End-To-End and Direct Human-Flying Robot Interaction

by

Valiallah (Mani) Monajjemi

M.Sc., Mechatronics Engineering, Amirkabir University of Technology, 2011
B.Sc., Electrical Engineering, Amirkabir University of Technology, 2008

Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2016

Approval

Name:	**Valiallah (Mani) Monajjemi**
Degree:	**Doctor of Philosophy (Computing Science)**
Title:	***End-To-End and Direct Human-Flying Robot Interaction***
Examining Committee:	**Chair:** Dr. Brian Funt Professor

Dr. Richard Vaughan
Senior Supervisor
Associate Professor

Dr. Greg Mori
Supervisor
Professor

Dr. Arrvindh Shriraman
Co-Supervisor
Assistant Professor

Dr. Nick Sumner
Internal Examiner
Assistant Professor

Dr. Steven Waslander
External Examiner
Associate Professor
Department of Mechanical and
Mechatronics Engineering
University of Waterloo

Date Defended: August 19, 2016

Abstract

As the application domain of Unmanned Aerial Vehicles (UAV) expands to the consumer market and with recent advances in robot autonomy and ubiquitous computing, a new paradigm for human-UAV interaction has started to form. In this new paradigm, humans and UAV(s) are co-located (situated) and use natural and embodied interfaces to share autonomy and communicate. This is in contrast to the traditional paradigm in Human-UAV interaction in which the focus is on designing control interfaces for remotely operated UAVs and sharing autonomy among Human-UAV teams. Motivated by application domains such as wilderness search and rescue and personal filming, we define the required components of *end-to-end* interaction between a human and a flying robot as *interaction initiation* (ii) *approach and re-positioning to facilitate the interaction* and (iii) *communication of intent and commands from the human to the UAV and vice versa.* In this thesis we introduce the components we designed for creating an end-to-end Human-Flying Robot Interaction system. Mainly (i) a fast monocular computer vision pipeline for localizing stationary periodic motions in the field of view of a moving camera; (ii) a cascade approach controller that combines appearance based tracking and visual servo control to approach a human using a forward-facing monocular camera; (iii) a close-range gaze and gesture based interaction system for communication of commands from a human to multiple flying UAVs using their on-board monocular camera; and (iv) a light-based feedback system for continuous communication of intents from a flying robot to its interaction partner. We provide experimental results for the performance of each individual component as well as the final integrated system in real-world Human-UAV Interaction tests. Our interaction system, which integrates all these components, is the first realized end-to-end Human-Flying Robot Interaction system whereby an uninstrumented user can attract the attention of a distant (20 to 30m) autonomous outdoor flying robot. Once interaction is initiated, the robot approaches the user to close range ($\approx$ 2m), hovers facing the user, then responds appropriately to a small vocabulary of hand gestures, while constantly communicating its states to the user through its embodied feedback system. All the software produced for this thesis is Open Source.

Keywords: Human-Robot Interaction, Unmanned Aerial Vehicles, Flying Robots

Dedication

To my lovely parents. For their love, sacrifices and continuous support.

Acknowledgements

The process of earning a doctorate and writing a dissertation is a long journey, one full of adventures, challenges and unique experiences. I was truly blessed to share every moment of this quest with Shokoofeh Pourmehr, whose love and unconditional support helped me immensely to overcome all obstacles and meet my academic goals. Shokoofeh made this a journey to remember and I am deeply grateful for that.

I would like to sincerely thank my senior supervisor, Professor Richard Vaughan, for his continuous support, encouragement, advice, mentorship and above all, his patience. I appreciate his vast knowledge and domain expertise as well as his help in writing manuscripts, designing software systems and crafting new ideas. Without his guidance this research and dissertation would not have been possible.

I would also like to thank my supervisor, Professor Greg Mori, for supporting my research and providing me with thoughtful feedback at different stages of the project. He taught me how to be a critical thinker and encouraged me to always aim for the best. In addition, I would like to thank my PhD committee member, Professor Arrvindh Shriraman, for his support and valuable feedback.

Lastly, I would like to thank all my colleagues and friends at SFU Autonomy Lab who shared the graduate life experience with me and contributed to my research through their criticism, support and encouragement: Jens Wawerla, Abbas Sadat, Jake Bruce, Jacob Perron, Jack Thomas, Lingkang Zhang and Sepehr Mohaimenianpour.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

An *Unmanned Aerial Vehicle* (UAV) is a recoverable aerial vehicle that does not carry a human operator and is either remotely piloted or operates autonomously [127]. Historically, UAVs have been extensively used by military and governmental organizations for reconnaissance and surveillance missions, pilot training and offensive operations. In that context, UAVs are also known as Unmanned Aerial Systems (UAS) and Unmanned Combat Aerial Vehicles (UCAV). Since their early days, remotely operated UAVs have also been popular recreational vehicles. With recent advances in sensing, manufacturing and computing technologies, UAVs are becoming more affordable, capable and ubiquitous. These have led to introduction of new designs, capabilities and application domains for UAVs. Multi-rotor configurations, smaller form factors, improved flight autonomy and advanced on-board sensing capabilities are some examples of such new designs and capabilities. The new application domains include environmental and hazard monitoring, Wilderness Search And Rescue (WiSAR), aerial photography and filming, safety and infrastructure inspection, personal training, crop monitoring for agriculture, goods transportation and manufacturing.

There has been considerable interest in UAVs among researchers in the robotics and intelligent systems communities over the past two decades. Flight control, autonomous navigation, state estimation, intelligent sensing, vision based control and Simultaneous Localization and Mapping (SLAM) have been among the most popular research topics studied by UAV researchers. The goal is to make UAVs autonomous and self-contained, essentially turning them into *flying robots*. Commercial companies have also shown considerable interest in manufacturing UAVs, transferring knowledge from the research community and fostering of adoption of UAVs in the aforementioned application domains in recent years. Major technology powerhouses such as Amazon [1], Google [8, 72] and Facebook [155] have been trying to introduce UAVs to goods transportation and information distribution domains. Multi-rotor small form factor UAVs with high fidelity cameras are among the most popular consumer UAVs (also known as *drones*). During the most recent Consumer Electronics Show (CES 2016), 17 companies introduced 22 new UAV models [66], almost all of them fitting this category. In addition, there have been substantial investments in robotic companies that deal with drone related technologies in recent years. One recent study estimates the total amount of such investments in 2015 alone as $361.8 million (USD) [41].

We believe that interaction with humans is one of the new research opportunities that arise with widespread adaption of UAVs (and flying robots) to new application domains. In some of these new application domains such as infrastructure inspection, search and rescue, goods transportation, personal training and aerial photography, UAVs fly close to humans and may interact with users that are not necessarily their operators. The goal of this thesis is to identify the challenges associated with *Human-Flying Robot Interaction* (a subcategory of Human-Robot Interaction (HRI)) in these emerging application domains and provide solutions for them with the ultimate goal of designing end-to-end systems for situated and direct Human-Flying Robot interaction. Goodrich and Schultz [68] define, Human

Robot Interaction as “a field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans”. Traditionally, Human-UAV interaction happens remotely and in the form of human supervisory control. In that context UAVs are remotely controlled devices that extend the sensing capabilities of their fellow human teammates. Sharing the autonomy between humans and UAVs, managing the cognitive workload of human operators and designing effective remote interaction interfaces for human operators are among the active research topics in traditional Human-UAV interaction literature.

With the increasing level of UAVs’ autonomy and the emergence of new application domains such as search and rescue, and goods transportation, proximate (situated) interaction with UAVs is becoming a topic of interest in the robotics and UAV research communities. As an example, imagine a WiSAR scenario in which autonomous UAVs search an area for missing people, probably injured, alongside a ground search crew. Situated Human-Flying Robot interaction in this setting may happen in different forms. UAVs and their human teammates may communicate information and commands, UAVs may act as tele-presence robots for the ground crew to examine a victim or they may deliver first aid or supplies to victims or their human teammates. These types of situated interaction resemble the interaction scheme that is used by social living creatures (i.e. human-human or human-animal interaction). Interfaces that support such *natural* interaction schemes require naturalistic embodiment [191] and should preferably work when humans are not instrumented, i.e no operator control unit is required and the person carries no other dedicated equipment or clothing. For example, consider the UAV in the search and rescue scenario again. Requiring user instrumentation might limit the ability of the UAV to interact with people. This might be acceptable when the interaction happens with teammates, however it will make it impossible for a human that does not carry the instrumentation (such as the person in need) to interact with the UAV.

We can break down the main components of a situated interaction between a human and a UAV into (i) *interaction initiation*; (ii) *approach and re-positioning to facilitate the interaction*; and (iii) *communication of intent and commands from the human to the UAV and vice versa*. We define a Human-Flying Robot interaction system that implements all these three components as an *end-to-end* interaction system.

Definition 1. *A Human-Flying Robot interaction system is considered end-to-end if it includes all of the following components (i) interaction initiation; (ii) approach and re-positioning to facilitate the interaction; and (iii) communication of intent and commands from the human to the UAV and vice versa.*

Considering the search and rescue scenario again, the interaction between the UAV and the person of interest can be initiated *actively* by the human using an active stimuli such as gestures (e.g. body pose or movements) or auditory signals. Alternatively the UAV may always be running a human feature detector to find potential human partners. The

UAV may provide feedback to the user in forms of spatial maneuvers, auditory signals or visual feedback (i.e. using lights). In addition, the UAV may communicate her readiness for interaction by approaching the user or tracking her movements. At this state, *mutual attention* is created between the UAV and the human. The UAV may also approach the human to facilitate further close-range interaction with her.

Throughout the interaction session, the human may want to communicate its intents or commands to the UAV for execution. Example tasks in the context of the search and rescue scenario include commands to explore a certain area by the human teammate to the UAV or requests to deliver medical aid by the person in need. Feedback from the UAV to the user may also be useful to ensure that the human is fully aware of the current state and the next actions of the UAV. This is to improve the situational awareness of the UAV's interaction partner and to increase safety. For example, the UAV may provide visual, auditory or motion based feedback to the user upon receiving a command. It may also constantly communicate its next flying direction via visual or auditory signals so any human that shares the same workspace is aware of its existence and its next movement.

In what is to follow, we first survey the literature on state of the art in human-flying robot interaction. In Section 2.1, we provide an overview of related work in traditional human-UAV interaction research. In Section 2.2, we discuss related literature on situated Human-Flying Robot interaction, covering both human studies and practical systems.

In Chapters 3 to 5, we introduce methods and systems we developed for situated and direct interaction with flying robots and present experimental results. In Chapter 3, we introduce a close-range interaction system that enables a human to select and command a team of flying robots using gaze and hand gestures. The system described in this chapter is the first demonstration of un-instrumented and direct Human-flying robot interaction using on-board sensing. Next, in Chapter 4, we introduce a realtime computer vision pipeline that runs on-board a UAV that detects stationary periodic motions in UAV's field of view. We then describe how we use this pipeline to detect dual-arm waving gesture while a UAV is in flight to explicitly initiate the interaction with a flying robot. In Chapter 5, we describe our end-to-end human-flying robot interaction system that combines the components from Chapter 3 and 4 with a visual tracker, a cascade visual servo controller and a light based embodied feedback system. Using this system, a flying robot detects a human's explicit interaction initiation signal from distance, smoothly approach her and respond to her gestural commands while constantly communicating its intents and state to her. Finally, in Chapter 6 we conclude this thesis by discussing shortcomings of our proposed systems and providing possible solutions and ideas for future research.

1.1 List of Publications

The following list of publications contain the preliminary reports describing methods and findings of this thesis:

1. V. M. Monajjemi, S. Mohaimenianpour, and R. Vaughan. UAV, Come To Me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016
2. 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 *In Proceedings of IEEE/RSJ Intelligent Robots and Systems (IROS)*, pages 3614–3620, 2015
3. 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 *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, pages 106–106, 2014
4. 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 617–623, 2013

Other related publications made during the course of this thesis:

1. J. Bruce, V. M. Monajjemi, J. Wawerla, and R. Vaughan. Tiny People Finder: Long-range outdoor HRI by periodicity detection. In *Proceedings of Canadian Conference on Computer and Robot Vision, (CRV)*, 2016
2. V. M. Monajjemi, J. Wawerla, and R. Vaughan. Drums: A middleware-aware distributed robot monitoring system. In *Proceedings of Canadian Conference on Computer and Robot Vision, (CRV)*, pages 211–218, 2014
3. S. Pourmehr, V. M. Monajjemi, S. A. Sadat, F. Zhan, J. Wawerla, G. Mori, and R. Vaughan. "You Are Green": A Touch-to-name interaction in an integrated multi-modal multi-robot HRI system. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, pages 266–267, 2014
4. S. Pourmehr, V. M. Monajjemi, R. Vaughan, and G. Mori. You two! Take off!: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 137–142, 2013

Chapter 2

State of The Art in Human-Flying Robot Interaction

This chapter presents a review of current research within the field of Human-UAV (Unmanned Aerial Vehicle) interaction. We categorize the Human-UAV interaction systems based on the level of autonomy of the UAV system and the proximity of interaction partners into two general categories of *remote* and *situated* interaction. First, we study human factors and user interface designs for remote interaction with UAVs in Section 2.1. We introduce the common patterns for designing interfaces for remote interaction with UAVs (Section 2.1.2) as well as major human factor concerns for supervisory control of UAVs (Section 2.1.1). As the application domain of UAVs expands to the consumer market and with recent advances in UAV autonomy and ubiquitous computing, a new paradigm for Human-UAV Interaction has started to form. In this new paradigm, humans and UAV(s) are co-located (situated) and use natural and embodied interfaces to share autonomy and communicate.

The main focus of this chapter is to survey the literature on situated interaction with UAVs that exhibit high level of autonomy: *Flying Robots*. In Section 2.2 we introduce the three main components of situated Human-Flying Robot Systems: (i) interaction initiation; (ii) approach and repositioning to facilitate the interaction; and (iii) communication of commands from humans to flying robots and communication of intent from flying robots to humans. Interaction is initiated between humans and UAVs either implicitly, through human feature detectors, or explicitly, through active stimuli such as gestures. In Section 2.2.1 we survey related work that deal with these two types of interaction initiation and introduce major techniques for human and moving object detection from flying platforms. Once interaction is initiated, UAVs and humans are ready to communicate commands, state and intents. In Sections 2.2.2 and 2.2.3 we survey the literature and introduce notable techniques for such mutual communication between humans and UAVs. Our main focus is on techniques and systems that enable flying robots to interact naturally with humans using embodied sensing and preferably on-board computing. Natural interaction in this context, refers to interaction methods that resemble human-human and human-animal interaction and do not require any instrumentation of the human. Gaze, gestures and auditory signals (human to UAV) and visual or motion-based feedback (UAV to human) are some of the modalities commonly used in the literature for natural interaction between humans and UAVs. In Section 2.3 we provide a brief overview of challenges regarding approach and repositioning in Human-Flying Robot Interaction. We first introduce similarities between this task and target relative navigation in UAVs. Then we survey the state of the art systems that enable a UAV to follow a human and discuss their limitations.

2.1 Human Factors and Remote Interaction with UAVs

Military organizations were among the first users of Unmanned Aerial Vehicles. Since the early days of their utilization in surveillance, border control, reconnaissance and offensive

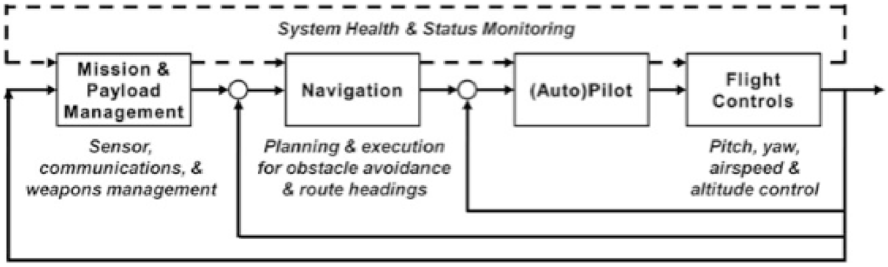

System Health & Status Monitoring
Mission & Payload Management
Navigation
(Auto)Pilot
Flight Controls
Sensor, communications, & weapons management
Planning & execution for obstacle avoidance & route headings
Pitch, yaw, airspeed & altitude control

Figure 2.1: Hierarchical model for human supervisory control of UAVs [67] (© 2015, Springer Science)

military operations, UAVs have been almost exclusively operated remotely by humans. This trend continued when UAVs started to emerge as a tool in other applications such as wilderness search and rescue, disaster damage assessment and environmental monitoring. In such settings, UAVs are tools to reduce operational risk and extend a human team's remote-sensing capabilities. In this context, this compound team of human operators, support staff and UAVs are called Unmanned Aerial Systems (UAS).

One of the key factors that affect the performance of any UAS and the success of its mission is how human operators and their remote flying teammates interact. This interaction has been the focus of "Human Factors Analysis" for UAS. Historically, this has been the first line of research to study Human-UAV interaction.

Parsons and Kearsley [132] summarize the main issues in human factors analysis of robotic systems as the devision of labor between robots and humans, the role of each partner for task execution, how they interact in the workplace and the general question of how machine (robot) and human should be combined. Wilson [185] identifies the goal of human factors analysis as defining design guidelines for human-UAV interfaces with the ultimate goal of improving a UAV operator's capabilities to control and supervise such vehicles. As suggested by Wilson [185], a good interface should increase an operator's situational awareness and manages her workload. Mouloua et al. [117] also identifies situational awareness, workload management and teaming concerns as the three key human factors' issues in development of UAS. Endsley [53] defines situational awareness as "the *perception* of the elements in the environment within a volume of time and space, the *comprehension* of their meaning, and the *projection* of their status in the near future".

2.1.1 Human Supervisory Control of UAVs

The way that humans and UAVs interact and the role of human operators in a UAS depends on the level of autonomy of a UAV. As this autonomy increases, the role of its operator(s) shifts from direct (low-level) control towards monitoring and supervision [24] also known as Human Supervisory Control (HSC) [164].

Goodrich and Cummings [67] propose a hierarchical model for human supervisory control of UAVs as represented in Figure 2.1. The control loops that govern guidance and motion of the UAV exist at the lowest level of this model (flight control and piloting). The operator's action at this level is short term, local and focused on keeping the UAV in a stable flight. At the next level, a navigation loop executes actions to satisfy mission constraints such as passing through specific way-points, obstacle and no-fly zone avoidance and target tracking. At the highest level, the mission and payload management control loop deals with decisions that need to be made to meet high level goals of the mission. The authors argue that since this level of decision making requires knowledge based reasoning and judgment, it can not be automated. The system health and status monitoring loop is the outermost loop in this model which represents the persistent supervision job that must occur when UAS is in operation, either by a human, an automated system or both.

As mentioned earlier, the level of autonomy of a UAS determines the role of the human operator, her mental workload and the interface required to interact with the UAV. As an example, if a UAV is only capable of performing basic flight control, the role of the human operator becomes guidance and motion control which is a demanding cognitive task and leaves the operator little room to interact with any higher level control loops. The interaction interface in this setting should provide the operator with low level and precise control over the UAV.

Goodrich and Cummings argue that increasing the level of autonomy in those three control loops boosts the effectiveness of human operators in UAS by reducing their workload. This would pave the way for cutting down the number of human operators in a UAS and increasing the number of UAVs that can be controlled simultaneously by a single operator. The challenging problem is "how, when, where, and what level of automation should be introduced" [67].

Two major strategies for managing the level of shared autonomy in UAS are management by consent and management by exemption [13]. In the management by exemption schema, the autonomous part of a UAS is allowed to make decisions and perform actions on its own. In this scheme, an operator may be given a short time window to veto this decision. In contrast, the management by consent requires that a human operator explicitly approves any decision made by the autonomous part(s) of the UAS prior to its execution. The management by exemption strategy requires less active human-UAV interaction but it demands constant monitoring by the human operator and may result in poor situational awareness. The management by consent on the other hand increases an operator's situational awareness while demanding more active communication between human operators and UAVs [130].

By surveying different studies on the effect of these two management strategies, Goodrich and Cummings [67] conclude the following results about the level of automation and where to apply those in a UAS:

- Intermediate levels of management by consent are preferred over fully manual or fully autonomous behaviors in a UAS [154]

- Operator performance can degrade under a management by consent strategy when the workload increases [35]

- For high level tasks, management by exemption can improve the performance of an operator [34]

- Under a management by exemption scheme, operators are more likely to become over-dependent on automation, thus fail to check the correctness of the decision made by the system. This is called Automation Bias [116].

- In case of controlling multiple UAVs, management by consent outperforms management by exemption by providing better situational awareness [153, 154]

In a meta analysis on previous studies on multiple UAV control by a single human operator, Goodrich and Cummings further investigate the question of what level of autonomy (LOA) should be introduced in a UAS. The authors first introduce a scale similar to SV-LOA scale by Sheridan and Verplank [165] for characterization of the LOA in the different control loops of their proposed Human Supervisory Control models (Figure 2.1). This scale consist of six levels, from full manual control by a human operator (level I) to fully autonomous control loops (level IV). At level II of this scale, the computer suggests a complete set of action/decision alternatives, while at level III, the computer is able to prune this set to include only decision/action candidates that meet a certain criteria. Level IV and V of this scale represent management by consent and management by exemption respectively.

According to this meta-analysis, the authors conclude that:

- Controlling more than one vehicle by a single operator requires a fully autonomous inner motion control loop (Level VI)

- The operator capacity to control multiple UAVs converges to 4-5 vehicles given some sort of assistance (level II/III) or management by consent (level IV) in the navigation/autopilot loop. This number jumps to 8-12 vehicles when management by exemption is introduced in this loop.

The authors anticipate that when the LOA is at the highest level for the motion and autopilot control loops, the control architecture shifts from centralized and per vehicle to decentralized and task based. This will put an operator solely in charge of mission management and supervisory monitoring, which should increase the number of vehicles that can be controlled by a single operator.

2.1.2 Interfaces for Remote Interaction with UAVs

As mentioned in the previous section, the payload and mission management loop requires knowledge based reasoning and judgment, thus it demands a high mental workload from the operator, especially in case of multiple UAVs [75]. In a UAS with low level of autonomy in lower level control loops, this task is off-loaded to a separate operator known as the sensor operator. A considerable body of research in human factors of Unmanned Aerial Systems investigates methods that can reduce this mental workload, mainly through improving how information about the flight, the mission and the environment is presented to an operator and how commands from an operator are perceived. This can be particularly beneficial in application domains in which operating UAVs with the minimum number of operators is critical. As an example, in Wilderness Search And Rescue (WiSAR), it is desirable to have as many well-trained staff as possible in the ground search task due to the size of the area that needs to be covered [30].

In an effort to combine the role of the pilot and the sensor operator in WiSAR scenarios, Cooper and Goodrich [30] designed an integrated information display system for piloting and monitoring a UAV. The authors classify the three main paradigms in designing information display systems for UASs as *pilot centered*, *traditional* and *integrated*. The pilot centered paradigm tries to simulate the cockpit environment of a manned aircraft for the remote pilot. The traditional display consists of multiple windows each dedicated to visualize a certain type of information such as maps, the mission, live video feeds and raw sensory information. Integrated displays on the other hand, are mixed-reality displays that integrate satellite imagery, the state of the of the UAV, sensor footprints and live video feeds all together and project them onto a simulated three dimensional terrain.

In a series of informal user studies, Cooper and Goodrich first showed that localizing targets with respect to the UAV is a quite challenging task for sensor operators when the information display system is traditional. This task is one of the main duties of sensor operators in WiSAR scenarios and is crucial to the success of the whole mission. The authors hypothesize that an integrated information display system can reduce this mental workload to the level that the pilot can perform this task.

They also found out that even when the information is displayed in an integrated manner to the sensor operator, they are mostly unaware of the flight path and can not recover that reliably. In search and rescue missions, effective execution of the search strategy depends on an operator's awareness of the flight path. The authors' second hypothesis was combining the role of the pilot and the sensor operator will increase the awareness of the flight path since the sensor operator is now in charge of generating the flight path as well.

In a series of experiments in a medium fidelity flight simulator and with an integrated information display system (Figure 2.2), Cooper and Goodrich studied the performance of minimally trained operators performing a simulated search task. The UAS provides a high

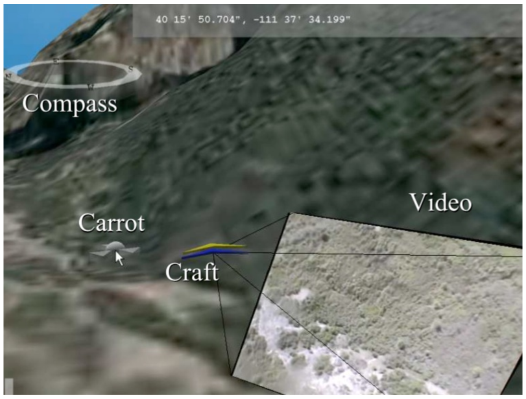


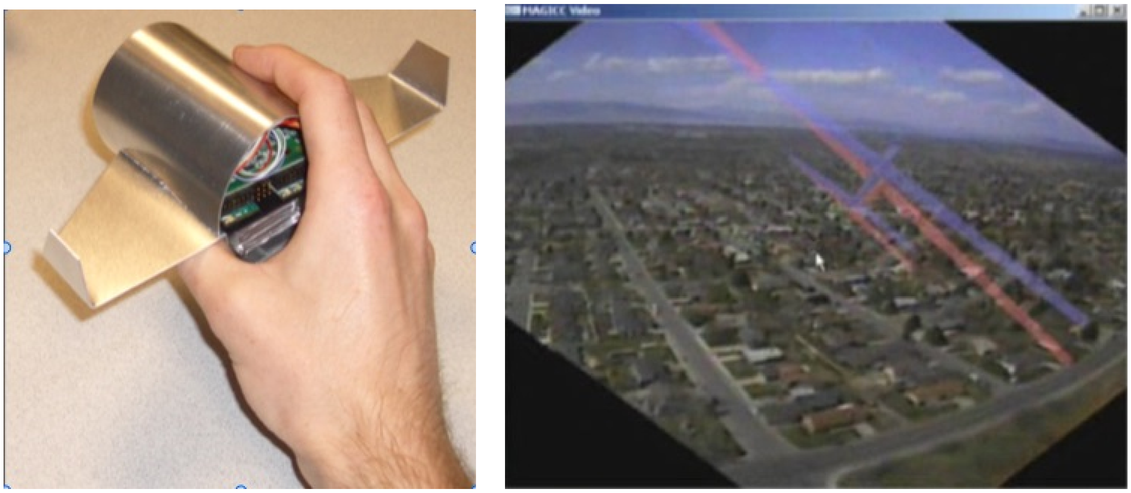
Figure 2.2: Integrated mixed-reality interface of Cooper and Goodrich [30] (© 2008, IEEE)

level of autonomy for all the flight control loops (Figure 2.1) so the pilot could control the UAV by changing its target location (projected to the world) using a computer mouse. The task was to identify and localize colored objects distributed randomly in the world in the presence of random distracting objects. They evaluated their integrated display system under four control perspectives (*chase*, *north-up*, *track-up* and *split*) and subject to three distributions for objects: uniform, Gaussian and rectangular pattern.

In the *chase* perspective, the user sees the UAV and the integrated virtual world from behind and above the UAV, similar to the perspective used commonly in first person shooter video games. The *north-up* and *track-up* perspectives similarly show the UAV from directly above it. The difference is how the map is oriented. While the *north-up* always orients the map towards north, the *track-up* orients the map such that the heading of the UAV is always towards to the top of the screen. The *split* perspective is similar to the *north-up* but renders the UAV from much more distance. This implies that the operator needs to rely on a secondary video display since the integrated video contains few details.

The results of the experiments do not strongly suggest that using integrated displays will enable the roles to be combined for this task. That is mainly due to the high LOA in the control loops which simplifies the piloting task, the medium fidelity nature of the simulator and the low complexity of the identification task. However the results suggest that if a UAS can afford to provide these capabilities, this might become an achievable goal. In addition, the experiments show that the *chase*, *north-up* and *track-up* perspectives exhibit almost the same level of performance while the *split* display performs significantly worse. They also indicate that the performance of those three perspectives varies under different distributions of objects. This implies that the choice of perspective depends on the search scenario. For example, the *chase* perspective is well suited for reactive (hasty) search while the *north-up* perspective is better suited for exhaustive searches.

Mixed reality interfaces are similar in concept and design to interfaces used to interact with virtual worlds in video games. A taxonomy for classification of UAV interfaces by



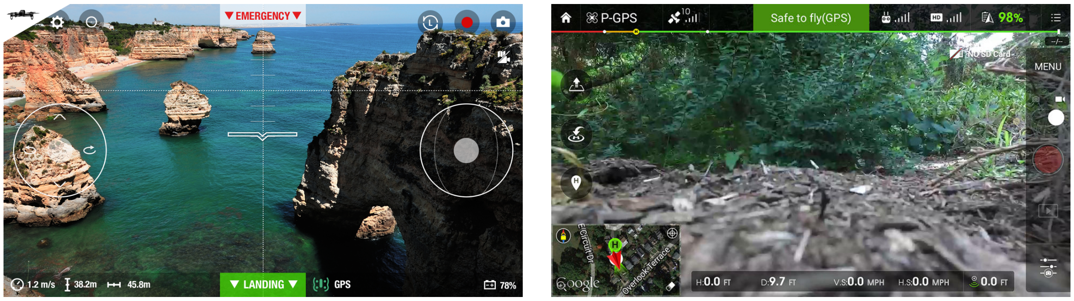
(a) Physical Icon: The operator controls the UAV by manipulating the model airplane (Physical Icon) of the UAV

(b) Mixed reality interface: The interface shows the world-centeric view of UAVs video stream. Two superimposed blue and red icons show the current and desired state of the UAV respectively

Figure 2.3: Physical Icon mixed-reality interface of Quigley et al. [146] (© 2004, IEEE)

analogies to video game interfaces has been proposed by Richer and Dury [94]. Based on this taxonomy, the *chase* perspective is a camera of type "external attachment to a primary object". The special case when the operator looks through a UAV's camera is defined as "internal attachment to a primary object". In the *north-up* and *track-up* perspectives, the camera is categorized as "not attached to a primary object" with "free" or "fixed" movement constraints. According to Richer and Dury's taxonomy, the *Split* display consist of two views, a camera "not attached to a primary object" and a camera with "internal attachment to a primary object".

Another notable example of mixed-reality interfaces is the interface proposed by Quigley et al. [146] for remote operation of UAVs in WiSAR scenarios. Their system consists of a so-called Physical Icon (Figure 2.3a) for direct manipulation of UAV's attitude through inertial sensors and a *chase*-perspective mixed reality interface that superimposes the desired and actual state of the UAV on the stabilized live video feed from the vehicle (Figure 2.3b). The interface tries to reduce the operator's mental workload by stabilizing the video feed with respect to the ground plane such that even when the UAV banks, the image is aligned with the horizon. The authors were the first to combine direct manipulation with mixed reality for single user operation of UAVs, the idea which has became popular in designing interfaces for remote interaction with consumer UAVs. Such interfaces utilize inertial sensors and touch screen of consumer smart-phones and tablets as physical interfaces to control the UAV and/or their on-board cameras and use their display for real-time mixed-reality visualization of video and telemetry data (Figure 2.4).



(a) Parrot®FreeFlight3 Interface (Piloting Mode)

(b) DJI®Go Interface

Figure 2.4: Example consumer UAV interfaces that run on smart-phones and tablets. These mixed reality physical interfaces take advantage of the rich set of input and output modalities offered by such devices.

With increased LOA in the higher level control loops in Figure 2.1, the need for interfaces that support higher level of controls for UAVs emerges. For example, if the UAV is capable of autonomously flying to certain GPS way-points, the interface has to provide efficient means for systematic navigation, mission manipulation, and monitoring to the human operator. Such interfaces usually provide some sort of *map-based* perspective in plain or mixed-reality manner. Interfaces similar to Cooper and Goodrich [30] in *north-up* configuration can also be considered as a special form of *map-based* perspectives. Goodrich and Cummings [67] identify the main benefit of using *map-based* perspectives as the ability to localize the UAV with respect to the world's landmark.

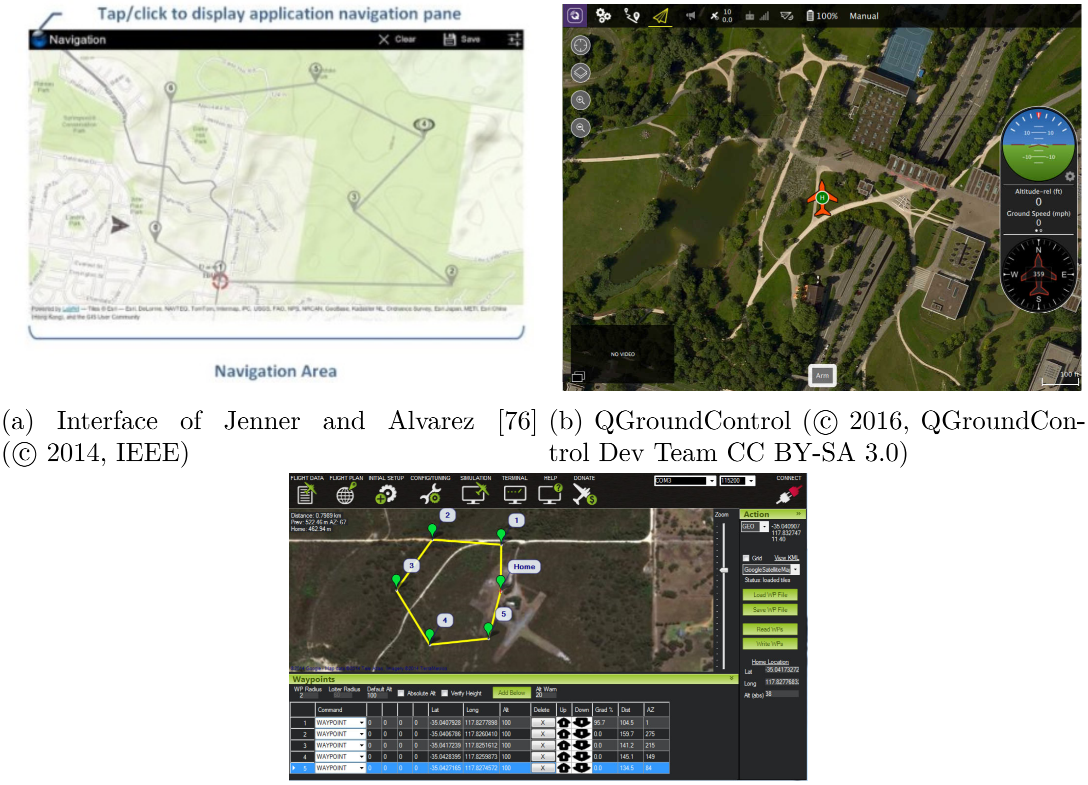
Jenner and Alvarez [76] developed a cross-platform interface to command, control and monitor a fleet of unmanned aerial/ground/underwater vehicles. The navigation part of this interface provides a *map-based* view of all vehicles' missions (Figure 2.5a). It also provides context-based information such as the current location, the target way-points and their associated tasks, the planned flight path and the actual flight trajectory. The user is able to modify the mission parameters, the target way-points and their associated actions through a touch screen interface. This is the concept that is used by many similar off-the-shelf ground control software for consumer UAVs, such as QGroundControl [1], APM Planner [2] and Parrot FreeFlight3 [3] (Figure 2.5). For technical details on available software technologies for developing such integrated interfaces with *map* and *chase* views, see the work by Perez et al. [133].

Based on the findings by Fong and Thorpe [59] which indicate that in complex and dynamic environments, tele-operation interfaces need to be multi-modal and provide support for high level navigation and command generation, Crescenzio et al. [33] designed an end-to-end multi-modal ground control station for remote operation of a single UAV. The

[1] http://qgroundcontrol.org/

[2] http://planner.ardupilot.com/

[3] http://www.parrot.com/ca/apps/



	Command					Lat	Long	Alt	Delete	Up	Down	Grad %	Dist	AZ
1	WAYPOINT	0	0	0	0	-35.0407928	117.8277898	100	X			95.7	104.5	1
2	WAYPOINT	0	0	0	0	-35.0406786	117.8260410	100	X			0.0	159.7	275
3	WAYPOINT	0	0	0	0	-35.0417239	117.8251612	100	X			0.0	141.2	215
4	WAYPOINT	0	0	0	0	-35.0428395	117.8259873	100	X			0.0	145.1	149
5	WAYPOINT	0	0	0	0	-35.0427165	117.8274572	100	X			0.0	134.5	84

(a) Interface of Jenner and Alvarez [76] (© 2014, IEEE)

(b) QGroundControl (© 2016, QGroundControl Dev Team CC BY-SA 3.0)

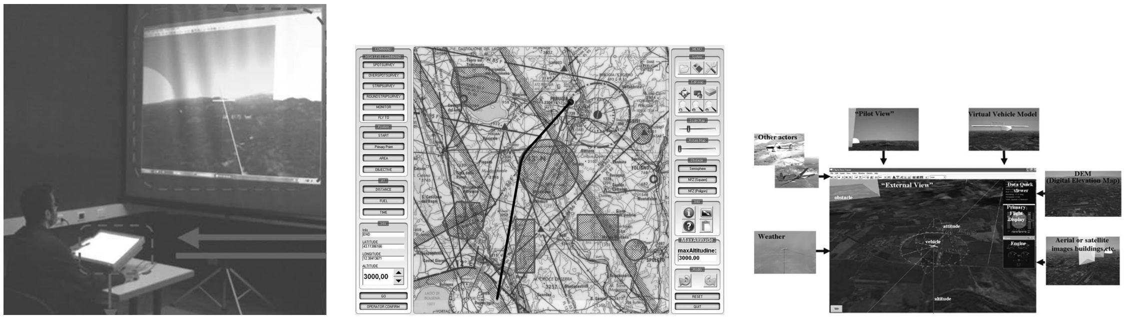
(c) APM Mission Planner (© 2016, ArduPilot Dev Team CC BY-SA 3.0)

Figure 2.5: Example *map*-based interfaces

system consist of two main components: a command panel running on a touch screen enabled computer and a stereoscopic visualization projected to a big screen in front of the operator.

The command panel provides a *north-up* oriented *map* based navigational display with touch based controls to manipulate high level mission parameters such as tasks (survey or monitor), way-points and priorities (Figure 2.6b). The interface provides a management by consent planning engine which generates a plan based on the operator's inputs, then gives her the option to accept, reject or modify the plan. The stereoscopic visualization is a 3D virtual reality display that integrates data coming from multiple sources and projects them to a virtual train map (similar to the interface of Cooper and Goodrich [30]). The video feed from the vehicle in *chase* or external perspective, the telemetry data, weather information, the mission and the flight path are all integrated in this display (Figure 2.6c). The interface provides audio feedback to the user when a command is sent to the vehicle, when there is an unexpected change in the data coming from the UAV and to communicate the general state and the progress of the mission.

Crescenzio et al. evaluated their ground control station design by performing a user study with 12 participant under three different LOA for the mission (re)planning compo-



(a) The overall design of the Ground Control Station

(b) The command panel

(c) The 3D virtual reality display

Figure 2.6: The Multi-modal ground control station of Crescenzio et al. [33] (© 2009, MIT Press)

nent. The users were asked to operate the UAV in a dynamic environment in which random obstacles were being added to the environment. The re-planning strategy was subject to three different autonomy levels: manual (no automatic re-planning), management by consent and fully autonomous (no control from the user). The study shows in general that the system provides a satisfactory level of situational awareness using its integrated virtual reality display and auditory feedback. The users found the command panel a good tool to manage the mission. However the most important finding has been that the management by consent schema for shared autonomy provides the best balance between situational awareness and operator's workload. This result is in agreement with conclusion made by Ruff et al. [154] and Goodrich and Cummings [67].

The multi-modal design paradigm for Ground Control Stations has been further studied by Maza et al. [103]. In their study, the authors examined the effect of different modalities on reaction time of users operating a ground control station for UAVs. The abstract task was to select random "yes" buttons that show up on three monitors emulating a ground control station while ignoring "no" buttons. The study evaluated the reaction time of users and the percentage of their right and wrong actions when under different combination of control (input) and feedback modalities. The system could receive input from a computer mouse or a touch screen and could provide feedback about the presence of new buttons to the users using speech synthesis, 3D audio interface and tactile interface. For the latter two, the feedback indirectly includes positional feedback about the location of the button with respect to the user. The results indicate that, compared to baseline of a tablet interface without any feedback, multi-modal feedback increases the mean response time of the users by about 14%. The best result was obtained using all three feedback modalities.

Most interfaces we have surveyed in this section deal with remote interaction of UAVs under human supervisory control. Some consensus has emerged in the design of the user interfaces, into either vehicle-centered (pilot-like) modes, or world-centered modes, both augmented with a variety of sensor data. As the application domain of UAVs expands to the consumer market and with advances in robot autonomy and ubiquitous computing, a new paradigm for human-UAV interaction has started to form. In this new paradigm, humans and UAV(s) are co-located (situated) and use natural and embodied interfaces to share autonomy and provide feedback. We will discuss this new paradigm in the following section.

2.2 Situated Interaction With UAVs

We introduced the common patterns for designing interfaces for remote interaction with Unmanned Aerial Vehicles as well as major human factor concerns for supervisory control of UAVs using these remote interaction systems in the previous section (Section 2.1).

In this section we focus on proximate interaction between humans and UAVs mainly outside the scope of human supervisory control. This type of interaction happens when humans and UAVs are co-located (situated) and UAVs exhibit the level of autonomy that makes them operate under minimal or no remote control. As an example, consider an autonomous UAV in a search and rescue scenario which surveys an area, locates the person in need, communicates its location, provide tele-presence for first responders and provides first aid kit or food to her upon request. The same UAV may also be summoned by ground search crew and commanded to modify its mission. In case that the UAV requires help (e.g. battery replacement or maintenance), it may look for a human team member for assistance.

These types of situated interaction resemble the interaction scheme that is used by social living creatures (i.e. human-human or human-animal interaction). Interfaces that support such *natural* interaction schemes require naturalistic embodiment [191] and should preferably work when humans are not instrumented, i.e no operator control unit is required and the person carries no other dedicated equipment or clothing. For example, consider the UAV in the search and rescue scenario again. We argued earlier that requiring user instrumentation may greatly limit the ability of the UAV to interact with people. This might be acceptable when the interaction happens with the teammates, however it will make it impossible for a human that does not carry the instrumentation (such as the person in need) to interact with the UAV. In this section, we survey situated interaction with UAVs and corresponding challenges. Our focus will be mainly towards the methods and systems that facilitate *situated*, *natural*, *embodied* and *un-instrumented* interaction with UAVs.

We can break down the main components of a situated interaction between a human and a UAV into (i) *interaction initiation*, (ii) *approach and re-positioning to facilitate the interaction* and (iii) *communication of intent and commands from the human to the UAV and vice versa.* Considering the scenario introduced earlier, the interaction between the UAV and the person of interest can be initiated *actively* by the human using an active stimuli such as gestures (e.g. body pose or movements) or auditory signals. Alternatively the UAV may always be running a human feature detector to find potential human partners. The UAV may provide feedback to the user in forms of special maneuvers, auditory signals or visual feedback (i.e. using lights). In addition, the UAV may communicate her readiness for interaction by approaching the user or tracking her movements. At this state, *mutual attention* is created between the UAV and the human. The UAV may also approach the human to facilitate further close-range interaction with her.

Throughout the interaction session, the human may want to communicate her intents or commands to the UAV for execution. Example tasks in the context of the search and rescue scenario include commands to explore a certain area by the human teammate to the UAV or request to deliver medical aid by the person in need. Feedback from the UAV to the user may also be useful to ensure that the human is fully aware of the current state and the next actions of the UAV. This is to improve the situational awareness of the UAV's interaction partner and to increase safety. For example, the UAV may provide visual, auditory or motion based feedback to the user upon receiving a command. It may also constantly communicate its next flying direction via visual or auditory signals so any human that shares the same workspace is aware of its existence and its next movement.

The rest of this section surveys the literature in situated, natural, embodied and uninstrumented interaction between humans and UAVs. For each work, we will identify how it is related to the components we introduced earlier, the challenges it tries to solve and its technical contribution. When necessary we will provide an introduction to major techniques used by the paper and a quick survey about that.

2.2.1 Interaction Initiation

Interaction initiation between humans and UAVs mostly happens in two forms in the literature. In the first class of methods, the UAV utilizes human feature detectors to find potential interaction partners. Alternatively the user may try to attract the UAV's attention by using active stimuli such as gestures or body movements. Due to practical and safety considerations, most studied human-UAV interaction systems use micro or small-sized aerial multi-rotors with the ability to perform Vertical TakeOff and Landing (VTOL) and in-place hovering. These platforms are well-suited for research purposes in this area because they are easier to operate and control compared to fixed-wing vehicles and require less workspace. Furthermore their small form factor and ability to hover in place allows them to perform close-range interaction with humans. This comes at the cost of limited payload capability and less energy-efficient flying compared to fixed wing UAVs.

The limited payload carrying capacities of small form factor UAVs restricts the number and type of sensors and computational devices they can carry. Although some researchers opt to do computing off-board, any practical system requires on-board (and real-time) computing capabilities to truly perform embodied and situated interaction with humans. Due to these limitations, most interaction initiation techniques proposed in literature rely on light-weight sensing devices, mainly color or thermal cameras for sensing. Self-contained systems with on-board computing usually rely on fast computer vision techniques to find or detect a potential interaction partner. Before reviewing these papers, we briefly survey common techniques for pedestrian (human) detection using computer vision.

2.2.1.1 Vision based Pedestrian Detection

Pedestrian detection from visual data is studied under the general domain of object detection in computer vision research. In addition to human-robot and human-computer interaction, human detectors are used in surveillance, human activity recognition systems such as sports analytics and autonomous vehicles. The latter has been the main driving force of pedestrian detection for moving cameras research in recent years.

In general, contemporary vision based pedestrian detectors share the following components and steps. First, visual features to represent a human are selected. A model that represents the human (as the target object class) based on these visual features is then defined and trained using a usually large set of positive (human) and negative (non-human, background) examples. Machine learning algorithms commonly used in the literature to train this model are Boosting (mainly Adaptive Boosting (AdaBoost)) and Support Vector Machines (SVM) [12]. Human models are either *monolothic* meaning that they represent the human body (or upper body) as a whole or *part based* which model the human as a set of distinct parts. For *part based* models, the model captures the visual appearance of the parts in the feature space as well as their spatial or topological relationships. Once the model is defined and trained, the resulting classifier is applied to different locations of an input image, usually in a *sliding window* manner. Since the classifier is trained for a certain [pixel] size of humans, the input image is first up and/or down sampled to generate a pyramid of images of different sizes. Each layer of this pyramid, which corresponds to a certain size of a human, is searched for matches using the pre-trained classifier. Using the sliding window and multi-scale search, the classifier usually fires multiple times around a candidate detection. A non-maximal suppression post-processing step filters these results to form the output list of pedestrians. For most of the state of the art human detectors, the features are hand crafted prior to training. Recently, Deep Learning techniques such as Convolutional Neural Networks (CNNs) have been applied to automatically select the features during the training phase [161].

Viola and Jones's seminal work on face detection [179] was among the first papers to utilize the aforementioned pipeline to successfully detect human faces in real-time. In their work, Viola and Jones propose using Haar-like features to represent a face and a cascade of classifiers to perform the detection. Each classifier is a boosted classifier which aggregates the output of a set of weak classifiers. The weak classifiers are one level decision trees associated with a specific feature. They use AdaBoost algorithm to select the most distinctive examples from training set, adjust the weights of weak classifier and train the cascade. The authors later applied the same technique for detecting pedestrians by incorporating motion features into their model [180]. The next breakthrough came when Dalal and Triggs [38] proposed using the Histogram of Oriented Gradients (HOG) as the feature descriptor and linear SVM as the classifier for detecting humans. The proposed model consists of a vector of normalized HOG values calculated by concatenating overlapping HOG vectors over 8×8 cells across a fixed size template. Felzenszwalb et al. [57] introduced a deformable parts based multi-scale HOG model to represent objects. The proposed model consists of a root and a series of parts. The HOG pyramid is first formed by calculating HOG feature vectors over the entire image at different levels of the image pyramid. The root defines the boundaries of the human (object) within the coarser levels of the pyramid. Parts are defined with respect to the root and inside its boundaries at finer levels of the pyramid. Both the root and parts are so called *filters* that weigh the overlapping HOG feature vector at their corresponding pyramid level. Felzenszwalb et al. used a Latent SVM classifier to train the filters, their pyramid level and the placement of the parts with respect to the root for different classes of objects, including people.

In a recent evaluation by Beneson et al. [12], the authors studied 40 pedestrian detection methods and identify three main solution families: methods based on ensemble classifiers (boosted decision forests) similar to Viola and Jones's [179], methods based on Deep Networks and methods based on Felzenszwalb et al.'s Deformable Parts Model [57]. By evaluating the performance of these families of methods on the Caltech-USA pedestrian dataset [46], they conclude that all three families can reach current top performance in pedestrian detection. By analyzing the main approaches used by each detector to improve its performance, the authors deduced that, between the two most utilized classifiers (Support Vector Machines and decision forests), neither is believed to have an edge over the other in terms of improving the overall performance. Beneson et al. [12] also note that, in the context of pedestrian detection, DPM and Deep Learning based methods do not provide an advantage over decision forest based methods, except for better handling of occlusion by DPM based methods. The authors however found that better visual features by far are the largest contributing factor to improving pedestrian detection performance over the years.

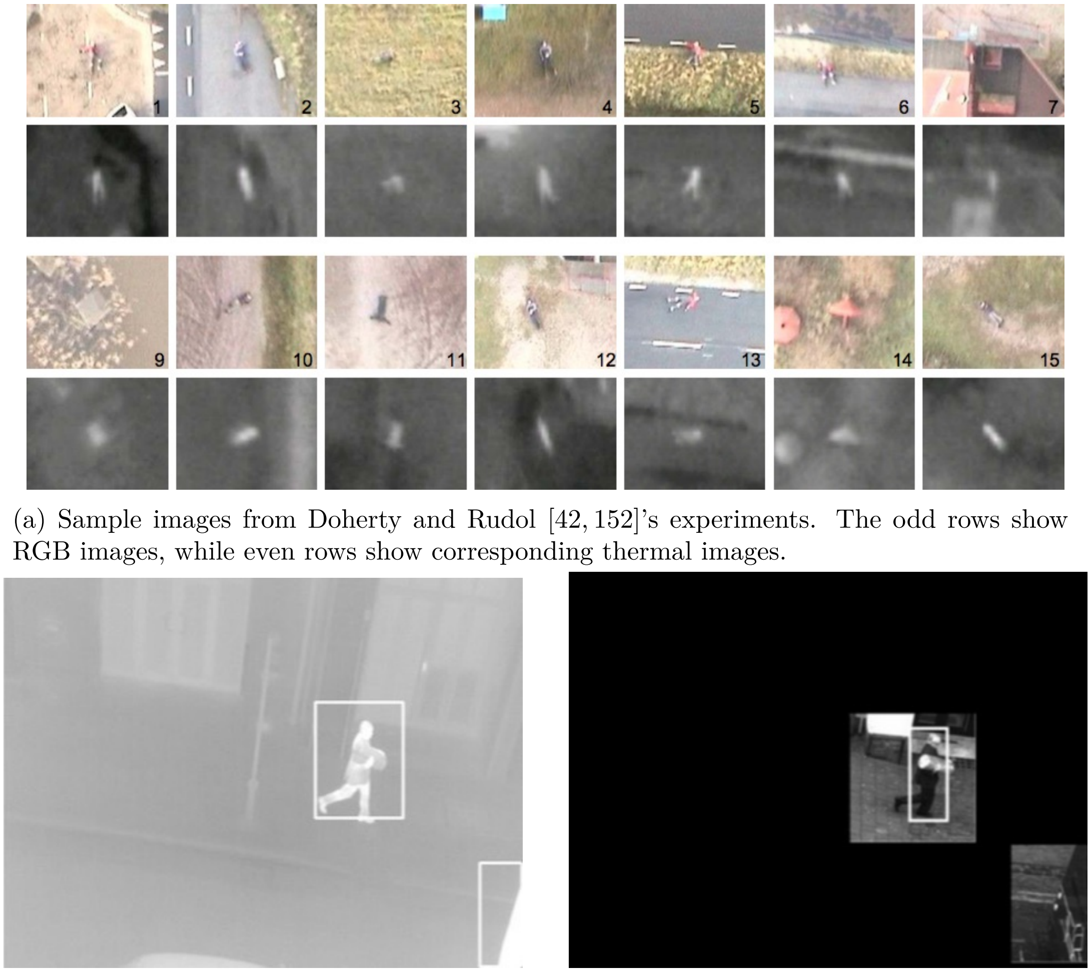
We briefly survey some notable recent works that have enhanced pedestrian detection by mainly enriching features and designing better classifiers. Dollar et al. [45] proposed generating channels from linear or non-linear transformation of the image and use the sum of regions in different channels as features. These summations can be computed efficiently using integral images. Their Integral Channel Feature (ICF) classifier, combines Dalal and Triggs's HOG features, gradient magnitudes and LUV color channels (10 channels in total). Their ICF detector (also known as *ChnFtr* in the literature) utilizes AdaBoost for feature selection and soft cascades, a variation of Viola and Jones's cascade to perform the detection. Benenson et al. [11] extensively analyzed every step of the ICF detector and made various optimizations to improve its performance. Their final classifier, named *Roerei* detector is one of the best performing classifiers on the Caltech-USA dataset. The authors proposed using all possible square rectangles inside the detection window instead of random pooling during the training process. Motion features [12, 131] and scene contexts [12] are further feature enhancements proposed by researchers to improve the ICF framework.

Pedestrian detectors based on monolithic models reach higher detection speeds compared to part based detectors. However the original CPU implementations of the ICF detector or HOG+SVM detector of Dalal and Triggs still do not run at framerate on VGA and higher resolution video streams. To speed up the original ICF detector, Dollar et al. [44] proposed generating coarser image pyramids at the detection time with $\frac{N}{k}$ levels instead of N levels. Features computed at these $\frac{N}{k}$ levels are used to approximate features at other $N - \frac{N}{k}$ levels. This detector is named as *Fastest Pedestrian Detector in the West* (FPDW) by the authors. Dollar et al. [43] further improved this detector by exploiting the correlation between responses of the detector when applied to neighboring sliding windows. The resulting detector, called *CrossTalk Cascade*, achieves 4× to 32× speedup over FPDW depending on the tolerable miss rate increase. Similar to FPDW's idea, Benenson et al. [10] proposed generating N classifiers for each scale at training time and use only one image scale during the detection phase. To speed up the training, they derived a method to train classifiers for $\frac{N}{k}$ scales and approximate classifiers for other scales. This classifier is known as *VeryFast* in the literature. By combining *VeryFast* with other enhancements such as GPU implementation and exploiting scene geometry through ground plane approximation, their detector achieves detection speed of 100+ frames per second on $640 \times 480px$ frames.

2.2.1.2 Interaction Initiation Through Pedestrian Detection on UAVs

Doherty and Rudol [42, 152] designed a UAV based system for autonomous detection of victims in a search and rescue operation. The goal of their system is to create a saliency map of the victims in a target area and then plan a path for the efficient delivery of medical services to the salient points in the map using multiple unmanned helicopters. The UAVs in this work look for human features in the data received from both a thermal camera and a color camera. The interaction in this work is *implicit* and is initiated via human detection by the UAV. The authors divide the mission into two legs. In the first leg, the saliency map is created by multiple UAVs. In the second leg, this saliency map is used to generate plans for delivering food or medical aid to the victims.

The assumption in this paper is that both camera planes are parallel to the ground and the victims are lying on a flat ground. This assumption allows the authors to use the cascade classifier of Viola and Jones [179] that is trained on frontal human bodies directly to detect the human. In order to speed up the detection, the authors employ a layered focus of attention system. The image from the thermal camera is first thresholded to find regions of human body temperature. A post processing step filters blobs based on their size and aspect ratio. Corresponding regions in the color image frame are then searched



(a) Sample images from Doherty and Rudol [42, 152]'s experiments. The odd rows show RGB images, while even rows show corresponding thermal images.

(b) Focus of attention system of Flynn and Cameron [58] thresholds the thermal camera's image to find regions of interest.

(c) Pedestrian detection applied to regions of interests of Figure 2.7b ([58])

Figure 2.7: Using thermal images to find regions of interest prior to running pedestrian detection (a) © 2007, Springer Verlag (b, c) © 2013, Springer International

with the cascade classifier to find a human body. The authors propose using the state of the UAV to find the corresponding image regions instead of performing image or feature matching between two frames. More specifically, the proposed technique first estimates the projection of the centroid of a blob in the thermal image in the world coordinate system by intersecting the ray that passes through the camera's center and the centroid with the (flat) ground plane and by employing camera intrinsic parameters, the attitude (roll, pitch and yaw) of the UAV and the orientation of the thermal camera. The point is then back projected to the image plane of the color camera. If the classifier fires a positive detection in that region, the world coordinate of the area is saved to a saliency map. The authors validated their system by doing a single large scale outdoor experiment with two helicopters. The experiment showed 100% success on finding 11 victims with the average geo-localization error of 2.5 meters in a $290 \times 185m$ area. Unfortunately the paper lacks more experiments to asses the performance of this vision pipeline. Figure 2.7a shows a few sample images from their experiments.

The idea of using thermal cameras to find regions of interest prior to applying human detection algorithms has also been studied by Flynn and Cameron [58]. Similar to Doherty and Rudol [42,152], the proposed method first thresholds the thermal image to find regions of interest. Since the camera is stationary and the scene is static in this study, the method applies a pre-calibrated homography transform to project each region of interest in the thermal camera's image plane to its corresponding location in the color camera's image plane. The parts based pedestrian detection method of Felzenszwalb [57] is then applied to these ROIs to find potential humans. Unsurprisingly, these focus of attention methods improved the speed of human detection algorithms (specifically slow ones such as DPM) substantially. For example, on the $1280 \times 960px$ frames of the dataset used by the authors, the processing time was reduced from 15 seconds per frame to approximately 4 seconds per frame.

Blondel et al. [14] identify two major application domains that existing human detectors are trained for: security monitoring and driver assistance, both assuming an upright human view. This assumption is not well suited for detecting people from UAVs, since the vantage point is different and time varying. Furthermore, the configuration of the camera is coupled to degrees of freedom of the UAV, specifically its roll and pitch, both changing over time. To compensate for these effects, Blondel et al. [14] propose to modify the training phase for ICF classifier to include variations in people's appearance caused by roll and pitch of the camera. In an earlier work by the same authors [15], they showed that a classifier that is trained only on frontal human bodies, starts to fail to correctly classify human images when the elevation angle (pitch) of the camera exceeds 40 degrees.

Similar to [42,58,61,152] the system of Blondel et al. uses a thermal and a color camera in stereo configuration, and utilizes thresholding of the thermal camera's image to narrow down regions of interest. To project from the thermal camera's image plane to the color camera's, they assume an infinite homography transform between these planes (Equations 2.1, 2.2)), which is a reasonable assumption when the human is far from the stereo rig ($> 10m$). In these equations K_1 and K_2 are matrices of intrinsic parameters of cameras, T is the rigid body transformation between them and u, v are pixel coordinates.

$$H_{\text{inf}} = K_2 \times T \times K_1^{-1} \tag{2.1}$$

$$\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = H_{\text{inf}} \times \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{2.2}$$

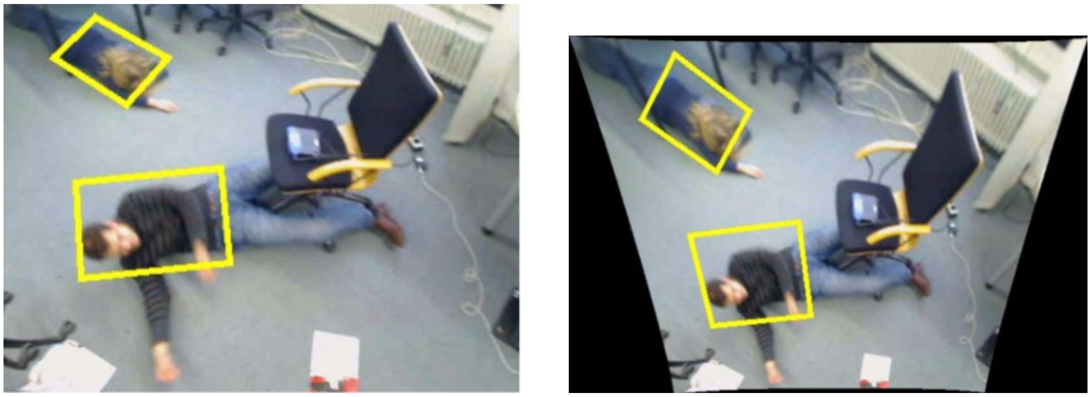
To accommodate the roll and pitch of the camera, the authors generated a dataset[4] of 3846 images taken from different pitch angles and rotated those images from -90 degrees to 90 degrees with 5 degrees steps to generate roll variations. The resulting dataset is then used to train a boosting classifier based on ICF. They name their classifier *Pitch and Roll-trained Detector (PRD).* In a series of tests on a testing dataset recorded from an Asctec Pelican UAV, the authors first show that the classic ICF approach fails on this dataset mainly due to variations in viewpoint as well as pitch and roll of the camera, while the PRD classifier shows reasonable performance when applied directly to the color image. The full pipeline also shows good performance on detecting people, while reducing computation time by 60%. Unfortunately the authors do not report the absolute timing measurements for their pipeline. In their earlier similar work [15], the authors explored the idea of using visual saliency maps instead of a thermal camera to reduce the search space of a human detector. They showed that saliency maps when combined with careful tuning of classifier parameters during training can help to increase the speed of HOG based human detector from 18.87 seconds per frame to 2.286 seconds for a $640 \times 480px$ frame.

In a study by Andriluka et al. [7], the performance of state of the art pedestrian detectors for detecting victims lying on the ground was evaluated. The authors generated a dataset of people lying on the ground in an office setting (220 images in total), some of them partially occluded by objects in the environments. Although the distance from the UAV to the victims is small (Figure 2.8), the results from this study are still applicable to more realistic scenarios. The authors first evaluated the performance of HOG based monolithic full and upper body detectors on this dataset with part based models. All classifiers were using their original trained model for human detection. In general, the part based models show better performance compared to monolithic models. As an example, the best performing monolithic model, upper body HOG, achieves 21.9% equal error rate (EER), while the best parts based model's EER is 51.5%. Although all classifiers are under-performing because they are exposed to images that are affected by roll and pitch of the UAV, the authors argue that part based models are performing better because they handle articulated poses and occlusions better.

Similar to Blondel et al. [14], the authors propose to use telemetry data (pitch and height from ground) coming from the UAV to improve the classification performance. Their method is based on compensating the distortion caused by pitch of the camera by back projecting the image to the ground plane, assuming that this plane is flat (Figure 2.8). This not only removes the distortion due to the perspective, but also provides a prior on the size of the person in the image plane. The authors use this prior to improve the accuracy of the *Pictorial Structures* detector [6], a part based pedestrian detector.

De Smedt et al. [169] propose to use the prior on the target object's height to reduce the search space of human detectors running on-board the UAVs. The idea is inspired by

[4] `http://mis.u-picardie.fr/~p-blondel/papers/data`



(a) Original image

(b) Ground-plane projected image

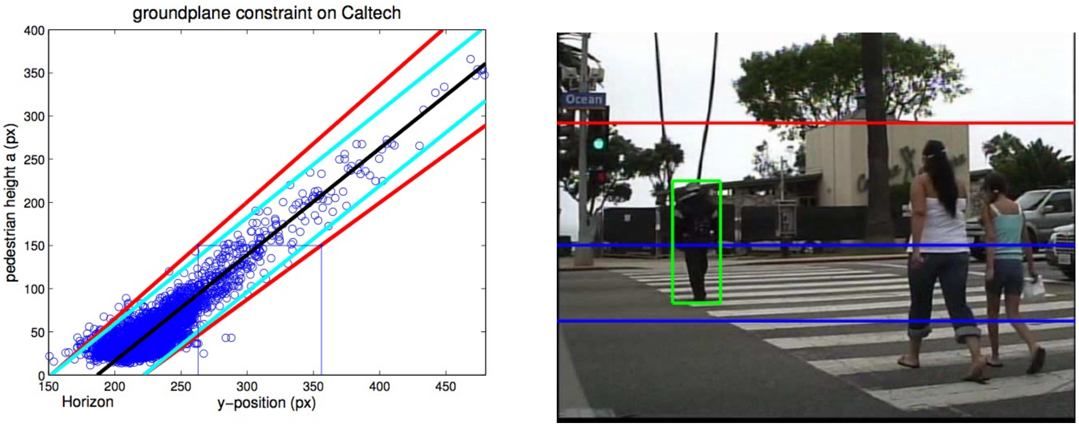
Figure 2.8: An example of using orientation estimation and telemetry data to compensate for UAV or camera's pitch variations by projecting the image to the ground plane, prior to using pedestrian detectors (from Andriluka et al. [7] © 2010, IEEE)

the method proposed in [172] which describes how the height of the object in the image plane can be estimated given the homography of the ground plane and a prior on its size. Instead of directly using the homography, De Smedt *et al.* adopt a data driven approach. Using annotated data from Caltech dataset [47], they estimate the boundaries of ground plane indirectly by fitting a linear function to the height of pedestrians with respect to the position of their feet in the image plane (Figure 2.9). This function is used to predict the horizontal boundaries of the search region in an image for a given target height.

The Caltech dataset is recorded with a camera mounted on a moving car. In this setting, the pitch, roll and altitude of the camera from the ground plane is almost fixed. This is not the case for a camera mounted on a UAV. As mentioned earlier, the roll, pitch and altitude of the camera all affect the ground plane estimation. Smedt et al. propose the following to take these parameters into consideration. Dealing with the roll of the camera is trivial, since the image can be rotated back with the inverse angle of the roll. The effect of the pitch and altitude on the location of the pedestrian in the UAV's field of view is shown in Figures 2.10a and 2.10b. The flat ground assumption (at zero pitch), imposes that the y-position of the horizon to be at the center of the image, thus the y position of the pedestrian feet will be at:

$$y = \frac{h_{im}}{2} + \frac{a}{A}B \tag{2.3}$$

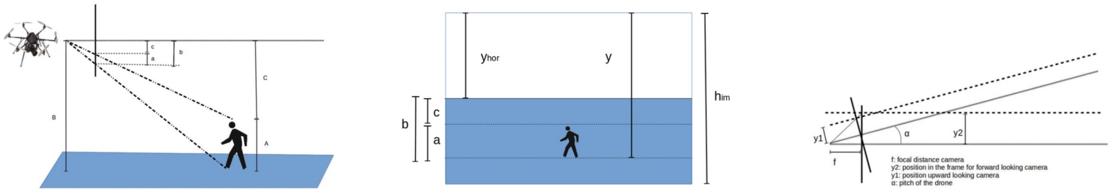
In the above equation, h_{im}, a, A and B refer to the height of the human in pixels (unknown), the real world size of the pedestrian (the prior) and the height of the UAV from the ground (measured), respectively. Setting $y = 0$ and solving the equation for minimum and maximum height of the pedestrian will result in the minimum and maximum height of the pedestrian in the image plane. These values are then passed to the linear mapping



(a) A linear model (black line) is fitted to estimate the y-position of pedestrian feet with respect to their height. Red lines are lower and upper boundaries on the y-position of the ground plane.

(b) Blue lines indicate the y boundaries of ground plane for $150px$ pedestrians. The red line indicates the upper boundary when the height itself is considered.

Figure 2.9: Using object height and ground plane estimation to reduce the search space for pedestrian detection (Smedt et al. [169] © 2015, IEEE)


c
a
b
C
B
A
y_hor
y
h_im
c
b
a
y1
α
y2
f
f: focal distance camera
y2: position in the frame for forward looking camera
y1: position upward looking camera
α: pitch of the drone

(a) The effect of altitude of the UAV on pedestrian's location on image plane

(b) The effect of altitude of the UAV on pedestrian's location on image plane

(c) The parameters used in Equation 2.4 to cancel out the effect of camera's pitch

Figure 2.10: Using telemetry data (altitude and pitch) received from the UAV to refine the ground plane boundaries (Smedt et al. [169] © 2015, IEEE))

function of 2.9a to obtain the region of interest. To cancel out the pitch, the authors provide the following formula based on the pinhole camera model and the estimated pitch angle (α) (Figure 2.10c):

$$y_2 = f\frac{fsin(\alpha) + y_1cos(\alpha)}{fcos(\alpha) - y_1sin(\alpha)} \tag{2.4}$$

One of the most related works to the task of pedestrian detection on-board UAVs is the recent work by Lim and Sinha [96]. In this work, the authors combine a feature based monocular SLAM pipeline with the ICF pedestrian detector [45] and an adaptive appearance based visual tracker [71] to reconstruct the 3D trajectory of a moving person from a flying camera. This system tracks the motion of the forward-facing monocular camera of the UAV using a standard feature based visual SLAM pipeline. The direction of the gravity

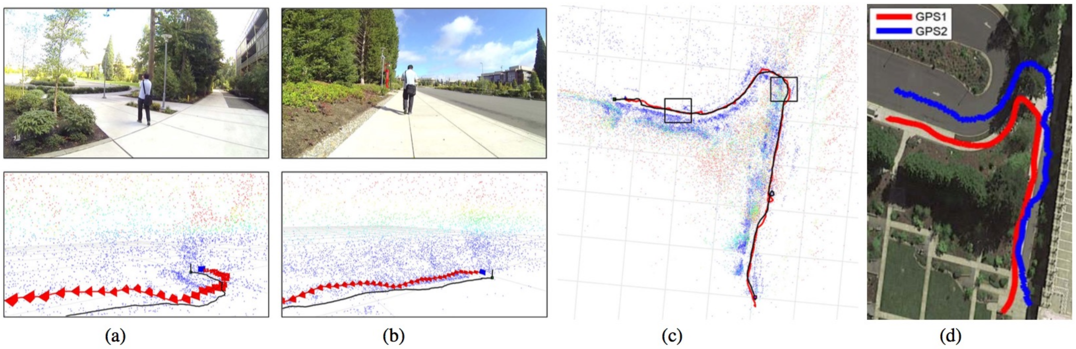


Figure 2.11: The outdoor open-loop experiments of Lim and Sinha [96] (a,b) Two snapshots from camera's FOV and the corresponding reconstructed trajectories (c) The top-view of the final trajectory (d) The ground truth trajectories of the pedestrian from GPS+INS tracker (GPS1) and GPS tracker (GPS2) (© 2015, IEEE))

vector reported by the on-board IMU is used by the system to detect the ground plane from the tracked 3D feature points. This is subsequently used to recover the absolute scale of camera's motion and the map.

The system runs ICF (pedestrian detector) and Struck (visual tracker) in parallel. Struck is initialized by the system using a heuristic policy based on the confidence of the pedestrian detector. The authors propose a dynamic programming based approach to combine the output bounding boxes of these two components and to decrease the overall false positive rate. To reconstruct the full 6 DOF trajectory of the detected person, the system back projects roll-corrected rays that correspond to the head and feet of the tracked person from feature points in the pedestrian bounding box to the image plane. Assuming that the height of the person is known and the person is in upright position, the system recovers the depth of the head and feet of the tracked person using simple geometry. The 3D pose of the head and feet location are tracked in the acquired map of the environment using a Kalman Filter.

Lim and Sinha performed a series of tests[5] on 9 video sequences acquired either by moving the UAV in indoor and outdoor settings by hand or by manually piloting the UAV. The authors show that their pedestrian detection and tracking hybrid performs better on average compared to the baseline tracker and detector. In addition, the authors examined the accuracy of the pedestrian trajectory in two *open loop* outdoor experiments in which a UAV was manually piloted to follow a person while the full trajectory reconstruction pipeline was running on-board the UAV (Figure 2.11). The mean 2D Euclidean error of two trajectories compared to ground truth data from a GPS Inertial-based tracker carried by the target was 2.57 ± 1.87 meters and 3.6 ± 3.22 respectively. The average framerate of the system during these experiments were 15 and 17 FPS respectively (640×480px image, Intel 4th generation Core i7 CPU).

[5] Dataset and demonstration video available at `http://research.microsoft.com/en-us/um/redmond/groups/ivm/mavloc/`

2.2.1.3 Interaction Initiation Through Moving Object Detection

One of the main concerns regarding executing machine learning based approaches on-board the UAV is the computational resources that these methods require. Even on fast desktop machines, state of the art CPU-only implementations do not run at frame-rate, particularly over long distances when pedestrians are smaller in size (approx. $50px$ in height) [47]. Using Graphical Processing Units (GPU) can speed up the processing time by at least one order of magnitude [44], however state-of-the art embedded computing platforms applicable to UAVs either do not include a powerful GPU or their Application Programming Interface (API) does not provide access to the GPU. As noted by Dollar et al. [47], when these pipelines are part of a larger system, ground-plane estimation or Region Of Interest (ROI) selection can be employed to speed up the detection. We have shown in this section that region of interest selection (focus of attention) methods based on additional sensory information are widely used by UAV researchers for detecting humans (e.g. [7, 42, 58, 169])

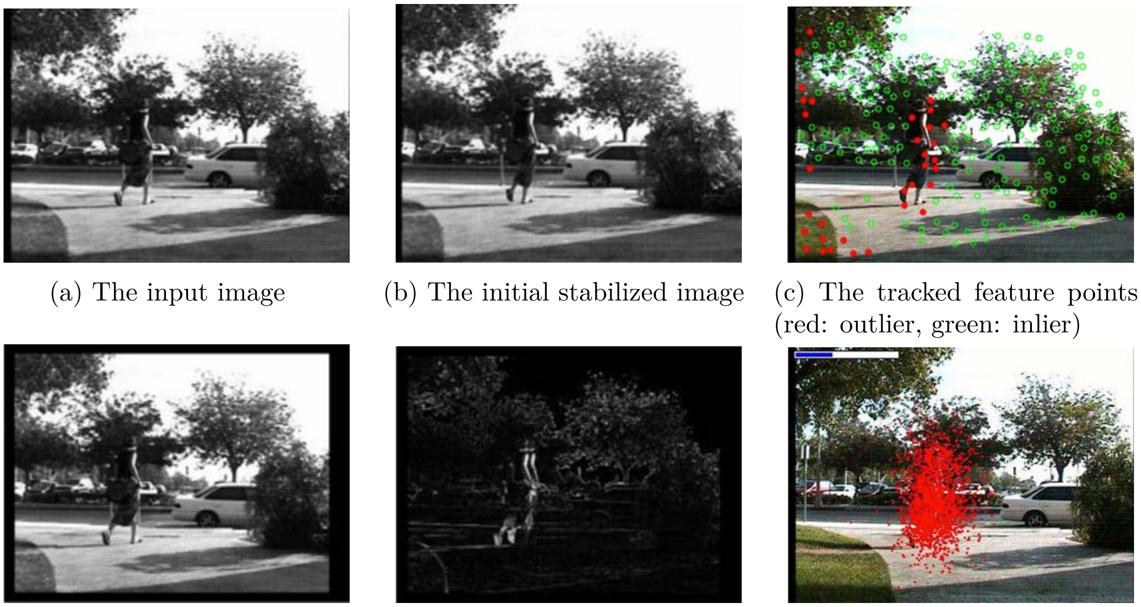
An alternative way to find regions of interest in an image is to use motion information to detect moving objects. Although the moving object based ROI selection filters out ROIs with potential stationary humans inside, they are still of great interest to the methods that detect movement based interaction initiation signals (i.e. hand gestures or follow-me when I move). Since the camera is attached to a moving platform, classical background subtraction methods [138] which assume a static environment are not applicable. Before concluding this section, we survey methods for detecting moving objects on-board a UAV. This problem is closely related to foreground-background segmentation in dynamic environments and camera ego-motion estimation and cancellation.

The pipeline proposed by Jung and Sukhatme [79, 80] deals with detection and tracking of independent motion on-board moving platforms in real-time using a monocular camera. The authors identify two main challenges for motion tracking from a mobile platform. The first challenge is canceling out the motion induced by the movement of the platform, known as the ego-motion. The second challenge is dealing with various types of noise caused by poor lighting conditions, camera distortions and the unstructured environment.

To compensate for the ego-motion, Jung and Sukhatme's pipeline selects a set of salient feature points [166] (F^{t-1}) in each frame (I^{t-1}) and tracks them using KLT optical-flow tracking method [18] to the next frame (I^t) to find the corresponding feature points (F^t). Once the correspondence is found, a bilinear motion model is fitted to the data using a least-squares method to recover the inter-frame motion of the camera (T^t_{t-1}):

$$\begin{bmatrix} f^t_x \\ f^t_y \end{bmatrix} = \begin{bmatrix} a_0 f^{t-1}_x + a_1 f^{t-1}_y + a_2 + a_3 f^{t-1}_x f^{t-1}_y \\ a_4 f^{t-1}_x + a_5 f^{t-1}_y + a_6 + a_7 f^{t-1}_x f^{t-1}_y \end{bmatrix} \tag{2.5}$$

To deal with outliers caused by features located on moving objects, a simple thresholding based outliers rejection method is employed. In the first pass, the full feature set is used



(a) The input image

(b) The initial stabilized image

(c) The tracked feature points (red: outlier, green: inlier)

(d) The refined stabilized image using estimation from inliers feature points

(e) The difference image

(f) The posterior probability

Figure 2.12: The different steps of Jung and Sukhatme [80]'s pipeline for detecting moving objects from a moving platform (© 2009, Springer Science & Business Media BV)

to find T^t_{t-1}. In the second pass, feature correspondences that show large re-projection error[6] are removed from the set as outliers. The motion of the camera is estimated again solely based on inlier correspondences. The input frame (I^{t-1}) is then warped using the inverse of the estimated transform ($I_c(x, y) = I^{t-1}{T^t_{t-1}}^{-1}(x, y)$). Finally, the pixel-wise intensity difference of the stabilized frame and the consecutive frame is calculated ($I_d(x, y) = |I_x(x, y) - I^t(x, y)|$) (Figure 2.12e).

The resulting difference image is not perfect due to the noise sources mentioned earlier (systematic errors) and also due to errors in camera ego-motion cancellation step (transient errors). The authors propose using an adaptive particle filter approach [60] to deal with the transient errors. The state of each moving object is defined as its position and velocity on the image plane. The posterior probability of these states are calculated recursively using an observation model defined over I_d assuming a constant velocity motion model.

Figure 2.12 shows the effect of different steps in the proposed pipeline and the resulting posterior probability. The particles are finally clustered into objects using a density based clustering algorithm. To track multiple objects, the pipeline maintains a particle filter for each moving object. Filters are initiated once all particles converge to an object and are destroyed when they diverge due to an object's disappearance. To prevent particle filters

[6] The threshold for detecting outliers is manually defined

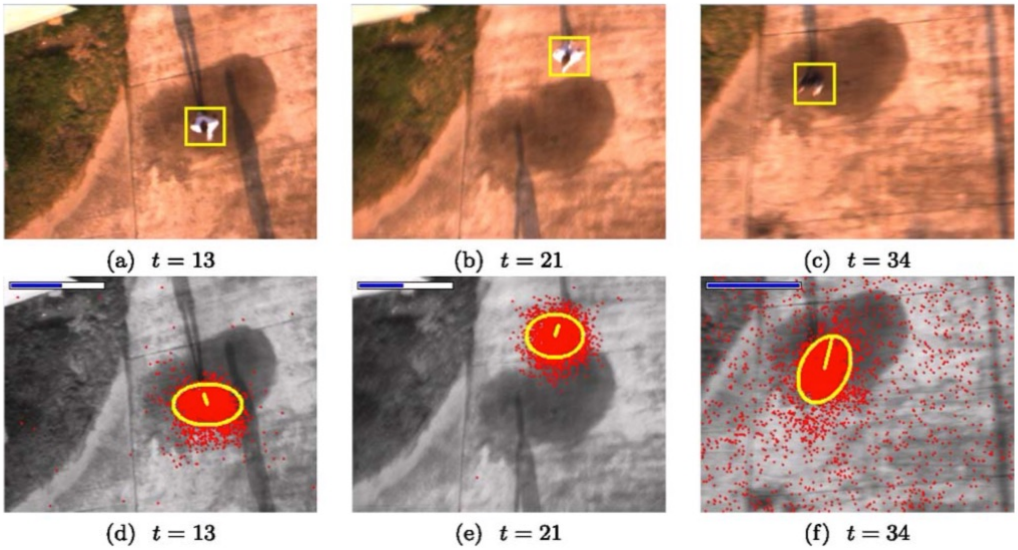


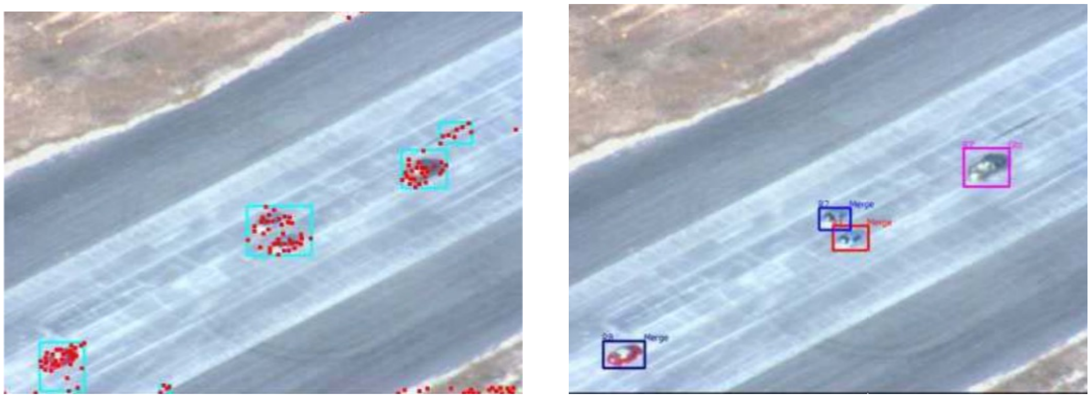
Figure 2.13: Single moving object detection and tracking on-board an autonomous helicopter using a downward facing monocular camera (Jung and Sukhatme [80] © 2009, Springer Science & Business Media BV)

converging on a single object, once a filter is updated, the difference image is modified to clear the motion of the tracked object.

The authors performed various types of experiments to evaluate different components of their proposed system. One experiment in particular evaluated the accuracy of single moving object detection and tracking on-board an autonomous helicopter using a downward facing monocular camera (Figure 2.13). The proposed pipeline achieved an 80% detection rate with an average tracking error of 11.9 pixels on a (rather short) dataset of 43 frames with an approximate computation speed of 5 frames per second[7]. The results on non-flying platforms were more promising, especially for the case of detecting and tracking pedestrians on-board a Segway RMP robot, which achieved a 96.15% detection rate on a dataset of 141 frames.

A similar framework is proposed by Siam and ElHelw [168] for detecting and tracking moving objects from UAV imagery. Similar to the previous work, this pipeline detects and tracks salient feature points between consecutive frames using the KLT optical flow method. Assuming a downward facing camera at relatively high altitude looking towards a flat ground, the tracked feature points can be approximated to be on the same plane. Under this assumption, the corresponding motion between two consecutive images can be approximated by a homography transform.

[7] Pentium III 1.0 GHz, $320px \times 240px$ frames



(a) Outlier feature points and initial cluster of objects

(b) Final list of objects after data association step

Figure 2.14: Moving object detector of Siam and ElHelw [168] (© 2012, IEEE)

$$\begin{bmatrix} f_x^t \\ f_y^t \\ w \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_3 & c_4 & c_5 \\ c_6 & c_7 & 1 \end{bmatrix} \begin{bmatrix} f_x^{t-1} \\ f_y^{t-1} \\ 1 \end{bmatrix} \tag{2.6}$$

Siam and ElHelw [168] adopted the LMedS (Least Median Square Estimator) [192] to estimate the homography between consecutive frames and find outlier feature points. Without stabilizing the camera or calculating a difference image, the pipeline directly clusters outlier features into candidate moving objects with the assumption that the effect of parallax effect and mismatches are negligible and most of outliers are caused by independent moving objects and belong to them. The pipeline uses a bank of Kalman filters to track multiple moving objects. The state vector is similar to what Jung and Sukhatme [79, 80] proposed. Instead of using a motion model, each object's neighborhood is searched for the best visual match in the consecutive frame using a cross correlation template matching approach. To associate objects (observations) to Kalman filters, Siam and ElHelw propose an overlap-rate-based data association method. Using a set of hand crafted rules and thresholds, this approach associates observations to trackers, merges or splits them based on the ratio of intersection area to detection area of each observation and the tracker. The pipeline achieves the overall average precision rate of 94.7% on the DARPA VIVID dataset [29] (Eglin-I and II) with average computation rate of 15 frames per second [8] (Figure 2.14).

Van Eekeren et al. [178] adopted a similar approach to the two previous works to detect moving objects, however they combine it with pedestrian detection. The pipeline consists of two parallel execution threads. The first thread calculates a stabilized difference image by tracking feature points between consecutive frames, calculating intra-frame affine transforms (under the same flat plane assumption of Siam and ElHelw [168]) based on feature

[8] 2.33 GHz Intel Core 2 CPU, 640 × 320px frame

correspondences, warping one image to another image plane and differentiating pixel intensities. The second thread runs the Fastest Pedestrian Detector in West (FPDW) [44] to detect pedestrians on the whole images. The stabilized difference image is then post-processed to detect objects (regions of interest) using morphological operations. These objects are finally merged with detected pedestrians based on overlap ratios of bounding boxes and confidence value of each detected pedestrian. A set of experiments on a subset of the UCF-ARG dataset [119] which consists of different human activities recorded from a remotely-controlled blimp, shows that combining these two methods increases both the accuracy of human detection and the performance of tracking compared to when only one method is applied.

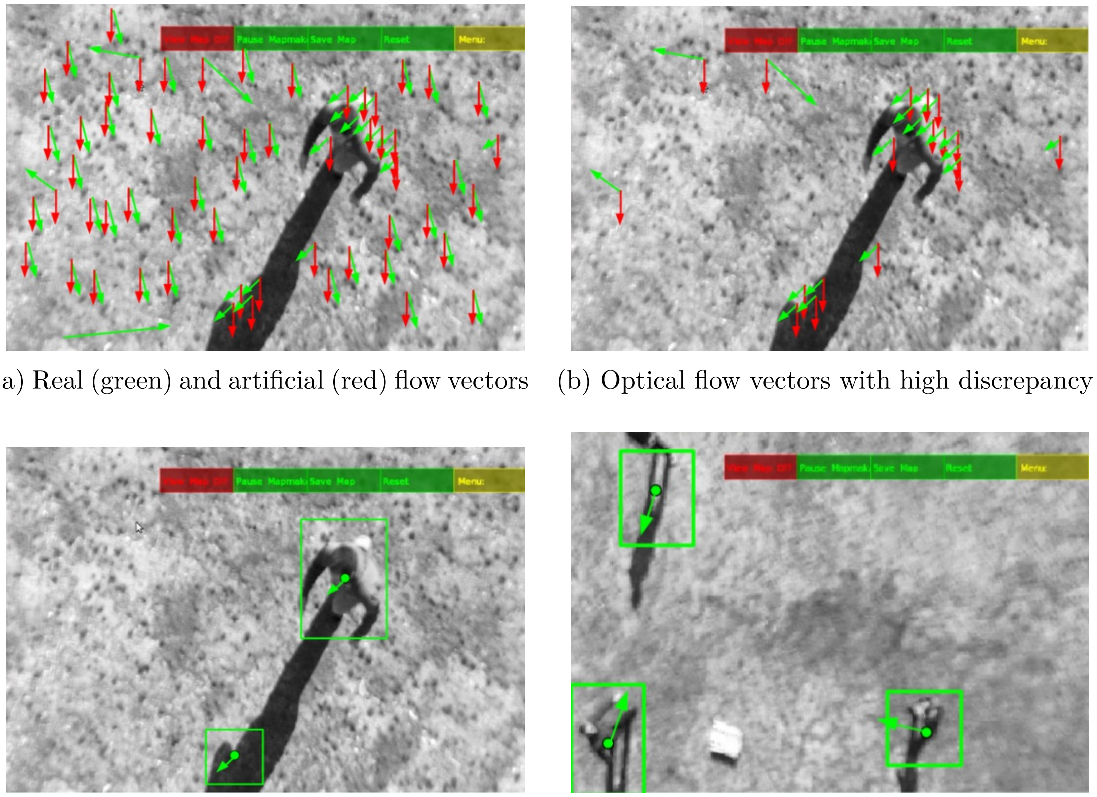
Rodriguez et al. [150] used the discrepancy between artificial optical flow field caused by the ego-motion of a camera and actual inter-frame optical flow field to detect moving objects on-board a UAV. The system simultaneously calculates two optical flow fields (real and artificial) from the monocular input video stream. The real optical flow field is calculated by tracking salient feature points over consecutive frames using KLT tracker (Figure 2.15a). The system tracks full 6 Degrees Of Freedom (DOF) of the camera using the Parallel Tracking and Mapping (PTAM) [87] method. PTAM is a keyframe-based monocular Simultaneous Localization and Mapping (SLAM) system that concurrently tracks the movement of a camera and generates a map of an unknown environment. The system directly employs the output of the tracking thread, the 6 DOF of the camera pose, as an estimation for camera's ego motion. To recover metric scale, PTAM is initialized with a marker of known size.

To calculate the artificial optical flow field, the proposed method assumes that the camera is downward facing and the scene is mostly flat. Hence, the frame to frame motion of feature points on the image planes can be approximated with a homography transform (similar to [178]). The system first reconstructs the homography transform from the estimated translation ($T_{t-1,t}$) and rotation ($R_{t-1,t}$) of the camera from frame $t-1$ to t as well as the distance from the camera plane to the ground (d) as follows: (n^T is the vector perpendicular to the ground plane)

$$H_{t-1,t} = R_{t-1,t} - \frac{T_{t-1,t}n^T}{d} \tag{2.7}$$

The feature points (from real optical flow field calculation step) are then re-projected from frame $t-1$ to t using the calculated homography $H_{t-1,t}$ to form the artificial flow field: (Figure 2.15b)

$$\begin{bmatrix} x_t \\ y_t \\ w \end{bmatrix} = H_{t-1,t} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{bmatrix} \tag{2.8}$$



(a) Real (green) and artificial (red) flow vectors

(b) Optical flow vectors with high discrepancy

(c) Detected moving objects and their corresponding velocity vector

(d) Moving objects detected on another example frame

Figure 2.15: Rodriguez et al. [150]'s method to detect moving objects from discrepancies between real and artificial optical flow vectors (© 2012, MDPI CC BY 4.0)

Up to this stage, two optical flow vectors are associated to each feature point, one real ($_r$) and one artificial ($_a$). To classify each feature point as static or dynamic (belonging to a moving object), the angular and modulus discrepancy between these two vectors are calculated for each feature point. Feature points for which their angular difference $|\alpha_r - \alpha_a|$ are less than 20 degrees and their relative magnitude ratios $|\frac{m_a - m_r}{m_a}|$ greater than $\pm 30\%$ are considered dynamic (Figure 2.15b).

To group the dynamic feature points, isolated dynamic pixels are removed first (assumed to be false outliers). The remaining feature points are clustered into groups based on their positions as well as the magnitudes and angles of their corresponding optical flow vectors (Figure 2.15c). Finally, objects with small number of associated feature points are discarded. A bank of Kalman filters with a greedy data association strategy track the position and size of the objects over time.

In a series of demonstrative experiments, the authors executed the described pipeline on-board an Asctec Pelican quadrocopter (on a single board computer equipped with a 1.6Ghz Intel Atom Processor and 1GB of RAM) to detect and track moving people. The UAV and the people were independently moving, meaning that the output of dynamic object detection system was not used to control the behavior of the UAV. To amplify the discrepancies of optical flow fields, the system runs at $\frac{1}{3}$ to $\frac{1}{6}$ of camera's 30fps framerate. The authors does not provide quantitative results about the success rate of the experiments and the run-time performance of the system. Figure 2.15 shows two sample tracking outputs of the system during the trials.

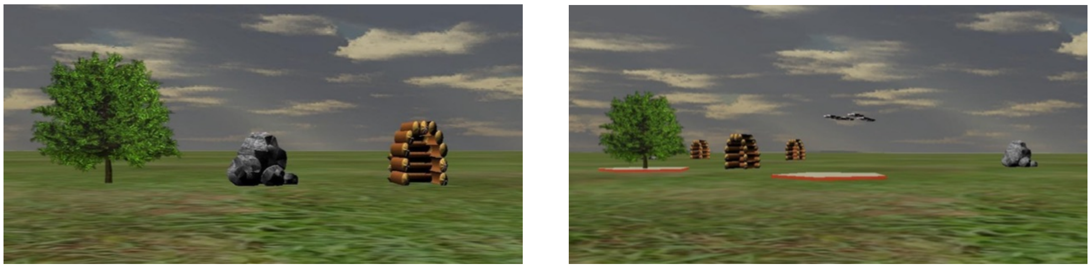
2.2.2 Communication of Intent and Commands to UAVs

Once the interaction between a human and a UAV (or a team of UAVs) is initiated, the human and the UAV(s) can interact more directly by communicating their intents. Yan et al. [189] defines a social robot as *"a robot which can execute designated tasks"* and the necessary condition to turn a robot into a social robot as *"the ability to interact with humans by adhering to certain social cues and rules"*. We believe that if a UAV exhibits the abilities we outlined in the example search and rescue scenario by interacting with humans naturally (adhering to social cues) and maintaining its safety distance while providing naturally understandable feedback (adhering to social rules) and executing designated commands, then it can be considered a social robot. As we will see in this section, although the state of the art in situated and natural interaction with UAVs is still far from introducing true social UAVs, the overall progress is towards this direction.

2.2.2.1 Human Studies

The four major perception modalities for natural and situated interaction with social robots are visual signals, audio signals, tactile signals and laser reading as identified by Yan et al. [189] in their recent survey. In a study by Jones et al. [77], the authors explored the first two modalities in the context of situated human-UAV interaction in search and rescue operations. The test-bed consists of a virtual reality environment that simulates a flock of flying robots in a natural scene populated with landmarks (Figure 2.16a). The task of each participant was to command the flock (from distance) using gestural (visual) or auditory commands. The system was not autonomous and the authors performed the study in a Wizard of Oz manner [149].

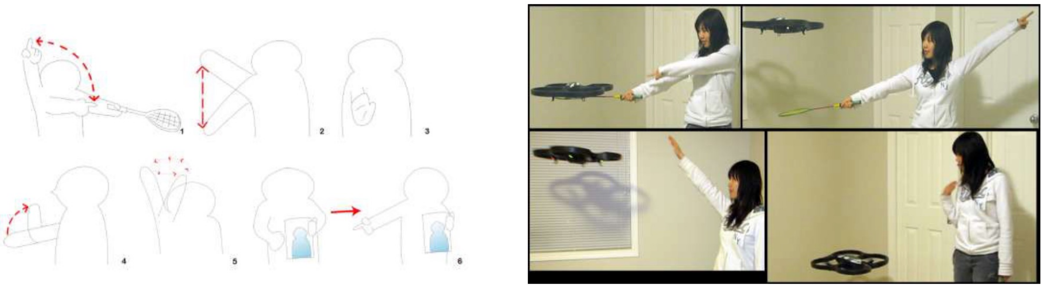
In the first part of the study, the participants were asked to command the flock (as a whole) to search the environment by directing them towards each landmark, simulating a search and rescue mission. When the flock reaches a landmark, the user is informed if the missing person is found or not through a message shown on the screen. The goal of this part of the study was to observe the natural commanding modalities that are preferred by users and identify the challenges for such an interaction scheme. The results indicate four major commanding modalities used by participants: high level voice commands such as *go to a landmark*, low level voice commands such as *go left.* herding gesture (*moving arms*



(a) The virtual scene and a few of its landmarks

(b) UAVs and their projected virtual shadows as seen by the user

Figure 2.16: Virtual reality environment used by Jones et al. [77] in their study (© 2010, IEEE)



(a) The gesture vocabulary (From 1 to 6): *take-off and land; raise and lower; stop; come; circle; and find*

(b) Communicating commands to a UAV through gestures

Figure 2.17: Ng and Sharlin [126]'s proposed idea for collocated and direct interaction with UAVs (© 2011, IEEE)

through space as if pushing UAVs in the desired direction) and pointing gesture (*Indicating direction by raising an arm*).

Major difficulties reported by the participants were lack of depth perception due to distance from the flock and lack of feedback from the UAVs, both made it difficult for the users to understand whether if their command is understood and executed by the flock. One interesting observation from this study was that herding gestures happened more frequently with low level voice commands while pointing gestures were associated with high level commands. In the second study, the authors fixed the depth perception issue by introducing false shadows (Figure 2.16b) and presented the valid multi-modal command combinations (pointing gesture/high level voice, herding gesture/low level voice) to the participants in advance. The results indicate that users significantly preferred the high level command combination to interact with the flock over the low level combination.

Ng and Sharlin [126] were among the first researchers to explore the idea of collocated and natural (direct) interaction with real flying robots. Inspired by interaction schemes used by humans to communicate with birds (e.g falconers and hawks), the authors performed a preliminary study on how natural interaction with flying robots are perceived by human

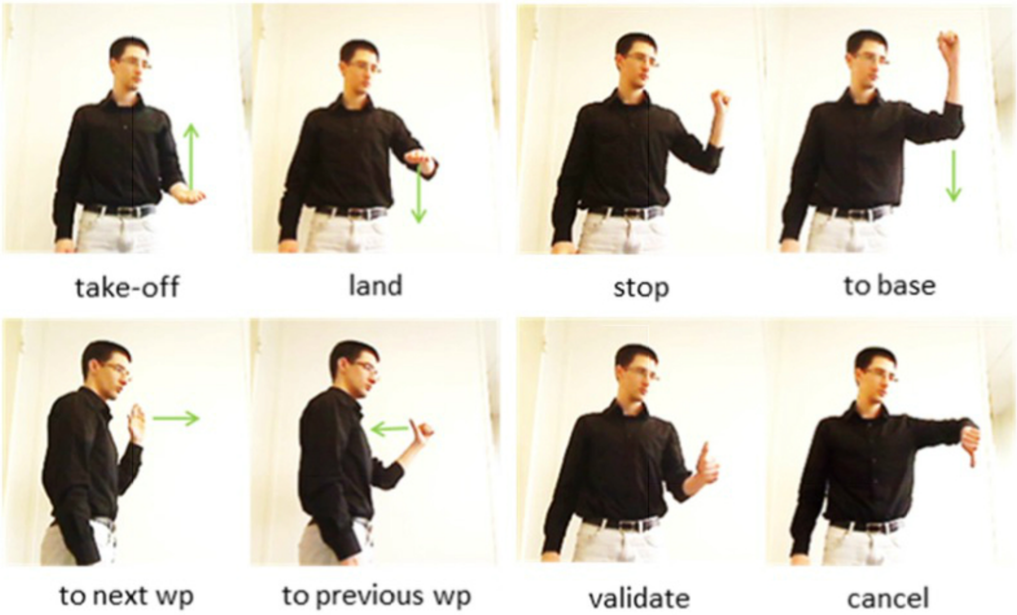
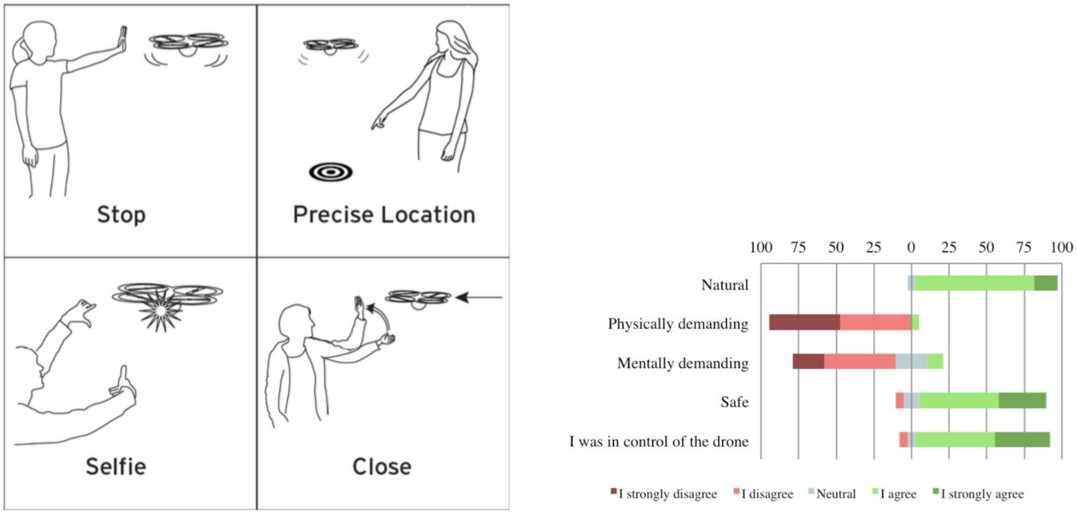


Figure 2.18: Gesture vocabulary designed based on Taralle et al. [175]'s study (© 2015, ACM)

users. Ng and Sharlin first crafted a set of hand and arm gestures to communicate commands to a UAV. The command set consists of *takeoff*, *land*, *raise (ascend)*, *lower (descend)*, *stop*, *come [to me] (approach)*, *circle* and *find* commands. The gesture vocabulary for executing each command is shown in Figure 2.17a. The authors then performed a series of Wizard of Oz experiments with two participants (one adult and one 11 year-old boy). In each experiment (Figure 2.17b), the participant was asked to command the UAV using the designed gestures set from close proximity while the UAV was flying. The UAV in use was Parrot AR-Drone quadrocopter which was being manually controlled by a human operator to execute participant's issued commands. The authors observed that participants were *very engaged* when performing gesture based interaction with a flying robot and interacted with it as if it was a pet. They conclude that natural interaction with a flying robot is *easy to understand and perform.*

Defining efficient gestural vocabularies for natural interaction with UAVs has been further studied both in the context of indirect and non-embedded human machine interfaces [136, 137] and direct and embedded human-UAV interfaces [175]. The goal of the latter work by Taralle et al. [175] is to define an efficient gesture vocabulary for interaction between infantrymen and their accompanying UAVs in battlefields using a bottom-up approach. The authors asked a group of 39 volunteers (10 civilian and 29 military personnel, divided into three groups) to propose, elect and evaluate gestures for the following tasks: *takeoff, land, to next waypoint, to previous waypoint, stop, to base, validate* and *cancel.* The gestures were required to be one-handed and intuitive, not to contain large movements and



(a) The getsure-action pairs with the highest agreement score among participants

(b) The participants' subjective ratings of their interaction experience

Figure 2.19: Results from the study by Cauchard et al. [25] (© 2015, ACM)

not to be too tiresome. The first group performed the proposal, the second group elected the most appropriate gesture for each task and the third group evaluated the understanding of each gesture by matching it to an action based on their interpretation. From the total of 160 proposed gestures, 46 unique ones were presented to the second group. The final eight elected gestures for each command are shown in Figure 2.18. The association score calculated based on the matching performed by the third group is 94%. Although the primary focus of the study was to define gestures to be used in the battlefield, the tasks are general enough so the proposed gesture vocabulary is applicable to other domains that involve issuing similar high level commands to UAVs such as search and rescue.

In a recent work by Cauchard et al. [25], the authors studied how users interact naturally with flying robots. Similar to [77], the authors asked a group of 19 participants to communicate a set of 18 commands *naturally* to a flying quadrocopter (DJI Phantom 2 with propeller guards) in an outdoor setting[9]. Similar to the settings of [77] and [126], the UAV was being controlled by a human operator and the users were aware of this fact. Tasks were split into five categories: *within body frame* commands such as *fly closer*, *outside body frame* commands such as *fly further away*, *general motion* such as *takeoff*, *relative to user* commands such as *follow* and *photo* commands such as *take a selfie*. Each participant first performed all the commands in random order, then completed a questionnaire about their interaction experience and finally performed four of the tasks again. In total, out of 414 valid interactions, 86% of them included a gesture (body, hand or arm gesture), 38% included a sound and 26% contained both. The authors calculated the agreement score for

[9] Video demonstration: `https://www.youtube.com/watch?v=vrWF3t7a_HU`

every 54 task-modality pair based on the method proposed in [186]. The results indicate that 44% of task-modality pairs have an agreement score greater than 0.5. This indicates that for 44% of the pairs, the users agree on using a certain modality for its corresponding task. Although the authors do not provide a fine-grained gesture and sound vocabulary for natural interaction with flying robots based on these data, they report four task-gesture pairs with highest agreement scores among participant (Figure 2.19a).

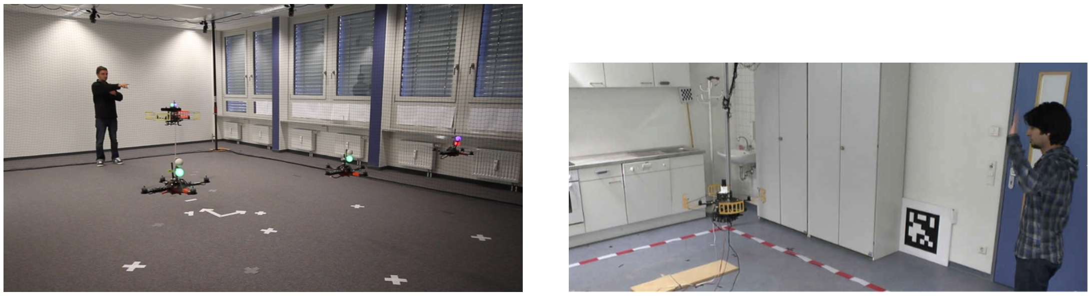
Based on their individual subjective ratings (Figure 2.19b), the participants found their interaction experience *natural*, *safe* and not *physically* or *mentally demanding*. Majority of the users expressed that they felt in total control of the drone. Similar to Ng and Sharlin [126], the authors observed that users treated the flying robot *as if it were an animate being: a person, a group of people, or even a pet*. Finally, preliminary analysis on proxemics reveal that [surprisingly] all participants felt safe enough to bring the UAV within 10ft ($\sim 3m$) of themselves. More specifically, 7, 9 and 3 participants brought the robot to their *intimate* (1.5ft), *personal* (4ft) and *social* (10ft) space (as defined by Hall [70]) respectively. This is particularly interesting since to best of our knowledge, this is the first work to provide data on safe approach distance with real flying robots. A previous study on this topic by Duncan and Morphy [51] does not provide a decisive result.

2.2.2.2 Practical Systems

The literature on practical and autonomous systems that allow direct and situated communication of intent and commands to UAVs is rather sparse. A major challenge in such systems is to perceive the commands issued by the user using usually limited computational resources available on-board. As we will show by surveying these works, visual cues such as gestures are by far the most dominant means of communication followed by audio and physical cues. In this section, we first briefly introduce common techniques used in human-computer and human-robot interaction systems to detect human gestures from visual input, then survey the state of the art on practical, direct and situated communication of intent from humans to UAVs.

In their survey on gesture recognition techniques, Mitre and Acharya [107] define gestures as *"expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of: 1) conveying meaningful information or 2) interacting with the environment."*. Gestures are either static (i.e a fixed posture) or dynamic (i.e. with temporal variation)[10]. They can also be broadly categorized as *hand and arm gestures*, *head and face gestures* and *[whole] body gestures* [107]. Gesture recognition techniques consist of three different phases: detection (of body parts), tracking and recognition [148]. Similar to what we have discussed so far in this chapter, detection can be done either through instrumenting the user or more naturally through embodied sensing.

[10] Please refer to [81] for a comprehensive taxonomy



(a) The prototyping environment of Lichtenstern et al. [95] (© 2012, IEEE)

(b) The UAV follows a human and responds to his hand gestures (Naseer et al. [125]) (© 2013, IEEE)

Figure 2.20: Two examples of using on-board RGB-D sensors (i.e. Microsoft Kinect) to interact with a UAV

Methods based on instrumentation usually rely on tangible interfaces such as gloves, special body suits or marker based optical tracking. Natural methods on the other hand rely on embodied sensors such as cameras to perform the detection. Robustness to body part occlusions as well as to viewpoint and intra-class variations are most important challenges for designing vision based gesture recognition systems [31].

Karam [83] found hand gestures as the most common gesture among humans for communication, hence he suggests those as the most natural gesture for human-computer interaction. To detect and recognize hand gestures (and body gestures in general), gestures can either be represented as 3D models such as 3D skeleton models or as appearance based models such as silhouettes and motion models [19]. In their recent survey on hand gesture recognition systems, Rautaray and Agrawal [148] identify color, shape, pixel values, 3D models and motion as the most common features used for vision based gesture detection in the literature. Hidden Markov models, Finite State Machines, Dynamic Time Warping and general classifiers such as K-Nearest Neighbor (KNN) and Support Vector Machines (SVM) are the most common techniques for the recognition part [107, 148].

Lichtenstern et al. [95]'s prototype system is one of the earliest examples of autonomous, situated and direct interaction with flying robots[11]. In their setup (Figure 2.20a), a Microsoft Kinect sensor is mounted on an Asctec Pelican quadrocopter hovering in front of the user. The skeleton tracking results obtained from Kinect's RGB-D data is used to detect and relay commands to a group of three Asctec Hummingbird quadrocopters. All flying robots use a motion capture system (in an indoor environment) to autonomously hover and navigate. In this work, interaction initiation is implicit and happens when the robot enters the field of view of the Kinect sensor (or equivalently the Pelican's field of view). Based on the user's tracked skeleton, her right arm pointing gesture is detected and used to select a corresponding Hummingbird. The user confirms her selection by touching her right arm

[11] Online demonstration video: `https://www.youtube.com/watch?v=oF3EcwNuO9Y`

with her left hand. When selection is done, the user's relative right hand movements are mimicked by all selected robots (e.g lifting the hand causes the robots to takeoff or ascend). Although the paper does not provide any technical details about the implementation, the computation is most likely done off-board the Asctec Pelican. Lichtenstern et al.'s work is similar in nature to bench-top RGB-D based human-machine interfaces that use skeleton tracking data to control and command a UAV (e.g. [40, 128, 136, 137, 158]). However, this is the first work to demonstrate embodied RGB-D based sensing for Human-UAV Interaction.

In the prototype environment of Lichtenstern et al. [95], the RGB-D sensor is almost stationary since the robot which carries the sensor is always hovering. Naseer et al. [125] further studies the problem of person detection, following and gesture recognition on-board a non-stationary flying platform. Most researchers rely on machine vision software provided by Microsoft[12] or the OpenNI project[13] to perform person detection and skeleton tracking on the data coming from a RGB-D sensor[14]. Naseer et al. noticed that the static background assumption made by those libraries (and their underlying algorithms) prevents them from performing reliable person detection and skeleton tracking when the depth camera is mounted on-board a non-stationary UAV. The solution proposed by Naseer et al. is to stabilize the depth image using the ego-motion estimation data coming from an on-board inertial visual navigation system.

The platform used in Naseer's work is an Asctec Pelican quadcopter equipped with a forward facing Asus Xtion Pro Live RGB-D sensor and an upward facing monocular camera. A set of ceiling mounted Augmented Reality markers help the UAV estimate its position in a world-fixed coordinate system. An Extended Kalman Based sensor fusion framework [184] fuses these data with inertial readings from the UAV's on-board sensors to estimate the position of the camera in world coordinates (T_{world}^{cam}). The position of the depth sensor at each time step k (T_{world}^{k}) is calculated using fixed and pre-calibrated transform from the depth sensor to the mono-camera:

$$T_{world}^{depth_k} = T_{world}^{cam} \times T_{cam}^{depth} \tag{2.9}$$

To cancel the ego-motion of the depth camera, a virtual static camera is initialized from its initial position (T_{world}^{static}). At each time step, the corresponding 3D point cloud of the depth image is re-constructed based on the pinhole camera model:

$$p_k(i, j, z) = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} \frac{(i-c_x)z}{f_x} \\ \frac{(j-c_y)z}{f_y} \\ z \end{bmatrix} \tag{2.10}$$

[12] https://dev.windows.com/en-us/kinect

[13] http://structure.io/openni

[14] For details about underlying algorithms, please refer to [167]

In the above equation, $p_k(i,j)$ is the corresponding position of an image pixel (i,j) with the depth of z in the cam_{depth_k}'s coordinate system. f_x, f_y, c_x and c_y are camera's intrinsic parameters indicating focal lengths and optical centers respectively.

At each time step, the current depth image is used to re-construct the respective 3D point cloud using camera intrinsics. Given the current transformation between the depth camera and the static virtual camera ($T_{static}^{depth_k} = T_{world}^{static^{-1}} T_{world}^{depth_k}$), each 3D point is first transformed into the virtual camera's frame: $p_{static}(i,j) = T_{static}^{depth_k} \times p_k(i,i)$, then back-projected into the virtual camera's image plane:

$$\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \frac{f_x x_{static_k}}{z_{static_k}} + c_x \\ \frac{f_y y_{static_k}}{z_{static_k}} + c_y) \end{bmatrix} \tag{2.11}$$

The resulting stabilized depth image (static camera's image) is first smoothed, then fed into OpenNI's skeleton tracker to detect a user who is facing toward the quadrocopter. The interaction is initiated when the UAV detects a person and successfully tracks its joints. At this point, the position of the torso of the human is transformed (from the static camera's frame) into the world coordinates to generate waypoints. These waypoints are tracked by the UAV's position controller which generates the follow-the-user behavior. While following the user, the relative depth between the left/right hand joints and the torso is used to detect raise left/right hand gestures. In a series of experiments, Naseer et al. [125] show that the depth stabilization increases OpenNI's tracking performance from 30% to 95% and 16% to 86% while the UAV is hovering and traveling a rectangular path respectively. The system was able to achieve a 92.5% gesture recognition rate. In an end-to-end demonstration of the system[15], the UAV would follow a user upon detecting her and respond to her left hand gesture command by taking a picture of her (Figure 2.20b).

Using RGB-D sensors exemplified by the Microsoft Kinect imposes some limitations on embodied human-UAV interaction systems. Firstly, the weight of these sensors exceeds the payload capacity of small UAVs and their accompanying libraries require powerful on-board computing resources. Secondly, since these types of sensors rely on structured light to estimate the depth, their performance degrades significantly in outdoor settings and in direct sunlight. We anticipate that with advances in sensing and computing technologies, these limitations will be gradually lifted. The second generation of Microsoft Kinect sensor, Google's Project Tango[16], Intel RealSense technology[17], Intel's Next Unit of Computing (NUC)[18] and NVidia's Jetson GPGPU computing platforms[19] are examples of advances

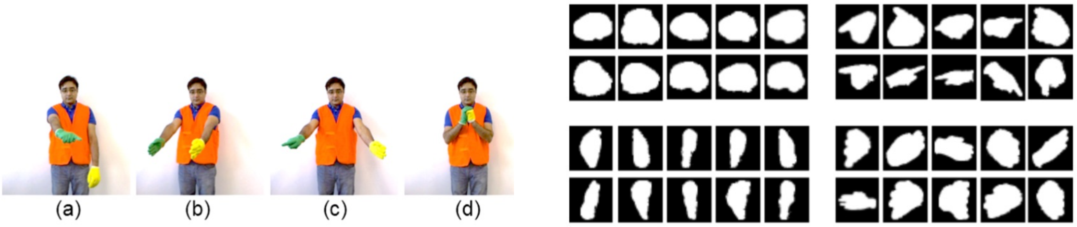
[15] https://www.youtube.com/watch?v=iaEKh4JYgqo

[16] https://www.google.com/atap/project-tango/

[17] http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

[18] http://www.intel.com/content/www/us/en/nuc/overview.html

[19] http://www.nvidia.ca/object/jetson-tk1-embedded-dev-kit.html



(a) The gesture vocabulary for robot selection. *(a) individual robots, (b) group of robots, (c) individuals and groups, (d) all robots*

(b) The features used to train the individual/group binary classifier for gesture *(a)* and *(b)* for Figure 2.21a respectively. Top row shows a few positive examples while bottom row show a few negative ones.

Figure 2.21: The gestures and features used by Nagi et al. [123] for their Human-multi UAV interaction system (© 2014, IEEE)

in underlying technologies which will benefit embodied sensing and computing on-board UAVs.

Nevertheless, the use of monocular cameras for human gesture and activity recognition on-board UAVs has also been explored by researches [31, 108, 123]. However, most of the state of the art systems simplify the computer vision problem of detecting human gestures by instrumenting users with tangible devices or special clothing. The prototype system of Miyoshi et al. [108, 109] is an example of such systems. In this work, the user's hand, which is covered with a glove of known color is detected by the UAV on its downward facing monocular camera video stream. A simple color segmentation based computer vision pipeline runs on-board the UAV to perform the detection. Another modality, the sound of a whistle, is used for interaction initiation and sending takeoff and land commands to the UAV. While hovering, the UAV flies "above the user's hand" while maintaining a fixed distance using its ultrasound distance sensor. The UAV also flies in the direction of hand movements. This way multiple users are able to pass around the UAV among themselves.

When a user is interacting with multiple UAVs, prior to communicating commands, the user needs to select the target UAV from the group. This selection cannot be made implicit if the user is in the field of view of multiple UAVs. Selecting, grouping and commanding multiple UAVs is the subject of a recent study by Nagi et al. [123]. The setup consists of a group of networked AR-Drone quadcopters with on-board forward facing cameras and off-board computing in an indoor environment. The UAVs perform color based segmentation (similar to Miyoshi et al. [108, 109]) to detect the human and position of her hands in their camera feed. In addition to color segmentation, the UAVs employ the cascade classifier of Viola and Jones [179] to detect the human's face.

The interaction is initiated when the user is in the field of view of the UAVs and his body, hands and face are detected by the UAVs. The UAVs actively maintain a human centric circular formation (uniform distribution, facing the user, fixed altitude) by estimating the pose of the human's face using a non-linear regression algorithm [121]. The gesture vocabulary for selecting UAVs consists of four static gestures (Figure 2.21a). These gestures are detected by each UAV independently and only when the motion of the human sensed by the UAV is negligible. This sensed motion is caused by the movement of the human and/or the motion of the flying robot. Each UAV maintains a circular buffer of pair-wise relative distances between the centroid of the gloves and the jacket (three components). This temporal estimation of optical flow of detected human parts are averaged and normalized by each UAV. The gesture detection pipeline is only executed when this value is below a certain threshold.

To detect gestures, the system extracts 30 geometrical shape features from the detected glove blobs for the left and right hands (i.e. convexity, aspect ratio, perimeter, etc). A multi-class Support Vector Machine (SVM) with non-linear Gaussian kernel classifier is first trained on a dataset of labeled ground-truth data. Each UAV uses an instance of this classifier to obtain a probabilistic decision vector that indicates the relative likelihood of each gesture in the vocabulary. Robots communicate their individually obtained vector with each other and run a distributed consensus algorithm [62] to agree on the issued gesture by the human.

As mentioned earlier, in such human-multi UAV setups, UAVs not only should agree on the issued gesture (command), they also need to agree on to whom (individual or group) the command is intended. To solve this problem, Nagi et al. trained a separate binary classifier for each three subset selection gestures of Figure 2.21a to determine if the gesture is towards an individual UAV or towards the whole group. As shown in Figure 2.21b, each classifier is trained on a subset of training data with positive examples of hand/palm gesture from the target robot's field of view and negative examples from the other robots' points of view. When a subset selection gesture is detected by the group, the second classifier corresponding to that gesture is executed by each UAV. If the individual selection gesture is detected, UAVs execute a distributed election algorithm based on the results obtained from the second classifier to agree on the selected UAV. This way the user can incrementally add UAVs to the selection. Similarly for group selection, the robots distributively agree on the robots that have the best view of the left and right hands (boundary robots), which consequently enables the user to simultaneously select all UAVs that are located within the selection cone.

Most of the experiments presented in the original paper by Nagi et al. [123] are performed on emulated and pre-recorded data. In general, the system performs well on detecting large human or scene motions, recognizing gestures and selecting individuals or groups of UAVs using them. A demonstration video (with an extended gesture vocabulary) is available from the following link: `https://www.youtube.com/watch?v=G2tyV2USjG8`.

Pourmehr et al. [142] took a multi-modal approach for selecting an individual or subset of UAVs from the group and commanding them. Similar to Nagi et al. [122, 123], a team

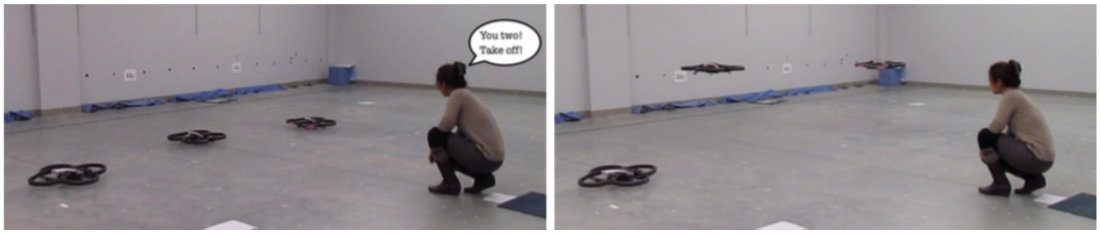
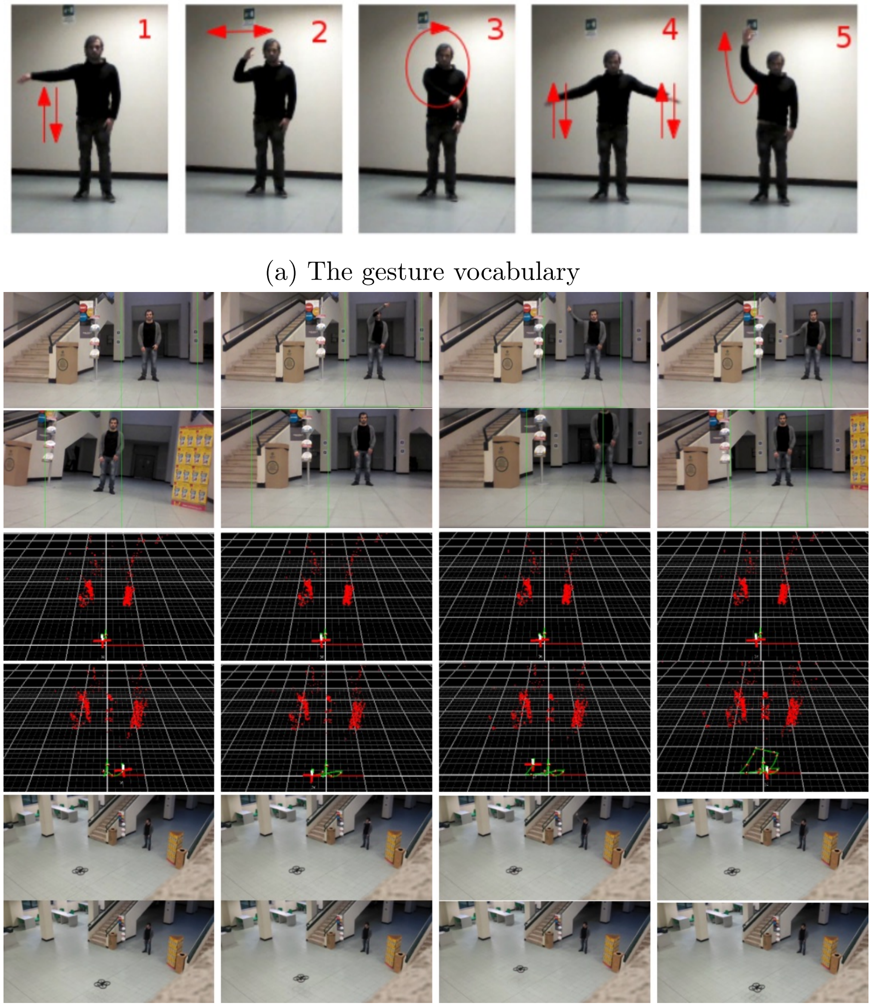


Figure 2.22: Multi-modal interaction system of Pourmehr et al. [142] for Human-mutli UAV interaction (© 2013, IEEE)

of multiple AR-Drones utilize their on-board forward facing camera to detect human faces and to initiate the interaction. In order to agree on which robot is being looked at, the UAVs communicate their so-called "face scores" distributively and perform a leader selection algorithm to reach consensus. "Face score", first introduced by Couture-Beil et al. [32] is defined as the number of neighboring detected windows by Viola and Jones [179]'s cascade classifier that cluster into a single face detection. Since the classifier is trained on frontal faces, when the user is directly looking at a robot, its "Face score" is higher compared to when she is not. A voice recognition module detects the spoken words from a predefined vocabulary uttered by the user. The grammar used for selecting the robots, consists of the word *you* followed by the desired number of robots. Pourmehr et al. [142] examined two methods of robot selection: sequential and simultaneous. In sequential selection mode, after saying the desired number of robots, the user individually looks at each robot and incrementally adds them to the group. Conversely in simultaneous selection mode, the user looks in the center of adjacent group of target robots after saying the selection phrase. The system allows the user to modify her selection by looking at a single robot and saying the phrase '*not you*' or '*and you*'. The selected group of robots can be further commanded to perform a task by subsequent voice commands such as '*take off*' (Figure 2.22). In a series of experiment [20], the authors showed that while incremental selection takes longer than simultaneous selection, this selection method leads to more accurate selection.

Costante et al. [31] propose a transfer learning approach for detecting and personalizing gestures for situated human-UAV interaction. Gestures are represented by encoding temporal variations of a Histogram of Optical Flows over region of interests in the input frames with a Fisher Kernel [106]. For each user, a short data gathering phase is performed to generate a small set of training data based on a vocabulary of five gestures (Figure 2.23a). The system uses face detection to define the region of interest in the image and face recognition to uniquely identify the user. Viola and Jones's face detector [179] and Ahonen et al. [3]'s Local Binary Patterns + SVM based face recognizer is utilized at this step. The core idea of the paper is to use a database of labeled and pre-recorded gestures gathered either from other users' interaction with the system or from the web to learn a set of user-

[20] Demonstration video: `https://www.youtube.com/watch?v=I8sJud-0Apw`



(a) The gesture vocabulary

(b) A sample interaction sequence: First two rows are from UAV's field, while the last two shows a third person view. The middle rows are UAV's trajectory. The user initiates a circle gesture which commands the UAV to follow a rectangular trajectory.

Figure 2.23: The transfer learning based gesture recognition system of Costante et al. [31] (© 2014, IEEE)

specific and personalized gesture classifiers. To achieve this, once the user is recognized, its personal training set is used to learn a set of distance functions to all the gestures stored in the database using a stochastic optimization approach. The user's novel gestures are then classified using these distance functions in a nearest neighbor manner.

The authors show that the proposed transfer learning based gesture recognition approach outperforms linear classifiers that are trained either on the user's small training set or the whole or subset of the gesture database excluding the user's training set. The database in those experiments consists of pre-recored data from three other users interacting with an airborne AR-Drone and the Keck 5,9 an 14 gesture recognition dataset [97]. In an end-to-end demonstration experiment (Figure 2.23b), the authors show that the system is able to successfully detect and track the user while responding to her commands.

2.2.3 Communication of Intent from UAVs to a Human

In collocated interaction between humans and UAVs, it is important that humans be aware of UAVs' internal state as well as its intents either through implicit or explicit feedback from the UAV. Jones and Rock [78] identify being able to "talk" as important requirement as being able to "listen" for an autonomous agent. Through proper feedback, the user can understand if the UAV correctly understands her intents and if the UAV is functioning properly. These in turn decrease user's cognitive workload and improve her awareness and safety. These requirements are generally true for any human robot interaction system. However, due to the inherently unstable nature of most UAV designs and their relatively higher momentum compared to ground or tabletop robots, the safety concerns are more important. As noted by Szafir et al. [173]:

"Any potential misunderstandings regarding robot intentions may damage human-robot rapport, prove detrimental to task efficiency, reduce trust in automation, and may even be dangerous for the human collaborator. Alternatively, being able to understand robot intentions and predict where, when, how far, and how fast it will move may enable users to work and collaborate with AAFs [Assistive Flying Robots] more effectively."

Drury et al. [48] define HRI awareness for single Human-Robot Interaction as "Given one human and one robot working on a task together, HRI awareness is the understanding that the human has of the location, activities, status, and surroundings of the robot; and the knowledge that the robot has of the human's commands necessary to direct its activities and the constraints under which it must operate.".

In case of multi-human multi-robot interaction systems, this definition is extended to cover pairwise human/robot awareness cases as well as human's overall mission awareness. When an HRI awareness information that should be provided is not provided by the HRI system, an *HRI awareness violation* occurs, as defined by the authors. By observing and analyzing participating teams at a major international robot search and rescue competition, Drury et al. conclude that all critical incidents happened during the competitions *were primarily due to a lack of human-robot awareness of location and surroundings.* Since the robots were remotely tele-operated in that setting, awareness of location and surrounding was crucial to successful and safe execution of the mission. Similarly, we believe that in the context of collocated and direct human-flying robot interaction, human awareness of status and activities of UAVs plays an important role in the successful interaction between humans and UAVs.

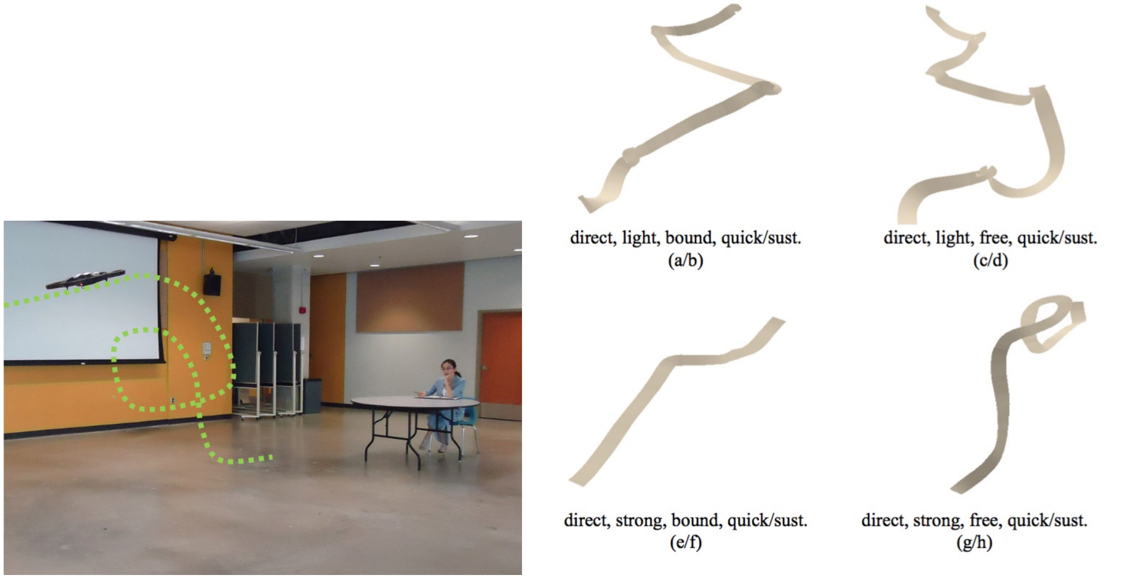
Two of the papers we surveyed in the previous section, Pourmehr et al. [142] and Nagi et al. [123], use LED lights on-board AR-Drone UAVs to communicate feedback (states) to the user. Others either do not use feedback at all or rely on the UAV's motion (i.e. when

Parameter	Definition	Extreme 1	Extreme 2
Space	Movement of the UAV in the space	Indirect: the robot meanders and wanders more while moving towards the next immediate goal	Direct: the robot moves towards the next immediate goal with little deviation in path
Weight	How the robot uses the impact of its body weight during a motion	Strong: robot moves towards the next immediate goal with power or force	Light: robot moves towards the next goal more effortlessly, being less influenced by gravity
Time	Speed-related aspects of a robotic motion	Quick: robot moves towards the next immediate goal by making hurried and urgent movements that are less time consuming	Sustained: robot moves towards the next immediate goal by making lingering (low speed) movements
Flow	The continuous and ongoing aspects of robotic motion	Bound: robot moves through the movements more carefully to execute the succession of the motion precisely	Free: robot moves through movements without caring about the precision

Table 2.1: Parameters of the Laban Effort System proposed by Sharma et al. [163] for composing communicative flight trajectories (© 2013, IEEE)

following the user) as an implicit feedback signal to communicate the status of the UAV to the user. In a recent paper by Sharma et al. [163], the authors study the communication of affects from UAVs to humans through flight path decomposition. In this study, Sharma et al. adopt the Laban Effort Systems to design flight paths for UAVs and study the perceived affect on humans. Laban Effort System is a component of the Laban Motion Analysis framework [181] used for human motion analysis in performing arts.

The Laban Effort System uses four parameters, each with two opposing extremes, to define a motion within space: *Space*, *Weight*, *Time* and *Flow*. Sharma et al.'s adaptation of these parameters and their extremes in the context of UAV flight path decomposition is listed in Table 2.1. The authors asked a Laban trained artist to compose 16 flight trajectories (all combinations of effort parameters) by moving a UAV by hand. The trajectories of the UAV was recorded using a motion capture system and were subsequently used to render a smoothed version of each trajectory (Figure 2.24b). A group of 18 participant then observed the UAV following each trajectory and measured the perceived affective states (emotions) on


direct, light, bound, quick/sust.
(a/b)
direct, light, free, quick/sust.
(c/d)
direct, strong, bound, quick/sust.
(e/f)
direct, strong, free, quick/sust.
(g/h)

(a) The experimental setup

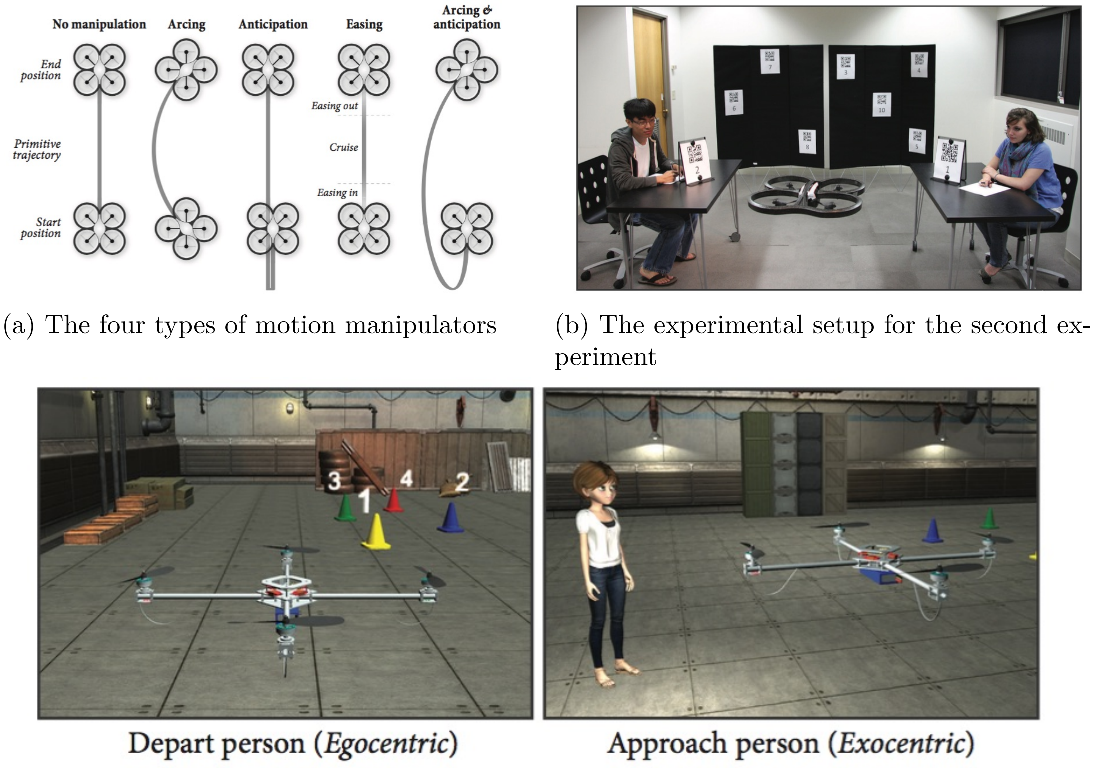
(b) Example rendered trajectories and their corresponding Laban Effort System's parameters

Figure 2.24: Communication of affect through flight path decomposition (Sharma et al. [163]) (© 2013, IEEE)

a two dimensional valence (pleasure) and arousal scale (Figure 2.24a[21]). Valence expresses how pleasant an emotion is while arousal indicates its energy and intensity. The study indicates strong correlation between the effort parameters and the perceived emotion by the participants. Most importantly, performing motions more quickly (*space indirect*), leads to increased arousal and vice verse. In addition, indirect use of space (wandering towards the goal) communicates happiness or excited state while low speed movements (*sustained time*) conveys fatigue or sadness.

Szafir et al. [173] examines how manipulation of motion primitives that define the flight path of UAVs affects the effectiveness of UAV to human communication. The authors define motion of the UAV as a composition of three components: *trajectory*, *velocity* and *orientation*. They identify 11 primitive motions for UAVs when they share an environment with a human. These primitives consist of four core motions: *Takeoff*, *Hover*, *Cruise*, *Land* and seven interactive motions: *Approach person*, *Avoid person*, *Depart person*, *Approach object*, *Avoid object*, *Depart object* and *Scan objects*. All the motion primitives are categorized as *Exocentric* by the authors since they can be perceived by any observer from a third person view. The three primitives that involve a person are additionally categorized as *Egocentric* since they can also be observed from the interaction partner's perspective (first person view). The motion manipulators used in this study are *arc* trajectories (curved trajectories as opposed to going through a straight line), *easing in and out* of velocity profiles (a velocity

[21] Demonstration video: `https://www.youtube.com/watch?v=44sdnjIUifE`



(a) The four types of motion manipulators

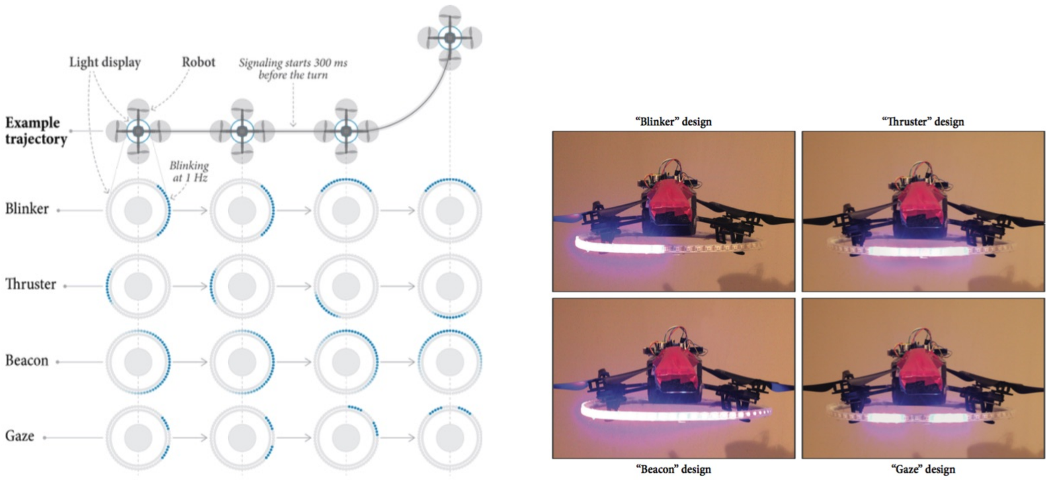
(b) The experimental setup for the second experiment

(c) The snapshots from two sample video renderings

Figure 2.25: Communication of intent from UAVs to a human through manipulation of motion primitives (Szafir et al. [173] © 2014, ACM)

profile with slow in and out) and performing *anticipatory* motions (moving in the opposite direction before starting the motion). These manipulators are shown in Figure 2.25a.

The authors first rendered a series of realistic videos of a UAV performing 7 interactive motion primitives in a shared warehouse environment. 10 motion scenarios (7 exocentric and 3 egocentric view) and 8 different combinations based on absence or presence of a particular manipulator resulted in total of 80 videos. Figure 2.25c shows two examples of such exocentric and egocentric renderings. In a study with 85 participants, the authors measured the response time and the accuracy of perception of intent by the participants. The result indicates that although no single manipulator can significantly improve the performance over the baseline, the combination of them can. Based on these findings, the authors performed a second study, this time with a real UAV, traversing a flight path in front of a participant (Figure 2.25b. In this study, Szafir et al. hand crafted a set of manipulators for each of 11 motion primitive (e.g. *arc* and *ease* for avoid person and *anticipate* for *approach object*) and asked 24 participants to rate their perceived usability, motion naturalness and safety with and without these manipulators. The results indicate that such manipulations not only significantly increase usability ratings, but also result in more natural motions and increased feeling of safety.



(a) The feedback signals to communicate the next flight direction

(b) THe physical implementation of each signal on an AR-Drone UAV

Figure 2.26: Communication of intent from UAVs to a human through light feedback (Szafir et al. [174] © 2015, ACM))

In a follow-up work, Szafir et al. [174] explored the possibility of using light feedback on-board UAVs to communicate its direction of movement. This is a critical safety aspect when UAVs are collocated with humans which are not necessarily familiar with interpreting UAVs actions based on their motions. The authors' proposed hardware consists of an array of individually addressable color LEDs mounted underneath a UAV on a circular fixture, providing a 360° viewing angle. Szafir et al. designed four different animated signals to communicate the UAV's planner motions to the user as well as its transition to and from hovering. These signals are named *blinker*, *beacon*, *thruster* and *gaze*. Figure 2.26a shows how these signals change over time for a particular planner trajectory. Figure 2.26b shows a snapshot of the physical implementation of these signals on an AR-Drone quadrocopter.

In a user study with 16 participants, the authors asked each participant to observe a UAV flying in close vicinity and mark if the UAV is flying to a predefined set of target locations correctly. The participants were unaware of the real intent of the study and the existence of the light feedback. The UAV would randomly fly to a wrong target location in each trial while always communicating its next [true] flying direction 300 milliseconds in advance. The study was designed to see the effect of four designed signals on speed and accuracy of each participant performing the monitoring task compared to the baseline (no feedback). The results show that the participants noticed the feedback design and recognized that it conveys robot's intent. They also indicate that all signaling methods (except *beacon*) significantly improved the performance results over the baseline. Qualitatively, participants found the robot's communication clear and intuitive. More specifically, the users rated *blinker* as highly intuitive and *thruster* as less intuitive. They also felt confident about their understanding of the robot's direction when the UAV used the *gaze*, *thruster* and *blinker* signals to communicate its direction. Under all four signal designs, the users perceived the UAV as a collaborative partner in a work environment, however they felt that the UAV made their task easier, compared to the baseline, only when it was communicating through *blink* and *gaze* signals.

2.3 Approach and Repositioning in Human-UAV Interaction

In the previous section we provided a few examples in which motion based feedback is utilized for communication of intents and affects from UAVs to their interaction partners. However this is not the only case where the movement or location of a UAV might be controlled in a Human-UAV interaction setting. In application domains such as search and rescue, personal photography, goods transportation and personal training the position or flight trajectory of the UAV can be controlled to facilitate further interaction or to fulfill the application goal. As an example, when the interaction is initiated by a human in need in a search and rescue scenario, the UAV approaches the human to provide her with a tele-presence link to first responders or to deliver medical aid. Another notable example is a personal filming drone which approaches and/or follows an athlete after interaction initiation or upon receiving a command from its interaction partner.

In general, solutions for tracking and following humans by mobile robots consist of two main components: A human localizer component which continuously estimates the location of the target human with respect to the robot and a control scheme. The controller [22] steers the robot to follow, track or approach the user given the location of the human and constraints imposed by the environment and the robot platform (i.e. kinematic and dynamic constraints).

Similar to the discussion in Section 2.2.2, one can either instrument the user or use external sensing devices to facilitate the human localization task, or use embodied sensing and direct methods to localize humans with respect to a robot. There exists many examples of Human-Ground Mobile Robot interaction systems that use embodied sensing to track and follow uninstrumented people (e.g. [9,64,80,124,139]). This is not the case for Human-Flying Robot interaction systems. In addition to the relatively young age of this field of research, the limited payload capacity of many consumer/research UAV platforms limit the type and fidelity of sensors and computing devices that these platforms can carry which in turn makes the task of human detection and localization more difficult. We provided a survey on challenges and methods related to employing human feature detectors on-board UAVs (in the context of interaction initiation) in Sections 2.2.1.1 and 2.2.1.2. This is the reason that many consumer UAVs rely on GPS or inertial-based tracking devices carried by their users (also known as *virtual tethers*) to perform person following and tracking. Example

[22] We use the term *controller* here as a general term that refers to both planners and reactive controllers

consumer UAVs that use virtual tethers to follow people include *Solo* by 3D Robotics[23], *Hexo+*[24], *AirDog*[25] and *Lily*[26].

As our survey in Sections 2.2.1 and 2.2.2 indicates, computer vision techniques that use monocular or depth sensing cameras are the most common techniques used by researchers for interaction initiation and communication of intents in situated and direct Human-UAV interaction systems. In systems that use pedestrian detectors to initiate the interaction, it is possible to directly use the location of the detected human in the image plane as an input to the tracking controller. However pedestrian detectors (like any vision-based object detectors) are prone to false positives and false negatives. A common approach used in practical systems to overcome the shortcomings of pedestrian detectors is to employ visual trackers to track the location of the human once detected. In Human-Flying Robot interaction systems, using this approach provides a major benefit. Regardless of which interaction initiation method is used by the system, the visual tracker can track the location of the target human after the interaction is initiated. In the remainder of this section we survey related work on uninstrumented human tracking with UAVs. Later on, in Chapter 5 we show how we integrate a state of the art visual tracker into our approach controller to bring a flying robot to close proximity of a user once the interaction is *explicitly* initiated.

Tracking and approaching humans with UAVs using vision-based perception can be considered as a special case of *visual target detection and tracking* with UAVs. In a recent comprehensive survey on navigation and control of unmanned rotor-craft systems, Kendoul [85] categorizes research projects related to application of visual target detection and tracking for unmanned rotor-craft systems as (i) vision-based landing on a known target; (ii) vision-based landing on an unknown target; (iii) vision-based static target tracking; (iv) vision-based mobile target tracking; (v) target tracking for automatic landing; (vi) horizontal target approach; and (vii) moving ground target tracking.

From these categories (i), (iii) and (vi) are closely related to the approach and repositioning components of end-to-end Human-Flying robot interaction systems. A common practice for designing systems to perform automatic landing or static target tracking is to use predefined targets (or *helipads* in case of landing) to facilitate the visual state estimation process. This way the position (with metric scale) of the UAV can be recovered with respect to the target using a monocular camera. The automatic landing systems of Saripalli et al. [159], Shakernia et al. [162], Yang et al. [190] and Ling et al. [98] are notable examples of systems that fit this category. For Human-Flying Robot Interaction systems that employ human feature detectors this approach can be used to design tracking or approaching controllers. Compared to a specifically crafted target, a bounding box in the image plane returned by a human feature detector or a visual tracker provides fewer constraints for

[23] `https://3dr.com/follow-me-mode/`

[24] `https://hexoplus.com/`

[25] `https://www.airdog.com/`

[26] `https://www.lily.camera/`

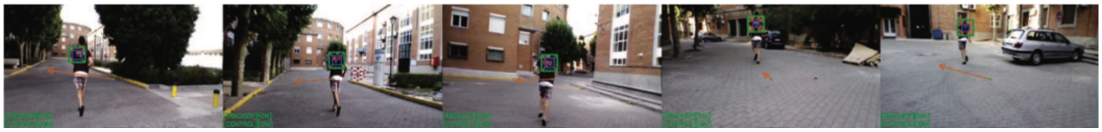


Figure 2.27: A sample sequence from Pestana et al. [134, 135]'s experiments where an AR-Drone 2.0 quadrocopter followed a pre-selected human for 45 seconds covering 120-140 meters of distance (© 2013, IEEE)

recovering the pose of the UAV with respect the user. Therefore practical systems usually impose some simplifying conditions or assumptions about the environment or target to overcome this issue. We provide some examples of these assumptions in the context of Human-UAV interaction systems in the remainder of this section as well as in Chapter 5.

To facilitate the state estimation and control of the UAV while landing on (or tracking) an object, control systems might also fuse other sources of information (such as inertial or GPS) with the data obtained from a vision system (e.g. Proctor and Johnson [144] and Hermansson et al. [74]). This is also a viable approach for control systems designed for tracking or approaching humans in Human-UAV interaction systems.

The special case of approaching a human from distance as part of a Human-Flying Robot Interaction scenario is similar in nature to the horizontal target approach category. However, it must be noted that despite many similarities between these categories, approach to a horizontal target is more challenging, specifically when a monocular camera is used. This is due to the fact that landing targets are usually placed on the ground level which makes it straightforward to use the estimated altitude of the UAV directly as the depth (distance) of the object. Recovering the depth of an object which is positioned freely with respect to the UAV using a monocular camera without any prior on the target size or configuration is not possible. Without proper depth estimation (or approximation) safe and smooth approach towards a human is not feasible.

Pestana et al. [134,135] designed an Image Based Visual Servo (IBVS) controller for the task of tracking (following) user-defined objects using a low-cost consumer quadrocopter. The user first defines the region of interest on the video feed streamed by the UAV. The system then uses the Tracking-Learning-Detection visual tracker of Kalal et al. [82] to track the location of the selected object on the image plane while refining its appearance model over time. The location and size of the tracked object in the image plane is the input to the IBVS controller. The controller's task is to keep the object in the center of UAV's field of view and fly at a fixed distance with respect to the object. The authors use the normalized size of the object as an approximation for its depth and propose a heuristic decoupling strategy that uses the pitch and yaw angles of the UAV and intrinsic parameters of the camera to map image-based error vectors to the four controllable degrees of freedom of the quadrocopter. Four Proportional-Derivative (PD) controllers generate the reference velocity vectors for the on-board flight controller to track. In a series of demonstrative

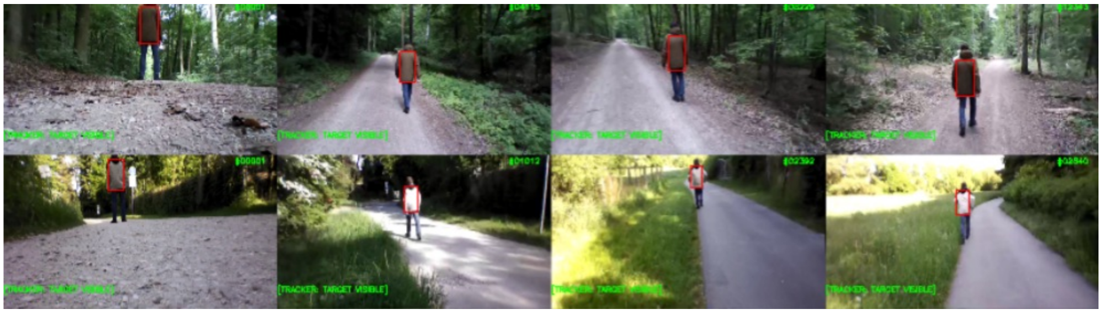


Figure 2.28: Sequences from Haag et al. [69]'s long-term human following demonstrative experiments using an AR-Drone 2.0 quadrocopter (© 2015, IEEE)

experiments[27] in outdoor settings, the authors showed how their system could track moving objects including people. For the task of people following, the operator would select the logo on the t-shirt of the user for the controller to track (Figure 2.27).

Haag et al. [69] modified this system and replaced the short-term tracking component of the TLD tracker with a state of the art correlation based visual tracker - with further improvements to the long-term tracking component - to achieve long-term human following by a quadrocopter in outdoor settings. In at least one of their demonstrative experiments, the UAV could follow a human jogging in a forest for 10 minutes (Figure 2.28). We integrated this long-term visual tracker in our end-to-end flying robot interaction system. We will provide more details about this tracker in Section 5.3.3.

The most recent example of a person following system in Human-UAV interaction is the active vision framework of Danelljan et al. [39]. The goal of this system is to implement a human following behavior on-board a quadrocopter equipped with a monocular camera. Unlike the previous two papers, in this system a HOG based human detector (ref. Section 2.2.1.2) is used to implicitly initiate the interaction (behavior). The authors propose to combine the output of the human detector with a visual appearance-based tracker and a probabilistic multi-object tracker. This is to improve the robustness of the system against false positives/negatives and to keep the focus of attention on the same person in cases where multiple people are in the field of view. Danelljan et al. also propose a depth estimation method based on the following assumptions (i) the size of pedestrian is known and (ii) the ground plane is flat. Given the size of the object, the height of the camera (UAV) from the ground plane and the tilt angle of the camera, they propose a formula to recover the distance of the target pedestrian to the camera. The method and the assumptions are conceptually similar to the method we propose for depth estimation in the end-to-end Human-Flying Robot Interaction system which we describe later on in Section 5.3.4.1. The authors use a Bayesian filtering approach to estimate the position and velocity of the target. These estimates are fed into a so-called *leashing control module* to generate reference

[27] Video, code and dataset available at `http://robotics.asu.edu/ardrone2_ibvs/`

velocities for the on-board flight control. This controller calculates a waypoint on the line connecting the UAV to the target at the desired following distance, then uses a proportional controller to generate desired reference velocities for linear and yaw degrees of freedom of the quadrocopter. In a series of indoor demonstrative experiments, the authors validated their leashing controller.

It is worth mentioning that none of the aforementioned state of the art uninstrumented human following systems with UAVs ([39,69,135]), report the number of conducted experiments, their failure rates and quantitative data about the performance of their systems.

2.4 Conclusion

In this chapter, we surveyed the state of the art in Human-Flying Robot Interaction. We first studied human factors and interface design for remote interaction with UAVs in Section 2.1. As discussed in that section, the level of shared autonomy in various control loops of an Unmanned Aerial System directly affects the cognitive load and effectiveness of its operators. We also discussed two major strategies for managing the level of shared autonomy in such systems: Management by Consent and Management by Exemption. Furthermore, we studied how different paradigms for presenting information in remote interaction systems can affect the performance of their operators and surveyed related work on ground control station designs for remote interaction with UAVs.

In the next section (Section 2.2), we focused on situated, embodied and natural interaction with UAVs. We introduced the main motivations for such interaction schemes, example application domains and a concrete example scenario. We broke down proximate interaction with UAVs into three components for initiating the interaction, approach and repositioning and communication of commands from humans to UAVs and communication of intent and states from UAVs to a human. We surveyed implicit interaction initiation methods through human feature detection and related human detection techniques in Section 2.2.1. As our survey shows, visual sensors are the most popular sensor on-board UAVs to detect humans. We also noticed that, most of the practical interaction systems that incorporate human detectors, utilize focus of attention techniques to reduce the computational load of human detectors. We then studied the literature on explicit interaction initiation methods through moving object detection in Section 2.2.1.3.

We introduced notable human studies that deal with effective natural and embodied communication of intent from humans to UAVs in Section 2.2.2.1. Defining gesture vocabularies, efficient interaction modalities and natural interfaces were the focus of the papers surveyed in that section. We then surveyed the rather sparse literature on practical systems for embodied and natural communication of intent and commands from humans to flying robots in Section 2.2.2.2. We introduced practical systems and human studies that focus on efficient communication of intent from UAVs to humans through motion decomposition or visual feedback in Section 2.2.3. Finally in section 2.3 we provided an overview of challenges associated with approach and repositioning in Human-Flying Robot Interaction and surveyed the few practical systems that enable a flying robot to follow an uninstrumented human. All three survyed systems operate in relatively close-range to people and use implicit interaction initiation to trigger the behavior.

In general, none of the surveyed systems fully demonstrate an integrated system that implements all three phases of Human-Flying Robot Interaction, thus can not be considered as *end-to-end* interaction systems (*cf.* Definition 1). Table 2.2 summarizes the most promising works from the state of the art and how close they come to implement an autonomous end-to-end Human-Flying Robot interaction system. As it is shown in this table, each system lacks one or more components of an end-to-end interaction system. Furthermore, none of the related work implement an explicit interaction initiation method. This is particularly important since in many of the application domains, the presence of a human does not necessarily mean that the human intends to interact with the UAV. Even in domains where implicit interaction initiation through human/face detection is acceptable, one needs to consider cases where multiple human/faces might be in the UAV's field of view. Given the state of the art, we see an opportunity to push it towards more realistic Human-UAV interaction scenarios such as in outdoor (natural) environments and over longer distances ($> 10m$).

In the following chapters we describe our methods to perform explicit interaction initiation with a flying robot using a monocular camera (Chapter 4), selecting and commanding teams of UAVs using gaze and gestural commands from close-range (Chapter 3) and our end-to-end interaction system that combines those two components with a cascade approach controller and light based feedback (Chapter 5) which works in outdoor environments over distances up to $25m$.

	Naseer et al. [125]	Nagi et al. [122]	Lim and Sinha [96]	Danelljan et al. [39]	Our goal
Year	2013	2014	2015	2015	-
Interaction Initiation	Implicit (Person Detection)	Implicit (Person Detection)	Implicit (Ped. Detection)	Implicit (Ped. Detection)	**Explicit**
Approach	No	No	No	No	**Yes**
Repositioning	Yes (4 DOF)	Yes (4 DOF)	Yes	Yes	**Yes**
Commands to UAV	Hand gestures	Hand gestures	No	No	**Hand gestures**
Feedback to Human	No	Yes (low bandwidth)	No	No	**Yes**
Non-instrumented	Yes	No	Yes	Yes	**Yes**
Embodied Sensing	Yes	Yes	Yes	Yes	**Yes**
Autonomous	Yes	Yes	No	Yes	**Yes**
Long-range Interaction (>10m)	No	No	No	No	**Yes**
Main sensor	RGB-D Cam.	Mono. Cam.	Mono. Cam.	Mono. Cam.	**Mono. Cam.**
Tested environment	Indoors	Indoors	Outdoors	Indoors	**Outdoors**

Table 2.2: Comparison of state of the art in situated and end-to-end interaction with flying robots with the goals of this thesis

Chapter 3

Close-range Situated Interaction with a Group of Flying Robots through Face Engagement and Hand Gestures

The work in this chapter represents the first demonstration of a close-range Human-UAV interaction system that enables an uninstrumented human to create, modify and command teams of flying robots. To create a team and to initiate the interaction with a UAV, the user focuses attention on an individual robot by simply looking at it, then adds or removes it from the current team with a motion-based hand gesture. Another gesture commands the entire team to begin task execution. Robots communicate among themselves by wireless network to ensure that no more than one robot is focused, and so that the whole team agrees that it has been commanded. Since robots can be added and removed from the team, the system is robust to incorrect additions. A series of trials with two and three very low-cost UAVs and off-board processing demonstrates the practicality of this approach.

3.1 Introduction

Selecting and commanding individual robots in a multi-robot system can be a challenge: interactions typically occur over a conventional on-screen human-computer interface (e.g. [104]), or specialized remote control (e.g. [37]). Humans, however, can easily select and command one another in groups using only eye contact and gestures. In this chapter we work towards a direct communication method for human-UAV interaction. In particular we avoid the need for the human to be instrumented in any way, and all interaction is mediated by the robot's on-board sensing and actuation.

The method we introduce here is an extension of the work by Couture-Beil et al. [32] that uses face engagement to select a particular robot from a group of robots. In that system, once selected, a single robot engaged in one-on-one interaction with the user. In this chapter we compose a multi-flying robot team from the population of flying robots by adding or removing the currently selected robot, then command the whole team at once. Couture-Beil et al. used wheeled mobile robots which were stationary for the human-robot interactions. In this chapter we use flying quadrotor robots which are continuously moving. The constant movement of cameras attached to flying robots make the problem of vision mediated human robot interaction much more challenging.

The contributions of this chapter are: (i) the first demonstration of HRI control of a flying robot by an uninstrumented human using only passive computer vision; (ii) the first demonstration of dynamically creating and modifying robot teams by an uninstrumented human; and (iii) the first demonstration of interaction initiation with a flying robot by face-engagement.

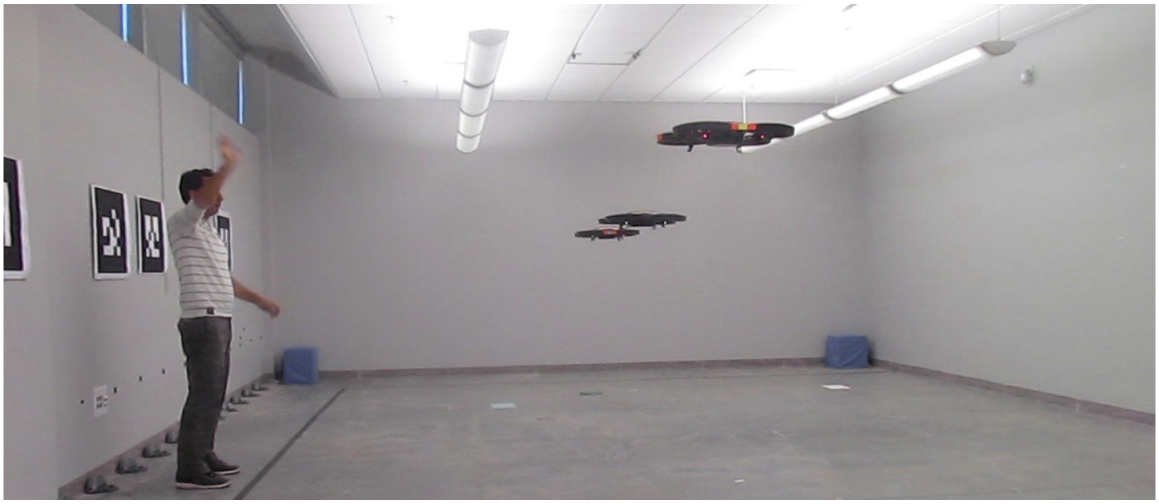


Figure 3.1: An uninstrumented person creates and commands a team of three UAVs using face-engagement and hand gestures

3.2 Background

Throughout this chapter, we will use the term *face engagement*, as coined by Goffman [65], to describe the process in which people use eye contact, gaze and facial gestures to interact with or engage each other.

3.2.1 Gaze and Gesture in Human Robot Interaction

Researchers have argued that exploiting stereotypical communication cues (without instrumentation) can achieve natural human-robot interactions [50]. Gaze and body movements (gestures) are two such communication cues.

There is a large literature on gaze tracking techniques; Morimoto and Mimica provide an in-depth survey [115]. Applications of gaze trackers can be found in fields ranging from psychology to marketing to computing science; many interesting examples are given in the survey provided by Duchowski [49].

In an experiment by Mutlu et al. [118], gaze is used to regulate conversations between, a humanoid robot, and two human participants. The study showed that (among other things) gaze was an effective tool for yielding speaking turns and reinforcing conversation roles. Kuno et al. [92] present a museum tour-guide that only responds when directly looked at. A telephoto lens is used to capture a high quality image; the robot then estimates if the user is looking at it by detecting if the nostrils are centered between the eyes. Couture-Beil et al. [32] showed that this method can be extended to select individual robots from a population by using explicit wireless communication between robots to perform a distributed election algorithm to unambiguously decide which robot (if any) was being looked at directly. Since the election is completed in a few tens of milliseconds and is essentially imperceptible to the user, the user's experience is simply that as you look from robot to robot, the selected robot is always "the one I am looking at right now". In the following sections we show that this method is also effective for flying robots.

We briefly introduced major computer vision-based gesture recognition techniques in Section 2.2.2.2. Several gesture-based robot interfaces exist; we do not attempt to provide an exhaustive survey, but rather mention some interesting examples. Systems may use static gestures – where the user holds a certain pose or configuration – or dynamic gestures – where the user performs a combination of actions. Waldheer et al. use both static and motion-based gestures to control a trash-collecting robot [183]. Earlier work by Kortenkamp et al. presents a mobile robot that uses an active vision system to recognize static gestures by building a skeleton model of the human operator; a vector of the human's arm is used to direct the robot to a particular point [89]. Giusti et al. [63] demonstrated how a swarm of mobile robots can cooperatively detect a static human gesture and act upon it.

We use simple motion-based gestures to issue commands to robots once they have been selected using face-engagement.

3.2.2 Robot Selection And Task Delegation

There is little work on human-robot interfaces for multi-robot systems. Examples can be broken up into two general cases:

3.2.2.1 World-Embodied Interactions

World-embodied interactions occur directly between the human and robot, through either mechanical or sensor-mediated interfaces. Key advantages of this approach compared to a conventional Graphical User Interface (GUI) include the possibility for users to walk freely among the robots rather than being tied to an operator station. Also since robots observe humans directly using their on-board sensing, they may not need to localize themselves in a shared coordinate frame. Examples include work by Payton that uses an omni-directional Infrared (IR) LED to broadcast messages to all robots, and a narrow, directional IR LED to select and command individual robots [37], work by Naghsh et al. [120] who present a similar system designed for firefighters, but do not discuss selecting individual robots, and work by Zhao et al. [193] which proposes the user leaves fiducial-based "notes" (e.g. "vacuum the floor" or "mop the floor") for the robots at work site locations. Xue et al. [188] introduced a fiducial design for imperfect visibility conditions and combined them with user-centric gestures.

3.2.2.2 Traditional Human-Computer Interfaces

Rather than interacting directly with robots, a traditional human-computer interface is used to represent the spatial configuration of the robots and allow the user to remotely interact with the robots. Examples of human-robot interactions which occur through a traditional interface include work by McLurkin et al. [104] that presents an overhead-view of the swarm in a traditional point and click GUI named "SwarmCraft", and work by Kato that displays an overhead live video feed of the system on an interactive multi-touch computer table, which users can control the robots' paths by drawing a vector field over top of the world [84]. Similar to Zhao et al.'s fiducial-based notes [193], Kolling et al. [88] designed a user interface that allows the operator to place virtual beacons in a simulated robot environment.

3.2.3 Human Robot Interaction with UAVs

Traditional human computer interfaces have been used extensively to design control interfaces for single [30, 33, 103] and multiple [133] UAVs. Uninstrumented interfaces have also been used to interact with UAVs. Song et al. [171] describes a method for recognizing aircraft handling signals from depth data, and tested their method on a database of videos collected from a stationary (non-airborne) camera. Lichtenstern et al. [95] describe a prototype system in which gestures directed at one UAV carrying a Kinect (active RGB-D) sensor can be used to control other UAVs. Jones et al. [77] performed a user study to investigate how different modalities can be used to control a swarm of simulated UAVs in a virtual reality environment. Naseer et al. [125] developed an autonomous system that enables a single quadrocopter to follow a human and respond to hand gestures using active RGB-D sensor with vision-based ego-motion cancellation.

Our method is different from the aforementioned works due to our use of vision-based gestures (obtained from a passive monocular camera) to select and command a team of airborne UAVs. Now that affordable UAVs are available we expect this area to grow rapidly.

3.3 Method

To demonstrate our approach, we use a group of unmodified AR-Drone 2.0 quadrocopters[1]. These inexpensive aircraft have a built-in attitude controller and a forward-facing 720p HD camera. Video from the camera and flight control data are streamed via 802.11 wireless network to a control computer. A practical challenge when using this setup is that all user software is run externally and is therefore subject to large network delays: we observe around 200 milliseconds end-to-end latency. Engel et al. [54] have shown that it is possible to explicitly model the communication delay and use monocular Simultaneous Localization and Mapping (SLAM) to accurately navigate a single quadrocopter. Another successful position controller is presented by Krajník et al. [90]. They determined the drone's dynamic model and implemented a PID controller that would hover the drone over a mobile target,

[1] http://ardrone2.parrot.com/

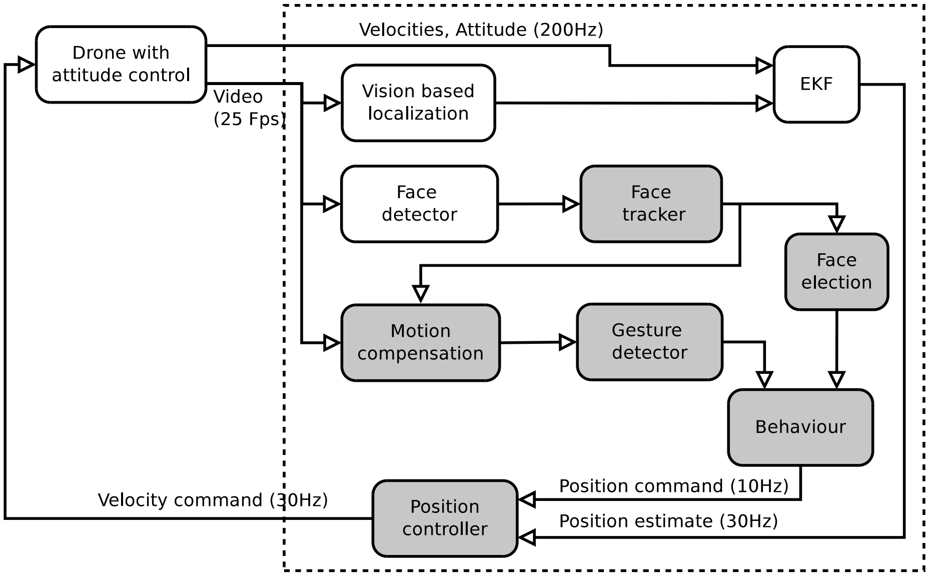


Figure 3.2: System overview, the dashed box (right) wraps the components that run on a laptop, the remainder (left) runs on-board the aircraft. Components in gray (lower right) are custom developed for this work, while third party modules with small adaptations are marked in white.

tracked by the downward facing camera. We use only the forward facing camera for HRI and localization, since the platform does not permit simultaneous streaming from both cameras.

Next we describe our approach, with an overview shown in Figure 3.2.

3.3.1 Position Estimate and Control

While the AR-Drone 2.0 is capable of generating 720p video streams, we use a lower resolution to save wireless channel bandwidth and allow us to use multiple robots. We experimented with two different 3D pose estimation methods for the robots: fiducial based and salient feature based. The fiducial based method uses the ALVAR library [182] to track the drone's position $(x, y, z, \phi, \theta, \psi)^T$ relative to fiducials mounted at known locations in the environment. Here x, y, z is the 3D location in the world frame and ϕ, θ, ψ are roll, pitch and yaw (heading), respectively. The feature-based method employs the Parallel Tracking and Mapping (PTAM) monocular SLAM system [87] to estimate each robot's pose. We use an Extended Kalman Filter (EKF) to fuse the vision based position estimate with inertial measurements from the drone's flight control computer to improve the accuracy of the estimated pose.

When robots use the fiducial based method, they are localized in the global coordinate frame, which makes the multi-robot formation control straightforward. However, this method is sensitive to fiducial occlusions. The feature-based method on the other hand is

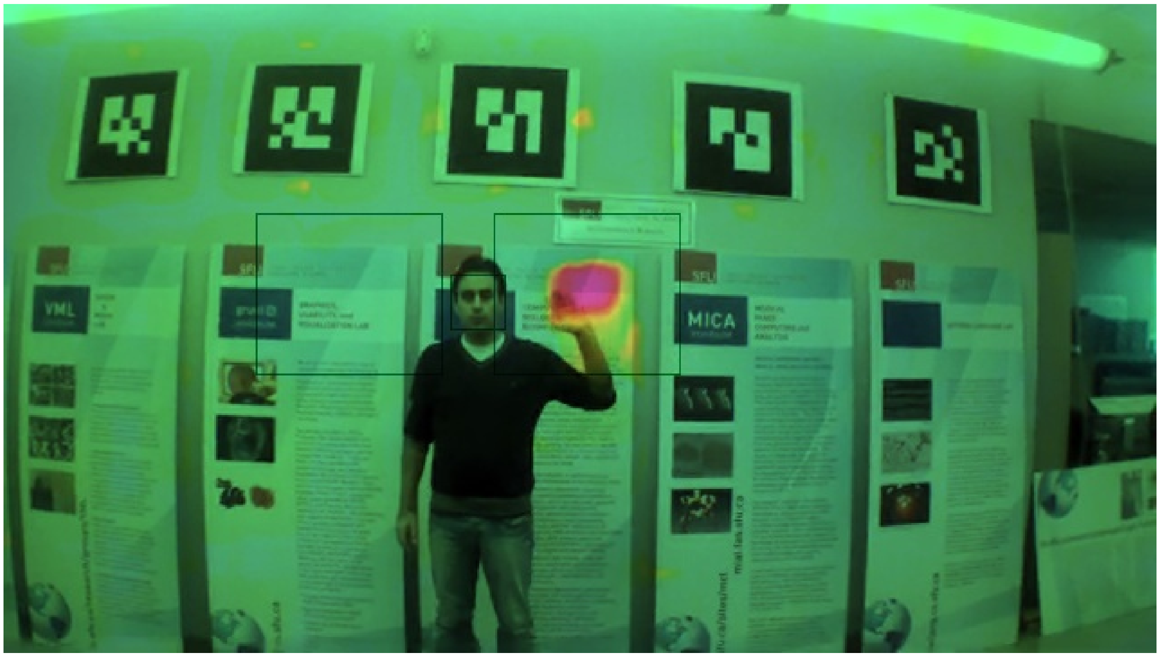


Figure 3.3: Face detection is used to locate the user, and to select the currently focused robot. Hand gestures change the state of the focused robot. This image is from the flying robot's point of view. The gesture detection regions are marked by a rectangle. (The stabilized optical flow magnitude's heat map is blended into the image.)

more robust to occlusions. However, the coordinate frame and scaling of pose estimates are not defined with respect to the world and depend on the PTAM initialization phase. Our system uses the method introduced in [54] to perform scale estimation using EKF. In our system, all robots use the same recorded video of the environment for PTAM initialization, and thus they all agree on the initial coordinate frame.

To control each drone, the position estimate and the 4-DOF target position $(x_T, y_T, z_T, \psi_T)^T$ are fed into four independent PID controllers, one for each directly controllable degree of freedom. The control output is then sent via the wireless network to the drone. In practice we find that this approach works well as long as there is sufficient distance ($> 3m$) between any two aircraft. When drones are too close together, turbulence from the down draft causes the drones to pitch and roll rapidly in an attempt to maintain their position, and the camera cannot be kept on-target for HRI. This fast movement cannot sufficiently be tracked by our position controller because of the network delay. We avoid this issue by enforcing a minimum distance of $3m$ between aircraft.

3.3.2 Face Detection and Tracking

To locate and track faces in the video stream, we use the OpenCV [21] implementation of the Viola-Jones [179] face detector. Because of the often rapid ego-motion of the airborne camera we might lose a detected face or detect several false positives. We address this prob-

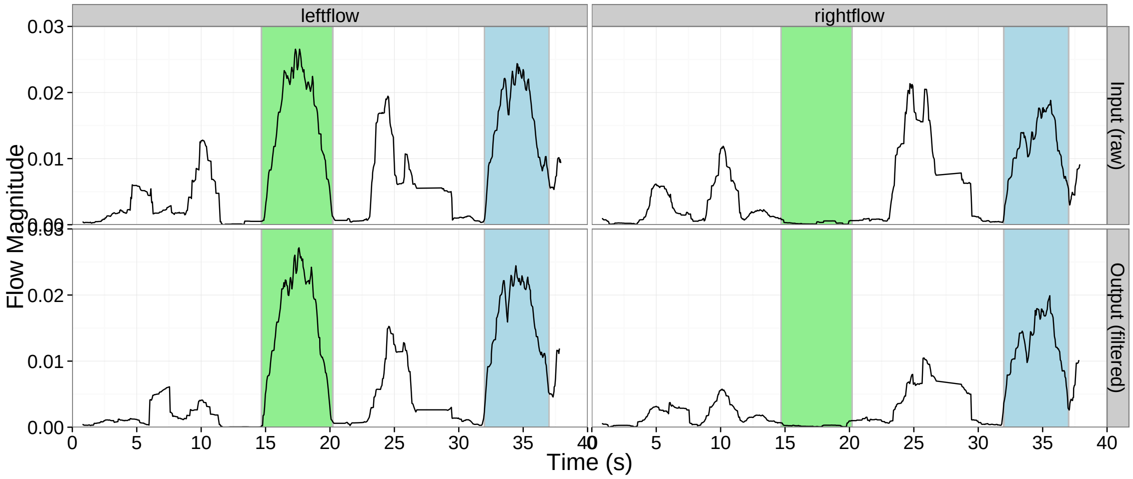


Figure 3.4: Optical flow in the left and the right hand zone; the top graph shows the unfiltered optical flow and the bottom graph shows the output of our multi-stage filter. Sections marked in green (left) correspond to left hand gesture, periods of both hands gesture are colored blue (right).

lem by using a Kalman Filter to smooth face position estimates. We use a nearest neighbor data association strategy to determine which detected face to use as the measurement input, using a Mahalanobis distance derived from the estimated covariance of candidate faces.

Information about the tracked face is used in two subsequent modules: first to partially cancel image flow due to ego-motion as described in the next section, and second to determine if the user is engaging in an interaction with the robot. Our HRI attention-focusing strategy is to engage one robot at a time out of the group by simply looking at it. Subsequent commands are addressed to the engaged robot. The challenge for the robots is to determine which robot is currently being looked at, as the user's face might be visible to several robots at the same time. As we mentioned earlier, we use a mechanism developed and successfully used earlier by Coutuere-Beil et al. [32]. The face detector is trained on frontal faces only, and we observe that the largest number of candidate face detections occur when the face is looking directly at the camera. Since the face detector is insensitive to small changes in scale or position, multiple candidate detections are often clustered around faces. We use the number of candidate detections in each cluster as a score to assess the quality of the detected face. To determine which robot sees the most frontal face, the robots perform a distributed election, each proposing their currently observed face score. If no robot has a score above a threshold, no robot is engaged, otherwise the robot with the highest score is the one being engaged by the human. Only the currently engaged robot will watch for gestural commands.

3.3.3 Motion Cancellation and Gesture Recognition

The system uses the magnitude of optical flow in fixed regions around the user's face to detect hand-wave gestures. In order to have reliable optical flow information, motion from sources other than user's hand movement in the video stream should be filtered out. We have to deal with three sources of motion in our video stream. The first is the motion of the camera caused by the motion of the aircraft stabilizing its attitude and controlling its position. The second is caused by user movements other than gesturing, and the third is a result of the hand gestures used to command the vehicle. The objective is to cancel the first two while not damping the gesture motion.

For motion cancellation and gesture recognition we define three zones in the image. The face zone is a bounding box around the face currently being tracked. The left and right hand zones are rectangles to the left and right of the face box respectively as shown in Figure 3.3. The size of the left and right zones is proportional to the size of the face zone. The hand zones are cropped if any of their corners exceed the image boundaries. This will happen when the human face is towards the edges of the image.

In the first step we mask all pixels in the hand zones to preserve the optical flow caused by waving the hands. We then calculate optical flow in the remainder of the image using the OpenCV [21] implementation of Franebäck's algorithm [56]. The median of this optical flow is an approximation of the ego-motion of the camera, which we can now remove from the original image.

Next, using the camera-motion-reduced image, we estimate the motion of the user by computing the median of the optical flow in just the face zone. The assumption is that motion of the face is a reasonable proxy overall non-gesture body motion. By removing the estimated user motion from the image we are left with an image that contains mainly the flow resulting from the gestures. The process is illustrated in Figure 3.4.

In the last step we average the magnitude of optical flow within the hand zones. For robustness to transient flow, the resulting signal is passed through a median filter with a window size of 15 frames. By thresholding the result we can detect left and right hand waving. This gives us a total of 4 states: no wave, left wave, right wave and two-hand wave. These gestures are then used by the behavioral module to command the aircraft.

3.3.4 Commanding the Vehicle

The user commands a robot by first engaging it (by looking at it) and then giving it one of the three gestures. A right hand wave means join the group. A robot that is part of the group increases its hover altitude by 0.2m. A left hand wave is the command to leave the group, consequently the aircraft returns to the original altitude. Waving both hands is the signal for the entire group to execute a mission. Note that only one robot has to be given the command to execute the mission; it will communicate this instruction to the

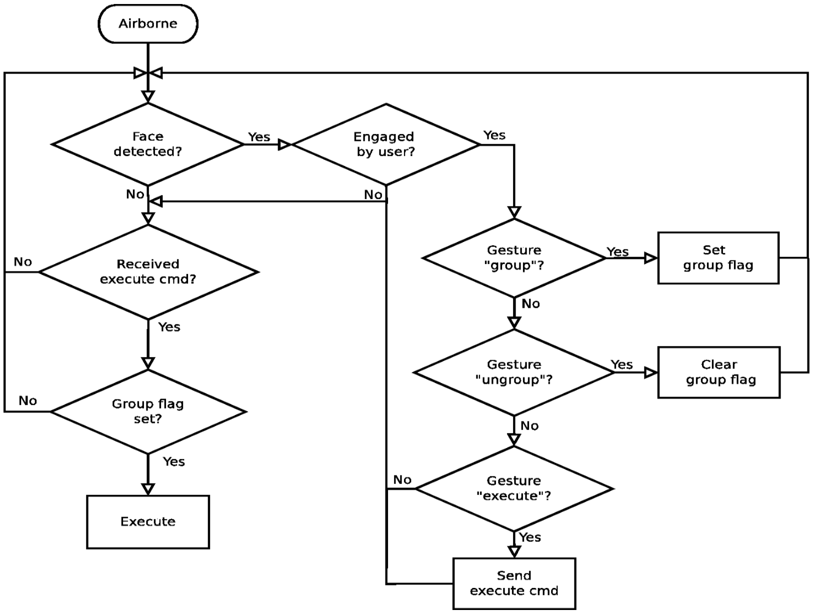
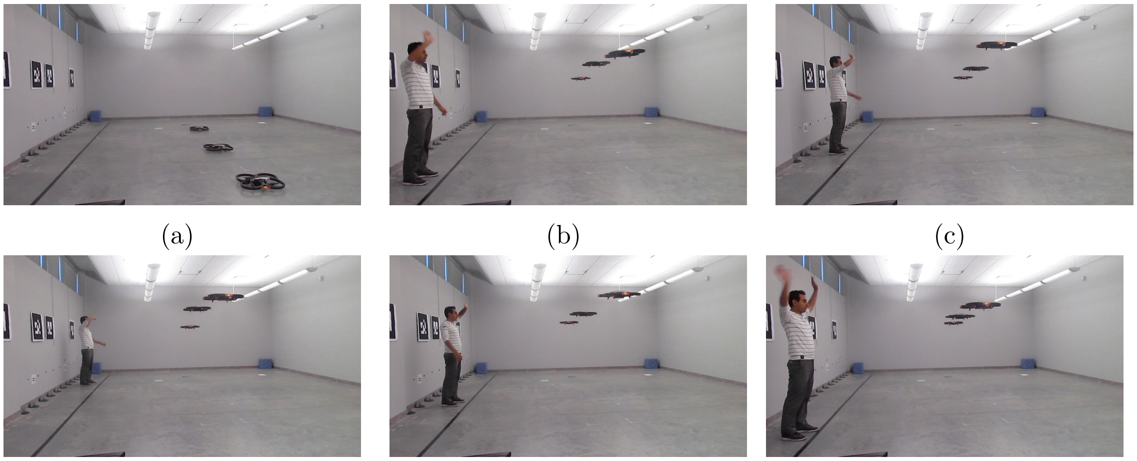


Figure 3.5: Flowchart outlining the decision tree for the robot's behavior.

others over the network and the group acts as one. In our demonstration the "mission" is either to land or perform a complete roll (flip) in place. These simple missions are a placeholder for a real mission such as search, patrol, mapping, etc. (In Section 3.4.3 we present demonstration of an autonomous UAV exploration system in which we use this interface to initiate the exploration mission). The robots also change the color and blinking frequency of their built-in LEDs to report their current state (being engaged or selected as part of the group) to the user. Informally, we found this direct feedback helps the user in the interaction process.

The flowchart of the controller is shown in Figure 3.5. We trigger take-off manually. Each aircraft, once airborne, autonomously flies to its predefined target location and tries to detect faces. If a face is detected as described above, the position controller tracks the face by steering the nose of the aircraft in the direction of the face. This is to ensure that the face is always in the middle of the image. This is not only a feedback mechanism to the user, but also keeps the hand zones from being cropped. Next, the face scores are communicated to all robots by wireless network. If a robot wins the face score election, it considers itself engaged by the user (hence the interaction is initiated) and accepts hand gestures. Left or right hand gestures set or clear a "belong to group" flag. If the *execute command* gesture is detected, the command is passed on to all other aircrafts via the wireless network. An aircraft receiving the execute command and belonging to the group will now execute the mission, i.e. land. The remaining aircraft stay airborne and wait for a user engagement.



(a) (b) (c)

(d) (e) (f)

Figure 3.6: Snapshots from a three robot experiment, in which a user is commanding three quadrocopters (Table 3.1).

3.4 Demonstration

To demonstrate this system, we performed two sets of trials with a group of flying robots and a human. All trials were performed by one expert user[2]. The arena is a 8 × 10 × 3m indoor lab environment clear of any static obstacles, shown in Figure 3.1. At startup, each robot is placed at a predefined position on the ground. During each trial, the robots take-off after receiving an external signal, then fly to their predefined target poses $(x_T, y_T, z_T, \psi_T)^T$. The main difference between two sets of experiments are the position estimation method used for each experiment and the number of participating robots.

3.4.1 Three-Robot Experiment with Marker-Based Localization

In our first experiment, we used the fiducial based position estimation method as described in section 3.3.1. Six unique 50×50cm ALVAR 2D tags were mounted on the wall behind the user as input to the ALVAR localization system. Due to low accuracy of heading estimates when the robots are looking at the fiducials with steep angles, initial poses for robots were set such that they look directly towards the fiducials. This led to a linear initial formation as shown in Figure 3.1. As a result, the human usually needs to walk along the wall into a robot's field of view first to get its attention. Once a face is seen by a robot, it yaws to track the face as described in section 3.3.2.

Fifteen trials with a total of 82 scripted interactions were executed. Table 3.1 summarizes the results. Robots were indexed from 1 to 3. In the table, the Scenario column contains a list of the interactions attempted by the user. S_i, D_i and C_i mean issue the Select (add to team), Deselect (remove from team) and Command (execute mission) gesture to the

[2] Video demonstration: `https://youtu.be/xHH3GvZ52xg`

Trial	Scenario	Gesture	Face	Success
1	S_1 S_2 C_1	3/3	3/3	Yes
2	S_1 S_2 C_3 C_2	4/4	4/4	Yes
3	S_1 S_2 S_3 C_3	4/4	4/4	Yes
4	S_1 S_2 D_1 C_2	4/4	4/4	Yes
5	S_2 S_3 ~~D_2~~ ~~C_2~~ ~~C_3~~	2/5	5/5	No
6	S_2 S_3 S_1 D_2 C_3	5/5	5/5	Yes
7	S_3 S_2 S_1 D_2 C_3	5/5	5/5	Yes
8	S_2 S_1 S_3 ~~D_3~~ C_2	4/5	5/5	No
9	S_2 S_3 C_1 S_1 C_1	5/5	5/5	Yes
10	S_1 S_2 S_3 D_1 D_2 C_3	6/6	6/6	Yes
11	S_1 S_2 D_1 ~~D_3~~ C_2 S_1 C_1	6/7	7/7	No
12	S_1 S_3 D_1 ~~S_2~~ C_3 S_1 C_1	6/7	7/7	No
13	S_3 S_2 S_1 D_1 D_2 D_3 C_2	7/7	7/7	Yes
14	S_3 S_2 S_1 D_2 D_3 C_2 C_1	7/7	7/7	Yes
15	S_2 S_3 ~~D_2~~ S_1 D_3 D_1 S_3 C_3	7/8	8/8	No
Total		75/82	82/82	10/15

Table 3.1: Result summary for three robot experiment, S_i, D_i and C_i mean issue the Select, Deselect and Command gesture to the ith robot. Unintended outcomes are marked by overstrikes.

ith robot, respectively. Unintended outcomes are marked by overstrikes. A trial with any unintended outcome is deemed to be unsuccessful. The ratio of successful to overall trials was 10/15. The success rate of individual interactions was 75/82.

To summarize the robot system behavior, we recorded each robot's altitude for the length of the trial. Figure 3.7 shows such a graph for experiment number 7. The script was to select robot 3, select robot 2, select robot 1, deselect robot 2, then command robot 3 to land. The plot shows the altitude of robot 3 is increasing at around 25 seconds, followed by robot 2 at around 30 seconds and 1 at 40 seconds, as each joins the team. The altitude of robot 2 decreases at around 45 seconds as it leaves the team. Robots 1 and 3 land at 60 seconds, while robot 2 remains hovering, as required by the trial script.

3.4.2 Two-Robot Experiment with Feature-Based Localization

In a second set of experiments we used the monocular SLAM-based pose estimation method. The main motivation was to let the robots create a formation in which they initially look at the same spot in the room. We could not arrange this with the ALVAR-based system as the robots needed to face directly towards a fiducial to maintain a stable hover. With the monocular SLAM method, this restriction is lifted and the user can stand on one spot and just look from robot to robot without moving in and out of the robots' field of view. The

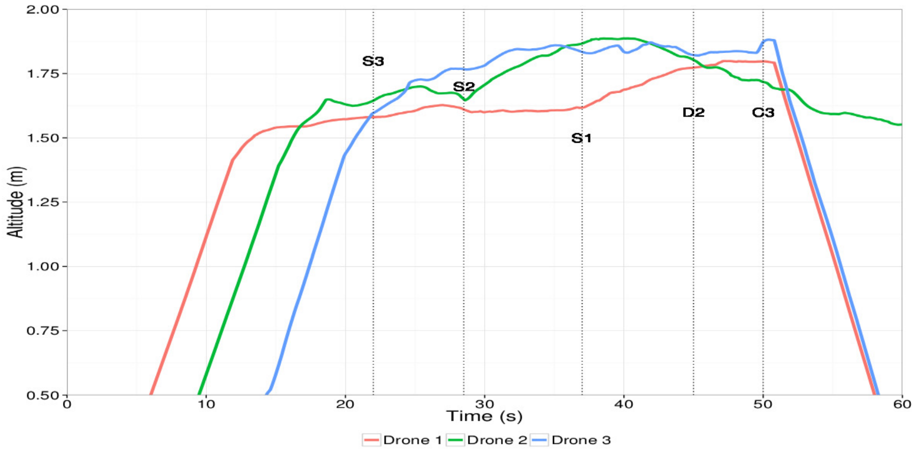


Figure 3.7: Plot of smoothed robot altitude over time during trial #7 (Table 3.1). Dotted vertical lines show the time that a specific gesture was performed. Select (S_i) adds robot i to the team, Deselect (D_i) removes robot i from the team. Team members hover 0.2m higher than non-team members. The Execute command (C_i) makes the team land.

other benefit of this method is that there is no need to instrument the environment with fiducial markers.

This system has its own limitations though. The PTAM system is not able to track the position of the robot well when the camera motion is mainly rotational. This situation happens when the robot is tracking the human's face while hovering close to its target position. We found empirically that to avoid this situation, the change in heading of the robot should be small while performing stable hovering and face tracking. This means that, as the heading angle of the robot with respect to human increases, the distance between human and the robot should increase. This new constraint, in addition to the minimum distance constraint discussed in 3.3.1, meant that we only had space for a two-robot experiment in our lab.

We performed a total number of 10 scripted trials with two drones. Table 3.2 summarizes the results. The ratio of successful to unsuccessful trials was 8/10. The success rate of individual interactions was 43/45. Figure 3.8 shows the snapshots of trial number 8.

3.4.3 Integrating Multi-Modal Interfaces to Command UAVs

In this section, we present an integrated human-robot interaction system that enables a user to select and command a team of two Unmanned Aerial Vehicles (UAV) using voice, touch, face engagement and hand gestures. This system integrates the close-range Human-Flying Robot interface that we introduced in this chapter with the "Touch-to-Name" interface of Pourmehr et al. [141] and "Feature-rich path planning algorithm" of Sadat et al. [156] and use it in a coherent semi-realistic scenario. The task of the UAVs is to explore and map a simulated Mars environment.

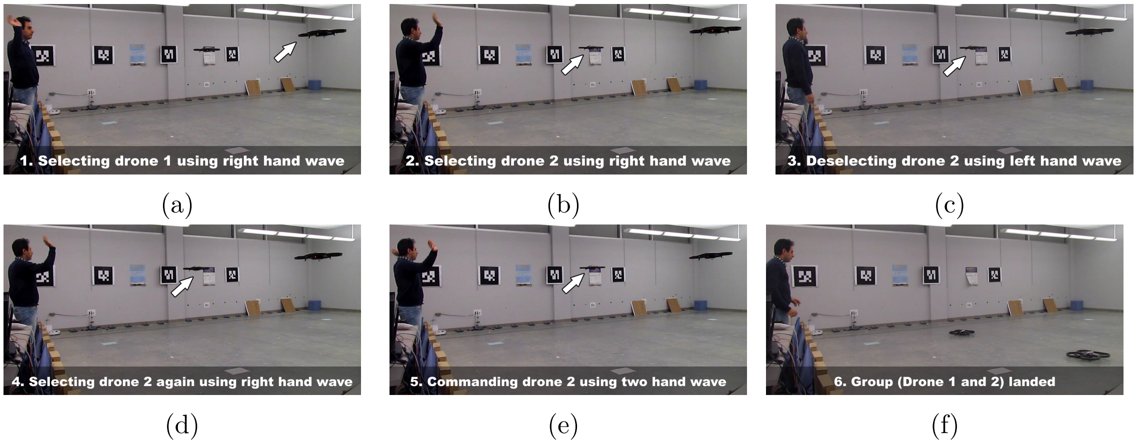

1. Selecting drone 1 using right hand wave
2. Selecting drone 2 using right hand wave
3. Deselecting drone 2 using left hand wave
(a)
(b)
(c)
4. Selecting drone 2 again using right hand wave
5. Commanding drone 2 using two hand wave
6. Group (Drone 1 and 2) landed
(d)
(e)
(f)

Figure 3.8: Snapshots from a two robot experiment, in which a user is commanding two quadrocopters (Table 3.2).

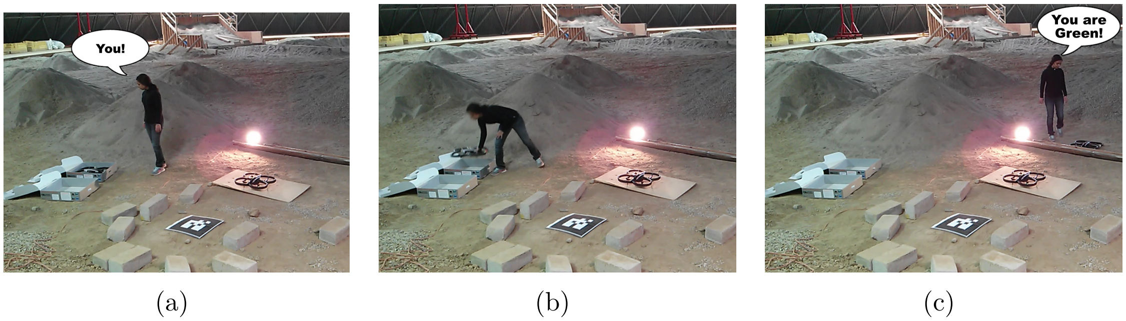


Figure 3.9: The touch-to-name interaction interface of Pourmehr et al. [141] used for interaction initiation during the multi-modal integrated demonstration (Section 3.4.3). (a) The user first announces the desired number of robots with "You" or "You N" where N is the desired number of robots (b) The user then handles the intended robot(s) (c) The user finally assigns a name to the selected robot(s) that can subsequently be used to address this robot or team.

To initiate the interaction (mission), the user needs to select a robot. To do this, we used the "Touch-to-Name" selection and naming interface of Pourmehr et al. [141]. In this method, the user first announces the desired number of robot(s) (e.g *"You"* or *"You Two"*), then gently moves the intended robot(s) iteratively. Robots compare their accelerometer readings over Wi-Fi to agree on which one is selected (Figure 3.9).

Once selected, the user names the selected robot using verbal commands (e.g *"You are Green"*). These names are then used to command the robots (e.g. *"Green Takeoff"*) [142]. Here, we use this interface with maximum group size set to one.

After taking off and while hovering, the flying robot uses the method described in this chapter to detect and track the user's face, maintain its yaw and respond to her hand gestures. A hand wave gesture (left or right) assigns an exploration task to the robot in the indicated direction.

Trial	Scenario	Gesture	Face	Success
1	S_1 S_2 C_1	3/3	3/3	Yes
2	S_1 S_2 C_2	3/3	3/3	Yes
3	S_1 S_2 D_1 C_2	4/4	4/4	Yes
4	S_1 S_2 D_2 ~~C_2~~ C_1	4/5	5/5	No
5	S_1 D_1 S_2 C_2	4/4	4/4	Yes
6	S_1 D_1 D_2 S_2 ~~C_1~~	4/5	5/5	No
7	S_2 S_1 D_2 D_1 S_2 C_2	6/6	5/5	Yes
8	S_1 S_2 D_2 S_2 C_2	5/5	5/5	Yes
9	S_1 S_2 S_1 C_2	4/4	4/4	Yes
10	S_1 S_2 D_2 D_2 S_2 C_1	6/6	6/6	Yes
Total		43/45	45/45	8/10

Table 3.2: Result summary for two robot experiment. S_i, D_i and C_i mean issue the Select, Deselect and Command gesture to the ith robot. Unintended outcomes are marked by overstrikes.

While exploring, each robot performs vision-based Simultaneous Localization and Mapping (SLAM) using their on-board monocular camera [87]. We used the "Feature-rich path planning algorithm" introduced by Sadat et al. [156] to robustly navigate the UAV while exploring an unknown environment. To terminate the mission, the user commands each robot to *come back* home (e.g *"Green come back"*). To come back, robots use the same algorithm to plan a *feature-rich* path to their takeoff position. Finally, The user asks robots to *land*. (e.g. *"Green land"*).

Similar to the setup described in Section 3.4.1 and 3.4.2, the system provides two types of feedback to the user during interaction sessions and mission execution. Robots change the color and blinking pattern of their LED lights to inform the user about their state (e.g. "tracking user's face", "exploring" or "being idle"). In addition, a text-to-speech (TTS) engine provides verbal feedback to the user whenever a robot's state changes. As an example, when the *Green* robot is asked by the user to comeback, it acknowledges by saying *"Green is coming back"*. The TTS is embedded within a general purpose web-based robot monitoring dashboard.

The video of this system in a demonstration experiment [3] shows a complete run-through of a two robot exploration mission in which the HRI worked perfectly.

3.4.4 Discussion

In all trials the face engagement subsystem was successful: the robots could successfully detect and track the user's face while running the distributed leader election algorithm. We note informally that this capability combined with the LED and altitude feedback made a

[3] `https://youtu.be/heiYPVGFnEM`

comfortable and natural-feeling method of interaction with the robots. The gesture recognition subsystem however had a total of 9 failures, 7 cases of false recognition and 2 cases of failed recognition. Examining the data, we found that false negative and incorrect recognitions occur when the motion cancellation happens to cancel a legitimate hand motion. The false recognition can also occur when the motion cancellation does not filter out all non-relevant motions.

The position control subsystem also had some failures when the marker based pose estimate of a robot became inaccurate either due to full occlusion of localization tags by the user's body or very fast human movements during an interaction. Although the robots could recover from these errors, their short-term instability forced the human to wait. After a few practice trials, the user learned to move his body so as to avoid these problems. While our goal is to design systems where such user adaptation is not necessary, we observe informally that a bit of user training can lead to a useful improvement in the performance of the current system. The occlusion was not a problem when using feature based pose estimates, however PTAM recovery after initialization from the pre-recorded video sometimes could take up to 15 seconds.

3.5 Conclusion

In this chapter we presented a computer vision-mediated human-robot interface whereby an uninstrumented user can create, modify and command a team of robots from a population of autonomous individuals in a multi-robot system from close-range. The user selects an individual as the current focus of attention by simply looking at it. The focused robot can be added/removed from the team by waving the right/left hand. The whole team is dispatched to a mission by waving both hands. We demonstrated the effectiveness of this method using a system of low-cost quadrotor robots with on-board attitude control and off-board computer vision-based 4-DOF position control. In a series of trials the robots achieved better than 90% correct execution of the user's intentions and 76% correct execution of trial interaction scripts.

The system we described in this chapter uses face engagement for implicit interaction initiation and is suitable for close-range direct interaction with a flying robot. This system partially implements elements of an end-to-end interaction system as we defined in Chapter 1. In addition to implicit interaction initiation, it provides gesture based methods for communication of commands from the human to the UAV and a low-bandwidth light and movement based feedback for communication of states from the flying robot to the human. In the following chapter we describe our method for explicit interaction initiation system through periodic motion detection that works over longer distances and in outdoor environments. In Chapter 5 we will integrate the close-range interaction system of this chapter into an end-to-end Human-UAV interaction system and introduce a more sophisticated light based feedback system for communication of intents from UAVs to a human. Furthermore, the system we described in this chapter does not implement an approach behavior and only maintains the vehicle's yaw and altitude based on the image-plane location of the user's face. We will introduce our visual servo-based approach controller as part of an end-to-end interaction system in Section 5.3.4.2.

Chapter 4

Explicit Interaction Initiation with a Flying Robot Using Stationary Periodic Motions

In this chapter we present the first demonstration of explicit interaction initiation and establishing mutual attention between an outdoor UAV in autonomous normal flight and an uninstrumented human user. We use the familiar periodic waving gesture as a signal to attract the UAV's attention. The UAV can discriminate this gesture from human walking and running that appear similarly periodic. Once a signaling person is observed and tracked, the UAV acknowledges that the user has its attention by hovering and performing a "wobble" behavior. Both parties are now ready for further interaction. The system works on-board the UAV using a single camera for input and is demonstrated working reliably in real-robot trials.

4.1 Introduction

As we discussed in Chapter 2.2.1, interaction between a Human and a UAV can be initiated either implicitly through human feature detection or explicitly through a triggering mechanism. For situated interaction scenarios using embodied sensing, no applicable solution exists in the literature. In this chapter, we propose a computer vision pipeline that runs on-board an autonomous UAV for this task.

We use the familiar periodic double-arm-waving gesture pictured in Figure 4.1 as a signal to attract the UAV's attention. The UAV can discriminate this gesture from human walking and running that appears similarly periodic. Once a signaling person is observed and tracked, the UAV acknowledges that the user has its attention by hovering and performing a distinctive "wobble" behavior. Both parties now know that they have the attention of the other and are now ready for further interaction. The system works on-board the UAV using a single camera for input and is demonstrated working reliably in real-robot outdoor trials.

The problem of detecting and tracking humans from a moving camera platform is non-trivial, and is a current research problem. It is essential for pedestrian detection in self-driving cars, and for automated surveillance from drones. We [114] and others [31, 95, 125] have previously demonstrated people-tracking for close-up HRI with small UAVs. Expanding this work outdoors and to a UAV that is flying rather than hovering, we find that the person is represented by only a few pixels in the image and may be in-view only briefly. Variable lighting conditions and fast camera motion also contribute to the challenge. Our approach is to use fast computer vision methods that require no training and can run in real-time on-board the UAV.

The periodic waving gesture is designed to be highly salient to an observer: it makes the user appear larger, and fast periodic motions are relatively rare in outdoor scenes. Thus it is amenable to computer vision detection.

The arm-waving signal is also very familiar, and we informally suggest that this is a natural way to attract the attention of any human, animal or robot that is looking for

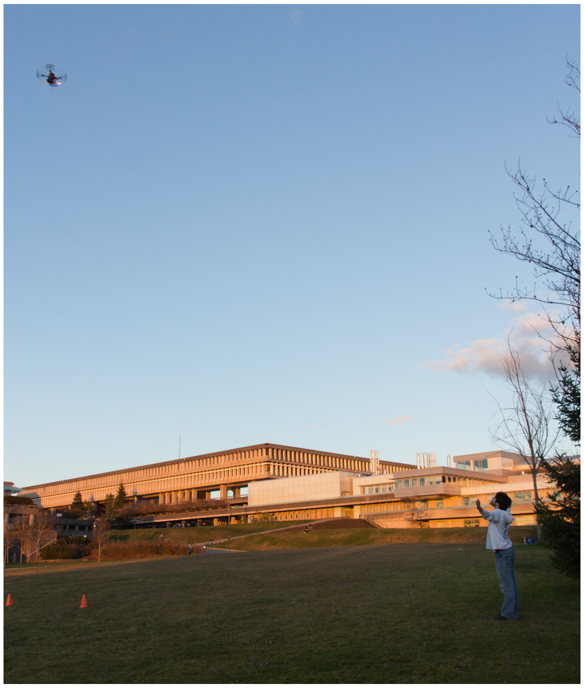


Figure 4.1: Our system in action, showing scale. A human is waving to a UAV.

you. In addition to being familiar and easy to perform, the user's intention to attract the robot can be correctly interpreted by human observers. Again, informally, we suggest that this behavior does not need to be taught to users, so our system could work in search and rescue scenarios where the subject has no robot training. In wilderness survival guides, this behavior is suggested as an effective signaling method to attract attention (e.g. [20]).

The robot signals back to the user with a high-frequency periodic "wobble" behavior as an approximation of the "wing-waggle" behavior conventionally used by fixed-wing aircraft pilots to show they have observed a person on the ground. Informally, this is readily perceived by the user as a confirmation of being attended to, probably because it looks deliberate yet distinct from the normal control motions of the vehicle.

Our vision system provides detection and tracking of candidate *humans-that-want-to-interact* for moving UAVs. It occasionally gives false positives and negatives, so should be used as part of a closed-loop system whereby the UAV has suitable failure/retry behaviors to achieve real-world robustness [143]. Our end-to-end Human-Flying robot interaction system, which we will present in the next chapter, is an example of such closed-loop system that integrates the pipeline we introduce here.

The contributions of this chapter are (i) a description of a fast approach to detect waving gestures based on a combination of well-known computer vision methods; (ii) a demonstration of the first fully autonomous UAV detecting the intention to interact from an uninstrumented person with all computation performed on-board. (iii) a complete implementation available online[1].

4.2 Background

As we surveyed in Section 2, interfaces to control UAVs can broadly be classified into two groups. Those that use conventional instrument-based Human-Computer Interfaces [101, 133] and direct and uninstrumented interfaces mostly based on computer vision techniques. Of the uninstrumented interfaces, a few have demonstrated fully integrated human-UAV interaction systems. Lichtenstern et al. [95] described a system in which gestures observed by a UAV carrying a Microsoft Kinect (indoor, active RGB-D) sensor are used to control other UAVs. Naseer et al. [125] developed an autonomous system that enables a single quadrotor to follow a human and respond to hand gestures using an active RGB-D sensor with vision-based ego-motion cancellation. Costante et al. [31] developed a person-specific gesture-based interface to command a UAV using monocular vision.

All these systems use vision-based body or face detectors to find the region of interest (ROI) in the robot's field of view for tracking the human and/or performing gesture recognition and to implicitly initiate the interaction. The RGB-D based solutions are not applicable to outdoor settings or long distances because sunlight overwhelms the projected infrared structured light field. Furthermore, state of the art human-detectors are too computationally intensive to run in real-time on a CPU and are unreliable when the person is distant (< 30 pixels tall) [47]. As a result, all existing approaches have been applied to close range interaction scenarios in indoor environments only. Our human-UAV interaction system is the first to work outdoors, while the UAV is translating rather than hovering, and over relatively long distances ($> 10m$) when the human occupies less than 5% of the image.

Instead of directly using human-detectors to find and track the human in the UAV imagery, it is possible to find and track objects using motion or other salient object detection techniques (also known as foreground object segmentation). Sokalski et. al [170] developed a system that combines contrast features, mean shift segmentation and multichannel edge features to detect static salient regions in UAV imagery. Rodriguez et. al [150] developed a real-time system to detect and track multiple moving objects from a UAV by constructing an artificial sparse optical flow field from estimated camera motion. Discrepancies between the real and artificial flow fields characterize moving objects. Camera motion is estimated by a monocular visual SLAM algorithm. Siam and Elhelw [168] developed a similar system to track multiple moving objects from a UAV by clustering feature points which are outliers in camera motion estimation step. The camera motion estimation is done by finding the homography transform between consecutive frames using tracked feature points. Kimura et al. [86] applied multi-view geometry constraints (epipolar constraint and flow-vector bound)

[1] `http://autonomylab.org/obzerver`

to tracked feature points between consecutive frames in order to detect moving objects from an airborne UAV. Our approach is similar to [86, 150, 168] in the sense that we also rely on explicit camera motion estimation and motion saliency to detect regions of interest. However since the arm waving gesture is not a strong motion cue from a distance, we also integrate tracked feature points to find salient objects.

To detect a dual arm waving gesture given a sequence of tracked ROIs, it is possible to apply two approaches: human activity recognition techniques and periodicity analysis. Human activity recognition is an active and vast area of research. Although there exist many promising human activity recognition algorithms, most are far from real-time on current hardware [2]. For the specific task of action recognition from distance, the computation time is dominated by the need for precise tracking (e.g [27, 52]) or an expensive motion feature extraction step (e.g [27, 187]).

Detecting and analyzing periodicity in image sequences has been explored in the human activity recognition community to classify cyclic human actions (e.g. walking or waving). Some earlier work [5, 177] relies on detection and tracking of specific points on the human body to detect periodicity. This method is not practical in our setting, since tracking feature points reliably on a small moving object from distance is not feasible. Methods based on temporal changes in individual pixel intensities use frequency domain analysis (e.g [99]), periodicity metrics such as periodograms (e.g [147]) or self-similarity (e.g [36]) to detect periodicity in regions of interest. Since these approaches require precise frame alignment, they become computationally expensive and inaccurate in the presence of tracking errors. Another approach to detect periodicity is to consider the ROI as a whole and perform frequency domain analysis of either its trajectory [17, 140] or mean pixel motion [176]. We found these methods less sensitive to tracking errors and thus a better fit for detecting periodicity using a moving camera. Similar to [176] we perform frequency domain analysis on the average motion per pixel of each ROI. Periodic motion detection has previously been successfully applied to outdoor human robot interaction: in Sattar et. al [160], an underwater robot follows a human diver by tracking the periodic motion of the diver's brightly-colored fins. The robot also uses blob tracking to compensate for tracking errors. This work is different, since we perform periodicity detection on-board a UAV, and it does not rely on strong color or other appearance priors.

4.3 Method

To detect arm waving signals from a flying platform, we first estimate the camera motion between consecutive frames—and hence the robot's ego motion—by tracking feature points between frames. It is also used later in the pipeline to estimate each salient object's movement over time with respect to a fixed reference frame. To find salient objects, we first cluster tracked feature points using a fast non-parametric clustering algorithm. Clusters are

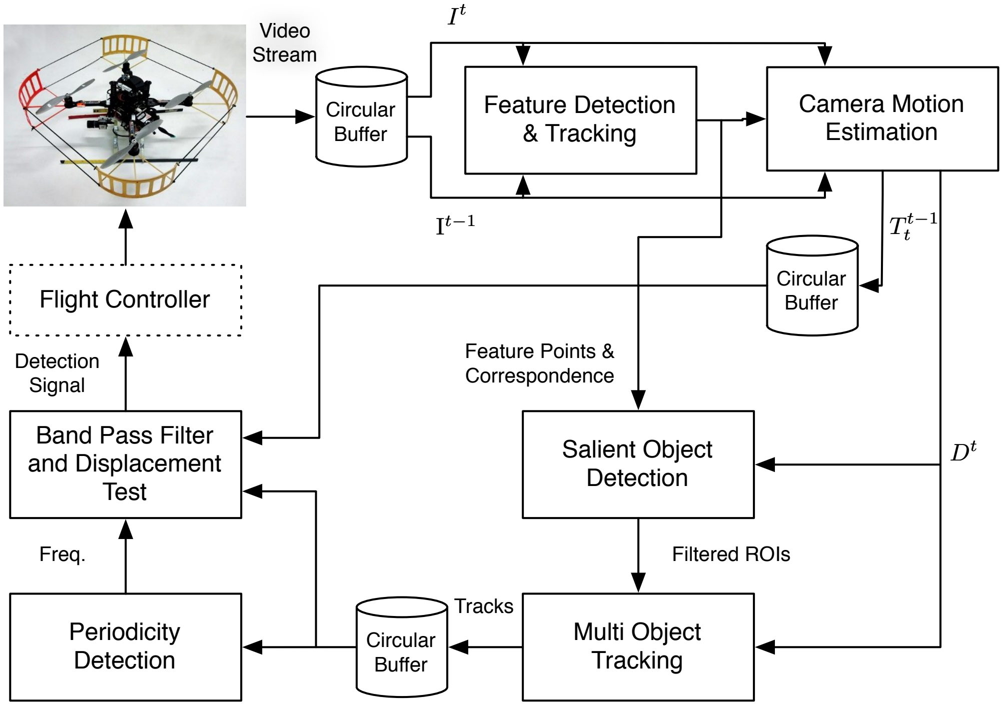


Figure 4.2: Block diagram of the system. Refer to Section 4.3 for the definition of each variable. Except flight controller, other components run on the vision processing unit (Section 4.3.5).

first pruned based on their size and motion characteristics and then tracked using a bank of Kalman filters. For each tracked object's ROI, we calculate average motion per pixel and use discrete Fourier transform and a statistical test to estimate the dominant frequency component of that signal. Tracks that are (i) stationary in the global reference frame and (ii) show periodicity in a specific frequency band are classified as positive detections. Figure 4.2 shows the overview of the detection pipeline. In the following sections we describe each component in more detail.

Throughout this section we introduce parameters for each component, but we defer the discussion on how to set these parameters to Section 4.3.4. Since our approach is a temporal method, we extensively use circular buffers of fixed size N to store a rolling history of different entities. We use the terms "sampling period" and "sequence length" interchangeably to denote N.

4.3.1 Camera Motion Estimation and Ego Motion Cancellation

Our camera motion estimation and stabilization method is based on techniques developed for visual odometry with monocular cameras [23, 129]. To estimate the camera motion at time t, the input frame is first converted to a grayscale image I^t. Then, we detect FAST corners [151] in I^t and store them in a list $F^t = \{f_i^t\}$. To limit computation time in scenes with large number of strong corners, we limit the number of stored feature points to N_F^{max}. We use a pyramidal implementation of the Lucas Kanade optical flow algorithm [18] to find matches between F^{t-1} and F^t. If the number of matches exceeds threshold N_F^{min}, we fit a full homography based motion model on the optical flow field using least median of squares regression. Otherwise, we consider motion estimation as failed for the current frame and do not execute other components of the pipeline. After pruning outliers, the result is further refined using Levenberg-Marquardt non-linear optimization. The resulting transform T_t^{t-1} from frame t to frame $t-1$ is stored in a circular buffer. To cancel out the ego motion of the vehicle, we warp I^t into I^{*t} using the inter-frame transform T_t^{t-1}. We calculate the absolute image difference of I^{*t} and I^{t-1}, adaptively threshold the result and pass it through a low-pass filter for smoothing and suppressing transient errors. We call the resulting image D^t *camera independent inter-frame motion image.*

4.3.2 Salient Object Detection and Tracking

In order to detect salient objects in each frame, we combine two cues: camera independent inter-frame motion and spatial density of feature points in that frame. Motion fields (similar to D^t) have been widely used by researchers to segment and track moving objects from mobile cameras (e.g [36]). However the segmentation quality is heavily dependent on the quality of the camera motion estimation, type of background and size of targets. For our specific application, the size of the target (two moving arms) can be as small as 10×10 pixels, the background is usually complex in outdoor settings, and there is often inevitable camera stabilization error. Camera stabilization error causes false motion blobs in areas with non-homogeneous background as well as around objects with feature points not lying on the ground plane. For those reasons, we found that the motion field alone is insufficient to segment arm waving motion from a distance.

Our approach for detecting salient moving objects is based on detecting dense clusters of feature points in motion salient areas of the image. We first use DBSCAN [55], a fast non-parametric density based clustering algorithm to detect dense clusters of feature points in the frame. As we will show in Section 4.4, it runs in real-time when clustering hundreds of feature points. DBSCAN only relies on two parameters: the maximum inner cluster distance ϵ and the minimum number of feature points per cluster N_c^{DBS}. For each cluster, elements that have zero motion are discarded ($D^t(X_{f_i^t}, Y_{f_i^t}) = 0$). Next a minimum axis-aligned bounding box is fitted to the remaining members. A post-pruning step filters out clusters that are smaller than $S_{min} = W_{min} \times H_{min}$, larger than $S_{max} = W_{max} \times H_{max}$ or have small average motion per pixel value. Given a bounding box B^t, the average motion per pixel ($D_{avg}^{B^t}$) is calculated as follows:

$$D^{B^t}_{avg} = \frac{\sum_{(x,y)\in B} D^t(x,y)}{W_B \times H_B} \tag{4.1}$$

We use a bank of Kalman filters with a constant-acceleration motion model to track the state of each cluster (position, velocity and size) over time. To cancel out the effect of ego motion in state transition, the state is warped using T^{t-1}_t before each Kalman prediction step. In other words, at time t, the previous state of each track is first transformed to the current frame's coordinate system using the inverse of the estimated camera motion, then the Kalman prediction step is applied. To associate observations to tracks we use the Hungarian matching algorithm with extensions proposed in [100]. Tracks without any associated observation are deleted after a timeout period.

To differentiate between stationary periodic actions such as hand waving gestures, and non-stationary periodic ones such as walking, we calculate the camera independent displacement of each track over the sampling period δ^t_{t-N-1}. To determine this value we first need to calculate the camera motion over the whole period:

$$T^{t-N-1}_t = \prod_{i=t-N}^{i=t} T^{i-1}_i \tag{4.2}$$

Then we remove the effect of camera motion from the position of each tracked object ($P^t = [x^t y^t]^T$):

$$P^t_s = T^{t-N-1}_t P^t \tag{4.3}$$

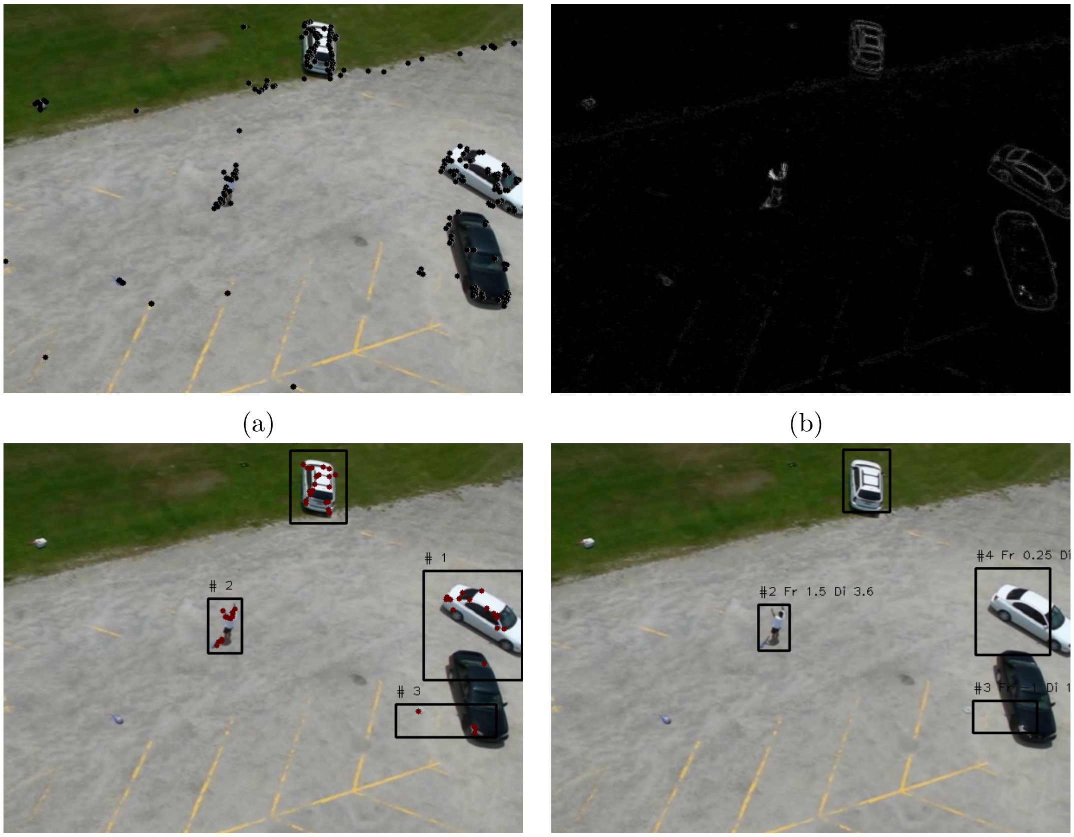
The Euclidean distance between P^t_s and P^{t-N-1} in image space is the camera independent displacement of the tracked object over the sampling period.

4.3.3 Periodicity Detection

To detect periodicity we perform frequency domain analysis on each track's average motion per pixel (Equation 4.1) over the sampling period. We chose this measure since it is fast to calculate and unlike pixel intensity based measures, does not require perfectly aligned tracks. The latter is important, because we found precise tracking to be difficult to achieve in real-time under fast camera motion in flight and when the tracked object is non-rigid.

For each track, the average motion per pixel signal $D^t_{avg}(t)$ is first de-trended and windowed with the Hann function. Using a discrete Fourier transform, we calculate the power spectrum of the signal and find its maximum normalized power component. If A_k where $k \in K = \{1..\frac{N}{2} - 1\}$ denotes the positive half of the energy spectrum, the maximum normalized component is calculated as follows:

$$A_M = \frac{arg\,max_{k\in K} A_k}{\sum_{k\in K} A_k} \tag{4.4}$$



(a)

(b)

(c)

(d)

Figure 4.3: The output of each component of the pipeline running on a sample from ARG Aerial dataset (a) Tracked feature points (b) Camera independent inter-frame motion image (c) Salient objects (d) Tracks

To test if A_M is statistically significant and thus is the dominant frequency of the signal, we apply the approximation to Fisher's exact test proposed by [4]. If A_M passes this test with confidence greater than 99.5%, we consider the track as periodic with frequency $f = \frac{k \times fps}{N}$.

If a track's dominant frequency is between f_{min} and f_{max} with small camera independent displacement $\delta_t^{t-N-1} < \delta_{max}$, we classify that track as stationary, periodic gesture.

Figure 4.3 shows the effect of each component in the pipeline to detect stationary periodic objects on a sample sequence from the ARG dataset (Section 4.4.1).

4.3.4 Tuning the Parameters

Our system is sensitive to two of the parameters described so far: the maximum inner cluster distance of DBSCAN (ϵ) which controls the size of objects of interest in the scene and the video frame-rate (FPS) that limits the accuracy of the periodicity detection component. Setting FPS is trivial because it is known in advance. We manually tuned ϵ for specific experiments. However, it is possible to tune this parameter automatically given the height above ground at which the UAV is flying, camera intrinsics and a prior on the size of objects of interest (people in our case). We set N_F^{max} and N_F^{min} to 500 and 10 feature points respectively. For smaller input sizes, we reduce this number. The sequence length N is set to four times the FPS value (100 to 120) to capture a few periods of the gesture. The parameters of band pass filter for periodicity detection is set to $f_{min} = 0.9$ Hz and $f_{max} = 3.0$ Hz to include the frequency range of human waving gestures. To reject small or large bounding boxes we set $S_{min} = 5 \times 10$ and $S_{max} = 100 \times 200$ pixels. Similar to ϵ, these two parameters can be inferred automatically. Finally we set the threshold to segment stationary and non-stationary tracks (δ_{max}) to 30 pixels.

4.3.5 Platform and Implementation

We run the entire system on board an Asctec Pelican quadrotor to create a fully autonomous system capable of establishing mutual attention with an uninstrumented human in outdoor settings. The pipeline runs on a small form factor PC with a dual core 4th generation Intel Core i7 CPU and 8GB of RAM. To capture images, we use a Point Grey Firefly MV color camera mounted on an actively stabilized gimbal. The Firefly MV is a global shutter camera which captures 640 × 480 color images up to 60 frames per second. The on-board computer controls the UAV by sending position and velocity commands to the flight controller. The total weight of the entire vision processing system (camera, small form factor PC and battery) is 400 grams. The pipeline is implemented in C++ and relies on an optimized build of OpenCV [21][2].

4.4 Experiments

In this section we first report the performance of the proposed method on two human action datasets, then we describe our experimental setup to demonstrate the effectiveness of this method in a human flying robot interaction scenario. As discussed in Section 4.3.4, two parameters have to be tuned for a specific data source: Framerate (FPS) and the maximum inner cluster distance of DBSCAN algorithm (ϵ). We tuned the latter for each dataset to achieve good performance.

4.4.1 Datasets

We tested the system on two human action datasets to evaluate the precision and performance of the proposed method in detecting arm waving gestures and rejecting periodic distractions such as walking and running people.

[2] All source code and configurations used to generate the results in this chapter are available for download at `http://autonomylab.org/obzerver/`. The commit hash of the code used to generate the results begins `1fc6bd8`.

Setup	Picture Size	FPS (Input)	ϵ	Avg. Exec. Time per frame (ms)
KTH	160×120	25	0.2	3.136
ARG	960×540	30	0.03	27.84
UAV	640×480	30	0.2	31.65

Table 4.1: Properties of video streams, parameters used for each experiment and average execution time per frame (Section 4.4.3)

The first dataset is the *KTH human action dataset* [93], which contains six actions performed by 25 actors. The camera is static and the background is homogeneous. Each action is performed four times by each actor in four different scenarios: static homogeneous background (SHB), SHB with scale variation, SHB with different clothes and SHB with lighting variations. The second dataset is the *UCF-ARG (University of Central Florida-Aerial camera, Rooftop camera and Ground camera)* [119]. We use the aerial component of the dataset which was recorded from a remote controlled helium balloon. It consists of 10 actions performed four times (in different directions) by 12 actors in an open parking lot. This dataset contains a set of challenges: fast and sudden camera motion in almost all video frames, shadows, variation in scale and clothing and small size of people in this dataset which occupy less than 5% of the whole 960×540 image. Table 4.1 lists the properties of the video stream in each dataset as well as the parameters we used to evaluate the system.

First we report the performance of our approach on detecting human hand waving gestures on these datasets. Table 4.2 summarizes the true detection rate, false positive rate and miss rate of the vision pipeline when applied to hand waving gesture subset of these datasets. A detection is considered correct if the detected bounding box is stationary, overlaps with the upper body of the actor and includes at least one hand. If the system detects a bounding box which is stationary but does not overlap with the body it is considered as a false positive. Non-stationary detections as well as no detections are considered as misses. The detection rate on KTH and ARG datasets are 78% and 56.25% respectively. Although, the false positive rate is low for both datasets (0% and 4.15%), the miss-rate is the major deficiency. We observe that salient object detection and tracking errors due to scale changes (KTH), small objects and fast camera motion (ARG) are the main causes of the high miss rate. Since the input video length is relatively short with respect to sequence length[3], the system does not have enough time to recover from bad/false tracks to detect periodic motions.

To evaluate the effect of non-stationary periodic distractions such as walking and running actions, we report the false positive detection rate of our approach when applied to the walking and running subset of UCF-ARG and KTH datasets. Table 4.3 shows the results. The pipeline shows zero false detections on either “running” sequence. However it exhibits

[3] Average duration of waving gesture sequences in KTH and UCF-ARG datasets are 21.5 and 10.2 seconds respectively.

Dataset	Number of Actions	DR	FDR	MR
KTH	100	78%	0%	22%
ARG	48	56.25%	4.16%	39.58%

Table 4.2: The accuracy of hand waving detection for each experiment (DT: Detection Rate, FDR: False Detection Rate, MR: Miss Rate)

Dataset	Action	Number of Actions	FDR
KTH	Walk	100	13%
KTH	Run	100	0%
ARG	Walk	48	16.67%
ARG	Run	48	0%

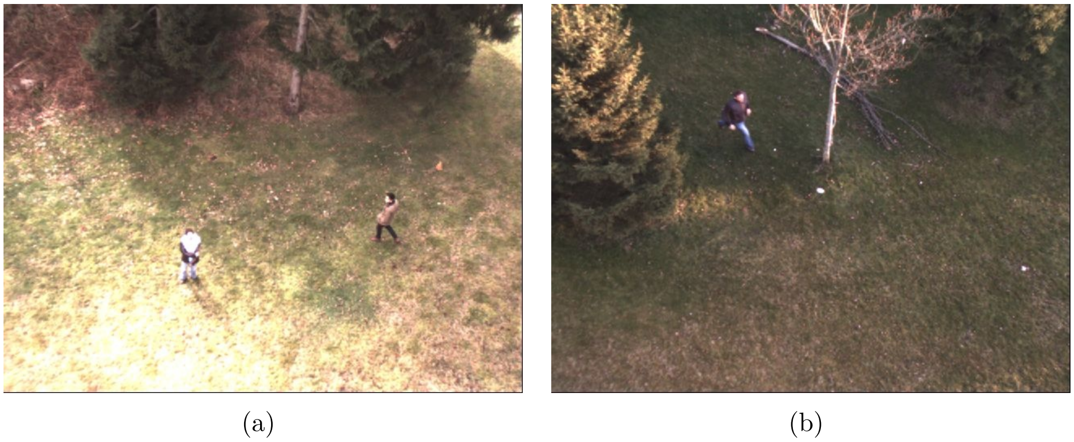
Table 4.3: False Detection Rate (FDR) for walking and running actions (UCF-ARG and KTH datasets)

a 13% and 16.67% false detection rate on the "walking" sequences of the KTH and ARG datasets respectively. This is mainly due to tracking and motion estimation error which causes a non-stationary periodic object to appear stationary. The false detection rate can be reduced by decreasing δ_{max} at the expense of a lower detection rate or slower response time. An alternative is to use robot behavior to reject false positives as discussed in Section 4.5.

4.4.2 Closed-loop experiments with an outdoor UAV

To demonstrate the effectiveness of our system in initiating interaction and establishing mutual attention between a flying robot and a human, we performed a series of 22 trials in outdoor settings[4]. The trials were carried out on three different days, under two lighting conditions (sunny and overcast), at two different locations and with different subjects. Both locations were open grass fields with trees and bushes at one side. In each trial the robot traversed a predefined path (a set of GPS waypoints) of length 10 meters back and forth at a fixed altitude and heading. We designed the UAV's flight path such that the vegetation be visible at all times. The altitude was varied from 10 meters to 15 meters during the trials. In each trial, a single human tries to initiate the interaction with the UAV and attract its attention by waving at it. Two types of distractions were present in the field of view of the UAV: walking, running or standing people and natural distractions such as trees and bushes often moving in the wind. Since the robot does not perform any active searching to find humans, the workspace in which the subject and human distractors are allowed to act is marked in advance. The robot is fully autonomous, untethered and self-contained except during take-off and landing, where it is controlled by a human safety pilot. The script for each trial is as follows:

[4] Video demonstration: `https://www.youtube.com/watch?v=KXmgBDI_6PE`



(a) (b)

Figure 4.4: Example images from the robot's perspective during experiments. Location 1 (left) Location 2 (right)

- Human distractors perform their act during the entire length of a trial and are instructed to stay with the UAV's workspace.
- The human subject chooses an arbitrary position in the workspace prior to the start of the trial.
- The UAV takes off and flies back and forth between two predefined points.
- The subject is instructed to stand still while the robot traverses the first leg (from A to B). This is to test that the system correctly handles the absence of gestures.
- Once the robot is on the return leg (after reaching point B) the subject starts waving.
- If the UAV detects this gesture it stops translating, hovers, and starts the "wobble" behavior. This indicates to the waving human that she is detected. This successfully concludes a trial.
- If the UAV reaches a waypoint without detecting a waving gesture it starts a new traverse back to the previous location. The subject is allowed to try again to get the robot's attention. We report the number of retries in the results section. Runs with more than 1 retries are considered failures.

For a few trials we asked the subject not to try to attract the UAV's attention so we can examine the system's resistance to false positives. Figure 4.4 shows the robot's field of view during trials on two different locations. Table 4.4 summarizes the conditions and results of all trials. The overall success rate of all trials was 81.8%. During all 22 trials, the UAV was never attracted to a false positive. In 6 successful runs with a waving human subject, it took the UAV one more traversal to find and acknowledge the subject.

Trial	Condition	Subjects and Distractors	Alt.	# of Tries	Outcome
1	C1	1,0	10	1	Success
2	C1	1,1(w)	10	1	Success
3	C1	0,1(w)	10	N/A	Success
4	C2	0,2(r)	10	N/A	Success
5	C2	1,1(w)	12	2	Success
6	C2	1,1(w)	12	4	Failure
7	C2	1,2(w)	12	3	Failure
8	C2	0,1(w)	12	N/A	Success
9	C2	1,0	12	1	Success
10	C3	1,1(w)	12	2	Success
11	C3	1,1(w)	12	4	Failure
12	C3	1,1(w)	12	1	Success
13	C3	0,3(w)	12	N/A	Success
14	C3	1,1(r)	15	1	Success
15	C3	1,1(r)	15	2	Success
16	C3	1,1(r)	15	1	Success
17	C3	1,1(r)	15	2	Success
18	C4	1,2(w)	15	4	Failure
19	C4	1,1(r)	15	1	Success
20	C4	1,1(r)	15	2	Success
21	C4	1,1(r)	15	1	Success
22	C4	1,1(w)	15	2	Success
Overall Success Rate				18 / 22 (81.8%)	

Table 4.4: Outcome of all trials. C1: Location 1, Late Afternoon, Overcast, C2: Location 1, Noon, Overcast, C3: Location 2, Noon, Overcast, C4: Location 2, Late Afternoon, Sunny, (r): running, (w): walking

Analyzing the experimental data we observe two major causes of failures. In two trials the human was on the edge or out of robot's field of view for the majority of time. Therefore the tracking of the human was not reliable enough to detect periodicity. This was mainly due to errors in the UAV's waypoint navigation and position control, which changed the robot's visible workspace. Since the robot is flying several meters away from the human and the camera is barely visible, the subject was not able to estimate the robot's field of view to correct her location. This emphasizes the importance of the robot providing behavioral feedback when the human is detected. In two other failed trials, the vision system was not able to detect a moving object. Either too few features were detected on the subject's body or they were too sparse to form a cluster.

	KTH	ARG	UAV
Pre-processing	0.179	2.14	2.51
Feature Detection & Tracking	0.529	13.10	13.69
Find Homography	2.30	10.55	10.10
Salient Object Detection	0.08	1.89	5.25
Object Tracking & Periodicity Detection	0.052	0.17	0.10
Total	**3.13**	**27.84**	**31.65**
Stddev	1.51	3.22	2.86

Table 4.5: Mean per-frame execution time breakdown for each component of the pipeline (in milliseconds).

4.4.3 Runtime Performance

For all three experiments, we measured the execution time per frame incurred by each step of the vision pipeline. The last column of Table 4.1 shows the average processing time per frame for each experiment. Table 4.5 shows the detailed breakdown of execution time for each component of the pipeline during each experiment. The processing time is less than the inter-frame time, so the system works at frame-rate.

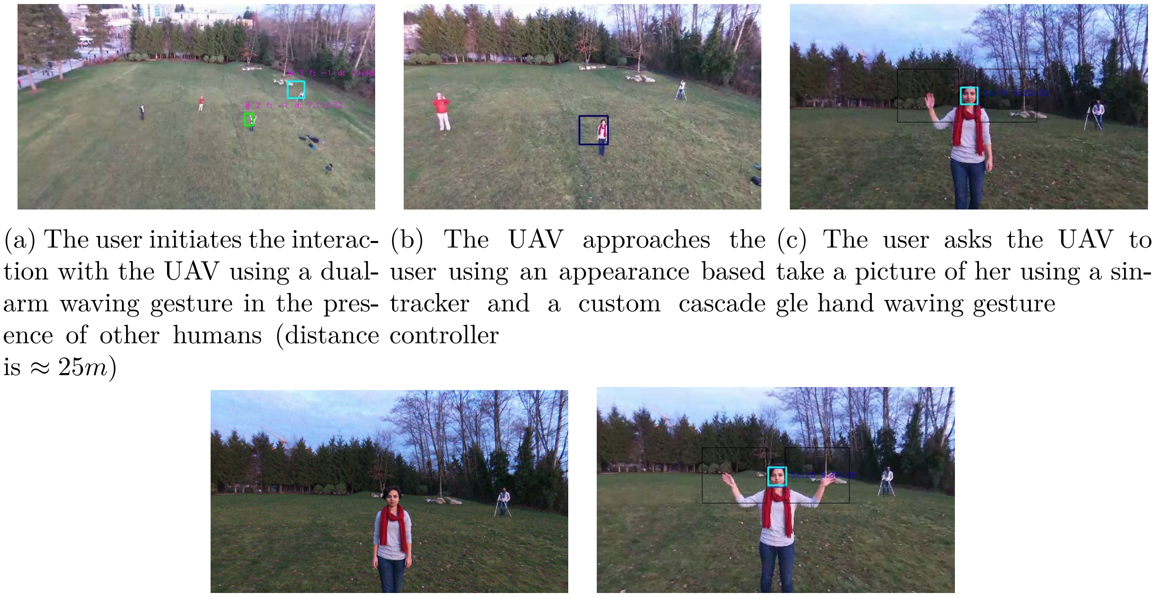
4.5 Conclusion

In this chapter we presented the first demonstration of human-UAV interaction in outdoor environments using real-time computer vision running entirely on-board. We show how a dual arm-waving gesture can be used to attract a flying robot's attention while being robust to similar distractions such as walking and running people. By acknowledging the user through a wing wiggle, the robot communicates its readiness for further interaction with the user.

The main limitation of this approach is that the UAV can become attracted to non-interesting stationary periodic motions caused either by other human actions (e.g. digging) or irrelevant extrinsic processes (e.g waving trees). In the following chapter, we will explore a solution to this problem based on using robot's behavior to approach the user and to perform close-range inspection/interaction.

Chapter 5

An End-To-End, Direct Human-Flying Robot Interaction System



(a) The user initiates the interaction with the UAV using a dual-arm waving gesture in the presence of other humans (distance is $\approx 25m$)

(b) The UAV approaches the user using an appearance based tracker and a custom cascade controller

(c) The user asks the UAV to take a picture of her using a single hand waving gesture

(d) The resulting portrait

(e) The user terminates the interaction by performing a dual hand waving gesture.

Figure 5.1: Our end-to-end human-flying robot interaction system in action during one of outdoor experiments (Section 5.4.2).

In this chapter we present the first demonstration of end-to-end far-to-near situated interaction between an uninstrumented human user and an initially distant outdoor autonomous UAV. The user uses an arm-waving gesture as a signal to attract the UAV's attention from a distance. Once this signal is detected, the UAV approaches the user using appearance-based tracking until it is close enough to detect the human's face. Once in this close-range interaction setting, the user is able to use hand gestures to communicate its commands to the UAV. Throughout the interaction, the UAV uses colored-light-based feedback to communicate its intent to the user. We developed this system to work reliably with a low-cost consumer UAV, with only computation off-board. We describe each component of this interaction system, giving details of the depth estimation strategy and the cascade predictive flight controller for approaching the user. We also present experimental results on the performance of the complete system and its individual components.

5.1 Introduction

In this chapter we show the first realized end-to-end Human-Robot Interaction system whereby an uninstrumented user can attract the attention of a distant (20 to 30 meters) autonomous outdoor flying robot, the robot then approaches the user to close range (≈ 2 meters), hovers facing the user, then responds appropriately to a small vocabulary of hand gestures.

The main contributions of the chapter are (i) the first demonstration of end-to-end interaction with a distant flying robot over multiple scales; (ii) a description of a robust integrated visual servo and predictive cascade controller design for smooth approach towards a human; and (iii) a case study in outdoor situated HRI with UAVs over multiple scales. Below we describe the components of our end-to-end situated interaction system. We describe how we use fast computer vision methods to detect the user's intention from distance using a monocular camera, how we estimate depth when approaching the user, a predictive cascade controller to follow a smooth trajectory towards the user despite the high latency of our off-board vision via WiFi link, our close-range interaction system for the communication of commands from the user to the UAV and our colored-light-based feedback system for communicating the UAV's state to the user. We present experimental results of this system in action, where an uninstrumented user can summon a Parrot Bebop Drone from distances over 20m and have the robot take a close range portrait photo - a *selfie* - of her. The scale change is such that the person initially appears around 15 pixels high in the UAV's 640 × 368 camera image, but the portrait taken features the person's torso and head in the center of the image (Figures 5.1 and 5.6).

5.2 Background

We previously presented systems that enable uninstrumented humans to perform close range situated interaction with UAVs through gaze and hand gestures in Chapter 3 and obtain a distant UAV's attention using stationary periodic gestures while the UAV is in flight in Chapter 4. In this chapter, we build upon those systems to provide an end-to-end interaction system for human-flying robot interaction.

In Chapter 1 we identified the components of an end-to-end interaction system as (i) interaction initiation; (ii) approach and re-positioning to facilitate close-range interaction; and (iii) communication of commands and intents from the human to the UAV and communication of intents from the UAV to the human (Definition 1).

As we discussed in the previous chapter (Chapter 4), uninstrumented interaction initiation between co-located humans and UAVs mostly happens in two forms. In the first form, the UAV utilizes vision-based human feature detectors to find potential interaction partners. Alternatively the user may try to attract the UAV's attention by using active stimuli such as gestures, sound or body movements.

Using vision-based human detectors on-board a UAV poses multiple challenges. First, when the UAV is flying far from the humans, features are either hard to detect or require high computational resources to be detected in real-time. Some researchers use extra sensors such as thermal cameras [14], scene information such as saliency maps [15] or the prior on the height of the human combined with ground plane estimation [169] to identify regions of interest in the image plane before executing vision-based human detection.

Most existing human detectors assume an upright human view [14]. The violation of this assumption caused by time-varying and different vantage point of UAVs causes the second issue for performing on-board pedestrian detection. In [15] the authors show that the performance of a conventional pedestrian detector can be improved by retraining it using a dataset that is recorded from a UAV and with synthetic variations of camera roll and pitch angles. In [7], the authors propose to compensate for this time-varying vantage point by estimating the ground plane using UAV's telemetry data and cancel out the distortion by projecting the image to the ground plane prior to using a pedestrian detector. Although none of the aforementioned methods were explicitly used for human-UAV interaction, they are applicable for implicit interaction initiation or as a building block for explicit interaction initiation. The same is true for methods such as [80] that utilize moving object detection to find regions of interest and potential interaction partners in the UAV's field of view. We provided a detailed survey on related work for implicit and explicit methods suitable for interaction initiation with flying robots in Section 2.2.1.

Once the interaction between a human and a UAV (or a team of UAVs) is initiated, the human and the UAV(s) can interact more directly by communicating their intents (Section 2.2.2). Uninstrumented, natural and situated communication of commands from humans to UAVs have been recently explored by various researchers in form of human studies and practical systems. Example human studies include [77] and [25] that investigate natural commanding modalities for collocated interaction between a human and a flock or a single UAV respectively.

To approach towards the user, the UAV should first track the location of the user, then constantly control its flight trajectory to reach the person. Recently, researchers have applied state of the art long-term appearance-based visual trackers and Image Based Visual Servo (IBVS) control for following an uninstrumented human with a UAV [69,135]. We use the same long-term visual tracker developed by [69] in our system. Similar to [135], we use a visual servo controller to generate approach trajectories for our target platform. However, since our system performs approaching towards the user, rather than following her, depth estimation of the target becomes more critical, thus we provide a solution to estimate depth of the tracked object using UAV's telemetry data and the intrinsic parameters of the camera. Furthermore, unlike [135], our system is not initialized by a human operator, instead it uses explicit interaction signals from the human to initialize the appearance-based tracker. Most relevant to our work is [39] in which the authors designed a self-contained person follower UAV that implements implicit interaction initiation through pedestrian detection, appearance based tracking, depth estimation and trajectory controller. As mentioned, we use explicit interaction initiation signals that helps the UAV steer its attention to a single person when multiple users are in its FOV. In addition, we explicitly address the two-way communication of intents and commands between a human and a UAV.

Practical systems for situated interaction with UAVs in the literature mainly utilize sound and gestural interfaces for communication of commands from humans to UAVs (Section 2.2.2.2). In the prototype environment of [95], the authors use a Microsoft Kinect sensor on-board a hovering UAV to transmit gestural commands to a team of flying robots in an indoor environment. In [125], the authors propose a solution for canceling the ego-motion of an RGB-D camera attached to a flying UAV and use the stabilized depth image to perform gesture recognition and person following in an indoor environment. In [31], the authors applied transfer learning to develop a person-specific gestural interface to command a UAV.

As argued in [78] being able to "talk" is as important requirement as being able to "listen" for an autonomous agent. Through proper feedback, the user can understand if the UAV correctly understands her intents and if the UAV is functioning properly. These in turn decrease user's cognitive workload and improve her awareness and safety. Recently, a few different modalities for communication of intent and affects from a UAV to its collocated human partners have been studied. These modalities include flight path manipulation [163, 173] and light-based feedback systems [174] (Section 2.2.3).

In the remaining of this chapter, we demonstrate the first end-to-end Human-Flying Robot interaction system that implements all the components of Definition 1 in outdoor settings. This system autonomously brings a flying robot from relatively long distances to a proximate distance to the user.

5.3 Method

Our proposed system consists of three hardware components and five major software blocks. We use the Parrot Bebop, a lightweight consumer UAV as our platform. The UAV transmits the live video stream of its front facing camera and flight telemetry data to an off-board computer over WiFi. This computer runs the core software components of the interaction system and sends the desired control commands over the same WiFi link to the UAV. A small form factor computer is mounted on top of the UAV to drive an array of 11 high intensity RGB LEDs mounted on the front side of the UAV and generate feedback signals (Figure 5.3). The off-board computer communicates the desired feedback to the embedded computer over a separate WiFi link based on the current state of the interaction. The five major components of the software stack are (i) the behavior generator and coordinator; (ii) the long-range periodic motion detector for initiating the interaction; (iii) the appearance-based object tracker; (iv) the cascade controller used for approach towards the user; and (v) the face engagement detector and motion based gesture recognizer for the close-range interaction phase. Figure 5.2 shows the overall architecture of our interaction system.

The system starts in the *searching* state, where it looks for periodic but net-stationary motions in camera's FOV as in Chapter 4. When a periodic signal is detected, the corre-

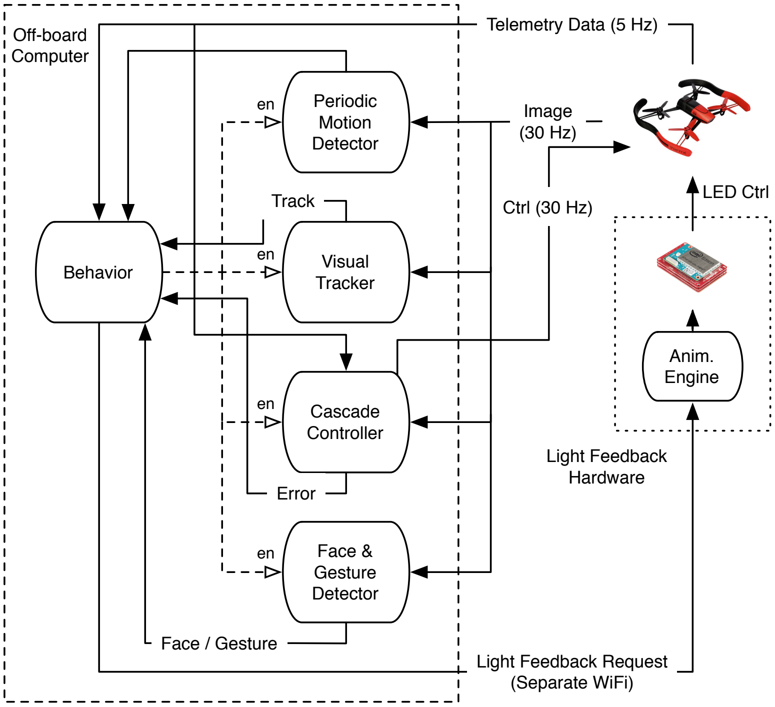


Figure 5.2: The block diagram of the system

sponding region of the image is fed into a long-term visual tracker which simultaneously tracks the object in the image plane and refines its appearance model. The track is piped into a cascade controller, which first estimates the distance of the target with respect to the image plane, then controls the flight of the UAV towards the target. The approach towards the target ends either when the target is in the center of the image plane and the UAV is within a predefined distance with respect to the target, or a human face detector finds a human face inside the target's bounding box in the image plane. In the latter case, the system transitions into *close-range interaction* state, where the UAV maintains the user's face in the center of its FOV and at a fixed distance from its camera (using the same cascade controller). In this state, a motion based gesture detector detects the left hand and right hand waving gesture of the human which is consequently used to command the vehicle to perform a certain action. The UAV constantly communicates its state and intentions to the user using its front facing colored-light-based feedback system. In the remainder of this section we describe each component of this system in more details.

5.3.1 Hardware Platform

One of the difficulties faced during the development of this system was to choose a UAV platform suitable for close-range situated interaction with a human. The main criteria for this platform were safety around the interaction partner, being able to perform stable hov-

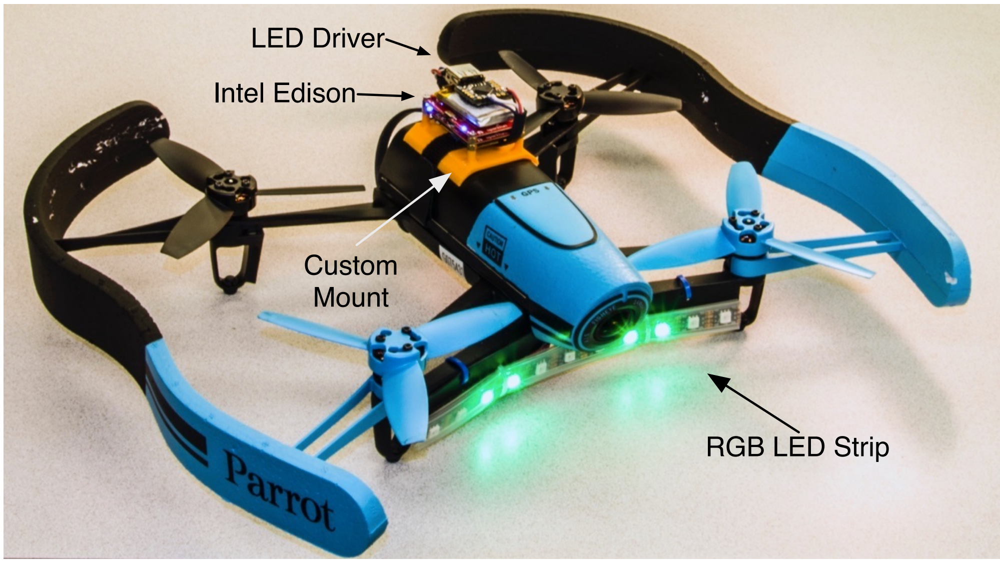


Figure 5.3: Our platform. Parrot Bebop Drone and color-light-based feedback system. The UAV is executing the *Gaze* feedback (Section 5.3.6).

ering and carrying enough payload for sensing, feedback and computation. Most consumer UAVs available in the market nowadays are multi-rotor flying platforms that are able to perform stable hovering. Many of these platforms are also powerful enough to carry small form factor sensing devices and computational units. However, not many of these UAVs provide the minimum safety measures to fly in close proximity of people. We believe any UAV platform that enters the *social space* ($\approx 3-4$ *meters)* [70] of a human or closer should be at least equipped with physical propeller guards and provide an automatic shutdown systems in case of contact between any of its propellers and an object.

We chose the *Parrot Bebop Drone*[1] as our UAV platform. Although this UAV provides the required minimum safety measures, its on-board [flight controller] computer is not powerful enough to execute our CPU intensive software stack. Due to its limited payload carrying capabilities, it is also not capable of carrying powerful small form factor computers. For these reasons we opted to control the UAV off-board over WiFi. Bebop is a small form factor consumer UAV with an on-board high definition camera and a fisheye lens with the field of view of 180 degrees. The video stream of this camera is digitally stabilized and rectified on-board prior to being transmitted over WiFi with the reduced resolution of $640 \times 368 px$ at 30 frames per second. The rectification target is limited to the FOV of $\approx 80°$ (horizontal) and $\approx 50°$ (vertical), essentially simulating a virtual pan/tilt camera with a stabilized gimball. The desired pan and tilt of this camera is also controllable over WiFi. Bebop transmits its telemetry data (i.e. altitude and attitude) over WiFi to the off-board computer at the rate of 5 Hz.

[1] `http://www.parrot.com/products/bebop-drone/`

5.3.2 Interaction Initiation using Periodic Gestures

To initiate the interaction with a distant human and while the UAV is in flight, we use the system previously presented in Chapter 4 to detect periodic salient motions on-board a UAV. The main software component of this system (Freely available at `http://autonomylab.org/obzerver/`) is a real-time computer vision pipeline that detects salient moving objects that exhibit periodic motion patterns in a moving camera's FOV. The dual-arm waving of a human is a periodic signal (with a dominant frequency of 1 to 4 Hz) which is detected by this component to initiate (trigger) the interaction. In Section 4.3 we provided a detailed description of this computer vision pipeline. We provide a brief summary of this pipeline here.

To detect periodic motions that are stationary with respect to the camera, this pipeline first detects and tracks salient feature points between consecutive image frames of the UAV's video stream. The correspondences between these tracked feature points are used to estimate the ego motion of the UAV's camera. After compensating the estimated ego-motion, the pipeline calculates the so-called "camera independent inter-frame motion image" which indicates the regions of interest in the image plane that contain motions from moving objects in the scene. Simultaneously, the pipeline clusters the salient feature points in motion-rich areas of the image plane into candidate objects and tracks these objects over time. For each candidate, the frequency spectrum of its average motion per pixel as well as its displacement in a world's coordinate system are calculated over a short time period of 4 to 6 seconds. If a track is stationary with respect to the world and exhibits a periodic motion with a dominant frequency of 1 to 4 Hertz, it is considered as the interaction initiation signal.

5.3.3 Visual Tracker

To track the location of the target detected by interaction initiation module in the image plane, we use the long-term visual tracker of Haag et al. [69]. This appearance-based tracker (named as *KCFTld* by the authors) combines the Kernelized Correlation Filter (KCF) tracker of Henriques et al. [73] for short-term tracking with Tracking-Learning-Detection (TLD) framework of Kalal et al. [82] for long-term tracking, target re-detection and loss detection.

The popularity of Correlation Filter-based Trackers (CFT) in the visual tracking community has been rising in recent years. CFT trackers have shown compelling results in various benchmarks (e.g. [91]) while running at faster than real-time speeds[2]. In general, when a correlation filter is applied (via convolution) to an image, it shows strong responses in the areas that the object of interest is located. CFTs use this property to localize the object of interest based on its appearance model. Bolme et al. [16] proposed an efficient

[2] For a recent experimental survey on CFTs please see [28]

method to model the appearance of the object of interest in the frequency domain from a few training examples called the Minimum Output Sum of Squared Error (MOSSE). Since the convolution can efficiently be done in the frequency domain, this tracker could operate at hundreds of frames per second. The KCF tracker of Henriques et al. [73] is a kernelized version of the MOSSE tracker that exploits the cyclic structure of dense sliding sampling windows used for refining the appearance model. The mathematical framework proposed by Henruques et al. [73] provides a way to efficiently use all sampling windows for training, apply the kernel trick to train non-linear models and use richer feature descriptors (such as HOG [38]) instead of raw pixels to improve both speed and accuracy of adaptive correlation filters.

KCF Tracker is a short term tracker, meaning that it can not deal with object re-detection when the object of interest is lost. Haag et al. [69] integrate KCF into Tracking-Detection-Learning (TLD) framework of Kalal et al [82]. TLD is a long-term tracker that performs tracking and detection simultaneously. The tracker tracks the object of interest between frames while the detector evaluates the whole frame to find instances of the target. While tracking, a semi-supervised learning pipeline monitors the detector and generates new positive and negative samples to decrease its false-positive and false-negative rate. The detector uses these samples to refine its appearance model of the target. When the target is lost, the detector is able to reinitialize the tracker.

In KCFTld, Haag et al. [69] use the learning and detection part of the TLD framework and use KCF as the tracking component. They also employ the Peak to Sidelobe Ratio (PSR) measure originally introduced by Bolme et al. [16] to detect target loss. Furthermore, KCFTld modifies the target re-detection policy of TLD framework such that the tracker should further confirm a re-detection ROI reported by the detection component.

5.3.4 Cascade Controller for Approaching the User

The task of the approach controller is to bring the UAV to a predefined distance of the user while keeping her in the center of its FOV. We designed a cascade controller in order to achieve this task. The input to the cascade controller is the current location of the tracked object in the image plane and the outputs are the desired set-point velocities for the on-board flight controller of the UAV. As a quad-rotor UAV, the Bebop has four controllable Degrees Of Freedom (DOF): *roll*, *pitch*, *yaw* and *altitude*. The on-board flight controller of Bebop offers velocity control for the latter two DOF. However, *roll* and *pitch* - which control the acceleration of the UAV - are set directly. The Bebop performs on-board visual-inertial state estimation and reports the estimated values for its attitude and velocity at 5 Hz. The high level controller in this cascade is an Image Based Visual Servo (IBVS) controller that receives the current position of the tracked ROI in the camera plane, estimates its depth based on the current state of the UAV, then calculates a set of reference velocities that would bring the camera to the desired location in front of the target. The angular and vertical

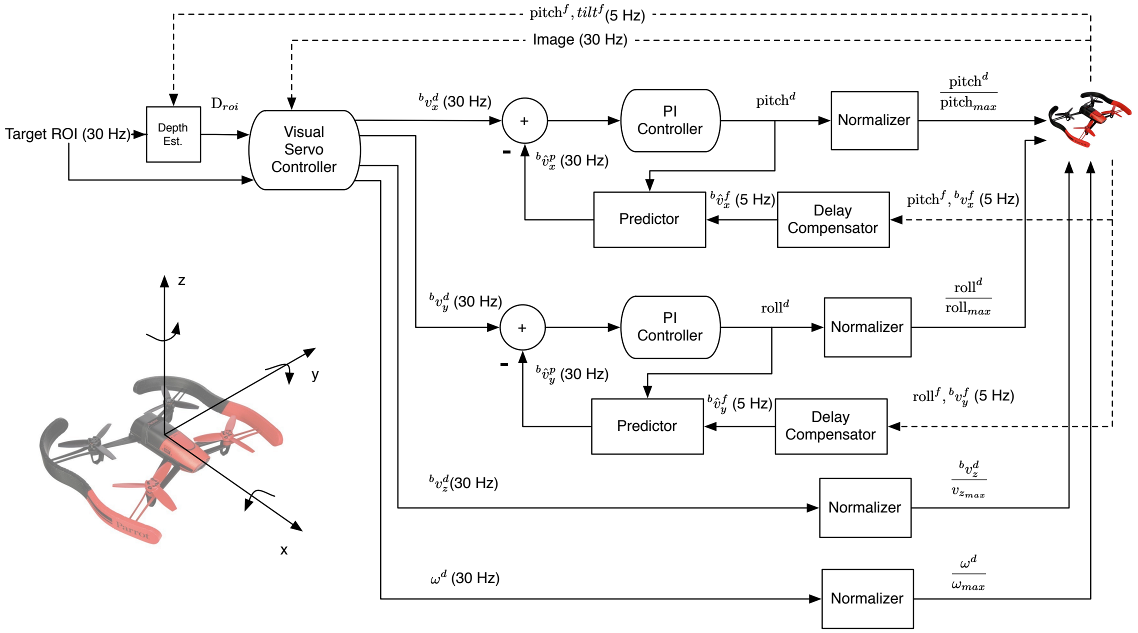

$\text{pitch}^f, tilt^f$(5 Hz)
Image (30 Hz)
Target ROI (30 Hz)
Depth Est.
D_{roi}
Visual Servo Controller
${}^b v_x^d$ (30 Hz)
+
-
${}^b \hat{v}_x^p$ (30 Hz)
PI Controller
pitch^d
Normalizer
$\frac{\text{pitch}^d}{\text{pitch}_{max}}$
Predictor
${}^b \hat{v}_x^f$ (5 Hz)
Delay Compensator
$\text{pitch}^f, {}^b v_x^f$ (5 Hz)
z
y
x
${}^b v_y^d$ (30 Hz)
+
-
${}^b \hat{v}_y^p$ (30 Hz)
PI Controller
roll^d
Normalizer
$\frac{\text{roll}^d}{\text{roll}_{max}}$
Predictor
${}^b \hat{v}_y^f$ (5 Hz)
Delay Compensator
$\text{roll}^f, {}^b v_y^f$ (5 Hz)
${}^b v_z^d$(30 Hz)
Normalizer
$\frac{{}^b v_z^d}{v_{z_{max}}}$
ω^d (30 Hz)
Normalizer
$\frac{\omega^d}{\omega_{max}}$

Figure 5.4: The block diagram of the cascade controller.

velocity components of the IBVS controller's output are sent directly to the UAV, while the lateral and forward velocity components are fed into a velocity controller which deals with the latency and slow update rate of the feedback signal. Figure 5.4 shows the architecture of the approach controller. Internally this controller uses the dynamic model of the UAV to compensate the delay and predict the feedback signal as well as a PI controller to track the reference velocity. We provide more details about this controller in the following sections.

Throughout this section, v_x, v_y, v_z and ω denote the forward, lateral, vertical and rotational (along z-axis) velocity of the UAV, respectively. In addition, d and f superscripts indicate *desired* and *feedback* values for a variable. Finally $\hat{v}$ stands for any estimated velocity.

5.3.4.1 Depth Estimation

Similar to the approach proposed by Danelljan et al. in [39], we use camera intrinsic parameters (specifically β, its vertical FOV), the prior on the size of the tracked object (H_1), the prior on the distance of the object from the ground plane (H_2), current tilt angle of the camera with respect to the inertial frame of the UAV (α), and the vertical pixel location of the ROI in the image plane (h_c) to estimate the distance of the center of the object (Z) from the image plane, under the assumptions that the ground plane is flat (horizontal) and the user's ROI is perpendicular to the ground plane (Figure 5.5). Using simple geometry we can derive Z as:

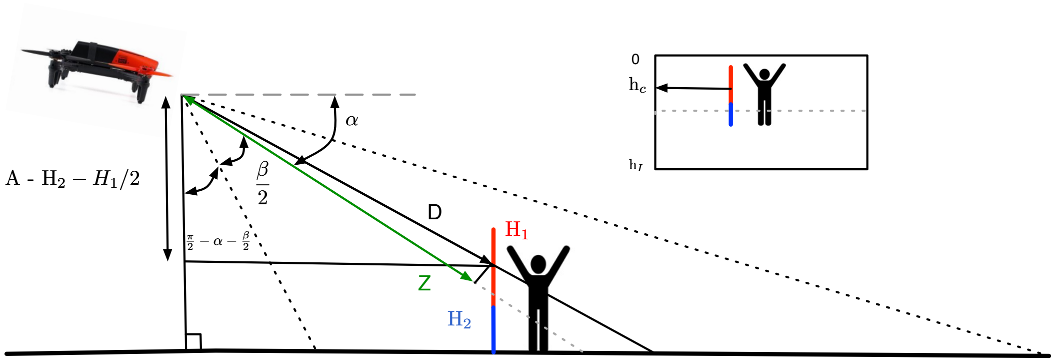


Figure 5.5: The proposed method for estimating the depth of the tracked object. Refer to the text (Section 5.3.4.1) for details.

$$Z = \frac{A - H_2 - \frac{H_1}{2}}{sin(\frac{\pi}{2} - \alpha - \frac{\beta}{2})} \times cos(\frac{h_c - \frac{h_I}{2}}{h_I} * \beta) \tag{5.1}$$

5.3.4.2 Visual Servo Control

Once the depth of the tracked bounding box is estimated, we use a classical IBVS controller [26] to calculate the desired velocity of the camera to approach the target. In the formulation proposed by Chaumette and Hutchinson [26] a visual servo controller minimizes the following error function between a set of visual feature points $s(\mathbf{m}(t), \mathbf{a})$ and a set of desired feature points $\mathbf{s}^*$ by generating appropriate control signals.

$$\mathbf{e}(t) = s(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \tag{5.2}$$

In this formulation $\mathbf{m}(t)$ is a set of image plane coordinates and $\mathbf{a}$ is additional knowledge about the system such as camera intrinsic parameters. For image-based visual controllers, $\mathbf{s}$ is computed directly from image data. The time derivative of $\mathbf{s}$ is coupled to the the linear and translation velocity of a camera ${}^c\mathbf{v} = ({}^cv, {}^c\omega)$ through the so-called *interaction matrix* $\mathbf{L}_s$:

$$\dot{\mathbf{s}} = \mathbf{L_s}{}^c\mathbf{v} \tag{5.3}$$

When $\mathbf{s}^*$ is constant, the time derivative of the error ($\dot{\mathbf{e}}$) equals to time derivative of the visual features ($\dot{\mathbf{s}}$), thus:

$$\dot{\mathbf{e}} = \mathbf{L_e}{}^c\mathbf{v} \tag{5.4}$$

Where $\mathbf{L_e} = \mathbf{L_s}$. Considering an exponential decrease of error ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$), the velocity of the camera can be controlled as follows to minimize the error in Equation 5.2.

$$^{c}\mathbf{v} = -\lambda \mathbf{L}_{\mathbf{e}}^{\dagger}\mathbf{e} \tag{5.5}$$

In this formula, $\mathbf{L}_{\mathbf{e}}^{\dagger}$ is the pseudo-inverse of the interaction matrix and λ is a constant gain.

In the classical Image-Based Visual Servo formulation, $\mathbf{m} = (i, j)$ are 2D image-plane locations of feature points in pixels and $a = (c_i, c_j, f, \alpha)$ is the set of camera intrinsic parameters (location of the principal point, the focal length and the pixel scale ratio). Assuming a pinhole camera model, these parameters convert $\mathbf{m}$ from pixel measurements to feature points $\mathbf{s} = (x, y)$ on the image plane. Considering $\mathbf{X} = (X, Y, Z)$ as the three dimensional location of feature points $\mathbf{s}$ in the camera plane, the relationship between $\mathbf{s} = (x, y)$ and $\mathbf{m} = (i, j)$ is as follows:

$$\begin{aligned} x &= \frac{X}{Z} = \frac{i - c_u}{f\alpha} \\ y &= \frac{Y}{Z} = \frac{j - c_v}{f} \end{aligned} \tag{5.6}$$

In [26], it is shown that the interaction matrix $\mathbf{L}_{\mathbf{x}}$ in this case has the following format:

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \tag{5.7}$$

In this setting, the interaction matrix requires the depth of each feature point to be known. In addition, since each feature point only provides two constraints to compute $\mathbf{L}_{\mathbf{x}}^{\dagger}$, more than one feature point (at least 3) is required to control all degrees of freedom of the camera. In case of multiple feature points, corresponding interaction matrices are stacked to form the final interaction matrix.

The input to our visual servo controller is the currently tracked region of interest by the visual tracking module (Section 5.3.3). This ROI is an axis aligned bounding box with a fixed aspect ratio (ζ). ζ is set when the tracker is initialized and is kept constant throughout the tracking. This ROI is represented as $(i, j, \zeta h, h)$, where i and j are the location of top left corner of the tracked ROI and h is the height of the ROI (in pixels). We calculate two image plane coordinates, $m_1 = (i, j)$ and $m_2 = (i + \zeta h, j + h)$ from this bounding box and augment those points with the estimated depth of the bounding box Z (Section 5.3.4.1) as the input to the visual servo controller. These values are used by the controller to calculate s_1 and s_2 and subsequently the interaction matrix:

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} \mathbf{L}_{\mathbf{1x}} \\ \mathbf{L}_{\mathbf{2x}} \end{bmatrix} \tag{5.8}$$

We define the desired feature points $\mathbf{s}^* = (s_1^*, s_2^*)$ based on the prior on the target height (H_1) and desired depth (distance) of the camera to the target (Z_d) as follows.

$$\begin{aligned} s_1^* &= (-\zeta\frac{H_1}{2}, -\frac{H_1}{2}, Z_d) \\ s_2^* &= (\zeta\frac{H_1}{2}, \frac{H_1}{2}, Z_d) \end{aligned} \tag{5.9}$$

These feature points represent a rectangle parallel to the image plane and provide only three independent constraints to the IBVS controller to calculate the desired velocity of the camera $(\mathbf{v}_c)$. This implies that one degree of freedom of the UAV is not controllable. We chose the lateral movement of the UAV $({}^b v_y)$ as the non-controllable degree of freedom and map ${}^c\mathbf{v}$ to Bebop's DOFs as follows. We transform ${}^c v_x$ and ${}^c v_z$ to the forward $({}^b v_x)$ and vertical $({}^b v_z)$ DOFs through the tilt angle of the camera $([{}^b v_x, {}^b v_z]^T = \Re(\alpha)[{}^c v_z, {}^c v_x])$, set $({}^b v_y)$ to 0 and control the angular velocity (ω) directly from the error between the horizontal center of the bounding box and the horizontal center of the image plane using a proportional controller.

This control schema flies the UAV to a semi-sphere with radius Z^d in front of the target in a configuration that keeps the object at the center of its FOV. If α is 0, the altitude of the UAV at the end of the approach trajectory will be $H_2 + H_1/2$. Although the semi-spherical shape of the final location with respect to the target might not be suitable for applications such as perching or landing on a moving platform, for human-robot interaction applications it is not a major concern since the user can re-position itself (changes her yaw (or gaze) direction) towards the UAV when the robot is flying towards her.

5.3.4.3 Velocity Controller

As mentioned in Section 5.3.4, ${}^b v_z$ and ω are directly sent to the on-board flight controller of the Bebop for execution. For lateral and forward velocities, we designed a velocity controller to control the roll and pitch angles of the UAV such that it tracks the desired velocity vector. Our objective was to design a controller that generates smooth trajectories towards the target. The major challenges towards designing such a controller are the latency and low update rate of the feedback signal. Our proposed controller uses a dynamic model of the UAV to compensate for this latency and predict the feedback signal. The dynamic model we used is a first-order non-linear system that relates the roll and pitch angles of Bebop to its lateral and forward velocities respectively (Equation 5.10).

$$\begin{aligned} \dot{v}_x^b &= C_x v_x^b + g \ \tan(pitch) \\ \dot{v}_y^b &= C_y v_y^b - g \ \tan(roll) \end{aligned} \tag{5.10}$$

In this equation, g is the gravitational constant ($\approx 9.81 s^{-2}$) and C_x and C_y are the free parameters. We performed a system identification step to find C_x and C_y by flying the UAV indoors and measuring true values for *pitch*, *roll*, v_x^b and v_y^b using a high precision and frequency ($\approx$ 120 Hz) motion capture system. The estimated values for these parameters are $C_x = 0.57633\ s^{-1}$ and $C_y = 0.58498\ s^{-1}$. By minimizing the squared error between measured velocities and feedback velocities over different time offsets, we estimated the latency of the feedback as $t_d \approx 262$ milliseconds. This latency is mainly caused by the WiFi transport delay as well as the down-sampling/buffering step performed by Bebop's firmware prior to transmitting the feedback over WiFi.

As shown in Figure 5.4, once the feedback is received, the controller utilizes the dynamic model of the UAV to predict the state of the system (${}^b\hat{v}_x$ and ${}^b\hat{v}_y$) from the feedback signals (${}^bv_x^f$, ${}^bv_y^f$, roll^f and pitch^f) which are t_d seconds delayed. The PI controllers calculate the desired control values of the system (pitch^d and roll^d) by calculating the error between the feedback velocity and the desired velocity (coming from the visual servo controller). Instead of relying on the low-frequency feedback to generate the control signal which would either decrease the output rate to 5 Hz or increase its jerk because of the periodically increasing delay between the last feedback signal and the true state of the system, the controller again utilizes the dynamic model of the UAV to predict the state of the system from the last received feedback signal and the latest desired control command. Once a new feedback signal is received, it resets the state of the predictor. This way, the predictor fills the 200 milliseconds gap between two feedback readings to provide a 30 Hz estimation of this signal for the PI controller.

5.3.5 Close-range Interaction

When the UAV enters the *Approaching State*, the behavior coordinator enables the close-range interaction component of the software stack. This component consists of the human face detector and the optical flow based gesture detector of Chapter 3. The UAV uses the cascade classifier of Viola and Jones [179] to detect human faces in the image plane. It only considers the faces which their corresponding bounding boxes overlap with the tracked object's region in the image. It also uses the so called *face score* [32] (Section 3.3.2) to filter out the faces that the classifier is not confident about them. Once a candidate face is detected, it is internally tracked with a Kalman filter and its bounding box is continuously fed into the cascade controller, replacing the input from the visual tracker. Compared to the output of the visual tracker, the tracked bounding box of the face region is more consistent with the prior on its size. therefore, in case the face is detected, the resulting depth estimation will be more accurate which subsequently leads to a more precise positioning in front of the user. The UAV maintains its position on a semi-sphere around the user while keeping her face in the center of its FOV.

Feedback Animation	**State**	**Metaphor**
Search	Searching	Radar Scanner
Approach	Approaching	Pointing
Engaged	Close-range	Gaze
Selfie	Close-range	Camera Timer
Bye	Close-range	Iris
Bad Video	Any	-
Lost	Approach & Close-range	Radar Scanner

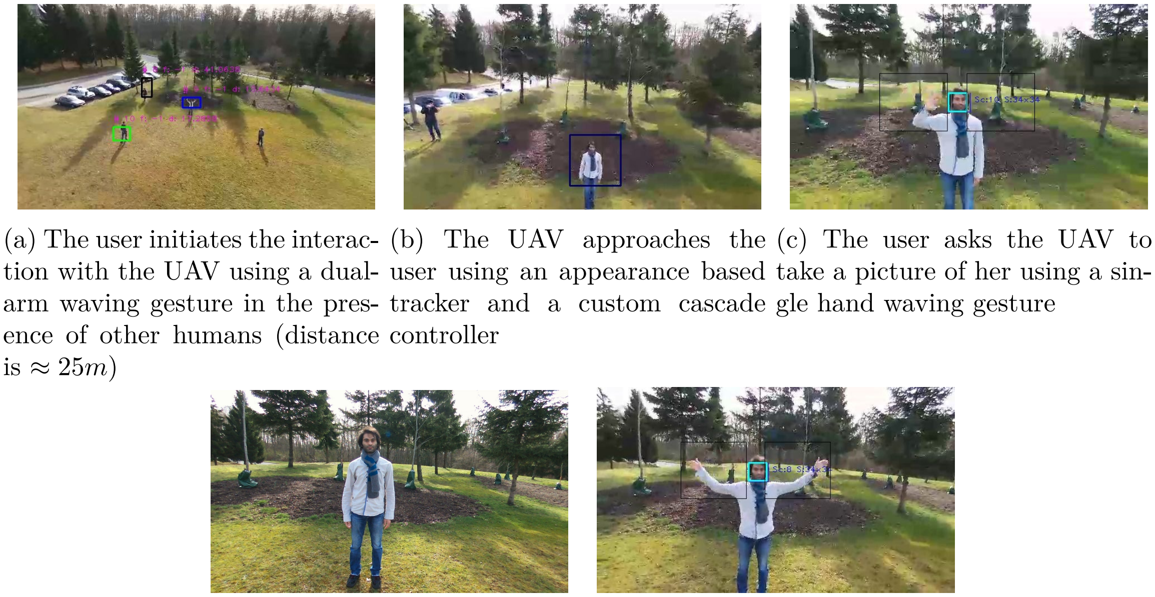
Table 5.1: Animation used for providing light-based feedback to the user, their corresponding state and metaphors.

While tracking the face, the close-range interaction component calculates the dense optical flow inside two regions around the human face. The size of these regions are linearly dependent on the size of the face and are placed such that they capture hand/arm movements. In order to cancel out the effect of ego-motion of the UAV, the median of magnitude of optical flow vectors inside the human face and the background regions are subtracted from all optical flow vectors inside the two gesture regions. A post processing step smooths the time variations of average optical flow per pixel inside the gesture regions, then applies a median filter and thresholding to decide if there is substantial motion in any of those regions. These motions are considered as left/right hand waving gestures by this component and used to communicate commands from the human to the UAV.

5.3.6 Communication of Intents from the UAV to the User

To communicate the state of the UAV and its intents to the user, we developed a custom color-light-based feedback system. This feedback systems consists of 11 individually addressable RGB LEDs mounted on the front side of the UAV, an Atmel AVR-based driver board and an Intel Edison embedded computer that executes the feedback generation software (Figure 5.3). The high level behavior coordinator (which runs on the off-board computer), communicates over WiFi to this embedded computer to request the execution of any of predefined animations based on the current state of the UAV and its next command. A custom key-frame-based animation engine runs on-board the Edison computer to generate the feedback signals. The total weight overhead of this feedback system is 55 grams.

As shown by Szafir et al. [174], using light feedback helps co-located humans deduce the flying intent of a UAV faster and more accurately (ref. Section 2.2.3). We believe this feedback modality is advantageous to other modalities previously used in this context such as sound [141] or LCD displays [105] over long distances, specifically for small form factor UAVs. In [174], the authors showed that animations based on *Gaze* and *[car] blinker* metaphors perform well to communicate the flying intention of a UAV. Inspired by these results, we designed a set of feedback signals to communicate the intent of the UAV to the



(a) The user initiates the interaction with the UAV using a dual-arm waving gesture in the presence of other humans (distance is ≈ $25m$)

(b) The UAV approaches the user using an appearance based tracker and a custom cascade controller

(c) The user asks the UAV to take a picture of her using a single hand waving gesture

(d) The resulting portrait

(e) The user terminates the interaction by performing a dual hand waving gesture.

Figure 5.6: Our end-to-end human-flying robot interaction system in action during one of outdoor experiments (Section 5.4.2).

user during each phase of the interaction process. These designed signals use colors and motion to convey the intent to the user. Table 5.1 provides a summary for all these feedback signals and their corresponding metaphors. Please refer to the supplementary video for the visualization of these signals (Section A.2).

5.4 Experiments

For all the experiments we used the platform described in Section 5.3.1. Except for the LED animation generator, all the software ran on a notebook computer with a specification matching the small form factor embedded computer we previously used for self-contained Human-UAV interaction [3] [110]. For the data intensive communication with the UAV over WiFi, we used a long-range IEEE 802.11ac external network card with a high gain antenna. We extensively used Robot Operating System (ROS) [145] to integrate different software and hardware components of this system. The cascade controller internally uses the ViSP library [102] to perform Image-Based Visual Servo.

All the source code for various components of this system (including the ROS driver for Parrot Bebop Drone, ROS bindings for the long-term visual tracker, the cascade controller, close-range interaction system, interaction initiation module and the animation generator

[3] Intel 5th generation Core i5 CPU, 8GB of RAM, SSD Storage

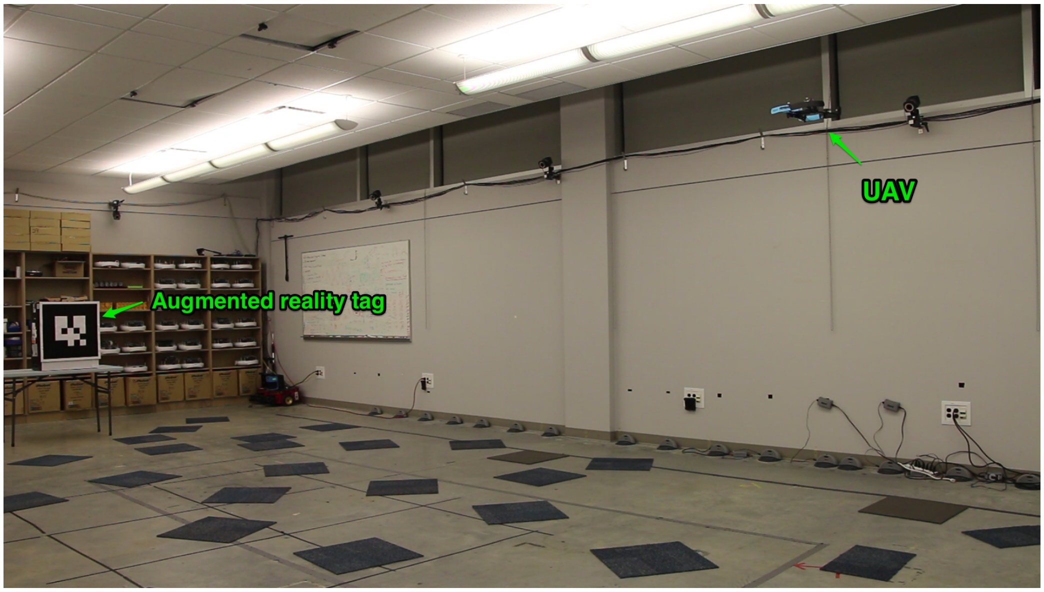


Figure 5.7: The setup of the indoor experiments for validating the approach controller (Section 5.4.1)

engine) are available online[4] and as supplementary materials of this manuscript. Please refer to Section A.1 for more information.

5.4.1 Approach Controller

The goal of this experiment was to validate the approach controller and assess its depth estimation accuracy as well as the resulting approach trajectories. We performed this experiment in a $7m \times 11m \times 3m$ indoor environment, equipped with a Vicon motion capture system. We put an augmented reality marker of size 56×56 centimeters in a fixed location of the arena (marked with `X` in Figure 5.8). The height of the center of the target from the ground was 1.175 meters. The augmented reality marker was used to bootstrap the long-range interaction initiation part of the system and to replace the visual tracker for one leg of the experiments, therefore we did not use the 6 DOF localization data these markers provide. Instead, we use the axis aligned bounding box of detected marker in the image plane to initiate or replace the visual tracker data.

In the first part of the experiment, the UAV was placed in one of 5 predefined starting locations in the room (Marked with `*` in Figure 5.8), either looking towards the target or looking forward (aligned with y-axis of the room). After takeoff, when the UAV first detects the marker, it transitions to the approaching mode and constantly use the consequent detections to feed the approach controller, replacing the visual tracker. The desired depth of the UAV with respect to the target and the camera tilt was set to 2.5m and 0° respectively. Once the sum of the velocity errors were below a certain threshold, the UAV would land.

[4] `http://autonomylab.org/bebop_hri`

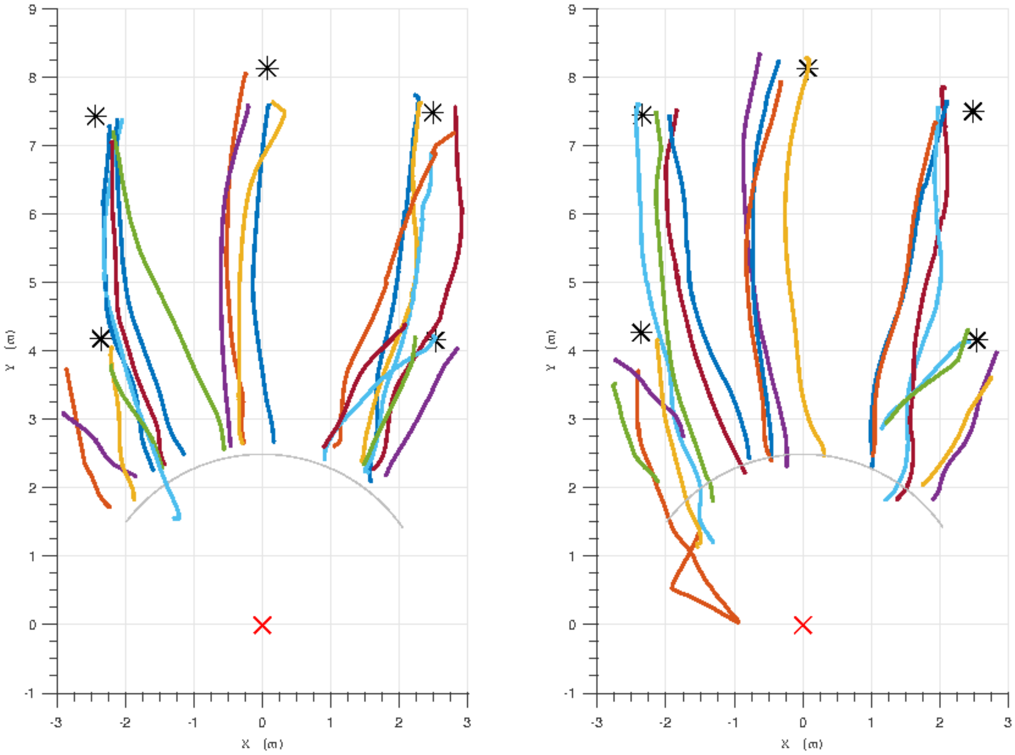


Figure 5.8: The indoor approach trajectories for leg 1 (left) and leg 2 (right)

The second leg of the experiment was similar in design with the first leg. The only difference was that the marker detection was only used once to initiate the visual tracker which would provide the reference bounding box to the approach controller. We repeated each leg of this experiment four times from each location (two for each orientation), resulting in total of 20 experiments for each leg.

Figure 5.9 shows the 2D top-down view of the ground-truth locations where the UAV decided to land relative to the target for both legs of the experiments. Since the camera was not tilted during these experiments ($\alpha = 0$, the target altitude of the UAV is expected to converge to the aforementioned height of the target center ($1.175m$). The root mean squared (RMS) error of distance and altitude error of the UAV for the first leg of the experiment was $0.242m$ and $0.064m$ respectively. The same errors measured for the second leg of the experiments (with the visual tracker in the loop) were $0.392m$ and $0.076m$. The RMS depth estimation error for the two legs of the experiments were $0.665m$ and $0.793m$ meters respectively. Figure 5.8 shows the trajectories the UAV flew to reach the target in two dimensions for each leg of the experiment.

Since in leg 1 the detection happens in every frame, the input to the approach controller more accurately corresponds to the true location of the target in the image plane, therefore the depth estimation accuracy is higher and the final location error is less for the first leg of the experiments. This means when an object detector is used to drive the approach

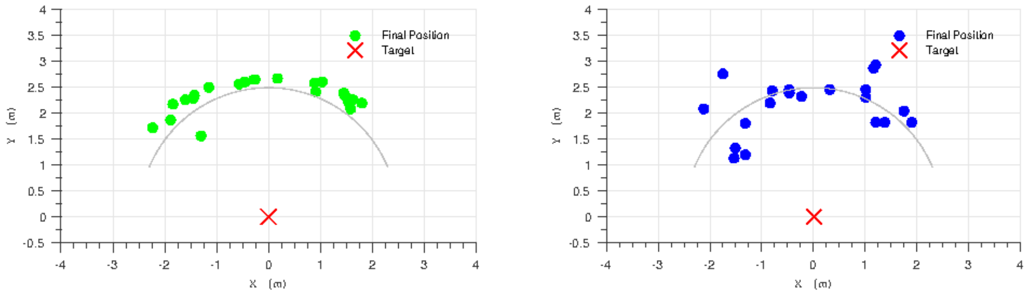


Figure 5.9: The indoor approach location error for leg 1 [continuous detection] (left) and leg 2 [detect and track] (right)

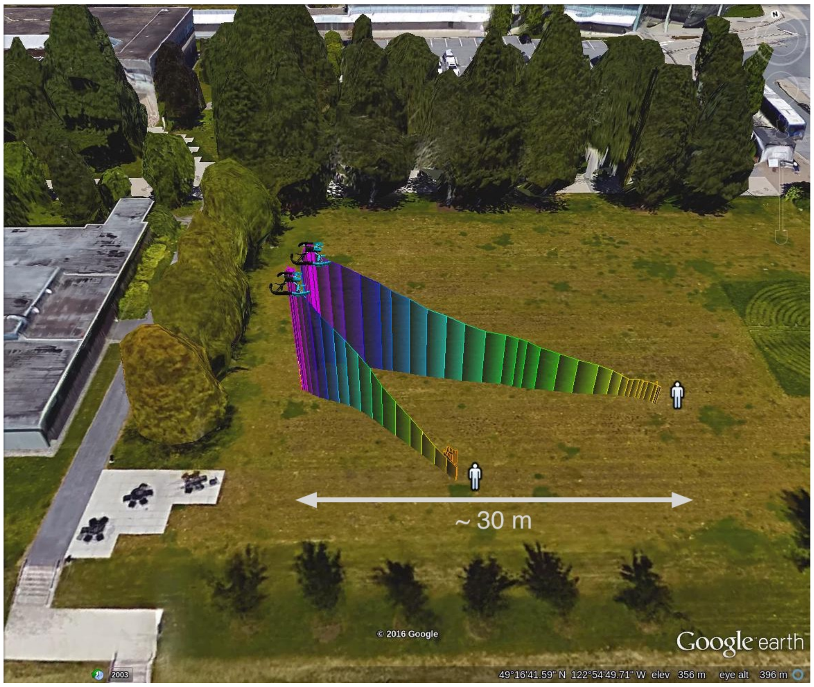


Figure 5.10: 3D rendering of two outdoor approach trajectories from actual GPS log data

controller or when the tracker does not drift much, and when the prior on the object size is precise, the final location of the UAV with respect to the target is more accurate.

5.4.2 Outdoor Experiments

To demonstrate and validate the proposed end-to-end Human-UAV interaction system, we performed a series of outdoor experiments with the platform and the setup previously described in Sections 5.3.1 and 5.4. The tilt angle of the [virtual] camera was set to 45° and this value was dynamically and independently being controlled by the behavior coordinator to smoothly tilt it to 0 towards the end of the approach trajectory. We tested the interaction system in three different locations, at three different times of the day (noon, early and late afternoon) and with 9 users. All the users were from our own research group, but

State	**Attempts**	**Success Count (%)**
Search	52	42 (80.77%)
Approach	42	40 (95.23%)
Close-range tracking	40	38 (95%)
Selfie Gesture	40	38 (95%)
Terminate Gesture	38	37 (97.37%)
Feedback System	52	51 (98.07%)
Total	**52**	**37 (**71.15%)

Table 5.2: The summary of failures in the end-to-end outdoor experiments

not necessarily familiar with the details of the interaction system in advance. In each experiment, the UAV would take off from a fixed location and towards a predefined direction. A safety pilot would correct the direction of the UAV after take-off to cancel out the yaw error during takeoff, then would put the UAV in autonomous mode. The UAV's search behavior was to hover at the fixed altitude of 12 meters and tilt its camera down to 45°. During each experiment, one user would try to attract the UAV's attention by using dual arm waving gestures. The other user(s) would act as distractors either by walking or standing in the FOV of the UAV. The UAV would execute the behavior described in 5.3.1 to find its interaction partner, approach and engage in close-range interaction with her, while constantly provide light-based feedback as described in Section 5.3.6. In close-range interaction mode, the single hand waving gesture of the user would cause the UAV to take a close-range portrait photo - a *selfie* - of her. The user then would ask the UAV to terminate the interaction and leave by performing a double hand waving gesture (*Bye Bye*). Upon receiving this command, the UAV would turn away, ascend and restart its behavior from the *searching state*. We briefed each user once in advance about the interpretation of each feedback signal. For the outdoor experiments we set the prior on the size of the region of the periodic motion (H_1) to $1.5m$ and the prior on its distance from the ground (H_2) to $1m$.

We consider an experiment to be end-to-end successful when the human and the UAV perform all steps of the interaction scenario described above. The incidents that would fail an experiment were: search behavior does not succeed in less than 45 seconds or detects a false positive, the approach behavior loses the target over the course of the approach and does not recover or takes more than 30 seconds, the UAV does not detect the user's face before getting closer than 0.5m to her and when any of gestural commands fail after more than one retries. Table 5.2 provides a summary of the failure points for all the 52 experiments. Taking all these failure points into account, 37 out of 52 experiments (71%) were successful end-to-end. Figures 5.1 and 5.6 show snapshots from the UAV's FOV during each phase of the interaction process during each experiment. Except for high resolution *selfie* shots, all other images are the actual image inputs to our interaction system.

Figure 5.10 shows two sample three-dimensional approach trajectories generated from GPS readings of the UAV.

As the breakdown in Table 5.2 shows, the major failure point of the system was in the search behavior for interaction initiation (periodic motion detector). Other components of the system performed with $> 95\%$ reliability. We also observed that for a few experiments this task took a relatively long time to find the person of interest. We measured the average and standard deviation of the response time of this component for successful runs as 34.82 and 17.02 seconds, respectively. From the 10 failures, 6 of them were due to false positives and 4 were due to the timeout (false negatives). We further analyzed the failure cases of this component by looking into the effect of different conditions on the failure. We observe that the variable frame-rate of the input video stream, the low contrast between the user and the background, the MPEG artifacts due to variable bitrate control were among the most affecting factors. The low-contrast between the user and the background were mainly caused by sunlight, failures in automatic white-balancing of the UAV's camera and the similarity of the color of user's clothes to the background which makes the user a less salient object in the environment. An immediate direction for future work is to improve this component to improve its performance in real-world settings and decrease its response. Similar to indoor trajectories (Figure 5.8), the flight trajectories of the UAV were smooth in outdoor settings, and were able to steer the UAV towards the user at the maximum speed of $\approx 2.5ms^{-1}$. The appearance based tracker performed well with occasional positional and scale drift. However, since upon detecting a face, the system would re-evaluate its estimated depth, those drifts did not cause major failures for the *approaching* behavior. We can informally report that the close-range interaction system was responsive and users found the color-light-based feedback system informative and intuitive.

5.5 Conclusion

In this chapter we presented the first demonstration of end-to-end human-UAV interaction in outdoor environments that implements (i) explicit interaction initiation; (ii) approach and re-positioning towards the user; (iii) close-range communication of commands from the user to the UAV; and (iv) communication of intents from the UAV to the user. We show how the user can use dual arm-waving gesture to attract a flying robot's attention from a distance, how an integrated visual tracking and servoing system can bring the robot to the close proximity of the user and how the user can perform close-range interaction with the UAV after the approach. Effective velocity control of the UAV based on computer vision was achieved despite a high latency control loop. We also describe how the UAV employs color-light-based feedback to keep the human informed about the intents of the UAV. We implemented this system on a low-cost consumer UAV that we believe meets the minimum safety requirements for the close-range interaction with a UAV. In a series of indoor and outdoor experiments we validated our integrated system, analyzed the accuracy of our depth estimation and approach trajectories and identified major failure points of the system.

Chapter 6

Conclusion and Future Work

In this dissertation we described our approach towards designing an end-to-end Human-Flying Robot Interaction system that relies only on embodied sensing capabilities of Unmanned Aerial Vehicles and requires no instrumentation for its users. Motivated by emerging application domains of flying robots such as Wilderness Search and Rescue, we first defined the essential components of an end-to-end interaction system for Human-Flying Robot interaction as (i) interaction initiation; (ii) approach and repositioning; and (iii) two way communication of intents and commands between flying robots and users (Def. 1). We then provided an in-depth survey of state of the art in situated Human-Flying Robot Interaction systems. We discussed how new form factors and the increasing level of autonomy of flying robots are affecting their application domains and changing the Human-Flying Robot interaction paradigm. We argued that in the new application domains for flying robots, they will start to operate in close proximity with people and with greater autonomy. This introduces new opportunities and challenges for Human-Flying Robot Interaction research, distinct from the concerns of traditional Human-UAV interaction research.

In our survey, we looked closely at related work on each component of the end-to-end interaction system. Our survey shows there exists a limited body of research on direct and uninstrumented human-UAV interaction. However, many of studies are only focused on a specific component of the end-to-end interaction system. Furthermore, we identified that explicit interaction initiation from distance and in outdoor settings, approaching towards an interaction partner using a forward-facing monocular camera and explicit communication of commands from humans to UAVs are among the least studied areas, specifically in a context of designing a practical and autonomous interaction system (Table 2.2).

In Chapter 3 we introduced the first demonstration of a situated close-range Human-UAV interaction system. We described our face engagement based method for interaction initiation and our optical flow based gesture detection method for a moving camera. In a series of indoor Human-Multi Flying Robot experiments we demonstrated the applicability of this approach and also designed a preliminary end-to-end Human-Flying Robot Interaction system with limited motion and light-based feedback.

In Chapter 4 we examined the problem of explicit interaction initiation with a UAV more closely. We introduced a fast computer vision pipeline that locates and tracks salient moving objects in a moving camera's field of view and analyzes their spatio-temporal motions to detect stationary periodic signals. We first showed the performance of the pipeline on two human activity recognition datasets, one being a challenging aerial dataset. Next we showed how this pipeline can be used to detect dual arm waving signal of a human that wants to attract a UAV's attention while being robust to non-stationary periodic motions such a walking or jogging humans. This was the first demonstration of a fully autonomous and self-contained Human-UAV interaction system in outdoor settings.

Finally in Chapter 5 we combined the interaction initiation component of Chapter 4 and the close-range interaction system of Chapter 3 with an approach controller and a light-based feedback system to design the first end-to-end Human-Flying Robot Interaction system. We provided details about different components of our proposed approach controller, namely depth estimation, visual tracking and cascade visual servo controller. We also introduced a custom made lightweight embodied light-based feedback system to constantly communicate the intent of the UAV to its fellow human interaction partner. Lastly, we tested our proposed end-to-end interaction system in outdoor settings with different users. The experiments showed that more than 70% of the time (35/52) (i) the user could successfully gain the attention of the UAV from distances up to ($25m$) by waving at it; (ii) the UAV would smoothly fly towards the user and hover $2.5m$ away from her while servoing to her face; and (iii) upon detecting her hand gestures would take a *selfie* of her (iv) all while continuously communicating its intents to her via light feedback.

As we will discuss in the following section, the components, systems and ideas we presented in this thesis require further refinements and experimentations for real-world deployments such as in search and rescue scenarios. Nonetheless, we believe that we have defined a new research direction for Human-Flying Robot Interaction. We have shown that situated and direct interaction interfaces with flying robots are practical with a broad range of applications. It also introduces interesting new challenges and research questions. The source code for all the components of our flying robot interaction systems are available as open source (Section A.1). We hope that this will help other researchers in this field reproduce our results and build upon our work to make real-world deployment of human-friendly flying-robots a reality.

6.1 Future Work

For all our Human-UAV interaction experiments, a proper user-study with naive participants would be required to justify a formal claim that these systems are "intuitive" or better than any other methods. Although we do not make this claim, but note *informally* that selecting a robot by looking at it and attracting a UAV's attention or sending commands to a UAV by waving at it is fun, and for our close-range interaction component it is responsive and feels easy and natural. As we discussed in Section 2.2.2.1, these types of formal user studies with flying robots have been mostly performed in Wizard-of-Oz manner by researchers. It would be valuable to study how practical, collocated and direct Human-UAV interaction systems are perceived by their potential end users, which interaction modalities are preferred by them and how those modalities perform in real-world scenarios. More specifically it is interesting to see if a flying-robot's autonomy affects the results reported by user studies in Section 2.2.2.1.

In the context of communication of commands from humans to UAVs, robust, embodied and fast solutions for detecting natural human gestures (and activities) from a flying robot, both in close-range interaction and long-range interaction scenarios are of great interest to the community. Our close-range gesture recognition pipeline uses a very small set of discrete gestures. In order to extend this vocabulary for real-world applications, further research is required on detecting gestures from a moving monocular camera. Furthermore, designing multi-modal interfaces as well as interfaces that are tailored for [multi-]human - [multi-]robot interaction scenarios are interesting research topics in this area. Many state of the art visual and non-instrumented gesture recognition systems rely on depth or stereo cameras to operate. In recent years, the weight and cost of depth and stereo sensors have decreased and the trend is likely to continue. One possible research direction is to adapt these techniques to work with moving cameras, similar to the work by Naseer et al. [125] which we described in Section 2.2.2.2.

As results of our interaction initiation (Section 4.4.2) and end-to-end interaction experiments (Section 5.4.2) indicate, the response time and false negative rate of the stationary periodic object detection pipeline need further improvement. Future research can focus on enhancing the individual components of to the pipeline i.e. background-foreground segmentation and salient object detection and spatio-temporal analysis. In addition, integrating state of the art human detectors into this pipeline can improve its performance. This adaptation should take into account the unique characteristics of the underlying flying platform. A few of these characteristics that we introduced in Section 2.2.1.2 are time-varying vantage points, limited on-board sensing and computational capabilities as well as the relatively small pixel size of humans. In addition, we believe that explicit interaction initiation through human activity recognition is of great value specially in environments where multiple humans might be present in flying robot's field of view. Focus of attention mechanisms based on saliency maps is one of other potential research topics in the area of interaction initiation. Furthermore, dealing with false positives during interaction initiation can be studied further. Our proposed approach in Section 5.4.2 relies on a face detector to further confirm the presence of the detected interaction partner during approach. Future research can focus on more advanced UAV behaviors such as active searching [157] to improve the robustness of the interaction initiation pipeline.

Other topics of interests are incorporating human-aware controllers for general UAV navigation and *follow the user/approach the user* scenarios as well as more studies on proxemics and safety issues in Human-Flying Robot interaction.

Overall, there are many opportunites for research and commercial development of Human-UAV interaction systems. We hope the work described in this thesis is a useful early contribution in this area.

Bibliography

[1] Amazon PrimeAir. http://www.amazon.com/b?node=8037720011, 2016. [Online; accessed 30-January-2016].

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):1–43, 2011.

[3] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 469–481. Springer Berlin Heidelberg, 2004.

[4] T. Aittokallio, M. Gyllenberg, O. Nevalainen, and O. Polo. Testing for periodicity in signals: An application to detect partial upper airway obstruction during sleep. *Journal of Theoretical Medicine*, (4), 2001.

[5] M. Allmen and C. R. Dyer. Cyclic motion detection using spatiotemporal surfaces and curves. In *Proceedings of 10th International Conference on Pattern Recognition*, pages 365–370. IEEE Computer Society Press, 1990.

[6] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1014–1021, June 2009.

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

[8] BBC News. Google plans drone delivery service for 2017. http://www.bbc.com/news/technology-34704868, 2015. [Online; accessed 30-January-2016].

[9] N. Bellotto and H. Hu. Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):167–181, Feb. 2009.

[10] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2903–2910. IEEE, 2012.

[11] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the Strongest Rigid Detector. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3666–3673, 2013.

[12] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 613–627, Cham, 2014. Springer International Publishing.

[13] C. E. Billings. *Aviation automation: the search for a human-centered approach.* CRC Press, 1997.

[14] P. Blondel, A. Potelle, C. Pegard, and R. Lozano. Fast and viewpoint robust human detection for SAR operations. In *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6. IEEE, 2014.

[15] P. Blondel, A. Potelle, C. Pegard, and R. Lozano. Human detection in uncluttered environments: From ground to UAV view. *Proceedings of 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 76–81, 2014.

[16] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2550. IEEE, 2010.

[17] P. V. K. Borges. Pedestrian Detection Based on Blob Motion Statistics. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):224–235, 2013.

[18] J.-Y. Bouguet. Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm. Technical report, Intel Corporation, 2001.

[19] A. Bourke, J. O'brien, and G. Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2):194–199, 2007.

[20] Boy Scouts of America. *Wilderness Survival Merit Badge Handbook.* Boy Scouts Of America Merit Badge Series. Boy Scouts of America, 1998.

[21] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly, 2008.

[22] J. Bruce, V. M. Monajjemi, J. Wawerla, and R. Vaughan. Tiny People Finder: Long-range outdoor HRI by periodicity detection. In *Proceedings of Canadian Conference on Computer and Robot Vision, (CRV)*, 2016.

[23] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotic Systems*, 54(1):137–161, 2009.

[24] M. Cahillane, C. Baber, and C. Morin. *Human Factors in UAV.* Research and Applications. John Wiley & Sons, Ltd, Chichester, UK, Apr. 2012.

[25] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay. Drone & Me: An exploration into natural human-drone interaction. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 361–365. ACM, 2015.

[26] F. Chaumette and S. Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, Dec. 2006.

[27] C. C. Chen and J. K. Aggarwal. Recognizing human action from a far field of view. In *Workshop on Motion and Video Computing (WMVC)*, pages 1–7, Dec. 2009.

[28] Z. Chen, Z. Hong, and D. Tao. An experimental survey on correlation filter-based tracking. *CoRR*, abs/1509.05520, 2015.

[29] R. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 17–24, 2005.

[30] J. Cooper and M. A. Goodrich. Towards combining UAV and sensor operator roles in UAV-enabled visual search. In *Proceedings of 3rd ACM/IEEE International Conference on IHuman-Robot Interaction (HRI)*, pages 351–358. ACM, 2008.

[31] G. Costante, E. Bellocchio, P. Valigi, and E. Ricci. Personalizing vision-based gestural interfaces for HRI with UAVs: a transfer learning approach. In *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3319–3326. IEEE, 2014.

[32] A. Couture-Beil, R. T. Vaughan, and G. Mori. Selecting and Commanding Individual Robots in a Multi-Robot System. In *Proceedings of 2010 Canadian Conference on Computer and Robot Vision (CRV)*, pages 159–166. IEEE, 2010.

[33] F. D. Crescenzio, G. Miranda, F. Persiani, and T. Bombardi. A First Implementation of an Advanced 3D Interface to Control and Supervise UAV (Uninhabited Aerial Vehicles) Missions. *Presence: Teleoperators and Virtual Environments*, 18(3):171–184, June 2009.

[34] M. Cummings and P. Mitchell. Predicting controller capacity in supervisory control of multiple UAVs. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(2):451–460, Mar. 2008.

[35] M. L. Cummings and S. Guerlain. Developing operator capacity estimates for supervisory control of autonomous vehicles. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(1):1–15, 2007.

[36] R. Cutler and L. S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000.

[37] M. Daily, Y. Cho, K. Martin, and D. Payton. World embedded interfaces for human-robot interaction. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Jan 2003.

[38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 886–893, 2005.

[39] M. Danelljan, F. S. Khan, M. Felsberg, K. Granström, F. Heintz, P. Rudol, M. Wzorek, J. Kvarnström, and P. Doherty. *A low-level active vision framework for collaborative Unmanned Aircraft Systems*, pages 223–237. Springer International Publishing, 2015.

[40] T. M. del Valle, C. R. del Blanco Adán, F. J. Núñez, and N. G. Santos. New generation of human machine interfaces for controlling UAV through depth based gesture recognition. In *Proceedings of SPIE Defense, Security and Sensing Conference*, volume 9084, May 2014.

[41] T. Deyle. Venture Capital (VC) Funding for Robotics in 2015. `http://www.hizook.com/blog/2016/01/12/venture-capital-vc-funding-robotics-2015`, 2016. [Online; accessed 30-January-2016].

[42] P. Doherty and P. Rudol. A UAV search and rescue scenario with human body detection and geolocalization. In *Proceedings of the 20th Australian joint conference on advances in artificial intelligence*. Springer-Verlag, Dec. 2007.

[43] P. Dollár, R. Appel, and W. Kienzle. Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *ECCV*, pages 645–659, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[44] P. Dollár, S. Belongie, and P. Perona. The Fastest Pedestrian Detector in the West. In *Procedings of the British Machine Vision Conference 2010*. British Machine Vision Association, 2010.

[45] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Proceedings of the British Machine Vision Conference*, pages 91.1–91.11. BMVA Press, 2009.

[46] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 304–311, June 2009.

[47] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: An Evaluation of the State of the Art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, Apr. 2012.

[48] J. L. Drury, J. Scholtz, and H. A. Yanco. Awareness in human-robot interactions. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 912–918, Oct. 2003.

[49] A. T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, and Computers*, 34(4):455–470, 2002.

[50] B. R. Duffy. Anthropomorphism and the social robot. *Robotics and Autonomous Systems*, 42(3–4):177–190, 2003.

[51] B. A. Duncan and R. R. Murphy. Comfortable approach distance with small Unmanned Aerial Vehicles. In *2013 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 786–792. IEEE, 2013.

[52] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision*, volume 2, pages 726–733. IEEE, 2003.

[53] M. R. Endsley. Automation and situation awareness. *Automation and human performance: Theory and applications*, pages 163–181, 1996.

[54] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrocopter. In *Proceedings of IEEE/RSJ International Cnference on Intelligent Robots and Systems (IROS)*, pages 2815–2821, 2012.

[55] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[56] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin / Heidelberg, 2003.

[57] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[58] H. Flynn and S. Cameron. Multi-modal People Detection from Aerial Video. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 815–824. Springer International Publishing, Heidelberg, 2013.

[59] T. Fong and C. Thorpe. Vehicle Teleoperation Interfaces. *Autonomous Robots*, 11(1):9–18, July 2001.

[60] D. Fox. KLD-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, pages 713–720, 2001.

[61] A. Gaszczak, T. P. Breckon, and J. Han. Real-time people and vehicle detection from UAV imagery. In J. Röning, D. P. Casasent, and E. L. Hall, editors, *IS&T/SPIE Electronic Imaging*. SPIE, Jan. 2011.

[62] A. Giusti, J. Nagi, L. Gambardella, and G. Di Caro. Cooperative sensing and recognition by a swarm of mobile robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 551–558, Oct. 2012.

[63] A. Giusti, J. Nagi, L. M. Gambardella, S. Bonardi, and G. A. D. Caro. Human-swarm interaction through distributed cooperative gesture recognition. In *Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 401–401, March 2012.

[64] R. Gockley, J. Forlizzi, and R. Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, HRI '07, pages 17–24, New York, NY, USA, 2007. ACM.

[65] E. Goffman. *Behavior in public places: Notes on the social organization of gatherings.* Free Press, Sept. 1966.

[66] J. Goldman. Drones hit new heights at CES 2016. `http://www.cnet.com/news/drones-ces-2016/`, 2016. [Online; accessed 29-January-2016].

[67] M. A. Goodrich and M. L. Cummings. Human Factors Perspective on Next Generation Unmanned Aerial Systems. In *Handbook of Unmanned Aerial Vehicles*, pages 2405–2423. Springer Netherlands, Dordrecht, Aug. 2014.

[68] M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[69] K. Haag, S. Dotenco, and F. Gallwitz. Correlation filter based visual trackers for person pursuit using a low-cost Quadrotor. In *Proceedings of Intertional Conference on Innovations for Community Services*, 2015.

[70] E. T. Hall. *The hidden dimension.* Doubleday, 1966.

[71] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 263–270. IEEE, 2011.

[72] M. Harris. Project Skybender: Google's secretive 5G internet drone tests revealed. `http://www.theguardian.com/technology/2016/jan/29/project-skybender-google-drone-tests-internet-spaceport-virgin-galactic`, 2016. [Online; accessed 30-January-2016].

[73] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with Kernelized Correlation Filters. *Pattern Analysis and Machine Intelligence*, 2015.

[74] J. Hermansson, A. Gising, M. Skoglund, and T. Schön. Autonomous landing of an unmanned aerial vehicle. In *Proceedings of Swedish Control Conference (Reglermöte).* Linköping University Electronic Press, 2010.

[75] M. Hou, R. D. Kobierski, and M. Brown. Intelligent adaptive interfaces for the control of multiple UAVs. *Journal of Cognitive Engineering and Decision Making*, 1(3):327–362, 2007.

[76] J. G. Jenner and L. M. Alvarez. Towards the development of 1-to-n human machine interfaces for unmanned aerial vehicles. In *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 211–221. IEEE, 2014.

[77] G. Jones, N. Berthouze, R. Bielski, and S. Julier. Towards a situated, multimodal interface for multiple UAV control. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1739–1744, 2010.

[78] H. Jones and S. Rock. Dialogue-based human-robot interaction for space construction teams. In *Proceedings of 2002 IEEE Aerospace Conference*, pages 73653–73645. IEEE, 2002.

[79] B. Jung and G. S. Sukhatme. Real-time Motion Tracking from a Mobile Robot. Technical report, Center For Robotics And Embedded Systems, University Of Southern California, 2005.

[80] B. Jung and G. S. Sukhatme. Real-time Motion Tracking from a Mobile Robot. *International Journal of Social Robotics*, 2(1):63–78, Dec. 2009.

[81] M.-B. Kaâniche. *Gesture recognition from video sequences.* PhD thesis, Université Nice Sophia Antipolis, 2009.

[82] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[83] M. Karam. *A framework for research and design of gesture-based human-computer interactions.* PhD thesis, University of Southampton, 2006.

[84] J. Kato, D. Sakamoto, M. Inami, and T. Igarashi. Multi-touch interface for controlling multiple mobile robots. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3443–3448, New York, NY, USA, 2009. ACM.

[85] F. Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, Jan. 2012.

[86] M. Kimura, R. Shibasaki, X. Shao, and M. Nagai. Automatic extraction of moving objects from UAV-borne monocular images using multi-view geometric constraints . In *International Micro Air Vehicles IMAV*, 2014.

[87] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, 2007.

[88] A. Kolling, S. Nunnally, and M. Lewis. Towards human control of robot swarms. In *Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 89–96, March 2012.

[89] D. Kortenkamp, E. Huber, R. P. Bonasso, and M. Inc. Recognizing and interpreting gestures on a mobile robot. In *In Proceedings of AAAI-96*, pages 915–921, 1996.

[90] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl. *AR-Drone as a platform for robotic research and education*, pages 172–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[91] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The Visual Object Tracking VOT2015 Challenge results. In *Proceedings of IEEE International Conference on Computer Vision Workshops*, pages 1–23, 2015.

[92] Y. Kuno, M. Kawashima, K. Yamazaki, and A. Yamazaki. Importance of vision in human-robot communication understanding speech using robot vision and demonstrating proper actions to human vision. In *Intelligent Environments*, Advanced Information and Knowledge Processing, pages 183–202. Springer London, 2009.

[93] I. Laptev and B. Caputo. Recognition of human actions. `http://www.nada.kth.se/cvap/actions/`, 2004.

[94] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bulthoff, and P. R. Giordano. Semiautonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles. *Mechatronics, IEEE/ASME Transactions on*, 18(4):1334–1345, 2013.

[95] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann. A prototyping environment for interaction between a human and a robotic multi-agent system. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction (HRI)*, page 185, New York, New York, USA, Mar. 2012. ACM.

[96] H. Lim and S. N. Sinha. Monocular Localization of a moving person onboard a Quadrotor MAV. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2182–2189, May 2015.

[97] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *Proceedings of IEEE 12th International Conference on Computer Vision*, pages 444–451, Sept. 2009.

[98] K. Ling, D. Chow, A. Das, and S. L. Waslander. Autonomous maritime landings for low-cost VTOL aerial vehicles. In *In Proceedings of 2014 Canadian Conference on Computer and Robot Vision (CRV)*, pages 32–39, May 2014.

[99] F. Liu and R. W. Picard. Finding periodicity in space and time. In *Proceedings of Sixth International Conference on Computer Vision*, pages 376–383, Jan 1998.

[100] F. Luetteke, X. Zhang, and J. Franke. Implementation of the Hungarian method for object tracking on a camera monitored transportation system. In *Proceedings of 7th German Conference on Robotics*, pages 1–6, 2012.

[101] T. Mantecón, C. R. del Blanco, F. Jaureguizar, and N. García. New generation of human machine interfaces for controlling UAV through depth-based gesture recognition. *SPIE Defense + Security*, 9084, June 2014.

[102] E. Marchand, F. Spindler, and F. Chaumette. ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics Automation Magazine*, 12(4):40–52, Dec. 2005.

[103] I. Maza, F. Caballero, R. Molina, N. Peña, and A. Ollero. Multimodal Interface Technologies for UAV Ground Control Stations. *Journal of Intelligent and Robotic Systems*, 57(1-4):371–391, 2010.

[104] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, et al. Speaking Swarmish: Human-Robot interface design for large swarms of autonomous mobile robots. In *Association for the Advancement of Artificial Intellegence*, 2006.

[105] M. Micire, T. Fong, T. Morse, E. Park, C. Provencher, E. Smith, V. To, R. J. Torres, D. W. Wheeler, and D. Mittman. Smart SPHERES: a Telerobotic Free-Flyer for Intravehicular Activities in Space. In *Proceedings of AIAA SPACE Conference and Exposition*, 2013.

[106] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, and N. Sebe. Time Matters!: Capturing variation in time in video using Fisher kernels. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 701–704. ACM, 2013.

[107] S. Mitra and T. Acharya. Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

[108] K. Miyoshi, R. Konomura, and K. Hori. Above Your Hand: Direct and natural interaction with aerial robot. In *ACM SIGGRAPH 2014 Emerging Technologies*, SIGGRAPH '14, pages 1–1, New York, New York, USA, 2014. ACM Press.

[109] K. Miyoshi, R. Konomura, and K. Hori. Entertainment Multi-rotor Robot that Realises Direct and Multimodal Interaction. *Proceedings of the 28th International BCS Human Computer Interaction Conference (BCS HCI 2014)*, 2014.

[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 *In Proceedings of IEEE/RSJ Intelligent Robots and Systems (IROS)*, pages 3614–3620, 2015.

[111] V. M. Monajjemi, S. Mohaimenianpour, and R. Vaughan. UAV, Come To Me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[112] 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 *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, pages 106–106, 2014.

[113] V. M. Monajjemi, J. Wawerla, and R. Vaughan. Drums: A middleware-aware distributed robot monitoring system. In *Proceedings of Canadian Conference on Computer and Robot Vision, (CRV)*, pages 211–218, 2014.

[114] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 617–623, 2013.

[115] C. H. Morimoto and M. R. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24, 2005.

[116] K. L. Mosier and L. J. Skitka. Human decision makers and automated decision aids: made for each other? In R. Parasuraman and M. Mouloua, editors, *Automation and Human Performance: Theory and Applications*, pages 201–220. CRC Press, 1996.

[117] M. Mouloua, R. Gilson, J. Kring, and P. Hancock. Workload, Situation Awareness, and Teaming Issues for UAV/UCAV Operations. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 45(2):162–165, Oct. 2001.

[118] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human Robot Interaction (HRI)*, pages 61–68, 2009.

[119] A. Nagendran, D. Harper, and M. Shah. University of Central Florida, UCF aerial camera, rooftop camera and ground camera dataset. `http://vision.eecs.ucf.edu/data/UCF-ARG.html`, 2008.

[120] A. M. Naghsh, J. Gancet, A. Tanoto, and C. Roast. Analysis and design of human-robot swarm interaction in firefighting. In *IEEE Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 255–260, 2008.

[121] J. Nagi, G. A. Di Caro, A. Giusti, and L. M. Gambardella. Learning symmetric face pose models online using locally weighted projectron regression. *Proceedings of 2014 IEEE International Conference on Image Processing (ICIP)*, pages 1400–1404, 2014.

[122] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella. *Human Control of UAVs using Face Pose Estimates and Hand Gestures.* controlling UAVs using face poses and hand gestures. ACM, New York, New York, USA, Mar. 2014.

[123] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. D. Caro. Human-swarm interaction using spatial gestures. In *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3834–3841, Sept. 2014.

[124] Y. Nakauchi and R. Simmons. A social robot that stands in line. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 357–364, 2000.

[125] T. Naseer, J. Sturm, and D. Cremers. FollowMe: Person following and gesture recognition with a quadrocopter. In *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, pages 624–630. IEEE, 2013.

[126] W. S. Ng and E. Sharlin. Collocated interaction with flying robots. In *2011 RO-MAN*, pages 143–149, July 2011.

[127] D. of Military and A. Terms. Unmanned Aerial Vehicle. In *Military and Associated Terms*. thefreedictionary.com, 2005. `http://www.thefreedictionary.com/unmanned+aerial+vehicle` [Online; accessed 29-January-2016].

[128] T. Ogioni Costalonga, L. Mendes Avila, L. Muniz, and A. Brandao. Gesture-based controllers to guide a quadrotor using Kinect sensor. In *Proceedings of 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol)*, pages 109–112, Oct. 2014.

[129] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado, and L. Merino. Motion compensation and object detection for autonomous helicopter visual navigation in the COMETS system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 19–24, 2004.

[130] W. A. Olson and N. B. Sarter. Management by Consent in Human-Machine Systems: When and Why It Breaks Down. *Human Factors*, 43(2):255–266, 2001.

[131] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár. Exploring weak stabilization for motion feature extraction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2882–2889. IEEE, 2013.

[132] H. M. Parsons and G. P. Kearsley. Robotics and Human Factors: Current Status and Future Prospects. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 24(5):535–552, Oct. 1982.

[133] D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero. A Ground Control Station for a Multi-UAV Surveillance System. *Journal of Intelligent and Robotic Systems*, 69(1-4):119–130, 2013.

[134] J. Pestana, J. L. Sanchez-Lopez, P. Campoy, and S. Saripalli. Vision based gps-denied object tracking and following for unmanned aerial vehicles. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6. IEEE, Oct 2013.

[135] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy. Computer vision based general object following for GPS-denied multirotor unmanned vehicles. In *Proceedings of 2014 American Control Conference*, pages 1886–1891, June 2014.

[136] K. Pfeil. *An exploration of unmanned aerial vehicle direct manipulation through 3D spatial interaction.* PhD thesis, University of Central Florida Orlando, Florida, 2013.

[137] K. Pfeil, S. L. Koh, and J. LaViola. Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles. In *Proceedings of 2013 international conference on intelligent user interfaces (IUL '13)*, pages 257–266, Santa Monica, California, USA, 2013. ACM.

[138] M. Piccardi. Background subtraction techniques: a review. *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104, 2004.

[139] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, 42(3-4):271–281, 2003.

[140] R. Polana and R. C. Nelson. Detection and recognition of periodic, nonrigid motion. *International Journal of Computer Vision*, 23(3):261–282, 1997.

[141] S. Pourmehr, V. M. Monajjemi, S. A. Sadat, F. Zhan, J. Wawerla, G. Mori, and R. Vaughan. "You Are Green": A Touch-to-name interaction in an integrated multimodal multi-robot HRI system. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, pages 266–267, 2014.

[142] S. Pourmehr, V. M. Monajjemi, R. Vaughan, and G. Mori. You two! Take off!: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands. In *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 137–142, 2013.

[143] S. Pourmehr, V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori. A robust integrated system for selecting and commanding multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[144] A. Proctor and E. Johnson. Vision-Only Approach and Landing. In *Proceedings of Guidance, Navigation, and Control and Co-located Conferences*. American Institute of Aeronautics and Astronautics, Reston, Virigina, Aug. 2005.

[145] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, 2009.

[146] M. Quigley, M. A. Goodrich, and R. W. Beard. Semi-autonomous human-UAV interfaces for fixed-wing mini-UAVs. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2457–2462. IEEE, 2004.

[147] Y. Ran, I. Weiss, Q. Zheng, and L. S. Davis. Pedestrian detection via periodic motion analysis. *International Journal of Computer Vision*, 71(2):143–160, 2007.

[148] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[149] L. D. Riek. Wizard of oz studies in HRI: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1), 2012.

[150] G. R. Rodríguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald. A Real-Time Method to Detect and Track Moving Objects (DATMO) from Unmanned Aerial Vehicles (UAVs) Using a Single Camera. *Remote Sensing*, 4(4):1090–1111, Apr. 2012.

[151] E. Rosten and T. Drummond. *Machine learning for high-speed corner detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[152] P. Rudol and P. Doherty. Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery. *Proceedings of 2008 IEEE Aerospace Conference*, pages 1–8, 2008.

[153] H. A. Ruff, G. L. Calhoun, M. H. Draper, J. V. Fontejon, and B. J. Guilfoos. Exploring automation issues in supervisory control of multiple UAVs. Technical report, DTIC Document, 2004.

[154] H. A. Ruff, S. Narayanan, and M. H. Draper. Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence: Teleoperators and virtual environments*, 11(4):335–351, 2002.

[155] A. Rutkin. Facebook unveils drone for beaming internet access from the sky. `https://www.newscientist.com/article/dn28003-facebook-unveils-drone-for-beaming-internet-access-from-the-sky/`, 2015. [Online; accessed 30-January-2016].

[156] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla, and R. Vaughan. Feature-rich path planning for robust navigation of MAVs with Mono-SLAM. In *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3870–3875, May 2014.

[157] S. A. Sadat, J. Wawerla, and R. Vaughan. Fractal trajectories for online non-uniform aerial coverage. In *IEEE Interntaional Conferehce on Robotics and Automation (ICRA)*, 2015.

[158] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri. A Kinect-based natural interface for quadrotor control. *Entertainment Computing*, 4(3):179–186, 2013.

[159] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Visually guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380, June 2003.

[160] J. Sattar and G. Dudek. Where is your dive buddy: tracking humans underwater using spatio-temporal features. In *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3654–3659. IEEE, 2007.

[161] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3626–3633, Washington, DC, USA, 2013. IEEE Computer Society.

[162] O. Shakernia, R. Vidal, C. S. Sharp, Y. Ma, and S. Sastry. Multiple view motion estimation and control for landing an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2793–2798, 2002.

[163] 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 *Proceedings of 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 293–300. IEEE Press, 2013.

[164] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control.* MIT Press, Cambridge, MA, USA, 1992.

[165] T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators. Technical report, DTIC Document, 1978.

[166] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[167] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[168] M. Siam and M. ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. *Proceedings of 11th International Conference on Signal Processing (ICSP 2012)*, 2:1060–1066, 2012.

[169] F. D. Smedt, D. Hulens, and T. Goedeme. On-board real-time tracking of pedestrians on a UAV. In *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8, June 2015.

[170] J. Sokalski, T. Breckon, and I. Cowling. Automatic salient object detection in UAV imagery. In *Proceedings of 25th International Conference on UAV Systems*, 2010.

[171] Y. Song, D. Demirdjian, and R. Davis. Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database. In *Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 500–506, Mar. 2011.

[172] P. Sudowe and B. Leibe. Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video. In *Computer Vision Systems*, pages 11–20. Springer Berlin Heidelberg, Berlin, Heidelberg, Sept. 2011.

[173] D. Szafir, B. Mutlu, and T. Fong. Communication of intent in assistive free flyers. In *Proceedings of the 2014 ACM/IEEE international conference*, pages 358–365, New York, New York, USA, 2014. ACM Press.

[174] D. Szafir, B. Mutlu, and T. Fong. Communicating Directionality in Flying Robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference*, pages 19–26, New York, New York, USA, Mar. 2015. ACM.

[175] F. Taralle, A. Paljic, S. Manitsaris, J. Grenier, and C. Guettier. *A Consensual and Non-ambiguous Set of Gestures to Interact with UAV in Infantrymen.* ACM, New York, New York, USA, Apr. 2015.

[176] X. Tong, L. Duan, C. Xu, Q. Tian, H. Lu, J. Wang, and J. S. Jin. Periodicity detection of local motion. In *Proceedings of 2005 IEEE International Conference on Multimedia and Expo*, pages 650–653, July 2005.

[177] P.-S. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic motion detection for motion based recognition. *Pattern Recognition*, 27(12):1591–1603, Dec. 1994.

[178] A. W. M. van Eekeren, J. Dijk, and G. Burghouts. Detection and tracking of humans from an airborne platform. *SPIE Security + Defence*, 9249:92490–92490, Oct. 2014.

[179] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:511–518, 2001.

[180] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of Proceedings of Ninth IEEE International Conference on Computer Vision*, volume 2, pages 734–741, Oct. 2003.

[181] R. von Laban. *Modern educational dance.* Princeton Book Co Pub, 1975.

[182] VTT Augmented Reality Team. ALVAR: Virtual and Augmented Reality Library. `http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html`. Accessed July 2013.

[183] S. Waldherr, R. Romero, and S. Thrun. A gesture based interface for Human-Robot Interaction. *Autonomous Robots*, 9:151–173, 2000.

[184] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 957–964, May 2012.

[185] J. R. Wilson. UAVs and the human factor. *Aerospace America*, 40(7):53–57, 2002.

[186] J. O. Wobbrock, H. H. Aung, B. Rothrock, and B. A. Myers. Maximizing the guessability of symbolic input. In *CHI'05 extended abstracts on Human Factors in Computing Systems*, pages 1869–1872. ACM, 2005.

[187] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[188] A. Xu, G. Dudek, and J. Sattar. A natural gesture interface for operating robotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3557–3563, May 2008.

[189] H. Yan, M. H. Ang Jr, and A. N. Poo. A Survey on Perception Methods for Human–Robot Interaction in Social Robots. *International Journal of Social Robotics*, 6(1):85–119, July 2013.

[190] S. Yang, S. A. Scherer, and A. Zell. An onboard monocular vision system for autonomous takeoff, hovering and landing of a Micro Aerial Vehicle. *Journal of Intelligent & Robotic Systems*, 69(1):499–515, 2013.

[191] T. Zhang, B. Zhu, L. Lee, and D. Kaber. Service robot anthropomorphism and interface design for emotion in human-robot interaction. In *Proceedings of IEEE International Conference on on Automation Science and Engineering (CASE)*, pages 674–679, 2008.

[192] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

[193] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi. Magic Cards: A paper tag interface for implicit robot control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 173–182, New York, NY, USA, 2009. ACM.

Appendix A

Supplementary Source Code and Media

A.1 Open Source Software

`ardrone_autonomy`: ROS driver for Parrot AR-Drone 1.0 and 2.0 quadrocopters

- Supplementary material filename (source code and documentation): `ardrone_autonomy_1_4_1.tgz`
- Online source code repository: `https://github.com/AutonomyLab/ardrone_autonomy`
- Online documentation: `http://ardrone-autonomy.readthedocs.io/en/latest/`
- Used in: Chapter 3

`bebop_autonomy`: ROS driver for Parrot Bebop quadrocopters

- Supplementary material filename (source code and documentation): `bebop_autonomy_0_5_1.tgz`
- Online source code repository: `https://github.com/AutonomyLab/bebop_autonomy`
- Online documentation: `http://bebop-autonomy.readthedocs.io/en/latest/`
- Used in: Chapter 5

`autonomy_human`: Face engagement and tracker module and optical flow-based gesture detector (ROS package)

- Supplementary material filename (source code and documentation): `autonomy_human_0ddb715.tgz`
- Online source code repository: `https://github.com/AutonomyLab/autonomy_hri/tree/dev/autonomy_human`
- Used in: Chapters 3 and 5

`obzerver`: Periodic salient object detector for interaction initiation (C++ library)

- Supplementary material filename (source code and documentation): `obzerver_cd33926.tgz`
- Online source code repository: `https://github.com/AutonomyLab/obzerver/tree/opencv-3.0`
- Online documentation: `http://autonomylab.org/obzerver/`
- Used in: Chapters 4 and 5

`obzerver_ros`: ROS Wrapper for `obzerver`

- Supplementary material filename (source code and documentation): `obzerver_ros_23e7d2.tgz`
- Online source code repository: `https://github.com/AutonomyLab/obzerver_ros`

- Used in: Chapters 4 and 5

`bebop_vel_ctrl`: Velocity controller for Parrot Bebop Drone (ROS package)

- Supplementary material filename (source code and documentation): `bebop_vel_ctrl_4962f39.tgz`
- Online source code repository: `https://github.com/AutonomyLab/bebop_vel_ctrl`
- Used in: Chapter 5

`bebop_vservo`: Visual servo controller for Parrot Bebop Drone (ROS package)

- Supplementary material filename (source code and documentation): `bebop_vservo_f25bfaa.tgz`
- Online source code repository: `https://github.com/AutonomyLab/bebop_vservo`
- Used in: Chapter 5

`autonomy_leds`: Firmware and animation engine for DotStar LED strips (ROS package)

- Supplementary material filename (source code and documentation): `autonomy_leds_1f9073a.tgz`
- Online source code repository: `https://github.com/AutonomyLab/autonomy_leds/tree/dev`
- Used in: Chapter 5

`bebop_hri`: The behavior generator code for the end-to-end experiments

- Supplementary material filename (source code and documentation): `bebop_hri_6d8857.tgz`
- Online source code repository: `https://github.com/AutonomyLab/bebop_hri`
- Used in: Chapter 5

A.2 Media

Chapter 3

- Close-range indoor experiments (Sections 3.4.1 and 3.4.2)
  - Supplementary material filename: `ardronehri_iros13.mp4`
  - Online video URI: `https://www.youtube.com/watch?v=xHH3GvZ52xg`
- Integrated multi-modal demonstration (Section 3.4.3)
  - Supplementary material filename: `integrated_dronehri_hri14.mp4`
  - Online video URI: `https://www.youtube.com/watch?v=heiYPVGFnEM`

Chapter 4

- Outdoor interaction initiation experiments (Section 4.4.2)
  - Supplementary material filename: `pelicanhri_iros15.mp4`
  - Online video URI: `https://www.youtube.com/watch?v=KXmgBDI_6PE`

Chapter 5

- Outdoor end-to-end Human-Flying Robot Interaction experiments and demonstration of light-based feedback signals (Section 5.4.2)
  - Supplementary material filename: `bebophri_iros16.mp4`
  - Online video URI: `https://www.youtube.com/watch?v=6kKuGHOB8XY`