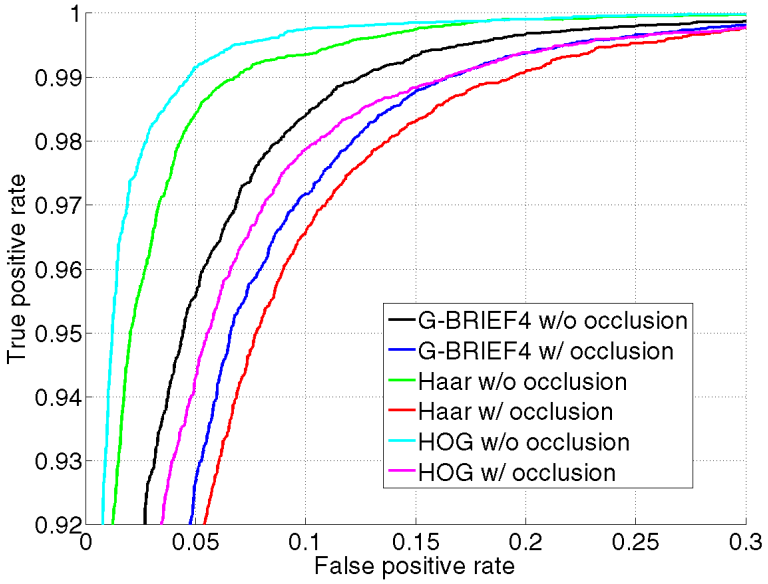

Figure 6.10: ROC curves of G-BRIEF4, Haar, and HOG with test dataset under synthetic occlusion. (Best viewed in color.)

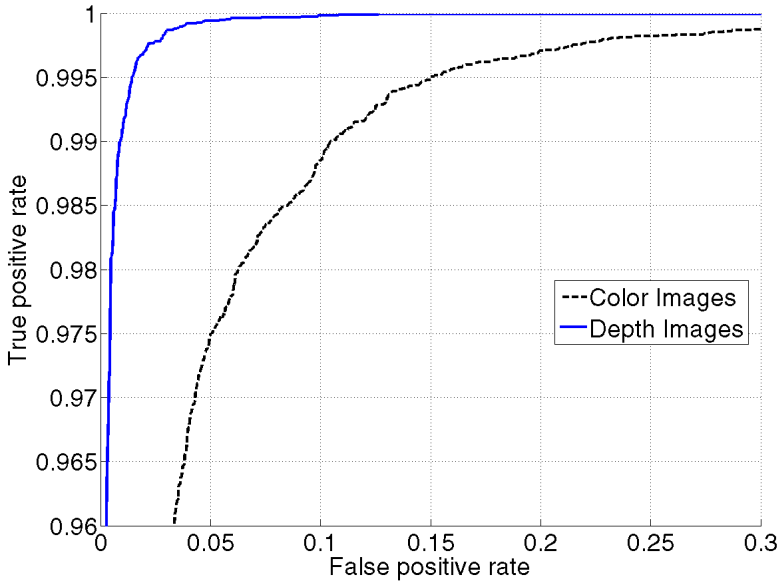

Figure 6.11: ROC curves of G-BRIEF2 with color and the corresponding depth datasets. (Best viewed in color.)

0). From **Fig. 6.5**, we can observe that G-BRIEF1 performs better than O-BRIEF. Second, Gaussian weights ensure robustness against image noise and enlarge the feature space. From **Fig. 6.5**, we can observe that G-BRIEF2 and G-BRIEF4 perform better than their counter alternatives of G-BRIEF1 and G-BRIEF3. Third, we find that binary coding is useful under ideal case but is susceptible to image noise. When train/test images contain large portion of noise, we find that G-BRIEF4 (without binary coding) performs better than its counter alternative of G-BRIEF2.

6.8 Conclusion

In this chapter, we focus on an appearance-based approach for hand detection. We proposed a new and simple feature extraction method that can be viewed as generalized BRIEF descriptor. Combined with AdaBoost, the new G-BRIEF descriptor is effective for hand detection and can be easily extended for other object detection or recognition tasks. We have compared G-BRIEF with existing feature descriptors with a large hand shape dataset that has challenging backgrounds. Experiment results show that G-BRIEF

performs comparably well with the Haar and HOG descriptors. When test images have partial occlusion, G-BRIEF outperforms the Haar descriptor. Moreover, G-BRIEF has very low computational cost since no image normalization or computation of integral image is required during runtime.

Chapter 7
Human Sensing Interface III: Facial Expression Recognition

7.1 Introduction

Facial expression recognition (FER) [225] has a great potential for improving our life quality. For instance, FER system is useful for medical applications, such as aiding patients with facial paralysis disease during rehabilitation treatment. FER system could also be used to analyze audience's facial expression for satisfaction survey. A robotic teacher could offer better learning experience by having a better understanding of students' feeling.

Designing a FER system is challenging due to the huge variability of face appearance, head pose, light condition, and partial occlusions due to hairs, sunglasses and masks. Over the years, researchers have proposed various techniques for robots/computers to recognize human facial expression. Previous works typically focus on proposing new feature descriptors and new classification methods for FER. In contrast, in this chapter, we aim to identify the best feature descriptors by performing an extensive comparison study. We empirically evaluate *five* popular feature descriptors (**Sect. 7.4**), namely Gabor [226], Haar [227], local binary pattern (LBP) [228], histogram of oriented gradients (HOG) [203], and binary robust independent elementary features (BRIEF) [229] descriptors. We then examine each feature descriptor by considering *six* classification methods (**Sect. 7.5**), such as k-nearest neighbors (k-NN), linear discriminant analysis (LDA), support vector machine (SVM), and adaptive boosting (AdaBoost) with *four* unique facial expression datasets. In addition to test accuracies (**Sect. 7.6**), we present confusion matrices of FER (**Sect. 7.8**). After that, we analyze the effect of combined features (**Sect. 7.9**) and image resolutions (**Sect. 7.10**) on FER performance. We also generalized our experiments to other datasets (**Sect. 7.11**), analyzed the computational efficiency of each feature descriptors (**Sect. 7.12**), and visualized the feature descriptors selected by AdaBoost classifier (**Sect. 7.13**).

7.2 Related Works

In this work, we focus on appearance approaches, in which we extract features from facial expression images by using *filters* or *transformations*, and apply classifiers to the

Table 7.1: Facial expression datasets, feature descriptors, and classification methods considered in our experiments.

Datasets	Descriptors	Classifiers
1. CK+	1. Gabor	1. k-NN
2. MUG	2. Haar	2. RFS + LDA
3. KDEF	3. LBP	3. PCA + LDA
4. JAFFE	4. HOG	4. SVM
	5. BRIEF	5. AdaBoost
		6. AdaBoost + SVM

extracted features. Our work is similar to a recent study [230] but we have more detailed analysis, including parameter sensitivity and image resolution analyses, generalization tests, and visualization of features.

On the other hand, geometric approaches use the locations of salient facial feature points such as eye corners, nose tip, and mouth corners for FER [231–234]. Geometric approaches normally require a high-resolution image for accurate localization of the salient facial feature points. Shan et al. has empirically shown that geometric approaches are more suitable for FER of high-resolution facial expression images [235].

Hybrid approaches also exist and independent works show that the combined geometric and appearance features can achieve higher test accuracies [236–238]. During our literature review process, we find that FER has a trend to combine the geometric and appearance features to achieve a more robust performance. Sadeghi et al. [239] first mapped the faces to a standard shape (i.e. geometric normalization) and then extracted appearance features for FER. Happy and Routray [240] extracted appearance features from salient facial patches (i.e. geometric features) to perform FER. Similarly, Zheng [241] proposed to perform FER by using sparse SIFT feature extracted from face feature points. Eleftheriadis et al. [242] also combined geometric and appearance features to learn a Gaussian process model for FER.

7.2.1 Feature Descriptors

Gabor descriptor [226] is one of the most common feature descriptors that has been used for FER [236, 237, 243–248]. Haar descriptor [227] is also popular for FER [249, 250]. Recently, LBP descriptor [228] has been adopted for FER [235, 239, 240, 242, 251] and HOG descriptor [203] has been examined for FER [252–254]. In this study, we also evaluate the performance a relatively new feature descriptor called BRIEF descriptor [229] in FER. To the best of our knowledge, we are the first to consider BRIEF descriptor in FER. Dense SIFT descriptor [255] has also been used for FER recently [256].

7.2.2 Classification Methods

SVM classifier [257, 258] is the most common classifier that has been applied to FER [235, 244–247, 249, 251–254]. Several works have reported that SVM with linear kernel produce similar test outcomes when compared to radial basis function kernel (RBF) [244–247]. These consistent outcomes indicates that feature descriptors such as Gabor and Haar filters are capable of transforming the original image data into a space with a higher linear separability between image classes.[1] On the other hand, AdaBoost [204] has also been used for FER [245–247, 250] and feature selection [235, 237, 247, 249]. Neural network approaches, including multilayer perceptron and radial basis function network, have also been explored in FER [236, 238, 248]. k-Nearest Neightbors (k-NN) classifier has also been used recently for FER [242]. Given sufficient training data, modern classifiers can normally achieve satisfactory test results. However, the best feature descriptor and classification method across different facial expression datasets remains unknown due to the lack of comparison study.

7.3 Facial Expression Datasets

Our experiments focus on four facial expression datasets, namely Extended Cohn-Kanade (CK+) Facial Expression Dataset [259, 260], Multimedia Understanding Group (MUG) Dataset [261], Karolinska Directed Emotional Faces (KDEF) Dataset [262], and Japanese Female Facial Expression (JAFFE) Dataset [263]. All faces in the four datasets were extracted with face detector of Computer Vision System Toolbox in MATLAB [264]. We use

[1] This process is conceptually similar to the kernel trick in SVM classifier.

the detected faces directly for FER without explicitly aligning facial feature points. Littlewort et al. have reported that explicit facial feature alignment does not improve the performance significantly [247].

7.3.1 CK+ Dataset

CK+ Database[2] [259, 260] is one of the most widely used facial expression databases. It has facial expression images of 210 adults aged between 18-50 years old. Participants were consisted of 69% female. All participants were requested to perform a series of facial displays with the help from an instructor. With careful selection criterion, 327 sequences were identified as one of the seven discrete facial expression, namely angry, contempt, disgust, fear, happy, sad, and surprise. Each sequence begins with a neutral facial expression and ends with a specific facial expression. For our comparison study, we excluded contempt class and focused on the Basic-6 facial expression [265]. We selected the first frame of all sequences as neutral images and used the last frame from all sequences as Basic-6 facial expression images. Overall, we have obtained 636 facial expression images (angry: 45, disgust: 59, fear: 25, happy: 69, sad: 28, surprise: 83, and neutral: 327). **Figure 7.1** shows a few example images of CK+ Dataset. The square boxes were obtained from face detector in MATLAB [264]. Red, orange, yellow, green, blue, indigo, and violet colored boxes represent angry, happy, fear, neutral, sad, surprise, and disgust facial expression respectively.

It is worth noting that CK+ Database has been used for FER under different setting. Shan et al. have used the first frame of all sequence as neutral images and used the last *three* frames of all sequences as Basic-6 facial expression images for FER [235], resulting in 1254 images (angry: 135, disgust: 177, fear: 75, happy: 207, sad: 84, surprise: 249, and neutral: 327). We have also tried this setting but found that it produces unfair comparison results. The last three extracted frames are almost identical to each other and subsequently cause significant overlapping in the training and test data during cross validation. Therefore, we opted to use only the last frame from all sequences as Basic-6 facial expression images.

[2] We use the term *database* to refer to the original resources offered by third party while the term *dataset* to refer to the images selected from the *database* for our experiments.

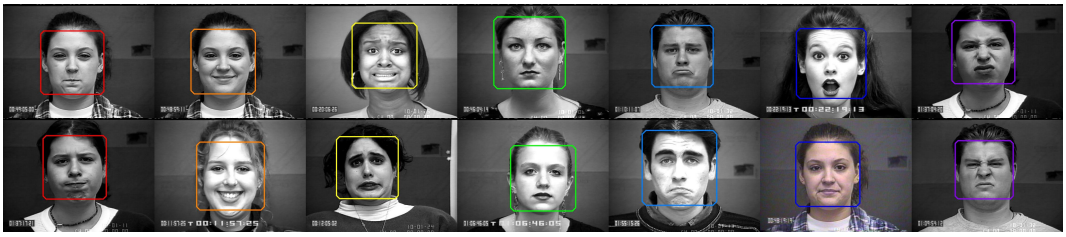

Figure 7.1: CK+ Facial Expression Dataset.

7.3.2 MUG Dataset

MUG Database [261] records facial expression images from 86 participants with Caucasian origin and aged between 20-35 years old. 59% of the participants were male. Some participants had beard or hair occlusions and 7 participants were wearing glasses. We have selected 919 facial expression images for our comparison study (angry: 157, disgust: 145, fear: 118, happy: 164, neutral: 52, sad: 124, and surprise: 159). Example images are not shown here due to license agreement term.

7.3.3 KDEF Dataset

KDEF Database [262] records facial expression images from 140 amateur actors (70 males and 70 females) at 5 different viewing angles. All actors aged between 20-30 years old. They have no beards, no mustaches, no earrings, no eyeglasses, and mostly no visible make-up during photo sessions. For our comparison study, we only consider frontal images, resulting in 980 facial expression images (angry: 140, disgust: 140, fear: 140, happy: 140, neutral: 140, sad: 140, and surprise: 140). **Figure 7.2** shows some example images from KDEF Dataset, where red, orange, yellow, green, blue, indigo, and violet boxes represent angry, happy, fear, neutral, sad, surprise, and disgust facial expression respectively.

7.3.4 JAFFE Dataset

JAFFE Database [263] contains 213 facial expression images of Basic-6 facial expressions and one neutral facial expression. All facial expression images were posed by ten Japanese female models. We use all 213 facial expression images for our comparison study (angry: 30, disgust: 29, fear: 32, happy: 31, neutral: 30, sad: 31, and surprise: 30). **Fig-**

Figure 7.2: KDEF Facial Expression Dataset.

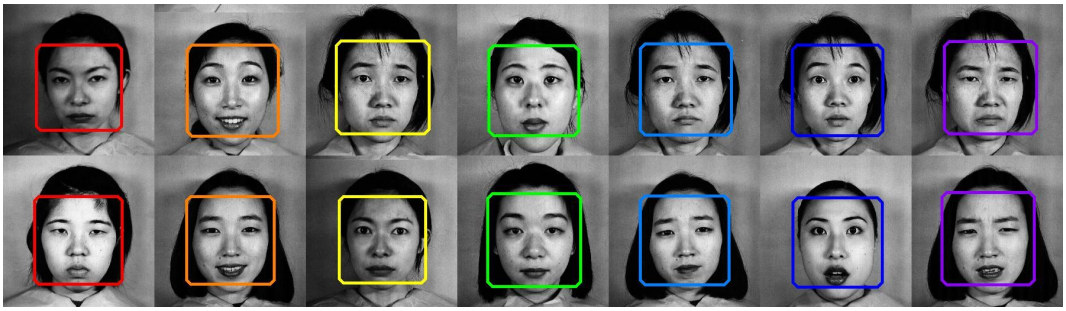

Figure 7.3: JAFFE Dataset.

ure 7.3 shows some example images from JAFFE Dataset, where red, orange, yellow, green, blue, indigo, and violet boxes represent angry, happy, fear, neutral, sad, surprise, and disgust facial expression respectively.

7.4 Feature Descriptors

Our main goal is to identify the best feature descriptor for FER. We empirically evaluate *five* feature descriptors, namely Gabor, Haar, LBP, HOG, and BRIEF descriptors in FER. In the following, we briefly review the computational process and advantages of each feature descriptor.

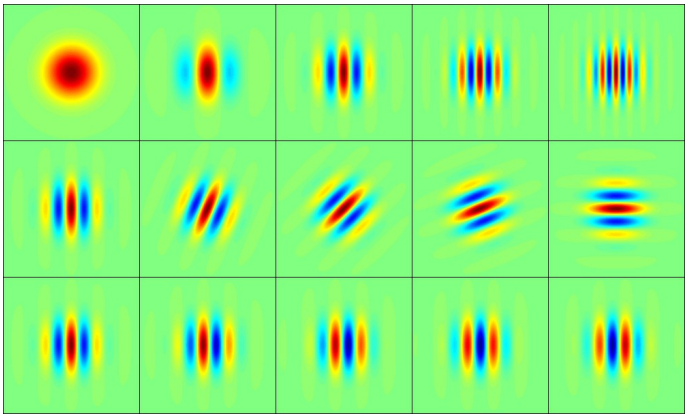

Figure 7.4: Gabor filter examples. First row shows filters with increasing frequency, second row shows filters with varying orientation, and third row shows filters with varying oscillation phase.

7.4.1 Gabor Descriptor

Gabor filter theory was first formulated by Dennis Gabor in 1946 [226]. John Daugman later discovered that Gabor functions can be used to model simple cells in the visual cortex of mammalian brains [266]. This discovery reveals that Gabor filters is similar to perception in human visual system and justifies the usefulness of Gabor filters in various computer vision applications such as iris recognition [267], fingerprint matching [268], and FER [245, 246]. We illustrate a few examples of 2D Gabor filters in **Fig. 7.4**, where the first row shows filters with increasing frequency, second row shows filters with varying orientation, and third row shows filters with varying oscillation phase. As shown in **Eq.** (7.1) and **Eq.** (7.2), Gabor filter has compact functions that relate filter size, oscillation frequency, orientation, and oscillation phase. From **Eq.** (7.1), we can observe that the Gabor function consists of two sub-functions—Gaussian and harmonic functions, where the Gaussian sub-function (a.k.a. envelope function) is responsible in defining spatial properties (x, y, σ, γ), while the harmonic function (a.k.a. carrier function) is responsible in governing oscillation frequency (λ), orientation (θ), and oscillation phase (φ) properties of the Gabor filters.

$$g(x, y, \sigma, \gamma, \theta, \lambda, \varphi) = exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \, cos(2\pi \frac{x'}{\lambda} + \varphi) \, , \tag{7.1}$$

$$\begin{aligned} x' &= \ (x - x_o)cos\theta + (y - y_o)sin\theta \ , \\ y' &= -(x - x_o)sin\theta + (y - y_o)cos\theta \ . \end{aligned} \tag{7.2}$$

In our experiment, we followed Bartlett et al. [245] and set the spatial resolution (x, y) in the range of 48×48 pixels, use 8 orientations (θ = 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°), and 5 oscillation frequencies ($\lambda = 4, 4\sqrt{2}, 8, 8\sqrt{2}$, 16 pixels per cycle). Ellipticity of the Gaussian function (γ) is set to 1 and oscillation phase (φ) is set to 0. Standard deviation of the Gaussian function (σ) is set according to half-response spatial-frequency bandwidths rule in **Eq.** (7.3), where the bandwidth (b) of the Gabor function is set to 0.5.

$$\frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{ln2}{2}} \times \frac{2^b + 1}{2^b - 1} \tag{7.3}$$

7.4.2 Haar Descriptor

Haar filter was first proposed by Papageorgiou et al. as a general framework for object recognition in 1998 [227]. Viola and Jones later popularized the Haar filter by showing its effectiveness along with integral image and cascaded classifiers in face detection problem [206]. Haar filter is a simple rectangular filter that represents the difference of sum of pixel intensities inside black and white regions. It has the key advantage of simplicity. Combined with integral image technique, Haar filter can achieve significant fast performance and make real-time face detection possible [206]. We followed Viola & Jones [206] and used the same five types of Haar filter in our experiments. **Figure 7.5** illustrates the five types of Haar filter, where each row shows type 1, 2, 3, 4, and 5 filters respectively with different sizes at different locations. We set the black and white regions to have the same size. We also increased the size of black and white regions with a step size of 2 pixels. As a result, we extracted 162,336 Haar values from a 48×48 image.

7.4.3 LBP Descriptor

Local binary pattern (LBP) descriptor proposed by Ojala et al. [228] is a powerful feature descriptor for image classification. Compared to Gabor or other image filters, LBP has the advantages of computational simplicity and robustness against illumination variations. LBP encodes information of local patterns such as edges, lines, and spots in each pixel. **Equa-**

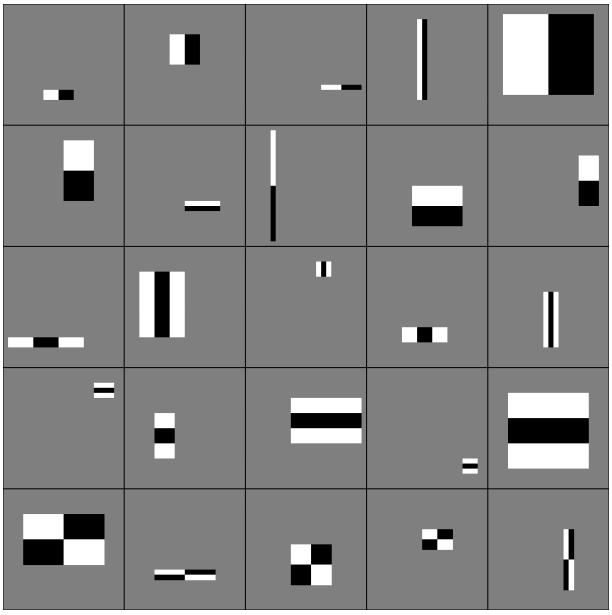

Figure 7.5: Haar filter examples. The rows show five types of Haar filter with varying width and height used in our experiments.

tion (7.4) and **Eq.** (7.5) summarize the encoding processes in a compact form while **Fig. 7.6** illustrates the encoding procedures for a $LBP_{P=8,R=2}$ operator in details. More precisely, LBP coding for every pixel is computed as follows:

1. Sample neighbor points, where P defines the number of neighboring points and R defines the radial distance between the center pixel to neighboring points.
2. Compute the difference of pixel values between center pixel and neighbor points.
3. Threshold the computed differences at zero.
4. Multiply the thresholded values with power of two consequently and sum all the values.

Note that the $LBP_{8,2}$ operator produces 2^P possible binary patterns and it has been shown that some binary patterns contain more information than others [228]. Ojala et al. named these binary patterns as uniform patterns, where they contain at most two bitwise transitions from 0 to 1 or vice versa (considering binary pattern in circular). For instance, 00000001,

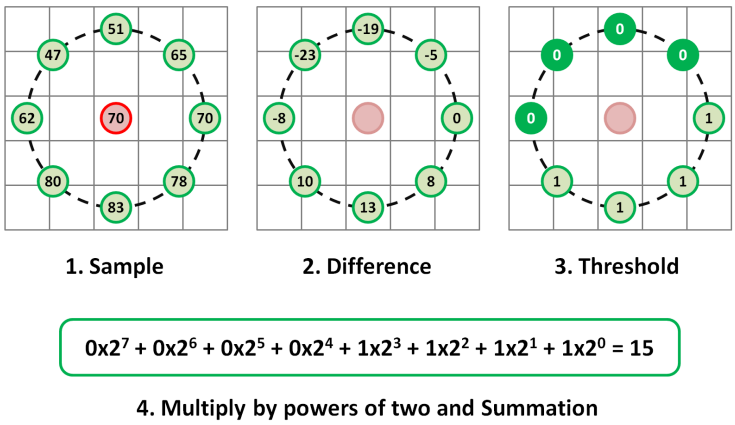

Figure 7.6: Coding procedures of $LBP_{P=8,R=2}$, where P stands for number of neighbors and R stands for radial distance between the center pixel and neighboring points.

11001111, and 11110011 are uniform patterns while 10100000, 00011101, and 11001100 are non-uniform patterns. Ojala et al. found that about 90% of all binary patterns are uniform and they proposed to accumulate all non-uniform patterns into single bin. Therefore, the original $2^8 = 256$ bins is reduced to 59 bins. We followed this practice in our experiments, similar to previous facial expression study [235] and face recognition study [269].

After LBP coding of every pixels, LBP values of pre-defined cells (**Fig. 7.7**) are stored in histogram form and eventually concatenated into a 1D feature vector. More precisely, we start with a window size of 12×12 pixels, compute the histogram of LBP values inside the window, and continue the process by shifting the current window to the right hand side by 3 pixels. If the right corner is reached, window will be shifted to the bottom side by 3 pixels and be restarted from the left hand side. Normalization of histograms is optional. In our case, we did not carry out normalization since all the considered windows are in the same size.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \tag{7.4}$$

$$s(x) = \begin{cases} 1, & if\ x \geq 0 \\ 0, & otherwise. \end{cases} \tag{7.5}$$

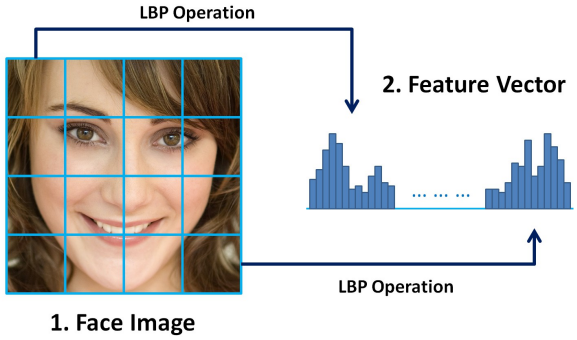

Figure 7.7: Face description with LBP descriptor. A face image [270] is first divided into cells and followed by LBP operation on each cell. Histogram of each cell is concatenated into a feature vector.

7.4.4 HOG Descriptor

Histogram of oriented gradients (HOG) feature was first described by Dalal et al. for human detection [203]. While being similar to scale-invariant feature transform (SIFT) descriptor [271], HOG represents dense coding of image and have some unique details such as using different number of histogram bins and different image block size [203]. HOG descriptor can be computed in the following five basic steps:

1. Gradient computation—image is convoluted with two Sobel filters, i.e. [-1,0,1] and $[-1,0,1]^T$, to form horizontal and vertical gradient maps. Following the common practice, smoothing and gamma normalization are omitted [203].

2. Magnitude and orientation computation—magnitude and orientation maps are computed based on the obtained horizontal and vertical gradient maps. Taking dx and dy as gradient value of pixel in the horizontal map and vertical maps, magnitude and orientation are calculated as: $Magnitude = \sqrt{(dx)^2 + (dy)^2}$ and $Orientation = tan^{-1}(\frac{dy}{dx})$.

3. Cell division—image is then divided into smaller cells. For example, we are using face images with size of 48×48 pixels in our experiments. These images are divided into 6×6 cells where each cell has size of 8×8 pixels (**Fig. 7.8**).

4. Cell quantization—the orientation values of each cell is quantized in histogram form with 9 orientation bins, where the magnitude represents voted weights, and we

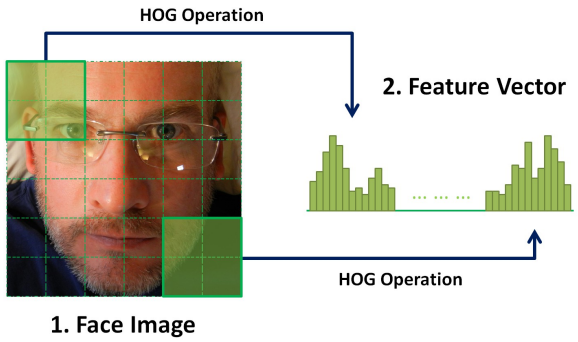

Figure 7.8: Face description with HOG descriptor. A face image [272] is first divided into cells and followed by quantization of gradient orientation on each cell. Histogram of each block of four adjacent cells is then locally normalized and concatenated into a full feature vector.

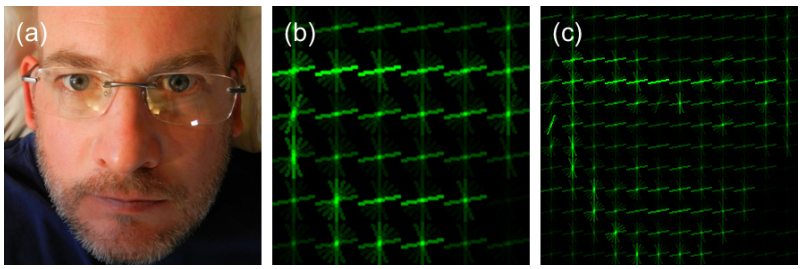

Figure 7.9: Visualization of HOG descriptors with divided cells. **(a)** An example face image [272]; **(b)** HOG visualization of 6×6 cells; **(c)** HOG visualization of 12×12 cells.

interpolate votes bi-linearly between neighboring bin center.

5. Block normalization—four adjacent cells form a block with 16×16 pixels (every block has 50% overlapping with the adjacent block). Orientation histograms of each block are locally normalized and concatenated into a feature vector.

Figure 7.9(a) shows an example face image in size of 48×48 pixels; **Fig. 7.9(b)** shows the corresponding HOG descriptors under 6×6 cells configuration (with cell size of 8×8 pixels); and **Fig. 7.9(c)** shows the corresponding HOG descriptors under 12×12 cells configuration (with cell size of 4×4 pixels). Overall, the standard HOG descriptor used in our experiments has a dimension of (5×5×4×9 + 11×11×4×9) = 5256.

7.4.5 BRIEF Descriptor

Binary robust independent elementary features (BRIEF) descriptor [246, 273] was first proposed for image matching with random forest [246] and random ferns classifiers [273]. BRIEF descriptor has the lowest computational cost among Gabor, Haar, LBP, and HOG descriptors because it only performs simple binary comparison test and uses Hamming distance (instead of Euclidean or Mahalanobis distance) for image classification [223].

BRIEF descriptor has two setting parameters – the number of binary pixel pairs and binary threshold. We used five binary pixel pairs and zero binary threshold in all our experiments. **Figure 7.10** illustrates computation process of a BRIEF descriptor. Firstly, five binary pixel pairs are randomly chosen from a given image. Secondly, binary pixel pairs are subjected to binary test,

$$f_n = \begin{cases} 1, & if(I_a - I_b) \geq 0 \\ 0, & otherwise. \end{cases} \tag{7.6}$$

where I_a and I_b are the intensity values of red circle pixel and blue circle pixel in **Fig. 7.10**, respectively. Thirdly, the binary values of five binary pixel pairs are summarized in binary code and subsequently converted into decimal value, which can be summarized in the following form,

$$F = \sum_{n=1}^{5} f_n \cdot 2^{n-1} \, . \tag{7.7}$$

In all our experiments, we use 10000 BRIEF values for FER.

7.5 Classification Methods

After feature extraction, we applied six different classifiers with the feature descriptors. We first considered k-NN classifier, followed by LDA classifier. Thirdly, we used PCA to reduce the size of feature descriptors and used LDA for FER. Fourthly, we used SVM and AdaBoost classifier for FER. Last but not least, we used AdaBoost for feature selection and SVM for FER.

We used two-class classifiers in all experiments. Specifically, we built seven one-vs.-all classifiers that are responsible in recognizing seven facial expressions, namely angry, disgust, *happy*, fear, neutral, sad, and surprise. We used common voting practice and combined seven

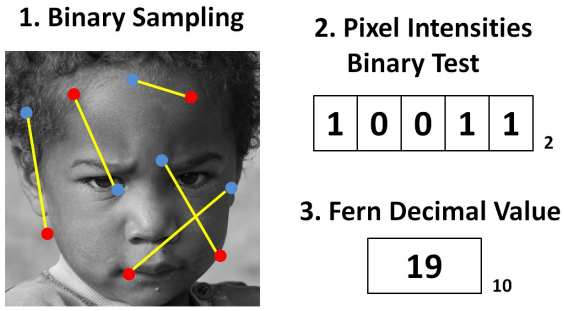

Figure 7.10: Face image [274] and its BRIEF descriptor. Five random pixel pairs are selected, followed by binary tests on pixel intensities, and conversion from binary code into decimal value.

test results in the end [235, 247]. For example, an image identified as positive by angry classifier will get +1 point for angry label and -1 point for other labels. Similarly, the same image identified as negative by neutral classifier will get -1 point for neutral label and +1 point for other labels. In order to avoid possible classification ties, each label is initiated with a random positive number that is smaller than 1. Label with the highest points after combining seven voting results will be elected as the final test label.

7.5.1 k-Nearest Neighbors (k-NN)

Given a test data in a feature descriptor space, 1-NN classifier tries to find a training data that is the closest to the test data, and consider the label of that training data as the test label. Similarly, given a test data in a feature descriptor space, k-NN classifier tries to find k training data that are the closest to the test data, and consider the label with the largest occurrence as the test label. In our experiments, we considered Euclidean distance and used exhaustive search. Denoting a particular training data as x_k and test data as x, the squared Euclidean distance between the training and test data can be computed as $d_k = (x_k - x)^T(x_k - x)$.

7.5.2 Linear Discriminant Analysis (LDA)

LDA assumes facial expression images to follow multivariate normal distribution model and all classes C have the same covariance matrix. During training process, LDA estimates

the means μ_c and covariance matrix Σ_c of every class c and tries to look for an optimal linear boundary between classes [275]. Given a test data $x \in \mathbb{R}^{D\times 1}$, the test label can be predicted by maximizing

$$\hat{c} = \arg\max_{c=1,\dots,C} \hat{P}(c|x) \,. \tag{7.8}$$

The posterior probability in **Eq.** (7.8) can be computed by using Bayesian rule

$$\hat{P}(c|x) = \frac{P(x|c)P(c)}{P(x)} \,, \tag{7.9}$$

where the $P(c)$ is class prior, the $P(x)$ is a normalization constant, and the likelihood can be computed by

$$P(x|c) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_c|}} \exp\Big(-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\Big) \,. \tag{7.10}$$

In practice, the number of our training data is smaller than the dimensions of feature descriptors. This condition would trigger singularity issue when we invert the covariance matrix during the training process. Moreover, the inversion of covariance matrix has large computational load due to the large feature descriptors (about 90,000 for Gabor descriptor and more than 160,000 for Haar descriptor). To maintain simplicity of LDA as well as comparison with PCA at later stage, we randomly selected two hundred features from each descriptors for FER.

While LDA is always used interchangeably with Fisher LDA (FLDA), we follow the naming convention in [275–277] to avoid confusion. We consider LDA as a *classification* method while FLDA as a *dimensionality reduction* method. Formally, LDA fits data with multivariate normal function with the assumption that each class shares a same covariance matrix. On the other hand, FLDA tries to maximize classes' separability by finding an optimal linear projection. Under this point of view, "LDA" mentioned in the previous facial expression studies [235, 247] are in fact FLDA method. For more details, please refer to Chapter 4.2.2 (LDA) and Chapter 8.6.3 (FLDA) in [275] or online resources [276, 277].

7.5.3 Principal Component Analysis (PCA)

In the previous section, we used random feature selection (RFS) to reduce the number of features in order to avoid singularity issue in LDA classifier and save computational

cost. In this section, we performed dimensionality reduction more systematically by using PCA [278]. In theory, PCA finds an orthogonal projection plane in a lower dimensional space that minimize the data projection error. From another point of view, PCA also maximizes the variances between data while projecting data to a lower dimensional space. Principal components can be computed by using eigen decomposition or singular value decomposition methods. In practice, the first few principal components normally capture most of the information in the original data. We used the first 200 principal components (retaining more than 98% of information) for LDA classification. In our experiments, we used power method of PCA [279] to reduce the computational load and speedup performance.

7.5.4 Support Vector Machine (SVM)

SVM was originally proposed by Vapnik and Lerner in 1963 [257] and the concept of soft margin in SVM was formulated by Cortes and Vapnik in 1995 [258]. Unlike LDA, SVM assume no prior knowledge about the distribution of the samples. As shown in **Fig. 7.11**, SVM looks for an optimal hyper-plane that maximizes the margin area between the two classes' closest training sample points (support vectors).

In practice, we have to decide a kernel function, such as linear, polynomial, or RBF for SVM [280, 281] and there is no analytical way to decide which kernel function works the best for any particular datasets. In this study, we focus on SVM with linear kernel (linear SVM) for two main reasons. Firstly, Littlewort et al. and Shan et al. have reported that SVM with radial basis function (RBF SVM) does not perform significantly better than linear SVM [235, 247]. These outcomes indicate that facial expression datasets are highly linear-separable upon feature transformation. Secondly, linear SVM has only one tuning parameters (soft margin) and has a lower computational cost than RBF SVM.

With linear SVM, we tuned the soft margin parameter by performing a grid search over range of $10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}$ and identified the parameter that leads to the best performance. Datasets were normalized prior to the classification. We used `svmtrain` and `svmclassify` functions in MATLAB Statistics Toolbox [282]. We also used sequential minimal optimization (SMO) method [283] since it performs significantly faster and produces similar results with the quadratic programming method.

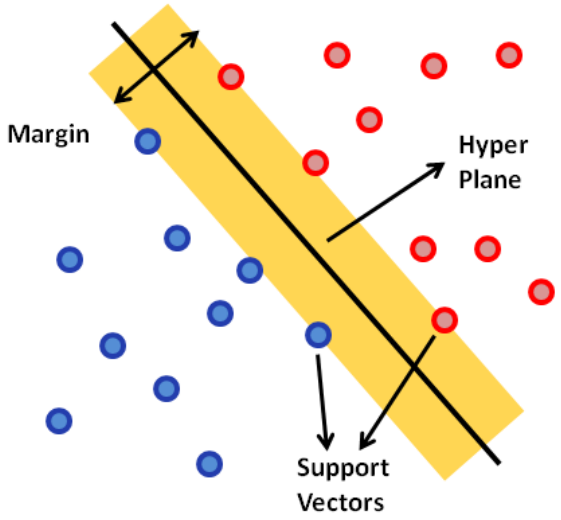

Figure 7.11: SVM classifier illustration. SVM find an optimal hyper-plane that maximizes the margin area in between two classes of data.

7.5.5 Adaptive Boosting (AdaBoost)

AdaBoost was formulated by Freund and Schapire in 1995 [204, 284]. While SVM tries to find the *training samples* that are the most difficult to classify (support vector) and form optimal hyper plane based on these training samples, AdaBoost looks for the best *training features* that are useful for its weak classifiers. AdaBoost iteratively select the best training feature on each training cycle. After each feature selection, sample weights are re-adjusted according to the local classification error. The same process repeats until a certain number of features are selected. Weak classifiers, normally linear decision stumps with only threshold and polarity parameters, are then combined together to form a strong classifier. Mathematically, the linear decision stump classifier can be written as

$$h_t(x, p, \theta) = \begin{cases} +1, & \text{if } p \cdot f(x) < p \cdot \theta\,, \\ -1, & \text{otherwise}\,. \end{cases} \tag{7.11}$$

where $f(x)$ is a feature descriptor, θ represents a threshold value, and p represents the direction of the inequality. We illustrate AdaBoost algorithm graphically in **Fig. 7.12** and

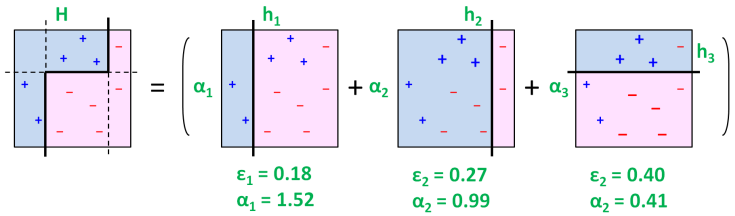

Figure 7.12: AdaBoost classifier illustration. AdaBoost classifier selects the best feature for each weak classifier h at one time, and then update the sample weights α based on the local classification error ϵ.

summarize the essential steps in **Algorithm 3**.

Algorithm 3 AdaBoost Algorithm

Step 1: Denoted as (x_i, y_i), training samples consist of a vectorized facial expression image x_i and training label $y_i = \{+1, -1\}$.

Step 2: Given P positive and N negative training data, set the weights of positive and negative samples to $w_i = \frac{1}{P}$ and $w_i = \frac{1}{N}$ respectively.

For $t = 1, \ldots, T = 80$:

Step 3: Normalize the weights of all training samples, $w_i \leftarrow \frac{w_i}{\sum_{i=1}^{k} w_i}$.

Step 4: By using **Eq.** (7.11) with $p = 1$ or $p = -1$ and different θ, compute the error rate ϵ_t of each decision stump $h_t(x)$.

Step 5: Select the decision stump $h_t(x)$ that has lowest error rate.

Step 6: Compute weight of selected decision stump, $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$.

Step 7: Update training sample weights, $w_i \leftarrow w_i \cdot \exp(-\alpha_t y_i h_t(x_i))$.

Step 8: The final strong classifier is $H(x) = \text{sign}[\sum_{t=1}^{T=80} \alpha_t h_t(x)]$.

7.5.6 AdaBoost with SVM (AdaBoostSVM)

AdaBoost is a special classifier as it performs feature selection and classification simultaneously by combining numerous weak classifiers in an additive manner. Useful features are identified by weak classifiers during the training process. A number of weak classifiers and features are then combined to form a strong classifier to carry out the image classification. Since SVM and AdaBoost design a robust classifier from different perspectives,

i.e. SVM selects *training samples* to be support vectors while AdaBoost identifies the best *training features* for weak classifiers, we expect that the integration of SVM and AdaBoost will lead to an even stronger classifier. One possible way is to use SVM as the weak classifier in AdaBoost. However, this would possibly increase both the training and test times in practice. Moreover, there is no guarantee that SVM will outperform decision stump since each weak learner only considers one feature. In contrast, we use AdaBoost to select the best training features and use SVM for classification. Specifically, all the features selected by all weak classifiers, i.e. decision stump, during the AdaBoost training process are adopted as the features for training the SVM classifier. Our concept is similar to [235, 237, 247, 249] but we fixed the number of selected features at 80 in our studies, which is close to the number of support vectors in our SVM classifiers. Note that we could only determine a very rough estimate of the number of support vectors since we are performing multiple one-vs.-all experiments in 20 validation cycles.

7.6 Classification Results

We report experiment results of each feature descriptors with six classifiers: k-NN, RFS+LDA, PCA+LDA, SVM, AdaBoost, and AdaBoostSVM classifiers in this section. We carried out repeated random sub-sampling validation in all our experiments. Specifically, we randomly selected 90% of all images for training and used the remaining 10% images for testing. We repeated this process for 20 times and report the mean and standard deviation of test accuracies in the followings. In the meantime, we focus our discussion on experiment results of CK+ Dataset. In **Sect. 7.11**, we will generalize our test results to the remaining datasets (MUG, KDEF, JAFFE Datasets) and a combined dataset (CK+ & MUG & KDEF & JAFFE).

7.6.1 k-Nearest Neighbors (k-NN)

Table 7.2 summarizes the test accuracies of five feature descriptors with k-NN classifiers applied to the CK+ Dataset. We tested performance of 1-NN, 5-NN, 10-NN, and 20-NN classifiers with an exhaustive search. While the overall performance is the worst among all classifiers, we found an interesting pattern in k-NN classification results. We find that when the number of nearest neighbors is small, LBP descriptor tends to produce the best

Table 7.2: Test accuracies of k-Nearest Neighbors (CK+ Dataset).

Descriptors	1-NN	5-NN	10-NN	20-NN
Gabor	57.7 ± 5.4	50.5 ± 7.3	59.5 ± 5.7	60.9 ± 5.6
Haar	52.7 ± 4.9	41.1 ± 6.5	47.3 ± 5.1	47.3 ± 5.1
LBP	**60.2 ± 5.1**	**58.1 ± 7.7**	60.5 ± 6.8	59.3 ± 6.3
HOG	56.5 ± 4.4	52.3 ± 5.9	**66.4 ± 4.3**	**68.4 ± 3.6**
BRIEF	55.6 ± 5.4	51.1 ± 6.6	57.7 ± 5.6	57.6 ± 6.0

results. On the other hand, when the number of nearest neighbors is large, we find that HOG descriptor tends to produce the best results. Moreover, the larger the number of nearest neighbors, the higher the test accuracies. However, a larger number of nearest neighbors would lead to a higher computational cost. In short summary, we considered 20 as the largest number of nearest neighbors in this experiment and identified HOG descriptor as the best feature descriptor for FER in k-NN classifier.

7.6.2 Linear Discriminant Analysis (LDA)

Table 7.3 summarizes the test accuracies of five feature descriptors with RFS+LDA and PCA+LDA classifiers applied to the CK+ Dataset. First of all, we observe that the performances of both RFS+LDA and PCA+LDA classifiers are better than all k-NN classifiers. Secondly, as expected, PCA performs better than RFS as PCA reduces the size of feature descriptors while capturing the most important information systematically. Last but not least, we find that HOG descriptor produces the best test accuracies in both RFS+LDA and PCA+LDA classifiers.

7.6.3 Support Vector Machine (SVM) and AdaBoost

Table 7.4 summarizes the test accuracies of five feature descriptors with SVM, AdaBoost, and AdaBoostSVM classifiers applied to the CK+ Dataset. We observe that once again, HOG descriptor produces the best test accuracies in all the three classifiers. Here, we examine the classifiers more carefully. First, we find that linear SVM classifier produces the best test accuracy among all the six classifiers in our experiments. Second, out of our expectation,

Table 7.3: Test accuracies of Random Feature Selection+LDA and PCA+LDA (CK+ Dataset).

Descriptors	RFS+LDA	PCA+LDA
Gabor	79.4 ± 5.2	82.7 ± 3.4
Haar	80.8 ± 4.9	83.4 ± 4.2
LBP	66.4 ± 6.0	86.4 ± 2.6
HOG	**82.5 ± 4.1**	**90.9 ± 3.2**
BRIEF	67.2 ± 5.9	83.7 ± 3.4

we find that AdaBoost classifier performs slightly worse than linear SVM classifier. Third, contrary to previous studies [237, 247], we find that in most cases, AdaBoostSVM classifier produces worse result than both linear SVM and AdaBoost classifiers. We believe that this discrepancy is caused by the uses of different experiment datasets, different tuning practices in SVM classifier, and different weak learners in AdaBoost classifier.

In addition to the test accuracies, we also analyzed the influence of the soft margin (hence the number of support vectors) on the SVM performance, as well as the influence of the number of weak learners on the AdaBoost performance. We focused on the HOG and BRIEF descriptors in order to save computational time. In the experiments, we varied the soft margin parameter over the range of $10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}, \infty$ and found that all the settings produce similar test accuracy results (within ±0.5%). This indicates that the facial expression images are highly linearly separable upon the feature extraction or transformation.

In the AdaBoost experiments, we varied the number of weak learners over the range of $40, 80, 120, 160, 200$. **Figure 7.13** shows the training and test accuracies of AdaBoost classifier at different number of weak learners (decision stump) with HOG (left) and BRIEF descriptors (right). We can observe that both descriptors start to achieve 100% training accuracies when the number of weak learners equal to 80. We also find that the test performances of HOG and BRIEF descriptors start to saturate when the number of weak learners are equal to 120 and 160 respectively. While we fixed the number of weak learners to 80 in the (default) comparison experiments, the differences between default and the best test accuracies are not large (less than 1% in HOG case and less than 2% in BRIEF case). Nevertheless,

Table 7.4: Test accuracies of Linear SVM and AdaBoost (CK+ Dataset).

Descriptors	Linear SVM	AdaBoost	AdaBoostSVM
Gabor	83.6 ± 3.4	81.1 ± 5.8	79.5 ± 4.0
Haar	80.2 ± 3.5	78.0 ± 4.1	74.7 ± 4.6
LBP	86.0 ± 4.0	81.2 ± 3.9	82.9 ± 4.6
HOG	**91.2 ± 3.2**	**85.7 ± 3.0**	**85.9 ± 4.0**
BRIEF	83.2 ± 3.4	79.7 ± 4.4	78.9 ± 4.9

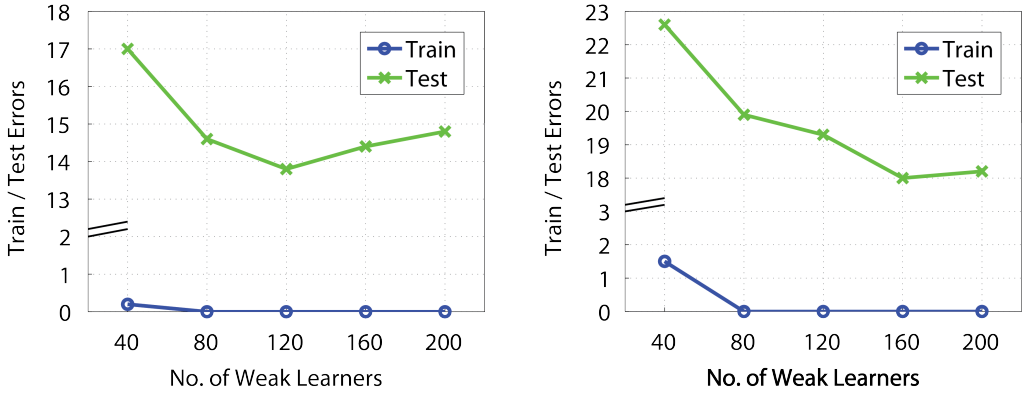

Figure 7.13: Training and test errors of AdaBoost classifier at different number of weak learners with HOG (left) and BRIEF descriptors (right).

these results suggest that we could increase the number of weak learners from 80 to 120 (in the HOG case) and to 160 (in the BRIEF case) in practice in order to achieve more accurate performance.

7.6.4 Overfitting Consideration

Since the CK+ Dataset that we are using is relatively small when compared to the dimensions of feature vectors, overfitting issue should be analyzed in order to consolidate the classification results. In addition to the previously mentioned repeated sub-sampling validation procedures, we further performed an overfitting test by plotting the training error and validation error as a function of the size of the training dataset. Since performing the experiments with all feature descriptors is time-consuming, we focused the overfitting test

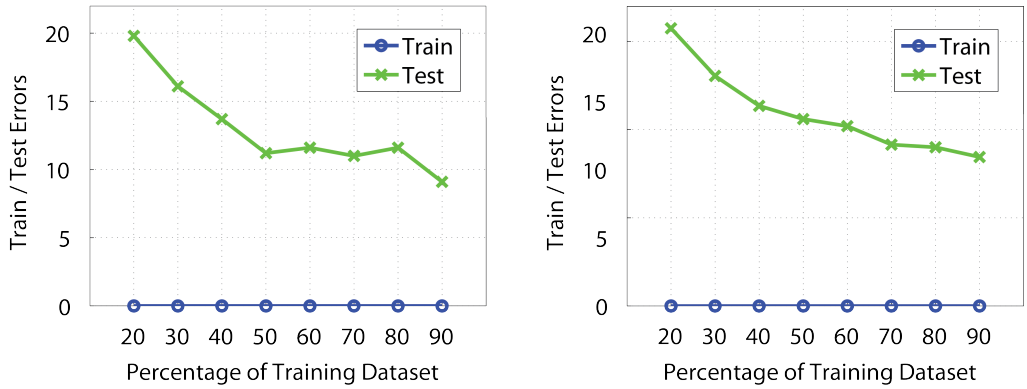

Figure 7.14: Training and test errors of SVM classifier at different percentage of training data with HOG (left) and BRIEF descriptors (right).

on HOG and BRIEF descriptors with SVM classifier. We chose HOG descriptor because it has the best test performance among other feature descriptors. We chose BRIEF descriptor because it has an out-of-expectation result in **Sect. 7.10**, where its classification performance outperform LBP and HOG descriptors when the image resolution is lower than 24×24 pixels. Note that the dimensions of both HOG and BRIEF descriptors are larger than the size of the training dataset. On the other hand, we chose SVM classifier because it does not perform feature selection and considers the full feature vector during the image classification.

Figure 7.14 shows the training and test accuracies of SVM classifier at different percentage of training data with HOG (left) and BRIEF descriptors (right). Note that all the experiment settings remain the same. We randomly sub-sample 10% of the full data for test, and vary the percentage of the remaining data to train the SVM. We repeated all experiments for 20 times and report the mean error results. From the plots, we can observe that both descriptors always have 0% training errors and have *decreasing* test errors, indicating that *no* overfitting occurs in our experiments. Moreover, the results suggest a lack of training data. With more training data, we expect an increase of training errors and further decrease of test errors.

7.6.5 Discussion

In this section, we focus on the basic version of all feature descriptors. From this point of view, we observe that most of the time HOG descriptor produces the best test accuracies

(except 1-NN and 5-NN classifiers). Compared to other descriptors, HOG descriptor is more powerful as it considers a few important feature extraction processes, such as the uses of image gradient, overlapping blocks, and histogram normalization.

The performance of Gabor descriptor in our experiments is also slightly different with works of Bartlett et al., possibly due to the use of different dataset[3] and different cross-validation approach [245]. Here, we use 20-round repeated random sub-sampling validation while Bartlett et al. use leave-one-subject-out cross validation.

7.7 Parameters Sensitivity Analysis

As described in **Sect. 7.4**, all Gabor, Haar, LBP, HOG, and BRIEF descriptors have a number of parameter choices that may significantly affect our experiment performances. We varied a few important parameters during the feature extraction processes in our experiments and report their sensitivities in this section. For all feature descriptors, we focus on PCA+LDA, SVM, and AdaBoost classifiers. All other experiment settings remain unchanged unless mentioned otherwise. Note that some results in this section are not exactly the same with the results in the previous section because we are using repeated random sub-sampling validation in our experiments.

7.7.1 Gabor Parameters

As shown in **Eq.** (7.1), Gabor filter has five parameter choices, where the differences in oscillation frequency (λ) and orientation (θ) can result in very different filters. In **Sect. 7.6**, we extracted Gabor descriptors by using 5 oscillation frequencies ($\lambda = 4, 4\sqrt{2}, 8, 8\sqrt{2}, 16$ pixels per cycle) and 8 orientations ($\theta = 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°$). Here, we denote this setting as $f = 5$ and $o = 8$. We varied these two parameters individually by first using 3 oscillation frequencies ($\lambda = 4, 8, 16$ pixels per cycle) or 1 oscillation frequency ($\lambda = 8$ pixels per cycle). Then, we used 4 orientations ($\theta = 0°, 45°, 90°, 135°$) or 2 orientations ($\theta = 0°, 90°$) to extract Gabor descriptors.

We summarized the classification results of the three classifiers in **Table 7.5**, where f and o represent the number of oscillation frequencies and orientations mentioned in the last paragraph. From the first three rows, we can observe that all three classification results

[3] We use the latest version of CK+ Database.

Table 7.5: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with varied oscillation frequencies and orientations when extracting Gabor descriptor. (See text for details.)

Gabor	PCA+LDA	SVM	AdaBoost
f=1, o=8	77.9 ± 4.4	73.3 ± 6.7	75.3 ± 6.2
f=3, o=8	83.2 ± 5.9	80.1 ± 6.1	78.8 ± 4.7
f=5, o=8	**83.3 ± 4.7**	**83.4 ± 5.5**	**79.8 ± 4.4**
f=5, o=4	81.3 ± 4.1	78.8 ± 6.5	79.0 ± 4.6
f=5, o=2	78.4 ± 4.3	73.8 ± 5.7	77.0 ± 5.3

improve steadily, indicating that increasing the number of oscillation frequencies has a positive impact towards the performance. From the last three rows, we can also observe that all three classification results improve steadily (from bottom to the third row), indicating that increasing the number of orientations has a positive impact towards the performance. In a short summary, these two parameters have high sensitivity towards the classification performance. While further increase the numbers of both parameters might continue to improve the classification performance, we limit $f = 5$ and $o = 8$ in our experiments in order to save computational cost.

7.7.2 Haar Parameters

Compared to Gabor filter, Haar filter has less parameter choices. In **Sect. 7.6**, we extracted Haar descriptors with 5 types of Haar filter and with a step size of 2 pixels. Here, we denote this setting as $s = 2$ and $t = 5$. Similar to the Gabor filter case, we varied these two parameters individually by first using step sizes of 4 or 3. After that, we used the first 4 types or the first 2 types of Haar filter (please refer to **Fig. 7.5**) to extract Haar descriptors.

We summarized the classification results of the three classifiers in **Table 7.6**, where s and t represent the number of step size and number of types of filter mentioned in the last paragraph. From the first three rows, we can observe that all three classification results improve steadily, indicating that increasing the number of step size has a negative impact towards the performance. Out of our expectation, from the last three rows, we observe that using more types of Haar filter indeed does not improve the results of all

Table 7.6: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with varied step sizes and filter types when extracting Haar descriptor. (See text for details.)

Haar	PCA+LDA	SVM	AdaBoost
s=4, t=5	82.9 ± 5.4	75.9 ± 6.3	64.9 ± 5.9
s=3, t=5	**84.0 ± 5.3**	78.1 ± 6.0	70.6 ± 6.5
s=2, t=5	83.7 ± 4.2	78.8 ± 5.7	76.7 ± 5.3
s=2, t=4	83.1 ± 4.6	79.4 ± 5.8	**77.6 ± 6.3**
s=2, t=2	83.1 ± 5.0	**79.7 ± 5.0**	77.5 ± 5.4

three classifiers. While more types of Haar filter could be helpful in differentiating face and non-face images [206], we find that using the first two types of Haar filter (the simplest Haar filter) is the best for FER. In a short summary, one should use a step size of 2 pixels of and use the first two types of Haar filter for FER.[4]

7.7.3 LBP Parameters

LBP descriptor has two parameter choices. In **Sect. 7.6**, we extracted LBP descriptors with 8 neighboring pixels at radial distance of 2 pixels. Here, we denote this setting as $p = 8$ and $r = 2$. Similar to the previous cases, we varied these two parameters individually by first using 6 or 4 neighboring pixels. After that, we used the radial distance of 3 or 4 pixels to extract LBP descriptors.

We summarized the classification results of the three classifiers in **Table 7.7**, where p and r represent the number neighboring pixels and radial distance. From the first three rows, we can observe that all three settings produce the best results when $p = 6$. From the last three rows, we find that PCA+LDA and SVM classifiers perform the best at $r = 2$ and $r = 3$ while AdaBoost classifier perform similarly and appears to be independent of radial distance. In contrast to a previous study [235] and our default parameter settings, one should use 6 neighboring pixels and different range of radial distance depending on the classifiers.[5]

[4] Since we perform this analysis at the later stage of our comparison study, we keep the results of the original setting, i.e. $s = 2$ and $t = 5$ in all other sections. Note that this does not affect our conclusion that HOG is the best descriptor for FER since different t produce similar test results.

[5] Similarly, we keep the results of the original setting, i.e. $p = 8, r = 2$ in all other sections. This does not affect our conclusion since the LBP descriptor with the latest settings still does not outperform HOG

Table 7.7: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with varied neighboring points and radial distances when extracting LBP descriptor. (See text for details.)

LBP	PCA+LDA	SVM	AdaBoost
p=4, r=2	86.8 ± 4.7	86.5 ± 5.2	77.8 ± 4.8
p=6, r=2	**89.2 ± 3.7**	**88.3 ± 3.9**	79.3 ± 4.2
p=8, r=2	85.8 ± 4.5	85.4 ± 5.4	79.4 ± 5.0
p=8, r=3	88.9 ± 4.0	85.9 ± 5.0	78.9 ± 5.5
p=8, r=4	87.4 ± 4.2	83.9 ± 4.7	**79.5 ± 5.1**

7.7.4 HOG Parameters

As described in **Sect. 7.4**, HOG descriptor is compact and has only one parameter choice—its cell size. In **Sect. 7.6**, we extracted HOG descriptors with cell sizes of 8 and 4, and stacked the two feature vectors together eventually. Here, we denote this setting as $c = 8$ and $c = 4$. Different with the previous cases, we varied the cell size and analyze the test accuracies by using individual feature vector or stacked feature vectors.

We summarized the classification results of the three classifiers in **Table 7.8**, where c represents the cell size in pixel values. The rows with multiple c represent stacked feature vectors with different cell sizes. From the first three rows, we can observe that all three classification produce the best classification results when $c = 4$. When $c = 4$, the resulting number of blocks and dimension of HOG descriptor is larger and can potentially capture more detailed edge information from the facial expression images. From the last four rows, we find that all classifiers produce similar (PCA+LDA and AdaBoost) or slightly better (SVM) test accuracies than HOG descriptor with one cell size. Note that we use $c = 8$ and $c = 4$ as our default experiment settings, which has similar outcomes to the best test accuracies in **Table 7.8**.

7.7.5 BRIEF Parameters

BRIEF descriptor has two parameter choices—the number binary pixel pairs and number of BRIEF values. In **Sect. 7.6**, we extracted 10000 BRIEF values with 5 binary pixel

descriptor.

Table 7.8: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with varied cell sizes and combination when extracting HOG descriptor. (See text for details.)

HOG	PCA+LDA	SVM	AdaBoost
c=8	84.7 ± 6.2	84.1 ± 4.4	80.8 ± 6.4
c=6	90.6 ± 3.5	89.1 ± 3.9	84.1 ± 5.0
c=4	**91.6 ± 3.6**	90.7 ± 3.9	**86.0 ± 3.9**
c=8, c=6	89.8 ± 3.6	89.8 ± 3.5	83.8 ± 5.4
c=8, c=4	91.2 ± 3.2	91.3 ± 4.0	85.2 ± 5.0
c=6, c=4	90.6 ± 3.7	90.6 ± 4.5	85.3 ± 5.8
c=8, c=6, c=4	91.2 ± 4.1	**91.6 ± 3.7**	84.8 ± 4.3

pairs. Here, we denote this setting as $b = 5$ and $f = 10000$. Similar to the Gabor, Haar, and LBP cases, we varied these two parameters individually by first using 2 or 8 binary pixel pairs while extracting 10000 BRIEF values. After that, we fixed binary pixel pairs to 5 and extracted 5000 or 2000 BRIEF values.

We summarized the classification results of the three classifiers in **Table 7.9**, where b and f represent the number of binary pixel pairs and number of BRIEF values. From the first three rows, we can observe that PCA+LDA and SVM perform the best when $b = 5$ while AdaBoost performs the best when $b = 8$. From the last three rows, we can observe that the test accuracies deteriorate with decreasing number of BBIEF values. In general, one should use a larger f if the computational cost is not an issue. On the other hand, we find that using $b = 5$ has a good balance in between computational cost and accuracy performance (except for the AdaBoost case, but nevertheless the difference is small.)

7.7.6 Advanced Variants

In addition to the analysis of feature parameters, it is also worth testing some advanced variants of the classic feature descriptors. We further compared the FER performance by extracting LBP histogram Fourier (LBP-HF) descriptor [285] and Gaussian BRIEF descriptor (G-BRIEF) [286] . The LBP-HF descriptor is a rotation invariant descriptor computed from Fourier transforms of LBP histogram. It has been reported that LBP-HF descriptor

Table 7.9: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with varied number of binary pixel pairs and descriptors when extracting BRIEF descriptor. (See text for details.)

BRIEF	PCA+LDA	SVM	AdaBoost
b=2, f=10000	83.1 ± 5.8	82.6 ± 5.5	80.6 ± 4.2
b=5, f=10000	**83.6 ± 4.8**	**83.0 ± 4.4**	79.5 ± 5.0
b=8, f=10000	82.9 ± 5.3	83.0 ± 4.9	**80.9 ± 4.0**
b=5, f=5000	82.7 ± 5.1	82.2 ± 5.4	77.5 ± 5.6
b=5, f=2000	80.1 ± 5.5	80.9 ± 5.2	76.4 ± 5.2

Table 7.10: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with LBP and LBP Histogram Fourier descriptors.

	PCA+LDA	SVM	AdaBoost
LBP	**86.4 ± 2.6**	**86.0 ± 4.0**	**81.2 ± 3.9**
LBP-HF	82.1 ± 6.2	80.6 ± 4.4	73.6 ± 4.9

outperforms classic LBP in texture classification and face recognition tests [285]. **Table 7.10** summarized the test results of the classic LBP and LBP-HF descriptors. Based on our experiments, we find that the LBP-HF descriptor produce worse test results when compared to the classic LBP descriptor. We speculate that the LBP-HF descriptor will perform better on datasets with in-plane rotations. Since our current datasets always have the same head pose, the LBP-HF descriptor does not prove to be very useful.

Table 7.11 summarizes the test results of the classic BRIEF and G-BRIEF descriptors. Based on our experiments, we find that the G-BRIEF descriptor perform slightly better than the classic BRIEF descriptor only in the case of SVM classifier. It has been reported that G-BRIEF is good at dealing with image Gaussian noise and partial occlusions. We speculate that G-BRIEF will performs better if there is Gaussian noises or partial occlusions in our FER datasets. To this end, we find that HOG descriptor is still the best descriptor for FER in consideration of the test accuracy.

Table 7.11: Test accuracies of PCA+LDA, SVM, and AdaBoost (CK+ Dataset) with BRIEF and Gaussian-BRIEF descriptors.

	PCA+LDA	SVM	AdaBoost
BRIEF	**83.7 ± 3.4**	83.2 ± 3.4	**79.7 ± 4.4**
G-BRIEF	81.7 ± 5.7	**83.9 ± 4.7**	78.3 ± 6.1

7.8 Confusion Matrices

We present confusion matrices of HOG descriptor applied with SVM classifier in order to identify the weak points of feature descriptors. **Table 7.12** presents the confusion matrix of HOG descriptor with linear SVM classifier applied to the CK+ Dataset, with *An*, *Di*, *Fe*, *Ha*, *Sa*, *Su*, and *Ne* represent angry, disgust, fear, happy, sad, surprise, and neutral facial expressions respectively. From **Table 7.12**, we observe that the *happy* class performs the best and achieves accuracy of nearly 97%. On the other hand, the *sad* classifier performs the worst and achieves accuracy of only about 44%, where most of the *sad* class images are mis-classified as *neutral* class. We find that this bad performance is caused by the small number of training samples in *sad* class. Note that the *fear* class also has a small number of training samples and performs second worst. In order to prove our speculation, we run the same experiment with KDEF Dataset, in which it has same number of images in all classes. **Table 7.13** presents the confusion matrix of HOG descriptor with linear SVM classifier applied to the CK+ Dataset. From the table, we observe that the *sad* classifier performs much better with the KDEF Dataset, proving that our speculation is correct. In **Sect. 7.11**, we will present more experiment results on KDEF, MUG, and JAFFE Datasets.

7.9 Feature Fusion Investigation

Since all feature descriptors possess different characteristics, one common question would be "Could the test accuracies be further improved by using multiple feature descriptors for image classification?". We have analyzed this issue by considering the following experiment. We combined the LBP, HOG, and BRIEF descriptors and tested the outcomes with AdaBoost classifier. It would be also interesting to realize this experiment by using SVM or other techniques such as multiple kernel learning (MKL) [287, 288]. Similar to AdaBoost,

Table 7.12: Confusion matrix of HOG descriptor with linear SVM classifier (CK+ Dataset).

%	An	Di	Fe	Ha	Sa	Su	Ne
An	**76.4**	6.8	1.1	1.1	1.1	2.3	11.2
Di	1.9	**90.4**	1.0	1.9	0.0	1.9	2.9
Fe	3.6	1.8	**71.3**	3.6	1.8	3.6	14.3
Ha	0.0	0.0	0.0	**97.1**	0.0	0.7	2.2
Sa	10.2	6.8	5.1	8.5	**44.0**	0.0	25.4
Su	0.0	0.0	1.2	0.0	0.6	**93.4**	4.8
Ne	0.7	0.0	0.6	0.5	0.7	0.2	**97.3**

Table 7.13: Confusion matrix of HOG descriptor with linear SVM classifier (KDEF Dataset).

%	An	Di	Fe	Ha	Sa	Su	Ne
An	**78.7**	7.1	6.4	1.4	1.1	2.5	2.8
Di	4.1	**80.7**	4.1	2.2	2.2	4.5	2.2
Fe	5.0	4.0	**65.6**	4.3	5.7	4.3	11.1
Ha	0.7	1.0	0.4	**95.2**	0.7	1.0	1.0
Sa	1.1	0.4	0.4	1.1	**93.8**	1.4	1.8
Su	3.1	5.8	6.2	2.7	5.0	**76.0**	1.2
Ne	0.3	1.0	6.3	0.7	3.3	1.6	**86.8**

Table 7.14: Test accuracies of AdaBoost classifier with combined features in CK+ Dataset.

Descriptors	AdaBoost
LBP	81.2 ± 3.9
HOG	85.7 ± 3.0
BRIEF	79.7 ± 4.4
LBP+HOG+BRIEF	**87.3 ± 3.8**

Table 7.15: Percentages of feature selected by AdaBoost in 20 validation cycles with CK+ Dataset.

Descriptors	An (%)	Di (%)	Fe (%)	Ha (%)	Sa (%)	Su (%)	Ne (%)
LBP	30	24	**35**	29	24	28	33
HOG	**40**	**54**	**35**	35	**41**	**42**	**42**
BRIEF	30	22	30	**36**	35	30	25

MKL has the ability to perform feature selection and classification simultaneously. However, our main objective is to analyze the description power of the feature descriptors by explicitly counting the number of feature descriptors selected by AdaBoost. To this end, we chose to consider MKL as our future work for FER.

Table 7.14 summarizes our experiment results. From the table, we can observe that the combined LBP & HOG & BRIEF descriptors perform the best, followed by HOG descriptor, LBP descriptor, and BRIEF descriptor. In addition, **Table 7.15** shows the number of feature descriptors selected by AdaBoost during the combined-features experiment. We observe that HOG descriptor almost always have the largest number of selection, indicating that HOG descriptor has more description power than other feature descriptors in the AdaBoost classification experiment.

7.10 Image Pre-processing Investigation

In this section, we investigate the effects of image pre-processing towards the FER test performances. Specifically, we investigated the effect of image resolutions towards the

test performances. In addition to the effect of image resolutions, we also compared the results under normal pre-processing, i.e. face detection and cropping procedures described in **Sect. 7.3**, to the results with additional pre-processing steps. We considered histogram equalization (HE) and median filtering (MF) in our experiments. HE could potentially enhances facial features that are not obvious in the low contrast images. On the other hand, MF could filter the image noises and smoothen the facial images. Note that we do not consider illumination normalization since the datasets we are using now are collected under controlled settings (i.e. have same illumination within datasets).

7.10.1 Effect of Image Resolution

We investigate the effect of image resolution on test accuracies in this section. Looking for an optimal image resolution that produces high test accuracy and low computational cost is important but is not commonly focused in research analysis. **Table 7.16** summarizes the size of all the five feature descriptors used in our experiments under different image resolution settings. The fourth column (48×48 = 2304) is the original setting in our previous experiments. Specifically, LBP operator has size of 12×12 pixels and shift horizontally and vertically by 3 pixels, resulting in a feature vector with size of 13×13×59 = 9971. HOG operator has two combined settings—block size of 16×16 pixels and 8×8 pixels respectively, resulting in a feature vector with size of (5×5+11×11)×36 = 5256. BRIEF descriptor was set to have 10000 values, which is close to the LBP descriptor size of 9971. We only considered LBP, HOG, and BRIEF descriptors in this section. The dimensions of Gabor and Haar descriptors of images at higher resolution are too large and significant amount of computational resources are required for the experiments.

Figure 7.15 summarizes the test accuracies of CK+ and KDEF Datasets under different image resolutions with AdaBoost classifier. Test results of both datasets are consistent. We observe that LBP and HOG descriptors produce test accuracies higher than BRIEF descriptor at high resolution images (≥ 48×48 pixels). In contrast, BRIEF descriptor produces higher test accuracies than LBP and HOG descriptors at low resolution images (< 48×48 pixels). Furthermore, BRIEF descriptor can produce surprisingly good test accuracies across all different image resolutions despite its much lower computational cost. We also observe that HOG descriptor almost always performs better than LBP descriptor across all image resolution settings. In practice, we recommend to normalize a high resolution image (≥

Table 7.16: Comparison of size of feature vectors at different image resolution.

Descriptors	12×12 = 144	24×24 = 576	48×48 = 2304	96×96 = 9216
LBP	3^2×59 = 531	8^2×59 = 3776	13^2×59 = 9971	19^2×59 = 21299
HOG	(1^2+2^2)×36 = 180	(2^2+5^2)×36 = 1044	(5^2+11^2)×36 = 5256	$(5^2+11^2+23^2)$×36 = 24300
BRIEF	2500	5000	10000	20000

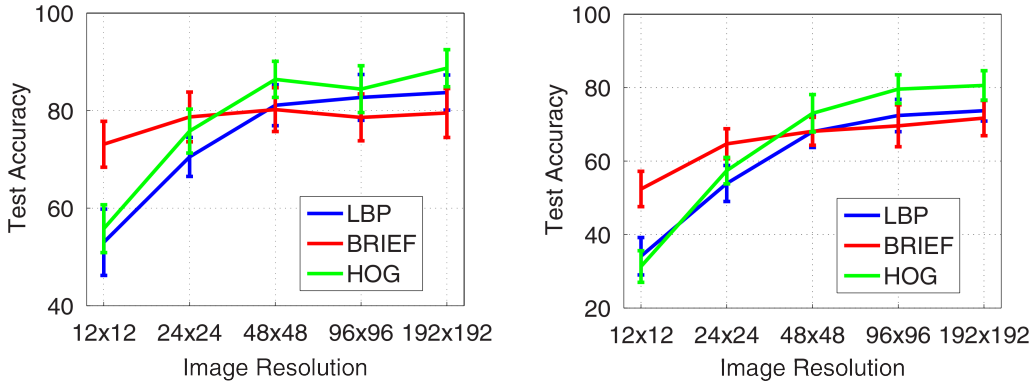

Figure 7.15: Test accuracies of AdaBoost classifier under different image resolution settings in CK+ Dataset (left) and KDEF Dataset (right).

48×48) to size of 48×48 pixels (because higher image resolution would produce similar performance) and use HOG descriptor for FER. When original image resolution is lower than 48×48 pixels, we recommend to use BRIEF descriptor.

7.10.2 Effect of Image Filtering

We find that comparing the results of HE and MF with the results of normal processing is challenging because their results strongly depend on the feature descriptors and classifiers. For the ease of understanding, we summarized the test results in five tables (**Tables 7.21–7.25**), in which each of them corresponds to Gabor, Haar, LBP, HOG, and BRIEF descriptors. For Gabor descriptor, we find that HE always produces the best test results, while MF always produces the worse test results in all three classifiers. This is reasonable as HE could enhances facial features that are not obvious in the low contrast images while MF smoothen the facial features and would deteriorate the test results.

Table 7.17: Test accuracies of Random Feature Selection+LDA and PCA+LDA.

Descriptors	MUG Dataset		KDEF Dataset		JAFFE Dataset	
	RFS+LDA	PCA+LDA	RFS+LDA	PCA+LDA	RFS+LDA	PCA+LDA
Gabor	**76.7 ± 4.3**	80.3 ± 5.0	69.2 ± 4.3	71.5 ± 3.4	62.9 ± 11.3	79.1 ± 8.2
Haar	75.4 ± 5.1	79.4 ± 5.1	68.8 ± 4.6	70.2 ± 3.5	**65.0 ± 11.2**	76.0 ± 10.9
LBP	56.7 ± 5.1	77.8 ± 3.7	52.4 ± 4.3	72.0 ± 4.3	34.3 ± 9.4	63.3 ± 6.2
HOG	71.7 ± 4.8	**82.6 ± 4.5**	**68.0 ± 4.7**	**77.1 ± 3.7**	54.3 ± 11.2	**85.5 ± 6.6**
BRIEF	54.1 ± 6.6	74.5 ± 3.6	50.6 ± 4.7	69.9 ± 3.8	37.4 ± 13.3	69.3 ± 8.2

Table 7.18: Test accuracies of SVM and AdaBoost.

Descriptors	MUG Dataset		KDEF Dataset		JAFFE Dataset	
	SVM	AdaBoost	SVM	AdaBoost	SVM	AdaBoost
Gabor	82.7 ± 3.4	**77.5 ± 4.1**	72.6 ± 5.5	71.3 ± 3.1	82.4 ± 7.1	**69.0 ± 10.3**
Haar	78.6 ± 4.7	73.5 ± 4.6	69.0 ± 6.4	67.6 ± 5.0	77.1 ± 10.2	67.4 ± 12.5
LBP	79.0 ± 4.2	67.9 ± 5.6	74.7 ± 5.2	66.7 ± 3.9	55.7 ± 9.8	49.5 ± 9.6
HOG	**85.3 ± 4.2**	77.0 ± 5.3	**80.2 ± 4.1**	**75.2 ± 4.0**	**89.5 ± 6.3**	64.0 ± 11.0
BRIEF	81.4 ± 4.2	71.7 ± 4.4	73.4 ± 5.7	68.8 ± 5.4	72.6 ± 10.4	62.1 ± 13.0

Table 7.19: Test accuracies of Random Feature Selection+LDA and PCA+LDA.

Descriptors	Combined Dataset		CK+ Dataset	
	RFS+LDA	PCA+LDA	RFS+LDA	PCA+LDA
Gabor	55.2 ± 2.7	56.6 ± 2.4	79.4 ± 5.2	82.7 ± 3.4
Haar	55.4 ± 2.8	58.6 ± 3.4	80.8 ± 4.9	83.4 ± 4.2
LBP	47.0 ± 3.5	63.7 ± 2.1	66.4 ± 6.0	86.4 ± 2.6
HOG	**60.5 ± 3.5**	**68.1 ± 3.2**	**82.5 ± 4.1**	**90.9 ± 3.2**
BRIEF	44.0 ± 3.2	59.3 ± 3.5	67.2 ± 5.9	83.7 ± 3.4

Table 7.20: Test accuracies of SVM and AdaBoost.

Descriptors	Combined Dataset		CK+ Dataset	
	SVM	AdaBoost	SVM	AdaBoost
Gabor	61.6 ± 2.4	62.8 ± 3.1	83.6 ± 3.4	81.1 ± 5.8
Haar	59.9 ± 2.8	60.0 ± 3.6	80.2 ± 3.5	78.0 ± 4.1
LBP	70.5 ± 2.7	59.6 ± 3.0	86.0 ± 4.0	81.2 ± 3.9
HOG	**73.3 ± 3.3**	**63.2 ± 3.2**	**91.2 ± 3.2**	**85.7 ± 3.0**
BRIEF	73.0 ± 2.7	59.3 ± 2.5	83.2 ± 3.4	79.7 ± 4.4

Table 7.21: Test accuracies of Gabor descriptor (CK+ Dataset) by using PCA+LDA, SVM, and AdaBoost classifiers with normal, histogram equalization, median filtering pre-processing.

	PCA+LDA	SVM	AdaBoost
Normal	82.7 ± 3.4	83.6 ± 3.4	81.1 ± 5.8
HE	**84.4 ± 3.4**	**84.7 ± 4.8**	**82.5 ± 4.7**
MF	80.0 ± 5.3	80.1 ± 4.5	78.8 ± 4.8

For HOG descriptor, we find that all three classifiers produce the best results without additional pre-processing. Different to Gabor filter, this indicates that HOG descriptor could capture the edge features in the facial images efficiently even without additional HE. This again would lead us to favor HOG descriptor in FER. For Haar, LBP, and BRIEF descriptors, the test performances vary depending on the chosen classifiers and we could not make a clear conclusion.

7.11 Generalization Tests

We performed three types of generalization test in this section. First, we applied the same experiments to the MUG, KDEF, and JAFFE Datasets. Second, we combined all the CK+, MUG, KDEF, and JAFFE Datasets into a large dataset (totally 2748 images), performed the same procedures, and report its test performance. Third, we trained the PCA+LDA, SVM,

Table 7.22: Test accuracies of Haar descriptor (CK+ Dataset) by using PCA+LDA, SVM, and AdaBoost classifiers with normal, histogram equalization, median filtering pre-processing.

	PCA+LDA	SVM	AdaBoost
Normal	83.4 ± 4.2	80.2 ± 3.5	**78.0 ± 4.1**
HE	**84.5 ± 4.3**	**81.1 ± 6.1**	77.1 ± 6.1
MF	82.7 ± 4.6	77.7 ± 5.6	74.5 ± 5.6

Table 7.23: Test accuracies of LBP descriptor (CK+ Dataset) by using PCA+LDA, SVM, and AdaBoost classifiers with normal, histogram equalization, median filtering pre-processing.

	PCA+LDA	SVM	AdaBoost
Normal	86.4 ± 2.6	**86.0 ± 4.0**	81.2 ± 3.9
HE	87.1 ± 5.3	84.6 ± 4.9	**81.3 ± 5.0**
MF	**88.0 ± 4.7**	84.4 ± 5.4	75.6 ± 6.7

Table 7.24: Test accuracies of HOG descriptor (CK+ Dataset) by using PCA+LDA, SVM, and AdaBoost classifiers with normal, histogram equalization, median filtering pre-processing.

	PCA+LDA	SVM	AdaBoost
Normal	**90.9 ± 3.2**	**91.2 ± 3.2**	**85.7 ± 3.0**
HE	88.9 ± 5.5	88.6 ± 4.0	82.0 ± 5.3
MF	88.3 ± 4.0	87.1 ± 6.0	83.1 ± 4.8

Table 7.25: Test accuracies of BRIEF descriptor (CK+ Dataset) by using PCA+LDA, SVM, and AdaBoost classifiers with normal, histogram equalization, median filtering pre-processing.

	PCA+LDA	SVM	AdaBoost
Normal	**83.7 ± 3.4**	83.2 ± 3.4	79.7 ± 4.4
HE	82.8 ± 5.0	**83.5 ± 4.5**	79.4 ± 3.8
MF	83.0 ± 5.5	83.0 ± 5.3	**80.6 ± 5.7**

and AdaBoost classifiers with all CK+ Dataset images and tested classifiers' performance with all MUG, KDEF, and JAFFE Datasets.

7.11.1 MUG, KDEF, and JAFFE Datasets

Table 7.17 summarizes the test accuracies of five feature descriptors with RFS+LDA and PCA+LDA classifiers applied to the MUG, KDEF, and JAFFE Datasets. Again, as expected, PCA performs better than RFS because PCA reduces the size of feature descriptors while capturing the most important information systematically. We also find that HOG descriptor produces the best test accuracies in PCA+LDA classifier.

Table 7.18 summarizes the test accuracies of five feature descriptors with SVM and AdaBoost classifiers applied to the MUG, KDEF, and JAFFE Datasets. We observe that SVM classifier performs better than AdaBoost classifier across all the three datasets and HOG descriptor produces the best test accuracies in SVM classifier. Moreover, we find that SVM classifier performs better than the PCA+LDA classifier in **Table 7.17**.

7.11.2 Combined Datasets

Table 7.19 summarizes the generalization performance of RFS+LDA and PCA+LDA classifiers in the combined dataset (we also repeat the test results of CK+ Dataset for direct comparison purpose). While the combined dataset is about 4 times larger than CK+ Dataset, we observe that both RFS+LDA and PCA+LDA classifiers perform worse in the combined dataset. This is not surprising, as the four datasets are collected under controlled indoor environment with different backgrounds and light settings. Moreover, the four datasets

have different demographical settings, e.g. CK+ Dataset was collected in North America, MUG and KDEF Datasets were collected in Europe, while JAFFE Dataset was collected in Asia. Our generalization test performance conforms with previous FER studies, suggesting that we need to collect more FER images in the wild in order to achieve a more robust FER [235, 245].

Table 7.20 summarizes the generalization performance of SVM and AdaBoost classifiers in the combined dataset (we repeat the test results of CK+ Dataset for direct comparison purpose). Similar to the preceding discussion, we observe that both SVM and AdaBoost classifiers perform worse in the case of combined dataset, suggesting that we need to collect more FER images in the wild in order to achieve a more robust FER.

7.11.3 Cross Datasets

In addition to the combined dataset, we also trained the PCA+LDA, SVM, and AdaBoost classifiers with all CK+ images and tested classifiers' performance with all MUG, KDEF, and JAFFE images. We summarized the test accuracies of the three datasets in **Table 7.26**. From the table, we can observe that in general, all test performances are much worse than the performances of individual CK+ Dataset and combined datasets cases due to the reasons discussed in **Sect. 7.11.2**. By referring to the dataset images (**Fig. 7.1–7.3**), we can observe that each dataset was collected under very different controlled indoor environment, which eventually lead to these unsatisfactory results. Similar to the previous sections, these test results suggest us to collect more FER images in the wild in order to achieve more robust FER.

In addition, we also observed that JAFFE Dataset produces much worse cross-dataset results than MUG and KDEF Datasets. Upon careful investigation, we found that JAFFE Database has a few ambiguous facial expressions posed by models. **Figure 7.16** illustrates a few faces with ambiguous facial expression. For instance, the first image has a label of angry but may be potentially perceived as sad expression. Since JAFFE Database is small, these ambiguity represents about 5% of the all 213 images. We believe that this is the main reason that JAFFE Dataset performs the worst in our cross-dataset experiments.

Images										
Labels	Angry	Sad	Disgust	Surprise	Angry	Disgust	Sad	Fear	Surprise	Disgust
Perception	Sad	Happy	Angry	Happy	Sad	Happy	Neutral	Neutral	Neutral	Sad

Figure 7.16: Ambiguous facial expression labels in JAFFE Dataset.

Table 7.26: Test accuracies of MUG, KDEF, and JAFFE Datasets (the 1st, 2nd, 3rd numbers in all the triplets) with PCA+LDA, SVM, and AdaBoost classifiers by using CK+ Dataset as training data.

	PCA+LDA	SVM	AdaBoost
Gabor	24.6 / 24.0 / 11.7	29.6 / 24.9 / 15.5	26.2 / 22.9 / 16.0
Haar	27.6 / 27.0 / 15.0	31.2 / 32.2 / 16.9	33.3 / 31.3 / 16.0
LBP	26.9 / 30.2 / 14.1	28.1 / 27.6 / 13.6	26.4 / 26.6 / 20.7
HOG	31.9 / 35.3 / 13.6	29.5 / 35.5 / 18.8	27.8 / 33.3 / 23.0
BRIEF	27.5 / 26.3 / 11.7	29.4 / 25.1 / 15.0	29.1 / 26.1 / 17.4

7.12 Computational Efficiency

Feature descriptor should be computationally efficient while leading to the best classification results. In this section, we analyze the computational cost of five feature descriptors in term of number of multiplication (NOM) and summation (NOS) operations. We first list up a few assumptions and describe our calculation process. In all five feature descriptors, we consider a grayscale image with size of 48×48 pixels.

Among the five feature descriptors, Gabor descriptor has the most expensive computational cost because it involves 48×48 summation and 48×48 multiplication operations. On the other hand, taking the computational cost of integral image into consideration, Haar descriptor has computational cost of only

$$NOS_{Haar} = 15 \times numberOfSelectedFeatures + imageSize^2 \times 3 \; . \tag{7.12}$$

The computation of LBP descriptor involves thresholding, binary mapping, and histogram binning operations. For simplicity, we assume one binary thresholding operation is equivalent to one summation operation. Besides, we assume that binary mapping is realized with Lookup Table (LUT) technique and is considered as one summation operation as well. We also assume that histogram binning operation is equivalent to one summation operation. Under these assumptions, our implementation of *one* LBP descriptor has computational cost of

$$NOS_{LBP} = (\frac{imageSize}{4})^2 \times (8 + 1 + 1) \; . \tag{7.13}$$

The computation of HOG descriptor involves convolution, square-root, arctangent, histogram binning, and normalization operations. However, it has surprisingly low computational cost when LUT technique is employed. Convolution process (by applying 1-D horizontal and vertical Sobel masks) seems to be expensive but it is in fact equivalent to one summation operation for each gradient map computation process. Square-root and arctangent operations can also be replaced with LUT technique since there are only 511×511 possibilities for an 8-bit grayscale image. Hence, we count these as two summation operations for every pixel when calculating gradient magnitude and gradient orientation maps. Subsequent histogram binning operations is considered equivalent to one summation operation. Local normalization involves 35 summation and 72 multiplication operations for each block. Under

these assumptions, our implementation of *one* HOG descriptor has computational cost of

$$NOS_{HOG} = ((\frac{imageSize}{4})^2 \times (2 + 2 + 1) + 35)\ ,\ NOM_{HOG} = 72\ . \tag{7.14}$$

BRIEF descriptor is well-known for its simplicity and fast computation. It involves only five comparison operations (equivalent to five summation operations) and one binary mapping operation (counted as one summation operations). BRIEF descriptor is also the only feature that has computation cost independent of image size. Overall, our implementation of *one* BRIEF descriptor has computational cost of

$$NOS_{BRIEF} = (5 + 1)\ . \tag{7.15}$$

In descending order, Gabor descriptor has the most expensive computational cost, followed by HOG, LBP, Haar, and BRIEF descriptors. Assuming that one multiplication operation is equivalent to ten summation operations, we show two interesting plots in **Fig. 7.17**. We investigated the number of summation operations by varying the number of selected features and changing the image size. We observe that all feature computational costs increase approximately linearly when number of features increase. On the other hand, LBP and HOG descriptors have exponentially increasing summation operations when image size increases. Nevertheless, all features have very low computation cost in modern computers. For instance, it takes less than 15 μs to compute one LBP and HOG descriptor respectively in MATLAB (C & C++ implementations). Note that Haar and BRIEF descriptors have even lower computational cost in the two cases shown in **Fig. 7.17**. Haar descriptor has a low computational cost thanks to the integral image technique while BRIEF descriptor has a computational cost independent of the image size.

7.13 Feature Visualization

Figure 7.18 and **Fig. 7.19** visualize Gabor and Haar descriptors selected by AdaBoost classifiers in CK+ Dataset. These figures show some insights about the size and position of the descriptors selected by the AdaBoost classifier. We can see that most Gabor and Haar descriptors are small and concentrate at the image center. In **Fig. 7.20**, we overlap all 80 feature descriptors selected by AdaBoost classifier. We can observe that all feature

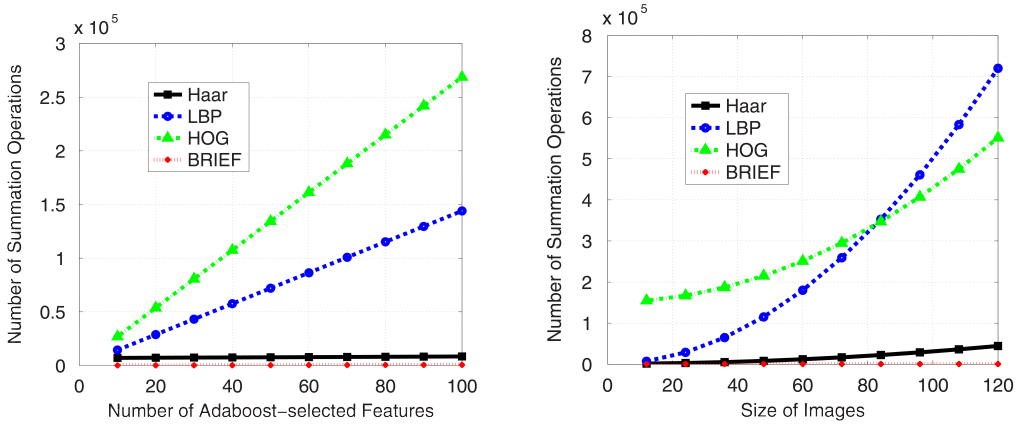

Figure 7.17: Number of summation operations of feature descriptors with varied number of features (left) and varied size of images (right).

descriptors concentrate at the image center. The observation also suggests us to put more weights on these location when we extract the feature descriptors from the face images. This concept is similar to the idea in [235], when LBP descriptor is extracted with different weighting effect based on their extraction location.

7.14 Conclusion

In this chapter, we empirically evaluate *five* feature descriptors, namely Gabor, Haar, LBP, HOG, and BRIEF descriptors in FER. We examine each feature descriptor by considering *six* classification methods, such as k-NN, LDA, SVM, and AdaBoost with *four* unique facial expression datasets. In the end, we identified HOG descriptor as the best feature descriptor for FER when image resolution of a detected face is higher than 48×48. On the other hand, when the image resolution of the detected face is smaller than 48×48, our experiment results show that BRIEF descriptor performs the best. In general, Gabor descriptor performs well but has a higher computational cost. In addition to the test accuracies, we presented confusion matrices of FER. We analyzed the effect of combined features and image resolutions on FER performance. We also generalized our experiments to other datasets, analyzed the computational efficiency of each feature descriptors, and visualized the feature descriptors selected by AdaBoost classifier.

In this study, we only consider frontal facial expression images. The use of facial

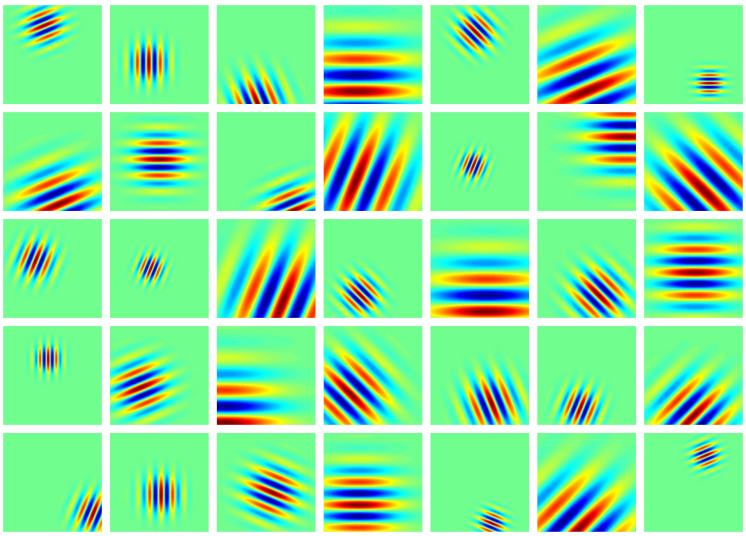

Figure 7.18: Visualization of Gabor descriptors selected by AdaBoost classifier. Each column represents the first five Gabor descriptors selected by *angry*, *disgust*, *fear*, *happy*, *sad*, *surprise*, *neutral* classifier respectively.

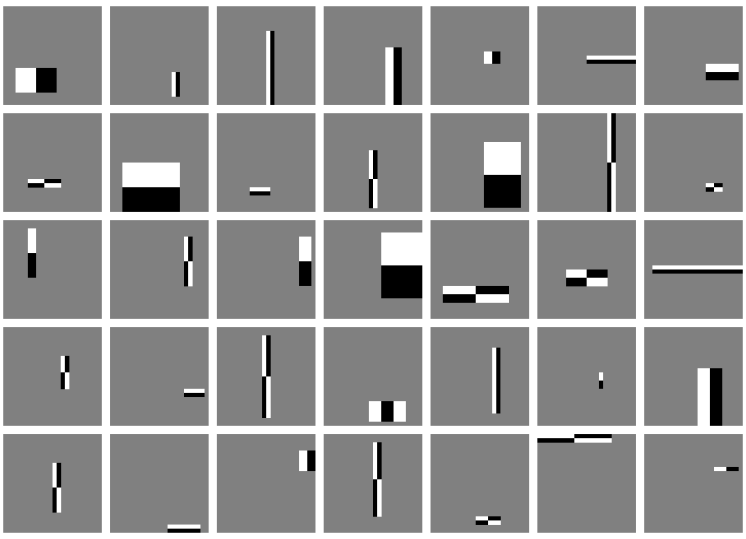

Figure 7.19: Visualization of Haar descriptors selected by AdaBoost classifier. Each column represents the first five Haar descriptors selected by *angry*, *disgust*, *fear*, *happy*, *sad*, *surprise*, *neutral* classifier respectively.

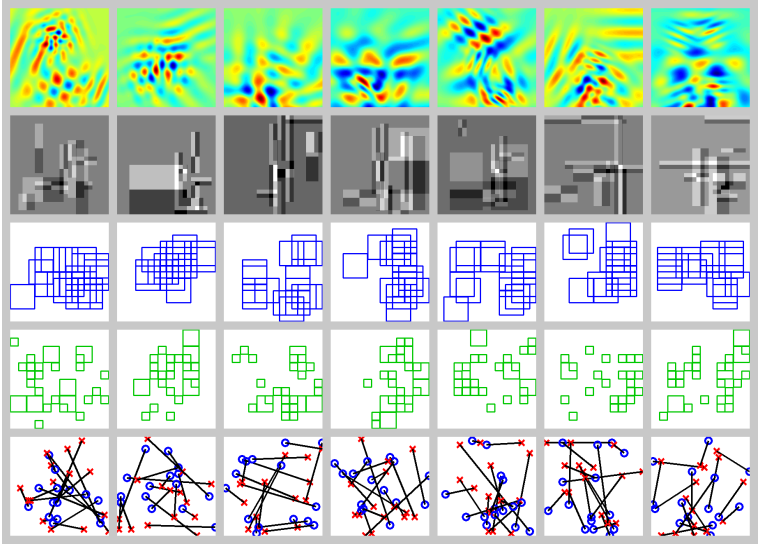

Figure 7.20: Visualization of 80 overlapping Gabor, Haar, LBP, HOG, and BRIEF descriptors (row-wise) of *angry*, *disgust*, *fear*, *happy*, *sad*, *surprise*, *neutral* classifier respectively.

expression dataset under different head poses is necessary to consolidate our findings. We also focus FER on single image. Temporal information provides strong clues about facial expression and should be carefully considered in the future.

Chapter 8
Human Sensing Interface IV: Face Alignment

8.1 Introduction

Face alignment, a process of locating facial feature points (**Fig. 8.1**), has been an active research topic because of its usefulness in vast applications such as face recognition and head pose estimation. However, automatic and real-time face alignment is very challenging due to the large variations of face shape, head pose, illumination, facial expression, and occlusions such as glasses or mustaches. A widely used approach for face alignment is parametric shape fitting, where each facial feature points are first estimated with independent local landmark estimators and then a global shape model is used to regularize the estimated local landmarks. Active shape model (ASM) [289, 290] and constrained local model (CLM) [291, 292] are two typical frameworks. In these frameworks, local landmark estimators can be any classifiers or regressors while point distribution model (PDM) [292, 293] is normally used as the global shape model. Recently, sparse representation model (SRM) [294–296] has also been proposed as the global shape model.

8.1.1 Our Approach

To the best of our knowledge, using the random forest regressor as local landmark estimators and using the PDM as global shape model [293] is currently the best framework among all parametric shape fitting approaches. Its training process is relatively simpler than other boosted regression approaches, and yet able to achieve accurate, fast, and robust alignment results. However, we find that it does not generalize well to face images with yaw angle larger than $\pm 15^{\circ}$. In this chapter, we propose to solve this problem by using a new local forest classification and regression (LFCR) framework. In particular, we add an additional classification step prior to the regression step. Verified by our experiments, we find that this additional classification step is useful in rejecting outliers prior to the regression step and improve the alignment results. We also further improve the LFCR framework by analyzing each system component, including the choice of feature descriptors, tuning parameters of random forest regressors, as well as the shape initialization and regularization processes.

Figure 8.1: Face alignment, a process of locating facial feature points, has been used in various applications such as face recognition, facial expression recognition, and head pose estimation. Seventeen selected facial feature points are plotted on images taken from BioID Dataset [297].

8.1.2 Our Contribution

Our contributions in this chapter is threefold. First, we propose a new LFCR framework for facial alignment that can generalize well to face images with yaw angle larger than ±15°. Second, we examine different initialization methods (such as sparse initialization and two-stage initialization), as well as different shape regularization methods (such as PDM and SRM) in our experiments. Third, we demonstrate the effectiveness of LFCR over state-of-the-art methods with two widely-used face alignment datasets—BioID Dataset [297] and MultiPIE Dataset [298].

8.2 Related Works

8.2.1 Parametric Shape Fitting Approaches

Parametric shape fitting are the most common approaches in face alignment problem, where the facial feature points are first estimated locally and the resulting face shape is regularized with a global shape model. These approaches are normally performed iteratively until convergence and are shown to be effective in practice. ASM [289, 290] and CLM [291, 292] are two typical approaches. In general, both ASM and CLM and their variants optimize

an objective function consisting of appearance likelihood and shape constraint. The shape constraint is typically represented by using PDM, i.e. $\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{Pb}$, where $\bar{\mathbf{x}}$ is a mean shape of training data, $\mathbf{P}$ contains t eigenvectors (eigenshapes) of the covariance matrix of training data, and $\mathbf{b}$ is a t dimensional vector. While being effective in practice, these approaches are widely known to be very sensitive to shape initialization. Parametric shape fitting approaches are normally initialized with mean shape $\bar{\mathbf{x}}$ and would likely to be fail if the face being aligned has a true shape that is very *far away* from the mean shape. It is tempting to enlarge search window of local landmark estimator but this would increase the computational cost exponentially and hinder real-time performance in practice.

In addition to PDM, SRM [294–296] has also been proposed as the global shape model. Instead of representing a face shape with mean shape eigenshapes computed from the training data, SRM represents a face shape with face shapes taken from the training data directly. Therefore, SRM has an advantage of representing a face shape that is not statistically significant in the training data. For example, if a test image has a true shape of fully-opened mouth while most training data have closed mouth, eigenshapes would most probably contain face shapes with only closed mouth and it would be difficult for PDM to regularize the face shape. In contrast, since SRM takes face shapes from the training data directly, SRM would still contains face shapes with closed mouth and is able to regularize face shape effectively.

8.2.2 Boosted Regression Approaches

In contrast to the parametric shape fitting approaches above, boosted regression approaches [299, 300] performs holistic face alignment, where features extracted from the face box are mapped to the facial feature points vector directly and the face shape constraint is implicitly modeled by the holistic regressors. Dantone et al. [301] proposed Conditional Random Forest to estimate facial feature points holistically by using appearance, gradient, and Gabor features. Xiong et al. [302] proposed Supervised Descent Method to optimize facial feature points search by using SIFT features. Sun et al. [303] and Zhou et al. [304] also proposed deep convolutional networks for face alignment.

Cao et al. [300] perform multiple initialization and consider mean of all results as final landmark location. In contrast, our method evaluates the quality of each initialization candidate with an objective function inspired by sparse model and carry on subsequent

processes with only the identified best candidate.

Valstar et al. [305] and Chen et al. [296] first detect prominent facial landmarks and then align mean shape to these landmarks with similarity transformation. However, this approach is not sufficient as facial landmarks could be occluded and possible false detection could lead to worse initialization.

8.2.3 Deformable Shape Approaches

The basic idea of deformable shape approaches [306–310] is to represent a face by a collection of face feature parts arranged in a deformable shape configuration. Specifically, the appearance of each face part is modeled separately while the deformable shape is represented by spring-like connections between pairs of face parts. In contrast of parametric shape fitting approaches, deformable part approaches optimize local appearance and shape deformation cost simultaneously. As a result, it normally can produce better convergence results but is more time-consuming. Felzenszwalb & Huttenlocher. [306] first demonstrated the idea of deformable shape (which they called it Pictorial Structure) in face alignment. Uricar et al. [307] treated the pictorial structure as a structured output problem and solved it with structured output Support Vector Machine. Aiming to capture face pose variation, Everingham et al. [308] improved the deformable shape approach by using a mixture of Gaussian trees. Zhu & Ramanan [309]. further extended the idea of mixtures of trees with a shared pool of parts. Instead of using all densely distributed facial feature points, Yu et al. [310] proposed a group sparse learning method to select the most representative facial feature points to improve the tracking speed performance. Ghiasi & Fowlkes [311] proposed a hierarchical deformable shape approach for face alignment that explicitly models occlusions of parts.

8.3 Framework Overview

Similar to previous parametric shape fitting approaches [293], we perform three major steps in our framework: (1) shape initialization, (2) local landmarks regression, and (3) global shape regularization. **Figure 8.2** illustrates this basic face alignment pipeline. We first describe the two shape initialization methods that we are using in our experiments in Sec. 8.3.1. Then, we explain the design process of local landmark regression in Sec. 8.3.2. Af-

Figure 8.2: Framework overview. During the test, our system starts from shape initialization, followed by local landmark searching and global shape regularization.

ter that, we explain two global shape regularization processes in Sec. 8.3.3.

8.3.1 Shape Initialization

We compare two types of shape initialization in our experiments: the mean shape initialization and sparse shape initialization [312]. Mean shape initialization uses the mean shape computed from the training data as the initialized shape. While being effective most of the time, it would most likely to fail if the face being aligned has a true shape that is very different or *far away* from the mean shape. On the other hand, sparse initialization method has been proposed to solve this problem [312]. **Figure 8.3** illustrates the sparse initialization method with one test image. Specifically, one performs initialization with n shape candidates that are selected heuristically within the face space of training data. By evaluating an objective function with response maps obtained from local landmark detectors, we are able to determine the best initialization shape candidate and enhance the alignment performance.

While one can randomly initialize a few possible shapes within face space by either varying elements in vector parameter $\mathbf{b}$ in PDM (hereafter *PC-shape*) or directly selecting available training shape (hereafter *AT-shape*), this is not effective in practice as most selected shapes might be similar. Instead, we select shapes within the face space based on simple heuristic. We select the first shape randomly and then select the second shape that has the largest Euclidean distance with respect to the first selected shape. After that, we select the third shape that has the largest Euclidean distance to the nearest shape among those previously selected (i.e. maximin strategy). The process continues until we obtain n required initialization shapes.

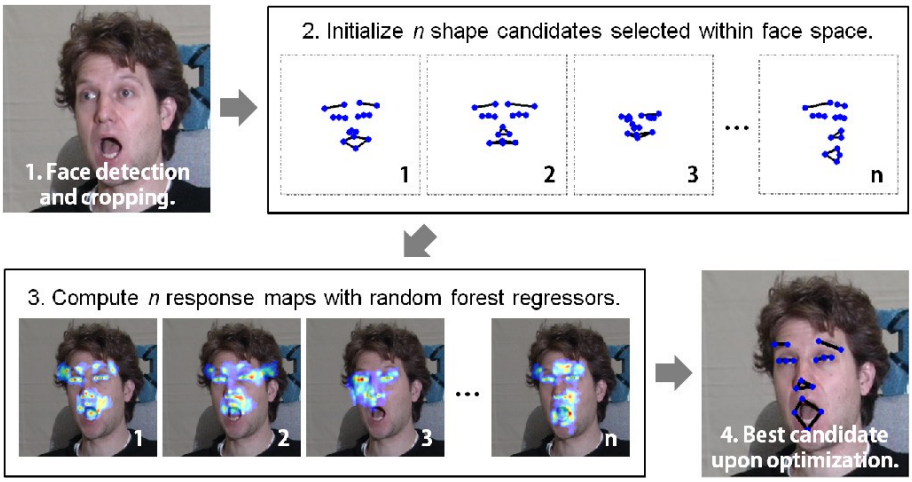

Figure 8.3: A process flow of sparse initialization for face alignment [312]. By optimizing an objective function inspired by sparse model, sparse initialization can identify the best candidate and enhance alignment performance.

8.3.2 Local Forest Regression

Random forest [207] is a powerful ensemble learning method for both classification and regression that works by combining outcomes of multiple decision trees. Cootes et al. [293] proposed to use random forest regressors to estimate the location of each facial feature point independently. Similarly, we design one random forest regressor for each landmark. However, prior to the regressor, we add one random forest classifier for each landmark. Since image patches are randomly sampled at each landmark, there is a chance where the sampled image patches contain significant occluded objects or belong to backgrounds. We find that the added classifier is good at rejecting these patches, where the patches essentially carry no useful information for the subsequent regressor. With the added classifier, the face alignment performance can be improved substantially.

During the training process, we sample k square image patches P_k around each ground truth facial feature point with a random displacement vector $d_k \in R^2$. We consider normal image patches as positive samples, while image patches that contain significant occluded objects or belong to backgrounds as negative samples. We then extract feature f_k from these image patches and learn a classifier. After that, we use the same features extracted from positive samples to learn a regressor $\phi(f_k) \rightarrow d_k$ that map the extracted features to the displacement vector. In this work, we report performance of both HOG features and BRIEF

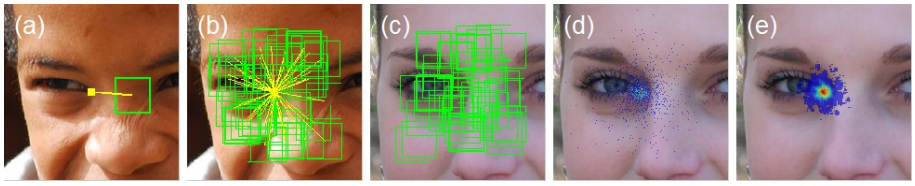

Figure 8.4: Image patches' sampling process around an eye corner point. **(a)** An image patch (the green square) is sampled around the ground truth point with a random displacement vector (the yellow line). **(b)** Pairs of image patches and displacement vectors are randomly sampled during the training. **(c)** Image patches are randomly sampled around the initialization point during the test. **(d)** Raw voting map by all image patches. **(e)** Filtered voting map after applying KDE.

descriptor. **Fig. 8.4(a)-(b)** show the image patches' sampling process around an eye corner point. During the test, image patches can be randomly extracted around the current facial feature point (**Fig. 8.4(c)**).

During the test process, we sample k' square image patches P'_k around the initialized/current updated landmark with random displacement vectors $d'_k \in R^2$. We then extract feature f'_k from the sampled image patches. With the pre-trained classifier, we can determine whether the sampled image patches are positive samples. All image patches that are considered as negative samples will be ignored thereafter. With the pre-trained regressor ϕ, we then estimate the displacement vector $\phi(f'_k)$. We use a single vote per tree per sampled image patch as suggested by [293]. All votes are accumulated in a 2D voting grid/map (**Fig. 8.4(d)**). Selecting the best regression location that has the most votes in the 2D voting map is one potential way [293] but the prediction might be prone to error. Chen et al. [296] avoid this error by taking the mean of landmark location (MLL) estimated by all random trees. In the experiments below, we select the peak density after applying kernel density estimation (KDE) with 5×5 average kernel to the 2D voting map (**Fig. 8.4(e)**). In Section 8.4, our empirical results indicate that KDE approach outperforms MLL approach.

8.3.3 Global Shape Regularization

After the local landmark search, we perform the global shape regularization with two purposes. First, global shape regularization lets strong landmarks such as eye and mouth corner points to guide weak landmarks such as eyebrow and nose tip points. Second, it

corrects strange alignment results and ensures that the final aligned shape stays within the face space of the training data.

We investigate two different shape regularization methods in our experiments, namely PDM [292, 293] and SRM [294–296, 313]. In PDM approach, the aligned shape is reconstructed with the representative eigenvectors (computed from PCA) of face shapes of the training data. On the other hand, in SRM approach, the aligned shape is reconstructed directly with the face shapes of the training data. While PDM approach is faster, we find that SRM approach can represent the face space of the training data better, especially when there are some face shapes that are statistically insignificant in the training data. For example, PDM approach would have difficulty in reconstruct a face shape that has yaw angle of ±45° if there is only a very small number of similar face shapes in the training data. However, SRM approach could reconstruct the face shape better than PDM approach, provided that the face shapes used for reconstructed have been selected properly.

8.4 Experiment Results

We start our experiments by first comparing each component in LFCR rigorously. In particular, we show that LFCR can obtain good performance over face images with yaw angle up to ±45°. We also compare the HOG [203] and BRIEF [223] feature extraction method explicitly. After that, we examine the effects of search window and image patch size, number of image patches, and number of trees toward the final alignment results.

Dataset. We carry out experiments by using the widely used MultiPIE Database [298]. The database consists of more than 750,000 images of 337 people under 15 view points and 19 illumination conditions. We select 4,478 near-frontal face images (within ±45° yaw angle) in the following experiment. Note that MultiPIE images do not have large in-plane rotation. We augment the original images by randomly rotating the face images up to ±45°. We also applied random scaling and random shifting effect, up to 10% and 5% of the face box size respectively, onto the selected 4,478 near-frontal face images. Augmented face images that cannot be detected by face detector [206] are re-augmented. All face boxes detected by the face detector are enlarged by 100% to ensure sampled image patches stay within the face box boundaries. We then normalize the face box to size of 240×240 pixels. These processes ensure that our dataset is very challenging, which includes large in-place rotation, scaling

and shifting variations with respect to the face boxes. From this dataset, we select and fix 50% of the full data as our training dataset in the following experiments.

Implementation details. In our experiment, image patches have size of 25×25 pixels[1] while searching window has size of 17×17 pixels, i.e. horizontally and vertically 8 pixels away from the current point. We set the number of patches k to 20 during both training and test processes in the beginning to save computational time. We use 10 random regression trees. In order to avoid overfitting problem, the random trees are fully expanded and then pruned to have lowest mean square error with respect to 10-fold cross validation loss.

Evaluation metrics. Following previous works [293, 301], we measure the alignment performance by computing the mean error of 17 facial landmarks shown in **Fig. 8.1** as a percentage of inter-ocular distance (IOD). Denoted as m_{17}, this measure is invariant to face size and is widely used in face alignment study:

$$m_{17} = \sum_{i=1}^{17} \frac{1}{17} \frac{1}{IOD} \|(x_i, y_i) - (\hat{x}_i, \hat{y}_i)\|_2, \tag{8.1}$$

where (x_i, y_i) and $(\hat{x}_i, \hat{y}_i)$ are coordinates of the estimated and ground truth facial feature points respectively.

8.4.1 LFCR Results

Figure 8.5 shows the mean errors of 17 facial feature points in 5 cycles with and without a classification step prior to the regression step, as well as the cumulative distribution function (CDF) of m_{17} error with and without a classification step. Both results show consistently that the additional classification step can improve the localization accuracy of the feature points. In the following experiments, we always include the additional classification step since its computational cost is small.

We also compare the localization accuracy of the feature points with HOG and BRIEF descriptors. As shown in **Fig. 8.6**, HOG descriptor perform slightly better than BRIEF descriptor. Since BRIEF descriptor has lower computational cost, we continue to use BRIEF

[1] Starting from the image patch center, we expand 12 pixels evenly to the top, bottom, left, and right directions, resulting in a singular number. When extracting HOG features, we normalize the image patches to 24×24 pixels to meet the cell size requirement.

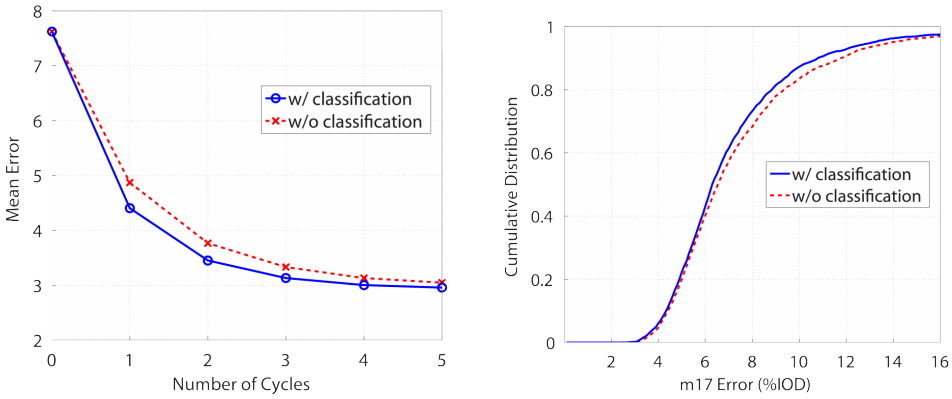

Figure 8.5: Left: Mean errors of 17 facial feature points in 5 fitting cycles with and without the classification step. **Right:** Cumulative distributions of m_{17} error with and without the classification step.

descriptor in the following experiments.

At the end of the regression of every feature points, all votes are accumulated in a 2D voting grid/map and the best regression location is identified with a filter. As shown in Fig. 8.7, our experiment results empirically show that the KDE filter outperforms MLL filter in both mean errors and CDF plots. We use KDE filter in all of the following experiments.

We find that the sizes of local search window and image patch are also critical to the LFCR performance. **Figure 8.8** shows the results of varied sizes of local search window and image patch. When the search window size is fixed, image patch size of 12 performs better than 6 because more information is available to train the regression trees. When the image patch size is fixed, window size of 8 performs better than 16 because of the lower uncertainty. We choose the image patch size and search window size to be 12 and 8 respectively, in line with the suggestions by Cootes et al. [293].

Next, we examine the effect of number of image patches during the training and test phases (**Fig. 8.9**). As expected, increasing the number of image patches during the training and test phases improves the performance. We keep the number of image patches during both the training and test phases to 20 in order to save computational time.

We also examine the effect of number of trees in the random forest (**Fig. 8.10**). For simplicity, we set the number of trees in the training and test phases to be equal. We find that increasing the number of tree does not improve the performance significantly. We keep

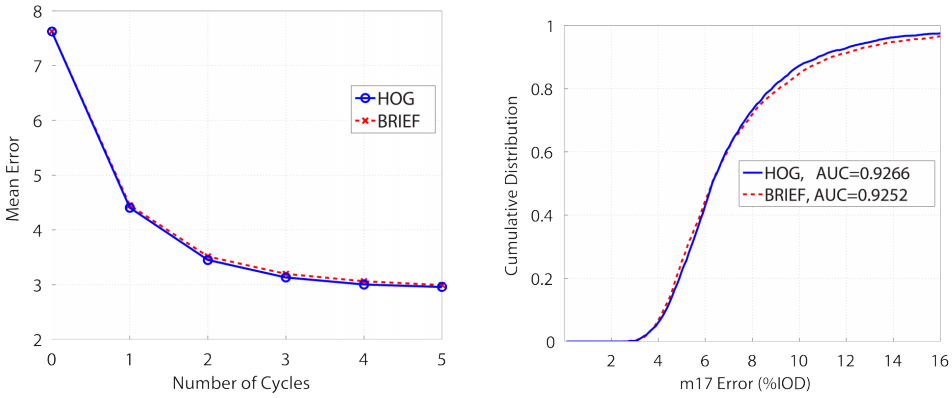

Figure 8.6: Left: Mean errors of 17 facial feature points in 5 fitting cycles by using HOG and BRIEF descriptors. **Right:** Cumulative distributions of m_{17} error by using HOG and BRIEF descriptors. AUC represents the Area Under the Curve.

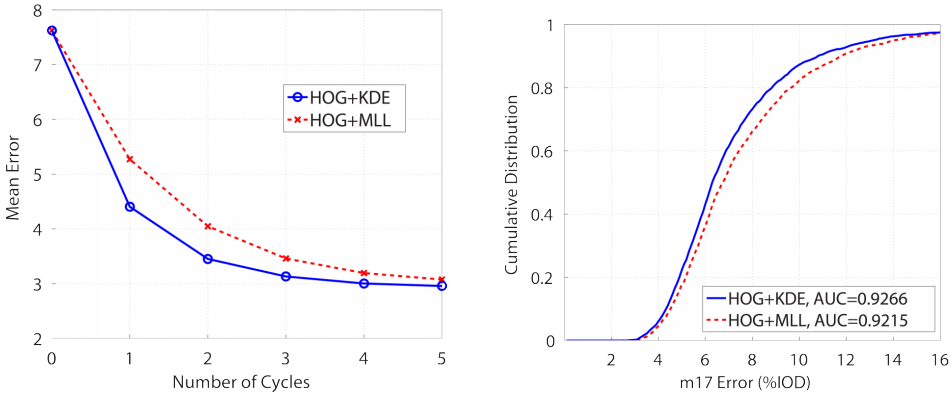

Figure 8.7: Left: Mean errors of 17 facial feature points in 5 fitting cycles by using KDE and MLL to filter fitting response maps. **Right:** Cumulative distributions of m_{17} error by using KDE and MLL to filter fitting response maps.

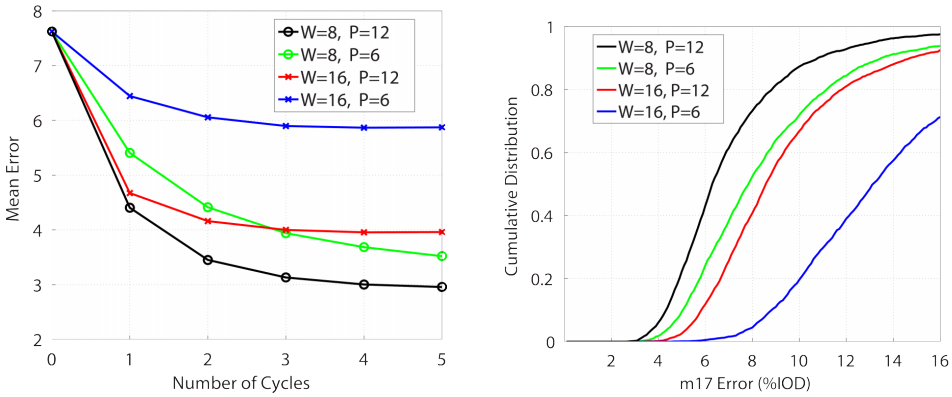

Figure 8.8: Left: Mean errors of 17 facial feature points in 5 fitting cycles with varying search window size (W) and image patch size (P). **Right:** Cumulative distributions of m_{17} error with varying search window size (W) and image patch size (P).

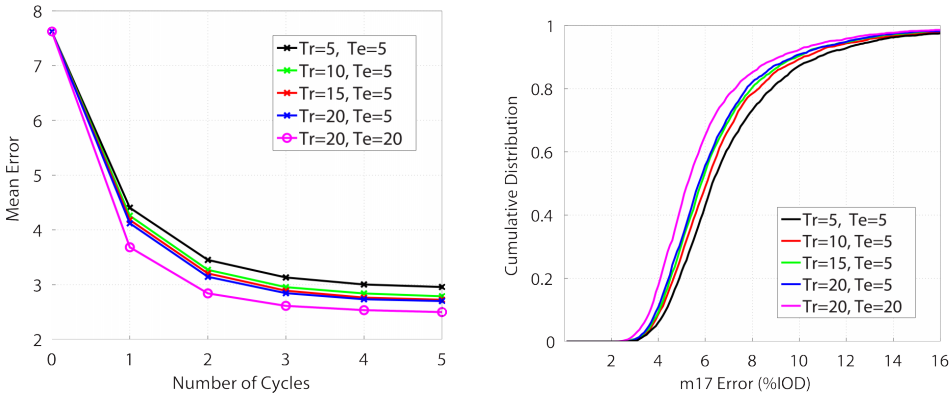

Figure 8.9: Left: Mean errors of 17 facial feature points in 5 fitting cycles with varying numbers of training patches (Tr) and test patches (Te) in each image. **Right:** Cumulative distributions of m_{17} error with varying numbers of training patches (Tr) and test patches (Te) in each image.

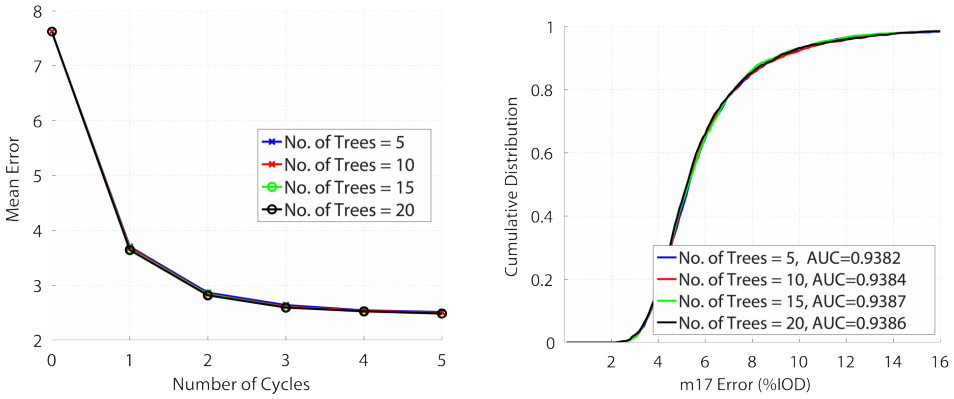

Figure 8.10: Left: Mean errors of 17 facial feature points in 5 fitting cycles with varying number of decision trees. **Right**. Cumulative distributions of m_{17} error with varying number of decision trees. AUC represents the Area Under the Curve.

the number of trees in both the training and test phases to 5 in the following experiments in order to save computational time.

8.4.2 Comparison of Shape Initialization Methods

In this section, we examine six different shape initialization methods, such as the conventional mean shape initialization [293], sparse initialization [312], multiple initialization [300], and two-stage initialization. In the conventional mean shape initialization method, the mean shape of all training data is used as the initialized shape directly. In the sparse initialization method, both PC-shapes and AT-shapes are computed from the training data and then the optimized initialized shape will be selected by maximizing an objective function [312]. In multiple initialization method, random training shapes are used as initialized shapes and the median of fitting results are taken as the final results. In the two-stage initialization method, LFCR is first performed on the test image with lower resolution (120×120), followed by a second round LFCR on the original image (240×240). In the two-stage-pa method, an additional Procrustes analysis step is included in between the first and second round of LFCR. Specifically, after the first round LFCR on the test image with lower resolution, the original image is transformed (with rotation and translation) based on the fitting result. After that, the second LFCR is performed on the transformed original image.

From **Fig. 8.11**, our findings show that two-stage initialization methods produce the

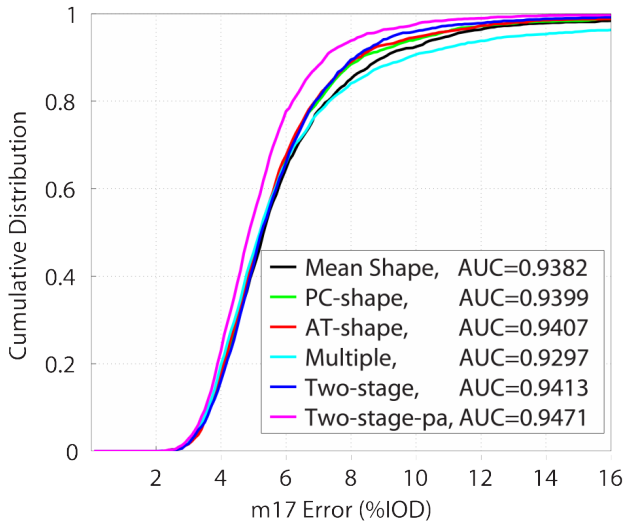

Figure 8.11: Cumulative distributions of m_{17} errors with different shape initialization methods.

best results in term of alignment accuracy and performance efficiency. With the additional Procrustes analysis step, we observe an additional boost of performance. Sparse initialization methods perform similar with the two-stage initialization method and slightly better than the mean shape initialization method, similar to the experiment results observed in the original paper [312]. Overall, the multiple initialization method perform the worse in term of area-under-the-curve (AUC) in **Fig. 8.11**. Multiple initialization method is good for boosted regression approaches due to their special consideration of random shape initialization in the training process but is not effective for parametric shape fitting approaches.

8.4.3 Comparison of Global Shape Regularization Methods

In this section, we compare two types of global shape regularization methods—PDM and SRM regularization. In the PDM regularization method, the LFCR fitting results are projected into the PCA space and then reconstructed with the principal components in PDM. In other words, the LFCR fitting results are filtered with the PCA. On the other hand, in the SRM regularization method, the LFCR fitting results are reconstructed with the training shapes directly under a sparsity constraint. From **Fig. 8.12**, we find that SRM regularization method performs slight better than PDM regularization method.

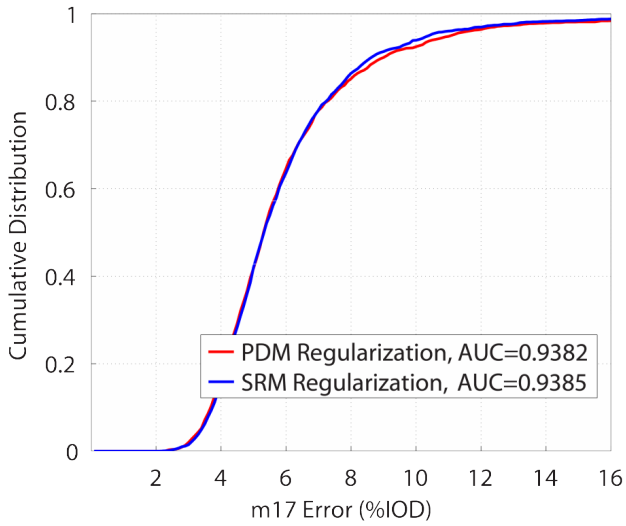

Figure 8.12: Cumulative distributions of m_{17} errors with different shape regularization methods.

8.4.4 Comparison to State of the Arts

In this section, we compare our results to the state-of-the-art results. We first report the comparison results on the BioID Dataset [297]. **Figure 8.13** shows the face alignment results of LFCR, Stasm [290], FaceTracker [292], BoRMaN [305], and LEAR [314]. From the figure, we can observe that LFCR performs the best, followed by Stasm, FaceTracker, LEAR, and BoRMaN. This indicate that LFCR is more effective than other approaches for frontal face alignment.

We also report the comparison results on the MultiPIE Dataset [298] for near frontal face alignment. **Figure 8.14** shows the face alignment results of LFCR1 (with mean shape initialization), LFCR2 (with two-stage-pa initialization), IntraFace [302] (boosted regression approach), and Zhu & Ramanan [309] (deformable shape approach). Note that the models of all methods are trained with MultiPIE Dataset. While IntraFace (boosted regression approach) produces the most accurate results, our LFCR method (parametric shape fitting approach) can be parallelized naturally (and hence achieve a speed advantage) thanks to the independent local forest classifiers and regressors. In contrast, IntraFace extract features directly from the whole face image and perform face alignment globally with pre-learned regressors, which makes them inherently difficult to parallelize. On the other hand, we find

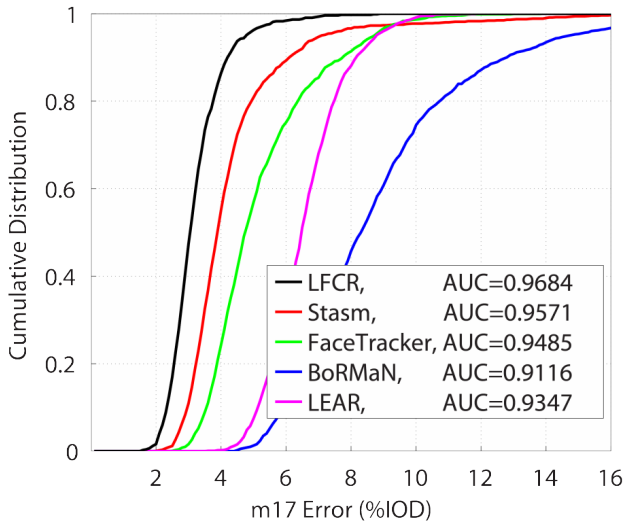

Figure 8.13: Comparison of cumulative distributions of m_{17} errors with different parametric shape fitting approaches.

that the deformable shape approach of Zhu & Ramanan do not perform well. We find that the model is good for face detection and achieve good alignment results around the face boundary, but not good for accurate face alignment on the me_{17} points.

8.5 Conclusion and Future Works

In this chapter, we propose a new LFCR framework for face alignment in order to handle face images with large yaw angles. Our framework is based on a recently proposed random forest regressor, where it is used as local landmark estimators and has robust performance when coupled with a global face shape regularizer. We analyze each system component through detailed experiments. In addition to the selection of feature descriptors and several important tuning parameters of the random forest regressor, we examine different initialization and shape regularization processes.

We also compare our best outcomes to the state of the arts and show that our method outperforms other parametric shape fitting approaches. This indicates that LFCR is effective in estimating local landmark and outperforms other local discriminative approach. Besides, its implementation is relatively easy and has very low runtime cost. The computation cost of random forest regressor can also be parallelized to achieve even faster performance.

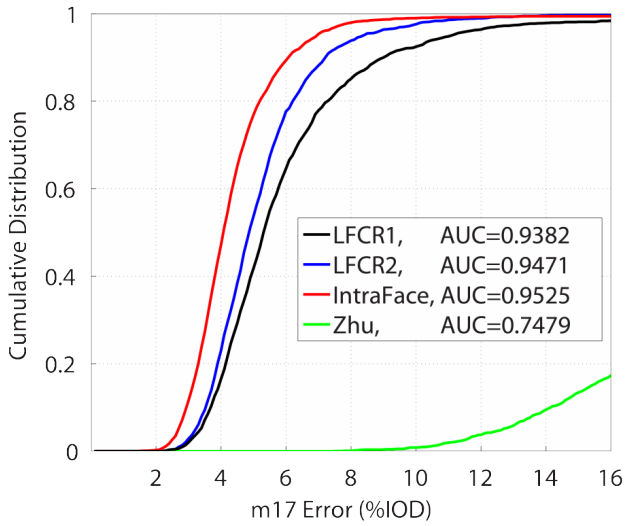

Figure 8.14: Comparison of cumulative distributions of m_{17} errors with Zhu & Ramanan and IntraFace, where the LFCR1 uses mean shape initialization and the LFCR2 uses two-stage initialization together with the Procrustes analysis.

Nevertheless, LFCR has some limitations, in particular it does not explicitly consider occlusions during the face alignment process. In our experiment, we also observe that boosted regression approaches, where all the feature extraction and face alignment processes are performed globally, produce more accurate results than LFCR.This indicates that face region that is far away from a face part also carries useful information for accurate face alignment. Combining features from multiple parts for local alignment could potentially boost accuracy of LFCR in the future.

Chapter 9
Conclusion

9.1 Summary

In this thesis, we propose the concept of a companion flying robot and discuss several challenges towards realizing this goal. Among the listed challenges, which includes topics on hardwares, control, robotics, machine learning, human-robot interaction (HRI), and design, we believe that (i) safety (of the flying robot and its user), (ii) natural HRI in between the flying robot and its user, and (iii) an interface for companion flying robot to understand human are the three most important topics towards our goal.

In the beginning of this work, we choose to use a multirotor-based flying robot over a blimp-based flying robot because a multirotor-based flying robot can handle more payload, has more responsive flights, and is less susceptible to wind disturbance. In order to enhance the safety of the companion flying robot, we develop a new holonomic hexacopter that could move horizontally without roll and pitch motions. From control's point of view, since the holonomic hexacopter always stay at or very close to the equilibrium point (hovering state), it is more stable and therefore safer for HRI theoretically. We verified our conjecture in actual flight experiments. We also received comments from novice user that flying holonomic hexacopter is more intuitive than a conventional flying robot, because it can move horizontally naturally like a car without roll and pitch motions.

In addition to the safety considerations, we develop an accompanying model based on a hierarchical finite state machine (FSM). The hierarchical FSM serves as a top-level model to unify several important HRI behaviors such as human approaching, human following, human circling, human leading, and side-by-side walking. To date, most companion mobile robots are equipped with only one human accompanying mode and it is rare for a companion flying robot to perform more than one accompanying mode. Therefore, we believe that hierarchical FSM is useful for our companion flying robot to perform rich human accompanying behaviors. Combined with a developed bottom-level relative positioning controller, our holonomic hexacopter can have a smooth and natural HRI with its user.

Last but not least, a companion flying robot needs to understand its user to achieve a meaningful HRI. We study and improve several computer vision techniques that are

useful for a companion flying robot, including human detection and body orientation estimation, hand shapes detection, facial expression recognition, and face alignment. We aim to integrated all the investigated computer vision techniques; however, due to the limited onboard computational resources, we focus on using human detection and body orientation estimation in the actual flight experiments.

9.2 Discussion and Future Works

While we focus on three topics toward realizing our goal, companion flying robotics is a broad research theme that involves many topics. In this section, we further discuss the safety and regulation issues, as well as some potential future works of companion flying robots.

9.2.1 Safety Considerations

Developing a holonomic hexacopter for companion flying robot is one step towards a safer HRI. In this section, we discuss three other possible ways to enhance the flight safety.

First, one natural way to enhance the flight safety is to make a smaller and lighter aircraft. We built a bigger aircraft in the beginning with the purpose to mount additional sensors onto the aircraft for various experiments. Therefore, we could optimize the design by only considering the necessary components. Moreover, in the current holonomic configuration, the motors are mounted on the upper side of the frame for test convenience. Mounting the motors on the bottom side of the frame would make the propellers less exposed to the users and therefore enhance the flight safety.

Second, there are several complaints reported by commercial drone users that their drones erratically fly away. Without a physical and tangible failsafe, it is very dangerous for a companion flying robot to fly along user in public space. We believe that attaching a tether to a companion flying robot is a potential solution to this problem. Attaching a tether to the flying robot does not only prevent the companion flying robot from erratically flying away (and then potentially hit nearby people), but also able to enable longer flight by supplying power through the tether. Moreover, combining the concept of tethered power and companion flying robot enables new HRI, reminiscent of a human walking with his/her pet.

Third, as mentioned in **Section 1.3**, a rotor- and a balloon-type flying robot are complementary to each other—the rotor-type flying robot has the merits of higher mobility,

compactness, and less susceptible to wind disturbance while the bolloon-type flying robot has the merits of more silent, safer, and longer flight. While we focus on a rotor-type flying robot in this thesis, designing a hybrid flying robot, e.g., using small balloons to create partial lifting thrust for multirotors, could help to enhance flight safety. With balloons, smaller propulsion units can be adopted and lighter aircraft is possible. Moreover, with balloons, the impact of the flying robots during emergency landing could be greatly reduced.

9.2.2 UAV Regulations

It is said that the current state of flying robots is similar to the state of automobiles back in 1900, where their rapid advancements lead to new regulations and laws. In this section, we discuss several drone regulations in United States, European Union, and Japan. Knowing these emerging regulations is important as we can ensure the flight safety by using the regulations as a safety checklist. Note that UAV regulations is a relatively new and broad topic. As UAV technology become more mature, reliable, and autonomous, we believe that the authority will amend the existing laws accordingly.

In USA, the Federal Aviation Administration (FAA) announced in December of 2015 that all flying robots weighting in between 250 g and 25 kg must be registered online. The regulations are different for commercial and recreational drones; in general, flying a companion flying robot at outdoor must adopt to the following guidelines [315, 316]:

- the UAV must weigh less than 25 kg,
- the UAV must operate during daytime and remain within visual line-of-sight,
- the UAV must stay within ground speed of 160 km/h (100 mph),
- the UAV must stay within altitude of 120 m (400 feet),
- no flight is allowed near airports and crowded places,
- pilot must have a remote pilot airman certificate,

Aiming to encourage technology innovation, EU has more flexible regulations [317–319] compared to USA. Depending on the level of risk, EU has separate regulation for three different categories. In general, as long as:

- the drone is less than 25 kg,
- the flight is within direct visual line-of-sight (500 m),

- the flight is within 150 m (above the ground or water),
- the flight is outside of specified areas like airports,

the flight is considered as a low risk drone operations and neither flight approval nor pilot license are required, even for commercial operations. Literally, this means that a companion flying robot, which is highly unlikely weigh more than 25 kg, can be flown freely in EU, since it always accompanies the user (within visual line-of-sight and height of 3 m). Nevertheless, drones must comply with industry standard such as having adequate safety measures when it is flown in populated area.

Compared to USA, Japan also has more flexible drone regulations [320, 321]. Essentially, neither registration nor approval is required as long as the flight operation:

- is 9 km away from airports,
- is outside of populated area,
- is within flight altitude of 150 m
- is within visual line-of-sight,
- is during daytime,
- is 30 m away from people, buildings, or other objects,
- does not carry dangerous object,
- does not drop ojbect from the UAV,

Similar to the case in USA, flying a companion robot in Japan is prohibited unless the user is staying at rural areas.

9.2.3 Multiple Flying Robots Interaction

In this work, we consider the case of one companion flying robot. It would be interesting to use multiple flying robots for more comprehensive HRI. For example, imagine we have three companion flying robot interacting with one user together: the first flying robot can guide (lead) the user in new place, the second flying robot can help to determine the user's position and share the information wirelessly with other flying robots, while the third flying robot can have face-to-face and closer interaction with the user. In addition, the companion flying robots can also have interaction with each other.

Multiple flying robots also enable new autonomy challenges that are technically interesting. For example, with wireless information sharing, companion flying robots can potentially localize the user like a GPS system. With Bayesian filtering or other similar techniques, the user position estimated from the GPS-liked multiple flying robots system can also be fused with the position information estimated from a vision-based system. On the other hand, each flying robot also has to know the position of each other in order to avoid collision in the air. Knowing positions of all the flying robots also enhances the performance of collaborative mapping and 3D reconstruction.

Furthermore, the multiple flying robots can also equipped with different sensors for task cooperation and system optimization. In general, it would be hard or make the aircraft undesirably bigger if we mount all the sensors onto one flying robot. Instead, for example, we can mount a high resolution color camera onto the first flying robot, a depth camera or lidar sensor onto the second flying robot, and a thermal camera or high-speed camera onto the third flying robot. By doing so, we could separate the payloads to multiple flying robots and potentially achieve the tasks cooperatively with some dynamic planning algorithms.

Bibliography

[1] J. Zufferey, A. Klaptocz, A. Beyeler, J. Nicoud, and D. Floreano, "A 10-gram microflyer for vision-based indoor navigation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 3267–3272.

[2] J. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions on Robotics (T-RO)*, vol. 22, no. 1, pp. 137–146, February 2006.

[3] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics (T-RO)*, vol. 23, no. 3, pp. 519–529, June 2007.

[4] X. Deng, L. Schenato, W. C. Wu, and S. S. Sastry, "Flapping flight for biomimetic robotic insects: Part I—System modeling," *IEEE Transactions on Robotics (T-RO)*, vol. 22, no. 4, pp. 776–788, August 2006.

[5] X. Deng, L. Schenato, W. C. Wu, and S. S. Sastry, "Flapping flight for biomimetic robotic insects: Part II—Flight control design," *IEEE Transactions on Robotics (T-RO)*, vol. 22, no. 4, pp. 789–803, August 2006.

[6] R. J. Wood, "The first takeoff of a biologically inspired at-scale robotic insect," *IEEE Transactions on Robotics (T-RO)*, vol. 24, no. 2, pp. 1–7, April 2008.

[7] H. Helble and S. Cameron, "OATS: Oxford aerial tracking system," *Autonomous Robots*, vol. 55, no. 9, pp. 661–666, September 2007.

[8] AscTec Pelican. [Online]. Available: http://www.rc-airplane-world.com/rc-ufo.html

[9] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proc. Digital Avionics Systems Conference (DASC)*, October 2004, pp. 1–10.

[10] D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz," in *Proc. IEEE*

International Conference on Robotics and Automation (ICRA), April 2007, pp. 361–366.

[11] DJI - The World Leader in Camera Drones/Quadcopters for Aerial Photography. [Online]. Available: http://www.dji.com/

[12] 3DR - Drone & UAV Technology. [Online]. Available: https://3dr.com/

[13] DJI Phantom (UAV). [Online]. Available: https://en.wikipedia.org/wiki/Phantom_(UAV)

[14] DJI Phantom Quadcopter. [Online]. Available: http://www.dji.com/product/matrice100/

[15] DJI Phantom Octocopter. [Online]. Available: http://www.dji.com/product/mg-1

[16] 3DR Solo Quadcopter. [Online]. Available: https://3dr.com/solo-drone/

[17] 3DR Pixhawk Flight Controller. [Online]. Available: https://store.3dr.com/products/3dr-pixhawk

[18] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, "Smartphones power flying robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1256–1263.

[19] Amazon Prime Air - Drone-based delivery system. [Online]. Available: http://www.amazon.com/primeair/

[20] Pix4D - Drone mapping software. [Online]. Available: https://pix4d.com/

[21] Secom to start security service using drones on Friday. [Online]. Available: http://www.japantimes.co.jp/news/2015/12/10/business/corporate-business/secom-start-security-service-using-drones-friday/

[22] Fire rescue drones. [Online]. Available: http://www.skyfireconsulting.com/

[23] Industrial & civil infrastructure inspection. [Online]. Available: http://www.asctec.de/en/industrial-civil-infrastructure-inspection/

[24] Volunteer Search & Rescue Network. [Online]. Available: http://sardrones.org/

[25] Parrot AR.Drone 2.0. [Online]. Available: http://www.parrot.com/products/ardrone-2/

[26] An easy educational drone kit for learning robotics! [Online]. Available: https://www.kickstarter.com/projects/dronesmith/eedutm-an-easy-educational-drone-kit-for-learning/

[27] Premier drone racing. [Online]. Available: http://thedroneracingleague.com/

[28] E. Graether and F. Mueller, "Joggobot: A flying robot as jogging companion," in *Proc. CHI Extended Abstracts on Human Factors in Computing Systems*, May 2012, pp. 1063–1066.

[29] M. Cooney, F. Zanlungo, S. Nishio, and H. Ishiguro, "Designing a flying humanoid robot (FHR): Effects of flight on interactive communication," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, September 2012, pp. 364–371.

[30] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli, "Vision based GPS-denied object tracking and following for unmanned aerial vehicles," in *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, October 2013, pp. 1–6.

[31] K. Higuchi, Y. Ishiguro, and J. Rekimoto, "Flying eyes: Free-space content creation using autonomous aerial vehicles," in *Proc. CHI Extended Abstracts on Human Factors in Computing Systems*, May 2011, pp. 561–570.

[32] C. Papachristos, D. Tzoumanikas, and A. Tzes, "Aerial robotic tracking of a generalized mobile target employing visual and spatio-temporal dynamic subject perception," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 4319–4324.

[33] Lily Camera. [Online]. Available: https://www.lily.camera/

[34] Hover Camera. [Online]. Available: http://gethover.com/

[35] Parrot Bebop Drone. [Online]. Available: http://www.parrot.com/products/bebop-drone/

[36] DJI Phantom 4. [Online]. Available: http://www.dji.com/product/phantom-4

[37] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a quadrotor MAV," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2182–2189.

[38] T. Naseer, J. Sturm, and D. Cremers, "FollowMe: Person following and gesture recognition with a quadrocopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 624–630.

[39] T. Higuchi, D. Toratani, M. Machida, and S. Ueno, "Guidance and control of double tetrahedron hexa-rotorcraft," in *Proc. AIAA Guidance, Navigation, and Control Conference*, August 2012, pp. 1–10.

[40] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics (T-RO)*, vol. 28, no. 1, pp. 90–100, February 2012.

[41] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 33–45, September 2012.

[42] A. Kalantari and M. Spenko, "Design and experimental validation of HyTAQ, a hybrid terrestrial and aerial quadrotor," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4445–4450.

[43] K. Kawasaki, M. Zhao, K. Okada, and M. Inaba, "MUWA: Multi-field universal wheel for air-land vehicle with quad variable-pitch propellers," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1880–1885.

[44] S. Mizutani, Y. Okada, C. J. Salaan, T. Ishii, K. Ohno, and S. Tadokoro, "Proposal and experimental validation of a design strategy for a UAV with a passive rotating spherical shell," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1271–1278.

[45] C. F. Liew and T. Yairi, "Designing a compact hexacopter with gimballed lidar and powerful onboard Linux computer," in *Proc. IEEE International Conference on Information and Automation (ICIA)*, August 2015, pp. 2523–2528.

[46] C. F. Liew and T. Yairi, "Towards a compact and autonomous hexacopter for human robot interaction," in *Proc. Annual Conference of Robotics Society of Japan (RSJ)*, September 2015, pp. 1–4.

[47] C. T. Raabe, "Failure-tolerant control and vision-based navigation for hexacopters," Ph.D. dissertation, Department of Aeronautics and Astronautics, The University of Tokyo, Japan, 2013. [Online]. Available: http://repository.dl.itc.u-tokyo.ac.jp/dspace/handle/2261/57479

[48] S. Huh, D. H. Shim, and J. Kim, "Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3158–3163.

[49] S. Salazar, H. Romero, R. Lozano, and P. Castillo, "Modeling and real-time stabilization of an aircraft having eight rotors," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1, pp. 455–470, March 2009.

[50] H. Romero, S. Salazar, and R. Lozano, "Real-time stabilization of an eight-rotor UAV using optical flow," *IEEE Transactions on Robotics (T-RO)*, vol. 25, no. 4, pp. 809–817, August 2009.

[51] A. Klaptocz, G. Boutinard-Rouelle, A. Briod, J. C. Zufferey, and D. Floreano, "An indoor flying platform with collision robustness and self-recovery," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3349–3354.

[52] A. Briod, P. Kornatowski, A. Klaptocz, A. Garnier, M. Pagnamenta, J. C. Zufferey, and D. Floreano, "Contact-based navigation for an autonomous flying robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3987–3992.

[53] J. Paulos and M. Yim, "Flight performance of a swashplateless micro air vehicle," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5284–5289.

[54] J. Paulos and M. Yim, "An underactuated propeller for attitude control in micro air vehicles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1374–1379.

[55] S. Bouabdallah, R. Siegwart, and G. Caprari, "Design and control of an indoor coaxial helicopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 2930–2935.

[56] W. Wang, G. Song, K. Nonami, M. Hirata, and O. Miyazawa, "Autonomous control for micro-flying robot and small wireless helicopter x.r.b," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 2906–2911.

[57] P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robust stabilization control of unknown small-scale helicopters," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 537–542.

[58] P. Y. Oh, M. Joyce, and J. Gallagher, "Designing an aerial robot for hover-and-stare surveillance," in *Proc. IEEE International Conference on Advanced Robotics (ICAR)*, July 2014, pp. 303–308.

[59] K. Kawasaki, Y. Motegi, M. Zhao, K. Okada, and M. Inaba, "Dual connected bi-copter with new wall trace locomotion feasibility that can fly at arbitrary tilt angle," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 524–531.

[60] S. Driessens and P. E. I. Pounds, "Towards a more efficient quadrotor configuration," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1386–1392.

[61] C. Papachristos, K. Alexis, and A. Tzes, "Model predictive hovering-translation control of an unmanned tri-tiltrotor," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 5425–5432.

[62] L. Daler, J. Lecoeur, P. B. Hählen, and D. Floreano, "A flying robot with adaptive morphology for multi-modal locomotion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1361–1366.

[63] R. Bapst, R. Ritz, L. Meier, and M. Pollefeys, "Design and implementation of an unmanned tail-sitter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1885–1890.

[64] P. E. I. Pounds and S. P. N. Singh, "Integrated electro-aeromechanical structures for low-cost, self-deploying environment sensors and disposable UAVs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4459–4466.

[65] M. Orsag, S. Bogdan, T. Haus, M. Bunic, and A. Krnjak, "Modeling, simulation and control of a spincopter," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2998–3003.

[66] L. Hines, D. Colmenares, and M. Sitti, "Platform design and tethered flight of a motor-driven flapping-wing system," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5838–5845.

[67] J. H. Park, E. P. Yang, C. Zhang, and S. K. Agrawal, "Kinematic design of an asymmetric in-phase flapping mechanism for MAVs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 5099–5104.

[68] Y. W. Chin, J. T. W. Goh, and G. K. Lau, "Insect-inspired thoracic mechanism with non-linear stiffness for flapping-wing micro air vehicles," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3544–3549.

[69] M. Hamamoto, H. Etoh, and T. Miyake, "Thorax unit driven by unidirectional USM for under 10-gram flapping MAV platform," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 2142–2147.

[70] K. Peterson and R. S. Fearing, "Experimental dynamics of wing assisted running for a bipedal ornithopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 5080–5086.

[71] M. Burri, L. Gasser, M. Käch, M. Krebs, S. Laube, A. Ledergerber, D. Meier, R. Michaud, L. Mosimann, L. Müri, C. Ruch, A. Schaffner, N. Vuilliomenet, J. Weichart, K. Rudin, S. Leutenegger, J. Alonso-Mora, R. Siegwart, and P. Beardsley,

"Design and control of a spherical omnidirectional blimp," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1873–1879.

[72] Y. Yang, I. Sharf, and J. R. Forbes, "Nonlinear optimal control of holonomic indoor airship," in *Proc. AIAA Guidance, Navigation, and Control Conference*, August 2012, pp. 1–12.

[73] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 5, pp. 664–674, April 2012.

[74] M. Hehn and R. D'Andrea, "A frequency domain iterative feed-forward learning scheme for high performance periodic quadrocopter maneuvers," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 2445–2451.

[75] A. E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero, "Aerial manipulator for structure inspection by contact from the underside," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1879–1884.

[76] G. Heredia, A. E. Jimenez-Cano, I. Sanchez, V. V. D. Llorente, J. Braga, J. Á. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014, pp. 3465–3471.

[77] M. Turpin, N. Michael, and V. Kumar, "Trajectory design and control for aggressive formation flight with quadrotors," *Autonomous Robots*, vol. 33, no. 1, pp. 143–156, August 2012.

[78] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 763–770.

[79] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3472–3479.

[80] W. Dong, G. Y. Gu, Y. Ding, X. Zhu, and H. Ding, "Ball juggling with an under-actuated flying robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 68–73.

[81] R. Ritz, M. W. Muller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 4972–4978.

[82] H. N. Nguyen, S. Park, and D. Lee, "Aerial tool operation system using quadrotors as rotating thrust generators," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1285–1291.

[83] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3465–3471.

[84] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1722–1729.

[85] I. Dryanovski, R. G. Valenti, and J. Xiao, "An open-source navigation system for micro aerial vehicles," *Autonomous Robots*, vol. 34, no. 3, pp. 177–188, April 2013.

[86] K. Schmid, T. Tomić, F. Ruess, H. Hirschmüller, and M. Suppa, "Stereo vision based indoor/outdoor navigation for flying robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3955–3962.

[87] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 2815–2821.

[88] M. Warren, P. Corke, and B. Upcroft, "Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles," *The International Journal of Robotics Research (IJRR)*, vol. 35, no. 4, pp. 381–403, April 2016.

[89] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 2815–2821.

[90] H. Lim, S. N. Sinha, M. F. Cohen, M. Uyttendaele, and H. J. Kim, "Real-time monocular image-based 6-DoF localization," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 476–492, April 2015.

[91] A. Wendel, A. Irschara, and H. Bischof, "Natural landmark-based monocular localization for MAVs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 5792–5799.

[92] Z. Fang and S. Scherer, "Real-time onboard 6DoF localization of an indoor MAV in degraded visual environments using a RGB-D camera," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5253–5259.

[93] R. Konomura and K. Hori, "Visual 3D self localization with 8 gram circuit board for very compact and fully autonomous unmanned aerial vehicles," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5215–5220.

[94] D. T. Cole, P. Thompson, A. H. Göktoğan, and S. Sukkarieh, "System development and demonstration of a cooperative UAV team for mapping and tracking," *The International Journal of Robotics Research (IJRR)*, vol. 29, no. 11, pp. 1371–1399, September 2010.

[95] D. Holz and S. Behnke, "Registration of non-uniform density 3D laser scans for mapping with micro aerial vehicles," *Robotics and Autonomous Systems*, vol. 74, pp. 318–330, December 2015.

[96] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1872–1878.

[97] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of MAVs using RGB-D sensors," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4021–4028.

[98] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 7, pp. 969–1002, June 2015.

[99] S. phane Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive UAV control in cluttered natural environments," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 1765–1772.

[100] S. Roelofsen, D. Gillet, and A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 4810–4817.

[101] J. Müller and G. S. Sukhatme, "Risk-aware trajectory generation with application to safe quadrotor landing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014, pp. 3642–3648.

[102] F. Mueller and M. Muirhead, "Jogging with a quadcopter," in *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, April 2015, pp. 2023–2032.

[103] K. Nitta, K. Higuchi, and J. Rekimoto, "HoverBall: Augmented sports with a flying ball," in *Proc. Augmented Human International Conference*, March 2014, pp. 13:1–13:4.

[104] SPARKED: A Live Interaction Between Humans and Quadcopters. [Online]. Available: https://www.youtube.com/watch?v=6C8OJsHfmpI

[105] SPARKED: Behind the Technology. [Online]. Available: https://www.youtube.com/watch?v=7YqUocVcyrE

[106] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[107] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 617–623.

[108] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann, "A prototyping environment for interaction between a human and a robotic multi-agent system," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2012, pp. 185–186.

[109] V. M. Monajjemi, S. Pourmehr, S. A. Sadat, F. Zhan, J. Wawerla, G. Mori, and R. Vaughan, "Integrating multi-modal interfaces to command UAVs," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2014, p. 106.

[110] V. M. Monajjemi, J. Bruce, S. A. Sadat, J. Wawerla, and R. Vaughan, "UAV, do you see me? Establishing mutual attention between an uninstrumented human and an outdoor UAV in flight," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 3614–3620.

[111] G. Costante, E. Bellocchio, P. Valigi, and E. Ricci, "Personalizing vision-based gestural interfaces for HRI with UAVs: A transfer learning approach," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014, pp. 3319–3326.

[112] M. Burke and J. Lasenby, "Pantomimic gestures for human-robot interaction," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 5, pp. 1225–1237, October 2015.

[113] A. S. Huang, S. Tellex, A. Bachrach, T. Kollar, D. Roy, and N. Roy, "Natural language command of an autonomous micro-air vehicle," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2010, pp. 2663–2669.

[114] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, July 2011, pp. 143–149.

[115] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "An exploration into natural human-drone interaction," in *Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, September 2015, pp. 361–365.

[116] M. Sharma, D. Hildebrandt, G. Newman, J. E. Young, and R. Eskicioglu, "Communicating affect via flight path: Exploring use of the laban effort system for designing affective locomotion paths," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2013, pp. 293–300.

[117] D. Szafir, B. Mutlu, and T. Fong, "Communication of intent in assistive free flyers," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2014, pp. 358–365.

[118] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A. Landay, "Emotion encoding in human-drone interaction," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2016, pp. 263–270.

[119] D. Arroyo, C. Lucho, J. Roncal, and F. Cuellar, "Daedalus: A sUAV for social environments," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2014, p. 99.

[120] D. Szafir, B. Mutlu, and T. Fong, "Communicating directionality in flying robots," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2015, pp. 19–26.

[121] J. L. Drury, L. Riek, and N. Rackliffe, "A decomposition of UAV-related situation awareness," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2014, pp. 88–94.

[122] R. R. Murphy, K. S. Pratt, and J. L. Burke, "Crew roles and operational protocols for rotary-wing micro-UAVs in close urban environments," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2008, pp. 73–80.

[123] R. R. Murphy, "Proposals for new UGV, UMV, UAV, and HRI standards for rescue robots," in *Proc. Performance Metrics for Intelligent Systems Workshop (PerMIS)*, September 2010, pp. 9–13.

[124] B. S. Morse, C. H. Engh, and M. A. Goodrich, "UAV video coverage quality maps and prioritized indexing for wilderness search and rescue," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2010, pp. 227–234.

[125] Parrot AR.Drone. [Online]. Available: https://en.wikipedia.org/wiki/Parrot_AR.Drone

[126] Aibot X6. [Online]. Available: https://www.aibotix.com

[127] Snap drone. [Online]. Available: https://vantagerobotics.com/

[128] Fleye flying robot. [Online]. Available: http://gofleye.com/

[129] D. Johnson, P. Papadopoulos, N. Watfa, and J. Takala, "Exposure criteria: Ooccupational exposure levels," in *Occupational exposure to noise: Evaluation, prevention and control*, B. Goelzer, C. H. Hansen, and G. A. Sehrndt, Eds. Federal Institute for Occupational Safety and Health, 2001, ch. 4, pp. 79–102.

[130] T. Bodin, M. Albin, J. Ardö, E. Stroh, P. Östergren, and J. Björk, "Road traffic noise and hyper tension: Results from a cross-sectional public health survey in southern Sweden," *Environmental Health*, vol. 8, 2009.

[131] European Agency for Safety and Health at Work. What problems can noise cause? [Online]. Available: http://osha.europa.eu/en/topics/noise/

[132] C. F. Liew and T. Yairi, "Quadrotor or blimp? Noise and appearance considerations in designing social aerial robot," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2013, pp. 183–184.

[133] SurveyMonkey. [Online]. Available: https://www.surveymonkey.com

[134] H. Jung, Y. L. Altieri, and J. Bardzell, "SKIN: Designing aesthetic interactive surfaces," in *Proc. International Conference on Tangible, Embedded, and Embodied Interaction (TEI)*, January 2010, pp. 85–92.

[135] G. Jiang and R. Voyles, "Mock-up of the exhaust shaft inspection by dexterous hexrotor at the DOE WIPP site," in *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, October 2015, pp. 1–2.

[136] G. Jiang and R. Voyles, "A nonparallel hexrotor uav with faster response to disturbances for precision position keeping," in *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, October 2014, pp. 1–2.

[137] R. Voyles and G. Jiang, "Hexrotor UAV platform enabling dextrous interaction with structures — Preliminary work," in *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, November 2012, pp. 1–7.

[138] B. Crowther, A. Lanzon, M. Maya-Gonzalez, and D. Langkamp, "Kinematic analysis and control design for a nonplanar multirotor vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1157–1171, July 2011.

[139] CyPhy LVL 1. [Online]. Available: http://www.cyphyworks.com/robots/lvl1/

[140] S. Rajappa, M. Ryll, H. H. Bülthoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4006–4013.

[141] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *Proc. IEEE International Conference on Robotics and Automation (ICRA) (in press)*, May 2016, pp. 1–6.

[142] M. Ramp and E. Papadopoulos, "On modeling and control of a holonomic vectoring tricopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 662–668.

[143] Y. Long and D. J. Cappelleri, "Complete dynamic modeling, control and optimization for an over-actuated MAV," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1380–1385.

[144] Y. Long and D. J. Cappelleri, "Linear control design, allocation, and implementation for the omnicopter MAV," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 289–294.

[145] A. Oosedo, S. Abiko, S. Narasaki, A. Kuno, A. Konno, and M. Uchiyama, "Flight control systems of a quad tilt rotor unmanned aerial vehicle for a large attitude change,"

in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2326–2331.

[146] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "First flight tests for a quadrotor UAV with tilting propellers," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 295–302.

[147] Festo Free Motion Handling - Autonomously flying gripping sphere. [Online]. Available: https://www.festo.com/group/en/cms/11957.htm

[148] Test review DJI E800 Tuned Propulsion: Sinewave (R) evolution of the middle class? [Online]. Available: http://www.rcgroups.com/forums/showthread.php?t=2408036

[149] iNav Open-Source Flight Controller Software. [Online]. Available: https://github.com/iNavFlight/inav

[150] iNav - Configurator GUI. [Online]. Available: https://github.com/iNavFlight/inav-configurator

[151] Control System Design, Lecture Notes for ME 155A by Karl Johan Åström (p.228). [Online]. Available: http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch6.pdf

[152] C. Reynolds, "Steering behaviors for autonomous characters," *Game Developers Conference*, pp. 763–782, 1999.

[153] Learning Guides - Understanding Steering Behaviors. [Online]. Available: https://www.skydio.com/

[154] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, E. A. Sisbot, R. Alami, and T. Siméon, "How may I serve you? A robot companion approaching a seated person in a helping context," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2006, pp. 172–179.

[155] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "A robot that approaches pedestrians," *IEEE Transactions on Robotics (T-RO)*, vol. 29, no. 2, pp. 508–524, April 2013.

[156] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans? - Strategies for social robots to initiate interaction," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2009, pp. 109–116.

[157] T. Kanda, D. F. Glas, M. Shiomi, and N. Hagita, "Abstracting people's trajectories for social robots to proactively approach customers," *IEEE Transactions on Robotics (T-RO)*, vol. 25, no. 6, pp. 1382–1396, December 2009.

[158] S. Satake, K. Hayashi, K. Nakatani, and T. Kanda, "Field trial of an information-providing robot in a shopping mall," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2015, pp. 1832–1839.

[159] Y. Kato, T. Kanda, and H. Ishiguro, "May I help you? - Design of human-like polite approaching behavior," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2015, pp. 35–42.

[160] P. Ratsamee, Y. Mae, K. Ohara, M. Kojima, and T. Arai, "Social navigation model based on human intention analysis using face orientation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1682–1687.

[161] A. Cosgun, D. A. Florencio, and H. I. Christensen, "Autonomous person following for telepresence robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4335–4342.

[162] E. J. Jung, J. H. Lee, B. J. Yi, I. H. Suh, S. Yuta, and S. T. Noh, "Marathoner tracking algorithms for a high speed mobile robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 3595–3600.

[163] A. Tani, G. Endo, E. F. Fukushima, S. Hirose, M. Iribe, and T. Takubo, "Study on a practical robotic follower to support home oxygen therapy patients - Development and control of a mobile platform," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 2423–2429.

[164] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2007, pp. 17–24.

[165] C. Granata and P. Bidaud, "A framework for the design of person following behaviors for social mobile robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 4652–4659.

[166] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2D laser scanners," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 726–733.

[167] S. Hemachandra, T. Kollar, N. Roy, and S. Teller, "Following and interpreting narrated guided tours," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2574–2579.

[168] N. Tsuda, S. Harimoto, T. Saitoh, and R. Konishi, "Mobile robot with following and returning mode," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, September 2009, pp. 933–938.

[169] J. J. Park and B. Kuipers, "Autonomous person pacing and following with model predictive equilibrium point control," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 1060–1067.

[170] Y. Morales, T. Kanda, and N. Hagita, "Walking together: Side-by-side walking model for an interacting robot," *Journal of Human-Robot Interaction*, vol. 3, no. 2, pp. 50–73, 2014.

[171] Y. Morales, S. Satake, R. Huq, D. Glas, T. Kanda, and N. Hagita, "How do people walk side-by-side? - Using a computational model of human behavior for a social robot," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2012, pp. 301–308.

[172] R. Murakami, Y. Morales, S. Satake, T. Kanda, and H. Ishiguro, "Destination unknown: Walking side-by-side without knowing the goal," in *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2014, pp. 471–478.

[173] D. Pucci, L. Marchetti, and P. Morin, "Nonlinear control of unicycle-like robots for person following," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 3406–3411.

[174] Y. Kobayashi, R. Suzuki, and Y. Kuno, "Robotic wheelchair with omni-directional vision for moving alongside a caregiver," in *Proc. IEEE Industrial Electronics Society Annual Conference (IECON)*, October 2012, pp. 4177–4182.

[175] E. J. Jung, B. J. Yi, and S. Yuta, "Control algorithms for a mobile robot tracking a human in front," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 2411–2416.

[176] A. Garrell and A. Sanfeliu, "Cooperative social robots to accompany groups of people," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 13, pp. 1675–1701, November 2012.

[177] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1688–1694.

[178] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014, pp. 1730–1735.

[179] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a person following robot with vision based target detection," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 5286–5291.

[180] J. B. Yoram Koren, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, April 1991, pp. 1398–1404.

[181] Learning Guides - Understanding Steering Behaviors. [Online]. Available: http://gamedevelopment.tutsplus.com/series/understanding-steering-behaviors--gamedev-12732

[182] Processing - A Flexible Software Sketchbook and a Language for Learning How to Code within the Context of the Visual Arts. [Online]. Available: https://processing.org/

[183] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a person following robot with vision based target detection," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006, pp. 5286–5291.

[184] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a quadrotor MAV," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2182–2189.

[185] C. Weinrich, C. Vollmer, and H. M. Gross, "Estimation of human upper body orientation for mobile robotics using an SVM decision tree on monocular images," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 2147–2152.

[186] I. Ardiyanto and J. Miura, "Partial least squares-based human upper body orientation estimation with combined detection and tracking," *Image and Vision Computing (IVC)*, vol. 32, no. 11, pp. 904–915, November 2014.

[187] F. Flohr, M. Dumitru-Guzu, J. F. P. Kooij, and D. M. Gavrila, "Joint probabilistic pedestrian head and body orientation estimation," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, June 2014, pp. 617–622.

[188] C. Chen, A. Heili, and J. M. Odobez, "Combined estimation of location and body pose in surveillance video," in *Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, August 2011, pp. 5–10.

[189] C. Chen and J. M. Odobez, "We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1544–1551.

[190] L. Rybok, M. Voit, H. K. Ekenel, and R. Stiefelhagen, "Multi-view based estimation of human upper-body orientation," in *Proc. 20th International Conference on Pattern Recognition (ICPR)*, August 2010, pp. 1558–1561.

[191] G. Yang, M. Iwabuchi, and K. Nakamura, "Real-time upper-body detection and orientation estimation via depth cues for assistive technology," in *Proc. IEEE Symposium on Computational Intelligence in Rehabilitation and Assistive Technologies (CIRAT)*, April 2013, pp. 13–18.

[192] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 12, pp. 2821–2840, December 2013.

[193] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3d pose estimation from a single depth image," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, November 2011, pp. 731–738.

[194] H. Y. Jung, S. Lee, Y. S. Heo, and I. D. Yun, "Random tree walk toward instantaneous 3d human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 2467–2474.

[195] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012, pp. 2389–2395.

[196] W. Liu, Y. Zhang, S. Tang, J. Tang, R. Hong, and J. Li, "Accurate estimation of human body orientation from RGB-D sensors," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1442–1452, October 2013.

[197] F. Shinmura, D. Deguchi, I. Ide, H. Murase, and H. Fujiyoshi, "Estimation of human orientation using coaxial RGB-Depth images," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, March 2015, pp. 113–120.

[198] D. F. Glas, T. Miyashita, H. Ishiguro, and N. Hagita, "Laser tracking of human body motion using adaptive shape modeling," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2007, pp. 602–608.

[199] Y. Kobayashi, Y. Kinpara, T. Shibusawa, and Y. Kuno, "Robotic wheelchair based on observations of people using integrated sensors," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2009, pp. 2013–2018.

[200] J. Shackleton, B. VanVoorst, and J. Hesch, "Tracking people with a 360-degree lidar," in *Proc. IEEE International Conference on Computer Vision Theory and Applications (VISAPP)*, August 2010, pp. 420–426.

[201] X. Shao, H. Zhao, K. Nakamura, K. Katabira, R. Shibasaki, and Y. Nakagawa, "Detection and tracking of multiple pedestrians by using laser range scanners," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2007, pp. 2174–2179.

[202] J. Ziegler, H. Kretzschmar, C. Stachniss, G. Grisetti, and W. Burgard, "Detection and tracking of multiple pedestrians by using laser range scanners," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 86–91.

[203] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 886–893.

[204] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, August 1997.

[205] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006, pp. 1491–1498.

[206] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2001, pp. 511–518.

[207] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[208] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE International Conference on Rehabilitation Robotics*, June 2011, pp. 1–7.

[209] C. F. Liew and T. Yairi, "A new RGB-Depth hand shape database for natural gesture interaction," in *Proc. 16th Meeting on Image Recognition and Understanding (MIRU)*, June 2013, pp. 1–2.

[210] M. Donoser and H. Bischof, "Real time appearance based hand tracking," in *Proc. 19th International Conference on Pattern Recognition (ICPR)*, December 2008, pp. 1–4.

[211] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1862–1869.

[212] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *Proc. British Machine Vision Conference (BMVC)*, September 2011, pp. 1–11.

[213] V. Spruyt, A. Ledda, and S. Geerts, "Real-time multi-colourspace hand segmentation," in *Proc. IEEE Conference on Image Processing (ICIP)*, September 2010, pp. 1862–1869.

[214] A. A. Argyros and M. I. A. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," in *Proc. 8th European Conference on Computer Vision (ECCV)*, May 2004, pp. 368–379.

[215] M. I. Boulabiar, T. Burger, F. Poirier, and G. Coppin, "A low-cost natural user interaction based on a camera hand-gestures recognizer," in *Proc. 14th International Conference on Human-Computer Interaction (HCI)*, July 2011, pp. 368–379.

[216] C. Zhang, X. Yang, and Y. Tian, "Histogram of 3D facets: A characteristic descriptor for hand gesture recognition," in *Proc. 10th International Conference on Automatic Face and Gesture Recognition (FG)*, April 2013, pp. 1–8.

[217] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proc. 19th ACM International Conference on Multimedia (MM)*, November 2011, pp. 1093–1096.

[218] H. Liang, J. Yuan, and D. Thalmann, "3D fingertip and palm tracking in depth image sequences," in *Proc. 20th ACM International Conference on Multimedia (MM)*, October 2012, pp. 785–788.

[219] A. Mittal, A. Zisserman, and P. Torr, "Hand detection using multiple proposals," in *Proc. British Machine Vision Conference (BMVC)*, September 2011, pp. 1–11.

[220] E. J. Ong and R. Bowden, "A boosted classifier tree for hand shape detection," in *Proc. 6th International Conference on Automatic Face and Gesture Recognition (FG)*, May 2004, pp. 889–894.

[221] M. Kölsch and M. Turk, "Robust hand detection," in *Proc. 6th International Conference on Automatic Face and Gesture Recognition (FG)*, May 2004, pp. 614–619.

[222] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958–1970, July 2007.

[223] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 7, pp. 1281–1298, July 2012.

[224] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face representation via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 2, pp. 210–227, February 2009.

[225] C. F. Liew, "Machine learning techniques for facial expression recognition - A comparison study of feature spaces and classification methods," Master's thesis, Department of Aeronautics and Astronautics, The University of Tokyo, Japan, 2013.

[226] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers*, vol. 93, no. 26, pp. 429–457, November 1946.

[227] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, January 1998, pp. 555–562.

[228] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. 12nd International Conference on Pattern Recognition (ICPR)*, October 1994, pp. 582–585.

[229] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 3, pp. 448–461, March 2010.

[230] C. F. Liew and T. Yairi, "A comparison study of feature spaces and classification methods in facial expression recognition," in *Proc. International Conference on Robotics and Biomimetics (ROBIO)*, December 2013, pp. 1294–1299.

[231] O. Rudovic, I. Patras, and M. Pantic, "Regression-based multi-view facial expression recognition," in *Proc. 20nd International Conference on Pattern Recognition (ICPR)*, August 2010, pp. 4121–4124.

[232] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing (IVC)*, vol. 18, no. 11, pp. 881–905, November 2000.

[233] A. Asthana, J. Saragih, M. Wagner, and R. Goecke, "Evaluating AAM fitting methods for facial expression recognition," in *Proc. 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops (ACII-W)*, September 2009, pp. 1–8.

[234] Y. Li, S. Wang, , Y. Zhao, and Q. Ji, "Simultaneous facial feature tracking and facial expression recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2559–2573, July 2013.

[235] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image and Vision Computing (IVC)*, vol. 27, no. 6, pp. 803–816, May 2009.

[236] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer

perceptron," in *Proc. 3rd International Conference on Automatic Face and Gesture Recognition (FG)*, April 1998, pp. 454–459.

[237] I. Gonzalez, H. Sahli, V. Enescu, and W. Verhelst, "Context-independent facial action unit recognition using shape and Gabor phase information," in *Proc. 4th International Conference on Affective Computing and Intelligent Interaction (ACII)*, October 2011, pp. 548–557.

[238] Y. li Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 2, pp. 97–115, February 2001.

[239] H. Sadeghi, A. A. Raie, and M. R. Mohammadi, "Facial expression recognition using geometric normalization and appearance representation," in *Proc. 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, September 2013, pp. 159–163.

[240] S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," *IEEE Transactions on Affective Computing*, vol. 6, no. 1, pp. 1–12, January 2015.

[241] W. Zheng, "Multi-view facial expression recognition based on group sparse reduced-rank regression," *IEEE Transactions on Affective Computing*, vol. 5, no. 1, pp. 71–85, January 2014.

[242] S. Eleftheriadis, O. Rudovic, and M. Pantic, "Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 189–204, January 2015.

[243] M. Lyons, M. Schuster, and S. Akamatsu, "Coding facial expressions with Gabor wavelets," in *Proc. 3rd International Conference on Automatic Face and Gesture Recognition (FG)*, April 1998, pp. 200–205.

[244] T. Wu, M. S. Bartlett, and J. R. Movellan, "Facial expression recognition using Gabor motion energy filters," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (CVPR-W)*, June 2010, pp. 42–47.

[245] M. S. Bartlett, G. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan, "Automatic recognition of facial actions in spontaneous expressions," *Journal of Multimedia (JMM)*, vol. 1, no. 6, pp. 22–35, September 2006.

[246] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Fully automatic facial action recognition in spontaneous behavior," in *Proc. 7th International Conference on Automatic Face and Gesture Recognition (FG)*, April 2006, pp. 223–228.

[247] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan, "Dynamics of facial expression extracted automatically from video," *Image and Vision Computing (IVC)*, vol. 24, no. 6, pp. 615–625, June 2006.

[248] C. C. Lee and C. Y. Shih, "Gabor feature selection for facial expression recognition," in *Proc. International Conference on Signals and Electronic Systems (ICSES)*, September 2010, pp. 139–142.

[249] C. Xu, C. Dong, Z. Feng, and T. Cao, "Facial expression pervasive analysis based on Haar-Like features and SVM," in *Proc. International Conference on E-business Technology and Strategy (iCETS)*, August 2012, pp. 521–529.

[250] S. U. Jung, D. H. Kim, K. H. An, and M. J. Chung, "Efficient rectangle feature extraction for real-time facial expression recognition based on AdaBoost," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, August 2005, pp. 1941–1946.

[251] G. M. Nagi, R. Rahmat, F. Khalid, and M. Taufik, "Region-based facial expression recognition in still images," *Journal of Information Processing Systems (JIPS)*, vol. 9, no. 1, pp. 173–188, March 2013.

[252] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based HoG features," in *Proc. 6th International Conference on Automatic Face and Gesture Recognition and Workshops (FG-W)*, March 2011, pp. 884–888.

[253] C. Orrite, A. Gañán, and G. Rogez, "HOG-based decision tree for facial expression classification," in *Proc. 4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, June 2009, pp. 176–183.

[254] S. Chen, Y. Tian, Q. Liu, and D. N. Metaxas, "Recognizing expressions from face and body gesture by temporal normalized motion and appearance features," *Image and Vision Computing (IVC)*, vol. 31, no. 2, pp. 175–185, February 2013.

[255] A. Bosch, A. Zisserman, and X. Muñoz, "Image classifcation using random forests and ferns," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, October 2007, pp. 1–8.

[256] A. Moeini, H. Moeini, and K. Faez, "Pose-invariant facial expression recognition based on 3D face reconstruction and synthesis from a single 2D image," in *Proc. 22nd International Conference on Pattern Recognition (ICPR)*, August 2014, pp. 1746–1751.

[257] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.

[258] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[259] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. 4th International Conference on Automatic Face and Gesture Recognition (FG)*, March 2000, pp. 46–53.

[260] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (CVPR-W)*, June 2010, pp. 94–101.

[261] N. Aifanti, C. Papachristou, and A. Delopoulos, "The MUG facial expression database," in *Proc. 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, April 2010, pp. 1–4.

[262] D. Lundqvist, A. Flykt, and A. Öhman, "The Karolinska Directed Emotional Faces - KDEF," in *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 1998.

[263] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 12, pp. 1357–1362, December 1999.

[264] "MATLAB R2013a Documentation - Face Detection and Tracking," http://www.mathworks.com/help/vision/examples/face-detection-and-tracking.html, [Accessed: 22th July 2013].

[265] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[266] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America (JOSA)*, vol. 2, no. 7, pp. 1160–1169, July 1985.

[267] J. G. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, January 2004.

[268] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Proc. IEEE Conference on Image Processing (ICIP)*, October 2001, pp. 282–285.

[269] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with Local Binary Patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 12, pp. 2037–2041, December 2006.

[270] Online resource, downloaded from http://www.flickr.com/photos/58842866@N08/5388738458/sizes/o/in/photostream/ under Attribution Creative Commons License on July 25, 2013.

[271] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, September 1999, pp. 1150–1157.

[272] Online resource, downloaded from http://www.flickr.com/photos/davedehetre/5328057917/sizes/o/in/photostream/ under Attribution Creative Commons License on July 25, 2013.

[273] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 9, pp. 1465–1479, September 2006.

[274] Online resource, downloaded from http://www.fotopedia.com/items/flickr-3714021304/ under Attribution and ShareAlike Creative Commons License on July 25, 2013.

[275] K. P. Murphy, "Machine learning: A probabilistic perspective Chapter 4," *The MIT Press*, 2012.

[276] "MATLAB R2013b Documentation - Discriminant Analysis," http://www.mathworks.com/help/stats/discriminant-analysis.html, [Accessed: 1st January 2014].

[277] "Wikipedia - Linear Discriminant Analysis," http://en.wikipedia.org/wiki/Linear_discriminant_analysis, [Accessed: 1st January 2014].

[278] L. I. Smith, "A tutorial on principal component analysis," http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, 2002, [Accessed 26th March 2014].

[279] N. Halko, P. G. Martinsson, Y. Shkolnisky, and M. Tygert, "An algorithm for the principal component analysis of large data sets," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2580–2594, 2011.

[280] D. Meyer, "Support Vector Machines - The interface to libsvm in package e1071," August 2015.

[281] C. J. Burges, "A tutorial on Support Vector Machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, June 1998.

[282] "MATLAB R2013a Documentation - Support Vector Machines," http://www.mathworks.com/help/stats/support-vector-machines.html, [Accessed: 16th July 2013].

[283] J. C. Platt, "Fast training of Support Vector Machines using sequential minimal optimization," *Advances in Kernel Methods, MIT Press Cambridge*, pp. 185–208, February 1999.

[284] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. 2nd European Conference on Computational Learning Theory (EuroCOLT)*, March 1995, pp. 23–37.

[285] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Proc. 16th Scandinavian Conference on Image Analysis (SCIA)*, June 2009, pp. 61–70.

[286] C. F. Liew and T. Yairi, "Generalized BRIEF: A novel fast feature extraction method for robust hand detection," in *Proc. 22nd International Conference on Pattern Recognition (ICPR)*, August 2014, pp. 3014–3019.

[287] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st International Conference on Machine Learning (ICML)*, July 2004, pp. 1–8.

[288] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2211–2268, February 2011.

[289] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *Proc. 10h European Conference on Computer Vision (ECCV)*, October 2008, pp. 504–513.

[290] S. Milborrow and F. Nicolls, "Active shape models with SIFT descriptors and MARS," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, January 2014, pp. 1–8.

[291] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *Pattern Recognition*, vol. 41, no. 10, pp. 3054–3067, October 2008.

[292] J. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shifts," *International Journal of Computer Vision (IJCV)*, vol. 91, no. 2, pp. 200–215, January 2011.

[293] T. F. Cootes, M. C. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *Proc. 12th European Conference on Computer Vision (ECCV)*, October 2012, pp. 278–291.

[294] Y. Li and J. Feng, "Sparse representation shape model," in *Proc. IEEE International Conference on Image Processing (ICIP)*, September 2010, pp. 2733–2736.

[295] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou, "Sparse shape composition: A new framework for shape prior modeling," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 1025–1032.

[296] C. Chen and G. Zheng, "Fully automatic segmentation of AP pelvis X-rays via random forest regression and hierarchical sparse shape composition," in *International Conference on Computer Analysis of Images and Patterns*, August 2013, pp. 335–343.

[297] O. Jesorsky, K. J. Kirchberg, and R. Frischholz, "Robust face detection using the Hausdorff distance," in *Proc. 3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, June 2001, pp. 90–95.

[298] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing (IVC)*, vol. 28, no. 5, pp. 807–813, May Multi-PIE.

[299] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 1513–1520.

[300] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2887–2894.

[301] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2578–2585.

[302] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2013, pp. 532–539.

[303] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 3476–3483.

[304] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. IEEE International Conference on Computer Vision Workshops (ICCV-W)*, December 2013, pp. 386–391.

[305] M. Valstar, B. Martinez, X. Binefa, and M. Pantic, "Facial point detection using boosted regression and graph models," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2729–2736.

[306] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision (IJCV)*, vol. 61, no. 1, pp. 55–79, January 2005.

[307] M. Uřičář, V. Franc, and V. Hlaváč, "Detector of facial landmarks learned by the structured output SVM," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, February 2012, pp. 547–556.

[308] M. Everingham, J. Sivic, and A. Zisserman, "Hello! My name is... Buffy - Automatic naming of characters in TV video," in *Proc. British Machine Vision Conference (BMVC)*, September 2006, pp. 1–10.

[309] X. Zhu and D. Ramanan, "Face detection, pose estimation and landmark localization in the wild," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2879–2886.

[310] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 1944–1951.

[311] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, December 2014, pp. 1–8.

[312] C. F. Liew, N. Yokoya, and T. Yairi, "Face alignment by using sparse initialization and random forest," in *Proc. IEEE International Conference on Image Processing (ICIP)*, October 2014, pp. 1–6.

[313] S. Zhang, Y. Zhan, and D. N. Metaxas, "Deformable segmentation via sparse representation and dictionary learning," *Medical Image Analysis*, vol. 16, no. 7, pp. 1385–1396, October 2012, special Issue on the 2011 Conference on Medical Image Computing and Computer Assisted Intervention.

[314] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic, "Local evidence aggregation for regression based facial point detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 5, pp. 1149–1163, May 2013.

[315] "Drones Guidelines by Federal Aviation Administration," http://www.topdronesforsale.org/drones-guidelines-by-federal-aviation-administration/, [Accessed: 13th July 2016].

[316] "FAA News - Summary of Small Unmanned Aircraft Rule (Part 107)," https://www.faa.gov/uas/media/Part_107_Summary.pdf, [Accessed: 13th July 2016].

[317] "EASA - Civil Drones (Unmanned Aircraft)," https://www.easa.europa.eu/easa-and-you/civil-drones-rpas, [Accessed: 13th July 2016].

[318] "Concept of Operations for Drones - A Risk Based Approach to Regulation of Unmanned Aircraft," https://www.easa.europa.eu/system/files/dfu/204696_EASA_concept_drone_brochure_web.pdf, [Accessed: 13th July 2016].

[319] "Proposal to Create Common Rules for Operating Drones in Europe," https://www.easa.europa.eu/system/files/dfu/205933-01-EASA_Summary%20of%20the%20ANPA.pdf, [Accessed: 13th July 2016].

[320] "MLIT - Flight Rules for Unmanned Aerial Vehicles (Drones)," http://www.mlit.go.jp/koku/koku_tk10_000003.html, [Accessed: 13th July 2016].

[321] "The Current Law on Drones in Japan," http://dronelawjapan.com/, [Accessed: 13th July 2016].