# VISION-BASED DETECTION AND DISTANCE ESTIMATION OF MICRO UNMANNED AERIAL VEHICLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATİH GÖKÇE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

**VISION-BASED DETECTION AND DISTANCE ESTIMATION OF MICRO UNMANNED AERIAL VEHICLES**

submitted by **FATİH GÖKÇE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Göktürk Üçoluk _____
Supervisor, **Computer Engineering Department, METU**

**Examining Committee Members:**

Assoc. Prof. Dr. Erol Şahin _____
Computer Engineering Department, METU

Prof. Dr. Göktürk Üçoluk _____
Computer Engineering Department, METU

Assist. Prof. Dr. Sinan Kalkan _____
Computer Engineering Department, METU

Assist. Prof. Dr. A. Buğra Koku _____
Mechanical Engineering Department, METU

Assist. Prof. Dr. Kutluk Bilge Arıkan _____
Mechatronics Engineering Department, Atılım University

**Date:** _____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: FATİH GÖKÇE

Signature :

# ABSTRACT

VISION-BASED DETECTION AND DISTANCE ESTIMATION OF MICRO UNMANNED AERIAL VEHICLES

Gökçe, Fatih

Ph.D., Department of Computer Engineering

Supervisor : Prof. Dr. Göktürk Üçoluk

September 2015, 95 pages

In this thesis, we study visual detection and distance estimation of Micro Unmanned Aerial Vehicles (mUAVs), a crucial problem for (i) intrusion detection of mUAVs in protected environments, (ii) sense and avoid purposes on mUAVs or on other aerial vehicles and (iii) multi-mUAV control scenarios such as environmental monitoring, surveillance and exploration. The problem is challenging since (i) a real-time solution is required, a burden when computational power is limited by the hardware carried by an mUAV, (ii) non-convex structure of the mUAVs causes the bounding box of mUAVs to include very different background patterns, (iii) background patterns from indoor or outdoor are very complex with different characteristics and can include moving objects, (iv) mUAVs tilt and rotate unavoidably resulting in very large changes in their appearances, (v) when the camera is not stationary, motion blur is a problem, and (vi) illumination direction and brightness changes cause different images. We evaluate vision algorithms for this problem, since other sensing modalities limit the environment or the distance between the mUAVs. We test Haar-like features, Local Binary Patterns (LBP) and Histogram of Gradients (HOG) using boosted cascaded classifiers. We also integrate a distance estimation method utilizing geometric cues with Support Vector Regressors. We evaluated each method on indoor and outdoor videos collected systematically and on videos with motion blur. Our experiments show that, using boosted cascaded classifiers with LBP, near real-time detection and

distance estimation of mUAVs are possible in about $60$ ms indoors ($1032 \times 778$ resolution) and $150$ ms outdoors ($1280 \times 720$ resolution) per frame, with a detection rate of $0.96$ F-Score. However, classifiers of Haar-like features lead to better distance estimation since they position the bounding boxes on mUAVs more accurately. Our time analysis yields that classifiers of HOG train and run faster than the other algorithms.

Keywords: micro UAV, computer vision, detection, distance estimation, cascaded boosted classifiers, Haar-like features, LBP, HOG

# ÖZ

MİKRO İNSANSIZ HAVA ARAÇLARININ BİLGİSAYARLI GÖRME TABANLI ALGILANMASI VE MESAFE KESTİRİMİ

Gökçe, Fatih
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Göktürk Üçoluk

Eylül 2015 , 95 sayfa

Bu tezde, mikro insansız hava araçlarının (mİHA) görsel olarak algılanması ve mesafe kestirimi üzerinde çalıştık. Bu problem, korunması gereken alanlara izinsiz giren mUAV lerin algılanması, mİHA lar veya diğer hava araçlarının algılama ve kaçınma sistemleri ve çoklu mİHA kontrol senaryoları için önemlidir. Şu nedenler bu problemi zorlaştırmaktadır: (i) Gerçek zamanlı bir çözüm gerekmektedir. mİHA ların taşıyabilecekleriyle kısıtlı donanımlar düşünüldüğünde bu oldukça zordur. (ii) mİHA ların konveks olmayan yapıları sebebiyle, mİHA ları içine alan görüntü penceresinde, değişik arka plan görüntüleri de bulunur. (iii) Arka plan görüntüleri karmaşıktır ve hareketli nesne içerebilir. (iv) mİHA ların eğilmeleri ve dönmeleri görünümlerini değiştirir. (v) Kamera sabit değilse, hareket bulanıklığı bir problemdir. (vi) Aydınlatma yönündeki ve parlaklıktaki değişiklikler, görüntülerde büyük farklılıklara sebep olur. Diğer yöntemler, ortam ve mesafeyi sınırlandırdığı için, problemimizin çözümünde görsel verilerin kullanımını değerlendirdik. Bu amaçla, HAAR benzeri öznitelikler, lokal ikili örüntü (LBP) ve yönlü gradyan histogramları (HOG) yöntemlerini kademeli sınıflandırıcılarla test ettik. Sistemimize aynı zamanda, mesafe kestirimi için destek vektör regresörü tabanlı bir yöntemi ekledik. Herbir yöntemi, iç ve dış ortamlarda sistematik şekilde topladığımız görüntülerle ve hareket bulanıklığı içeren görüntülerle test ettik. Testlerimiz, kademeli sınıflandırıcıların LBP ile kullanımıyla gerçek zamanlı çalışmaya yakın hızda (iç ortam:$60$ ms@$1032 \times 778$,

dış ortam:$150$ ms@$1280 \times 720$) ve yüksek hassasiyette ($0.96$ F-ölçütü) algılama ve mesafe kestiriminin mümkün olduğunu göstermektedir. HAAR benzeri özniteliklerin kullanımı, mİHA ların görüntü içerisinde bulunduğu alanı daha hassas şekilde konumlandırdığı için, daha iyi mesafe kestirimi sağlamaktadır. Zaman analizlerimiz HOG yönteminin öğrenme ve çalışma zamanları açısından diğer algoritmalardan daha hızlı çalıştığını göstermektedir.

Anahtar Kelimeler: mikro İHA, bilgisayarlı görme, algılama, mesafe kestirimi, kademeli kuvvetlendirilmiş sınıflandırıcılar, Haar-benzeri öznitelikler, LBP, HOG

*To my one and only Yasemin, our son Mustafa and our families*

# ACKNOWLEDGMENTS

First of all, I want to express my immeasurable appreciation and sincere gratitude to Erol Şahin, Göktürk Üçoluk and Sinan Kalkan (In alphabetical order by names, since I cannot make any separation among them.) without whom this study could not be possible. I am deeply thankful to Erol Şahin for his support, guidance, encouragement, invaluable comments and especially for preventing me from getting lost. I am extremely grateful to my supervisor Göktürk Üçoluk for his guidance, supervision and pushing me always forward with his immense wisdom; for helping me generously whenever I needed in every aspect of the study, especially for his great support during the outdoor experiments we performed starting very early in the mornings; and most importantly for trusting me more than myself even at the hardest times. I am immensely thankful to Sinan Kalkan for his never-ending support; for his guidance at every detail of this study with his great expertise; for his visionary advices when I faced struggles; for devoting his invaluable time generously for me whenever I needed; for keeping me fresh and strong with his encouragement. I want to also express my special thanks to these three great people for creating KOVAN Research Laboratory and for giving me the chance to work with them.

I am very thankful to my committee members A. Buğra Koku and Kutluk Bilge Arıkan for their supports, suggestions, valuable comments and encouragement.

I would like to express my special thanks to two great people: Hande Çelikkanat, for always being the source of energy in the laboratory, for lending a hand to me with her expertise whenever I need, for helping me during indoor and outdoor experiments, and for being a great companion during the long hour studies in the laboratory. Ali Emre Turgut, for his incredibly generous helps and supports and for being one of the biggest source of hope in my life. Great thanks to you for being such nice symbolic persons of true friendship.

I owe a debt of gratitude to Fatoş Tünay Yarman Vural for her supports and encouragement throughout my education in METU.

I am very grateful to all faculty members of the Department of Computer Engineering of METU, especially to Adnan Yazıcı, Uluç Saranlı, Attila Özgit, İ. Hakkı Toroslu, Volkan Atalay, Cevat Şener, Veysi İşler, Selim Temizer, Sibel Tarı, Ahmet Çoşar, İsmail Sengör Altıngövde, Ahmet Oğuz Akyüz, Ali Hikmet Doğru, Halit Oğuztüzün, Faruk Polat, Tolga Can, Onur Tolga Şehitoğlu, Pınar Karagöz and Murat Manguoğlu for helping me whenever I needed and for always encouraging me. It was a great

honour for me to work with you.

I am very thankful to Okan Tarhan Tursun for sharing his knowledge and experiences with me during our long conversations. He triggered new perspectives in my mind that are very critical for the success of this study.

I owe special thanks to Alperen Eroğlu, Selma Süloğlu and Mehmet Durna for all of their helps, encouragement and true friendship.

I would like to sincerely acknowledge Erol Öztaş, Özgür Çetin, Sultan Arslan, Perihan İlgün, Muteber Gökırmak, Murat İpek, Zafer Şanal, Bülent Özdemir, Kamber Aktaş, Mehmet Demirdöğen, Mehtap Ölmez and Sedef Araz for all of their helps.

I would like to express my sincere thanks to past and current members, and interns of KOVAN Research Laboratory: Erinç İnci, Sertaç Olgunsoylu, Osman Tursun, Mehmet Akif Akkuş, Güner Orhan, Nilgün Dağ, İlkay Atıl, Barış Akgün, Doruk Tunaoğlu, Mustafa Parlaktuna, Yiğit Çalışkan, Kadir F. Uyanık, Asil Kaan Bozcuoglu, Hacer Nihal Tarkan, Mustafa Mızrak, Alper Karamanlıoğlu, Irmak Doğan, Mehmet Çelik, Levent Bayındır, Yaman Çakmakçı, Onur Yürüten, Onur Soysal, Maya Çakmak, Mehmet Remzi Doğar, Erkin Bahçeci, Muhammed Pakyürek, Güven İşcan, Metin Balaban, Barış Özkuşlar, Mustafa Tülü, Ramil Agliamzanov, Burak Velioğlu, Mehmet Akif Akpinar, Rico Morasata, İlker Bozcan, Çağrı Erciyes, Çağlar Seylan and Fatih Semiz. I am specially thankful to four of them: Erinç İnci, for his contributions during MRC project of TAI. Sertaç Olgunsoylu, for installing and writing software to make sure the indoor camera used in this study could capture frames. Osman Tursun and Mehmet Akif Akkuş, for their supports and kindest companionship.

I am very thankful to Ergin Kılıç, Habil Kalkan and Özgür Başer for their supports and encouragement.

I am indebted to my all colleagues for supporting me, especially: Muhammed Çağrı Kaya, Merve Asiler, Barış Nasır, Saliha İrem Tanrıseven, Özgür Kaya, İlkcan Keleş, Aslı Geçtav, Murat Gençtav, Serdar Çiftçi, Burçak Otlu Sarıtaş, Hüsnü Yıldız, Dilek Önal, Mustafa Levent Eksert, Ali Fatih Gündüz, Sinem Demirci, Gülcan Can, Gökdeniz Karadağ, Burçin Sapaz, Ahmet Saçan, Ercan Bölükbaşı, Merve Aydınlılar, Çelebi Kocair, Orhan Fırat, Adnan Kılıç, Ayse Gül Yaman, Hilal Kılıç, İbrahim İleri, Aybike Şimşek Dilbaz, Alperen Dalkıran, Alev Mutlu, Alper Kılıç, Oral Dalay, Fatih Titrek, Ömer Nebil Yaveroğlu, Aykut Erdem, Erkut Erdem, Can Eroğul, Umut Eroğul, Utku Erdoğdu, Buğra Özkan, Mine Yoldaş, Özlem Erdaş, Gülşah Tümüklü Özyer, Ali Anıl Sınacı and Muhammed Emin Dursun.

My very special thanks is for Murat Tamer, Hasan Meydan, Behiç Demir, Halis Sözen, Murat Özbek, Gönenç Ülker, Selçuk Korkmaz, Fatih Bitmez, Hasan Güneş and Fatih Doğan. I am privileged for having friends like you.

I also wish to thank some special people from the guest house for their helps, encouragement and friendship: Ali Sinan Dike, Murat Özkaptan, Halime Aunt, Özer Zeybek, Alter Kahraman and Ayhan Öner Yücel.

My beloved wife, my one and only Yasemin, first more than five years of our marriage coincided with the hardest times of my doctorate. Most of the times, we could not even listen to the pouring rain on the same city. You generously postponed everything about you, devoted yourself to this success, shared the stress and pressure on me and patiently waited for me with your deepest understanding, love and trust. I want you to know that I always felt you and your strongest support with me during this period and always I will. My deepest gratitudes, appreciation, love and thanks is for you. My son Mustafa, you came into our life without being aware of how a big struggle you felt into. My little but big man, you may not remember much from these times, however, I want you to know that you did it perfectly and waited for me patiently, and that your father is loving you very much. These words are for you, sibling of Mustafa. We got your news at a hard time and you became a source of happiness for us. We are waiting for you! I want to express my deeply gratitude to you, my mother-in-law Lütfiye Bayram, father-in-law Yusuf Bayram, brother-in-law M. Burak Bayram, sister-in-law Zeynep Bayram Ertuğrul and brother-in-law Mehmet Ali Ertuğrul for your enormously generous helps and supports especially when taking care of Yasemin and Mustafa while I was far away home, and for your endless encouragement. I am hugely indebted to my mother, father, sister and sister's husband for their never-ending encouragement and support.

My special thanks go to five musicians/artists Erkan Oğur, Derya Türkan, Özer Özel, Cengiz Özkan and Ross Daly for whispering the hope into my ears in my hardest times even with their saddest musics.

I am currently enrolled in Faculty Development Program (ÖYP) in Middle East Technical University on behalf of Süleyman Demirel University, and I express my acknowledge to all people involved in Faculty Development Program.

For the experiments, I acknowledge the use of the facilities provided by the Modeling and Simulation Center of Middle East Technical University (Modelleme ve Simulasyon Merkezi - MODSIMMER). I also acknowledge Turkish Aerospace Industries, Inc. (TAI) for providing the quadrotor used in this study for a research project called MRC. However, the current study is not a part of the MRC project.

# TABLE OF CONTENTS

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# LIST OF ALGORITHMS

ALGORITHMS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AdaBoost | Adaptive Boosting |
| AR | Augmented Reality |
| BOW | Bag of Words |
| BRIEF | Binary Robust Independent Elementary Features |
| BRISK | Binary Robust Invariant Scalable Keypoints |
| C-HAAR | Cascaded Detector using Haar-like Features |
| C-HOG | Cascaded Detector using HOG |
| C-LBP | Cascaded Detector using MB-LBP |
| CMO | Close-Minus-Open |
| CSS | Chirp Spread Spectrum |
| CVPR | Computer Vision and Pattern Recognition |
| EKF | Extended Kalman Filter |
| FAST | Fast Corner Detection |
| FREAK | Fast Retina Keypoint |
| GDOP | Geometric Dilution of Precision |
| GFTT | Good Features To Track |
| GMM | Global Mapping Module |
| GPS | Global Positioning System |
| GPU | Graphical Processing Unit |
| HARRIS | Harris Corner Detection |
| HMM | Hidden Markov Model |
| HOG | Histogram of Oriented Gradients |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMU | Inertial Measurement Unit |
| IR | Infrared |
| IR LED | Infrared Light-Emitting Diode |
| J | Jaccard Index |
| LBP | Local Binary Patterns |

| | |
|---|---|
| LED | Light-Emitting Diode |
| MB-LBP | Multi-Block Local Binary Patterns |
| MOCAP | Motion Capture System |
| MSER | Maximally Stable Extremal Region Extractor |
| mUAV | Micro Unmanned Aerial Vehicle |
| OpenCV | Open Source Computer Vision Library |
| ORB | Oriented FAST and Rotated BRIEF |
| PnP | Perspective-n-Point |
| PR | Precision-Recall |
| RBF | Radial Basis Function |
| RSS | Received Signal Strength |
| RTK | Real-Time Kinematic |
| RTK-GPS | Real-Time Kinematic Global Positioning System |
| SAASM | Selective Availability Anti-Spoof Module |
| SIFT | Scale Invariant Feature Transform |
| SLAM | Simultaneous Localization and Mapping |
| SURF | Speeded-Up Robust Features |
| SVM | Support Vector Machine |
| SVR | Support Vector Regressor |
| SWaP | Size, Weight and Power |
| TDoA | Time Difference of Arrival |
| ToF | Time of Flight |
| TTC | Time to Collision |
| UWB | Ultra Wide Band |

# CHAPTER 1

# INTRODUCTION

Advances in the development of micro Unmanned Aerial Vehicles (mUAVs)[1] have led to the availability of highly capable, yet cheap flying platforms[2]. This has made the deployment of mUAV systems in surveillance, monitoring and delivery tasks a feasible alternative. The use of mUAVs in monitoring the state of forest fires where the mission spreads over a large region, and flying over the fire is dangerous [106], or in delivering packages in urban areas as a faster and cheaper solution [3] is being explored. Moreover, the widespread interest in the public has also resulted in mUAVs[3] showing up in places, such as the White House, where conventional security measures are caught unprepared [42], or in traffic accidents or in fires where the presence of mUAVs, flown by hobbyists or news channels to observe the scene, posed a danger to police and fire-fighter helicopters, and resulted in delays in their deployment [65]. mUAVs have also been employed in swarm robotics research where the aim is to exploit the availability of multiple robots to accomplish complex goals collectively, faster and more efficiently than a single robot [20]. In all of these cases, the need for the automatic detection[4] and distance estimation of mUAVs, either from the ground or from a flying platform (which can be another mUAV or a helicopter) against a possibly cluttered background is apparent.

The main objective of this thesis is the evaluation of vision as a sensor for detection and distance estimation of mUAVs. This problem poses a number of challenges: First,

[1] mUAVs are UAVs less than 5 kg [18].

[2] This chapter is partially published in [38].

[3] which are often referred to as *drones*

[4] In this study, detection is considered as both determining the presence of an mUAV and estimation of its bounding box in the image when it is used in the context of computer vision.

mUAVs are small in size and often do not project a compact and easily segmentable image on the camera. Even in applications where the camera is facing upwards and can see the mUAV against a rather smooth and featureless sky, the detection poses great challenges [25, 26]. In multi-mUAV applications where each platform is required to sense its neighbors and in applications where the camera is placed on a pole or on a high building for surveillance, the camera is placed at a height that is the same or higher than the incoming mUAV, and the image of the mUAV is likely to be blended against feature-rich trees and buildings, with possibly other moving objects in the background, so the detection and distance estimation problem becomes challenging. Moreover, in multi-mUAV applications, the vibration of the platform, as well as the size, power, weight and computational constraints posed on the vision system also need to be considered.

Within this thesis, we present our work towards the development of an mUAV detection and distance estimation system. Specifically, we have created a system for the automatic collection of data in a controlled indoor environment, proposed and implemented the cascaded approach with different features and evaluated the detection performance and computational load of these approaches with systematic experiments on indoor and outdoor datasets. We evaluated robustness of the approaches to motion blur on a dataset created by artificially blurring the indoor dataset. We also developed a method to estimate the distance of an mUAV using the size of the detection window. We performed indoor experiments to evaluate the performance of this approach in terms of both distance and time-to-collision estimation.

The main contribution of this thesis is a systematic analysis on whether a mUAV can be detected and its distance can be estimated using a generic vision system under different motion patterns both indoors and outdoors. The tested indoor motion types include lateral, approach-leave, up-down and rotational motions that are precisely controlled using the physical platform that we constructed. In the outdoor experiments, we tested the approaches on videos where the mUAV performs both “calm” and “agile” motions. The effect of moving objects in the background is also analyzed with another outdoor test video. Moreover, the effect of motion blur is also analyzed in a controlled manner. To the best of our knowledge, this is the first study that presents *comprehensive and systematical investigation* of the computer vision

for detection and distance estimation of mUAVs. We showed that using computer vision near-real time detection and distance estimation of mUAVs is possible with high accuracy. Furthermore, different from some earlier studies, whose details will be given later, reducing the problem to circular ring detection [31, 50] or augmented reality marker detection [71] by placing circular rings or markers on mUAVs, our approaches use the appearance of the mUAV itself without simplifying the problem via such special objects.

As another contribution, we are making our dataset, which we prepared for this study, publicly available. This is also crucial, since such datasets are very hard to prepare and to the best of our knowledge, no dataset is currently available for working the visual detection or distance estimation of mUAVs.

The thesis is organized as follows. In Chapter 2, we review the related literature. Chapter 3 presents the details of cascaded methods we utilized. Experimental setups and the details of our dataset is provided in Chapter 4. We present the results of our experiments in Chapter 5. We conclude the thesis in Chapter 6 by providing our conclusions, future works and related discussions.

# CHAPTER 2

# RELATED STUDIES

In this chapter[1], we review the relevant studies in two parts. In the first part, general computer vision approaches related with object detection are reviewed. The second part summarizes the efforts in the literature on detection and distance estimation of mUAVs using various modalities.

## 2.1 Object Detection Approaches with Computer Vision

In Computer Vision and Pattern Recognition (CVPR), object detection has been extensively studied (see [4, 16] for comprehensive reviews), with applications ranging from human detection, face recognition to car detection and scene classification [10, 11, 23, 61, 85, 94]. The approaches to detection and recognition can be broadly categorized into three: keypoint-based approaches, hierarchical approaches and cascaded approaches.

### 2.1.1 Keypoint-based Approaches

In keypoint-based methods, CVPR usually detects salient points, called interest points or keypoints, in the "keypoint detection" phase (See Figure 2.1). In this phase, regions in the image that are likely to have important information content are identified. The keypoints should be as distinctive as possible and should be invariant, i.e., detectable under various transformations. Popular examples of keypoint detectors include Fast

[1] This chapter is partially published in [38].

corner cetection (FAST) [79, 88], Harris corner detection (HARRIS) [39], Maximally Stable Extremal Region extractor (MSER) [66], Good Features To Track (GFTT) [86] - see [90] for a survey of local keypoint detectors.

In the next phase of keypoint-based approaches, which is the "feature extraction", intensity information at these keypoints is used to represent the local information in the image invariant to transformations, such as rotation, translation, scale and illumination. Examples of the keypoint descriptors include Speeded-Up Robust Features (SURF) [8], Scale Invariant Feature Transform (SIFT) [62], Binary Robust Independent Elementary Features (BRIEF) [15], Oriented FAST and Rotated BRIEF (ORB) [81], Binary Robust Invariant Scalable Keypoints (BRISK) [55], Fast Retina Keypoint (FREAK) [92].

Extracted features are usually high dimensional (e.g., 128 in the case of SIFT, 64 in SURF, etc.), which makes it difficult to use distributions of features for object recognition or detection. In order to overcome this difficulty, the feature space is first clustered (such as using k-means), and the cluster labels are used instead of high-dimensional features for example by deriving histograms of features for representing objects. This approach, called the *bag-of-words* (BOW) model, has become very popular in object recognition (see, e.g., [21, 70, 103]). In BOW, histograms of cluster labels are used to train a classifier, such as a Naive Bayes classifier or a Support Vector Machine [19], to learn a model of the object.

In the testing phase of BOW, a window is slid over the image, and for each position of the window in the image, a histogram of the cluster labels of the features in that window is computed and tested with the trained classifier. However, the scale of the window imposes a severe limitation on the size of the object that can be detected or recognized. This limitation can be overcome to only a certain extent by sliding windows of different scales. However, this introduces a significant computational burden, making it unsuitable for real-time applications. Moreover, topological information about the features are lost when the histograms are generated which is an important piece of information for a learning system.

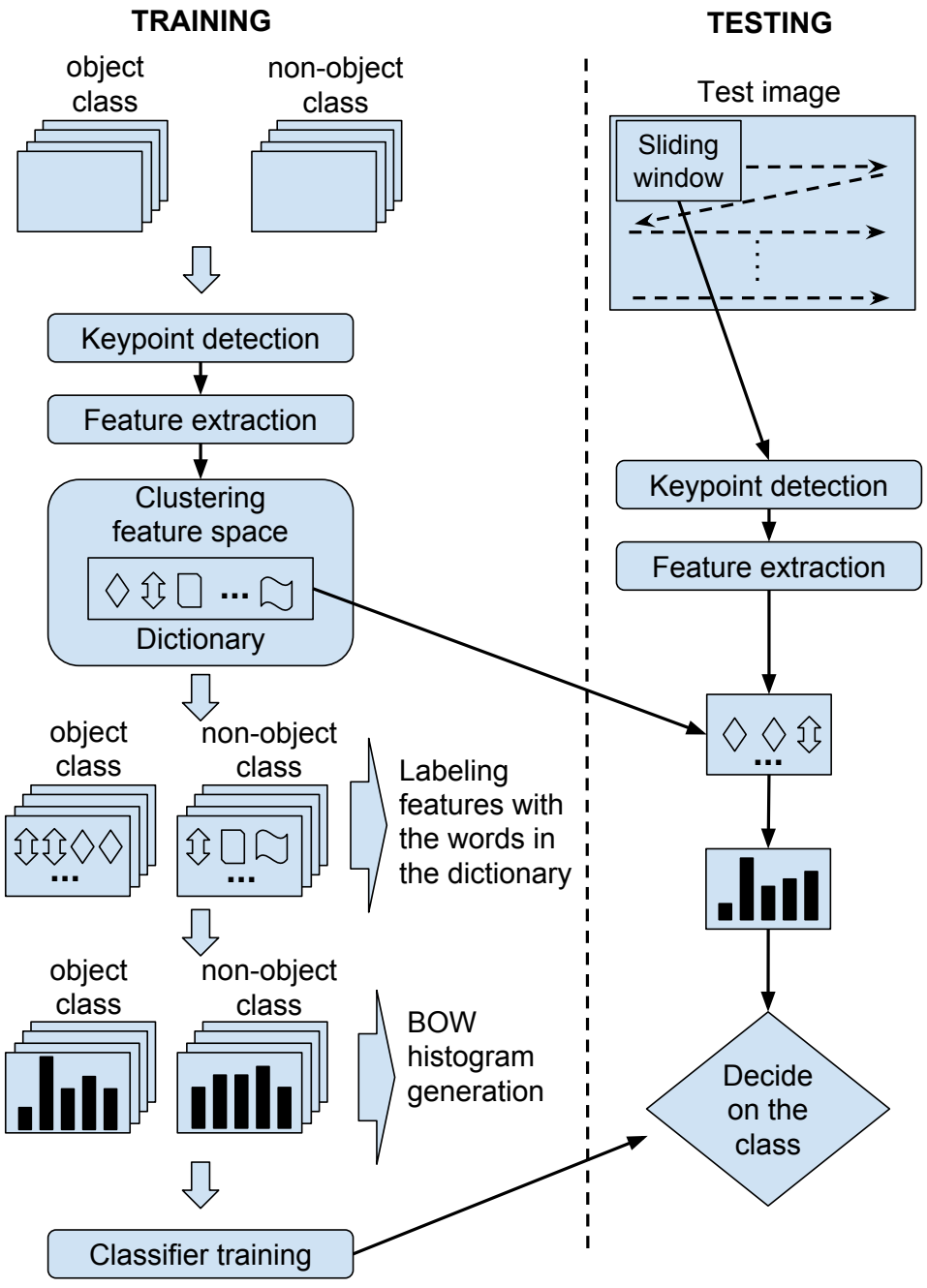



Figure 2.1: Phases of the keypoint-based bag-of-words (BOW) approach for object detection. Figure is adapted from [32] © 2005 IEEE.

### 2.1.2 Hierarchical Approaches

In these approaches, shape, texture and appearance information at different scales and complexities is processed, unlike the regular keypoint-based approaches. Processing at multiple levels has been shown to perform better than the alternative approaches (see, e.g., [51]).

In hierarchical approaches, such as the deep learning approaches [54], features of varying scale are processed at each level: in lower levels of the hierarchy, low-level visual information, such as gradients, edges etc. are computed, and with increasing levels in the hierarchy, features of the lower levels are combined, yielding corners or higher-order features that start to correspond to object parts and to objects. At the top of the hierarchy, object categories are represented hierarchically. In the hierarchical approaches, the information needs to be processed through all the levels for detection.

### 2.1.3 Cascaded Approaches

Cascaded approaches also keep a multi-level approach similar to hierarchical approaches but prune processing as early as possible if a detection does not seem likely. Those approaches are inspired from ensemble learning approaches [27] in machine learning, perform fast but coarse detection at early stages and pass only the candidate regions resulting from earlier stages on to higher stages where finer details undergo computationally-expensive detailed processing as illustrated in Figure2.2. These approaches benefit from speed ups by processing candidate regions that are highly likely to contain a match [80].

A prominent study by Viola and Jones [94, 95] which builds cascades of classifiers at varying complexities using Haar-like features and adopting the Adaboost learning procedure [34] forms the basis of our study. Viola and Jones [94, 95] applied their method to face detection and demonstrated high detection rates at high speeds. The approach was later extended to work with multi-block Local Binary Patterns (MB-LBP) for face detection [107], and Histogram of Oriented Gradients (HOG) for human detection [109], which are more descriptive and faster to compute than Haar-like features.

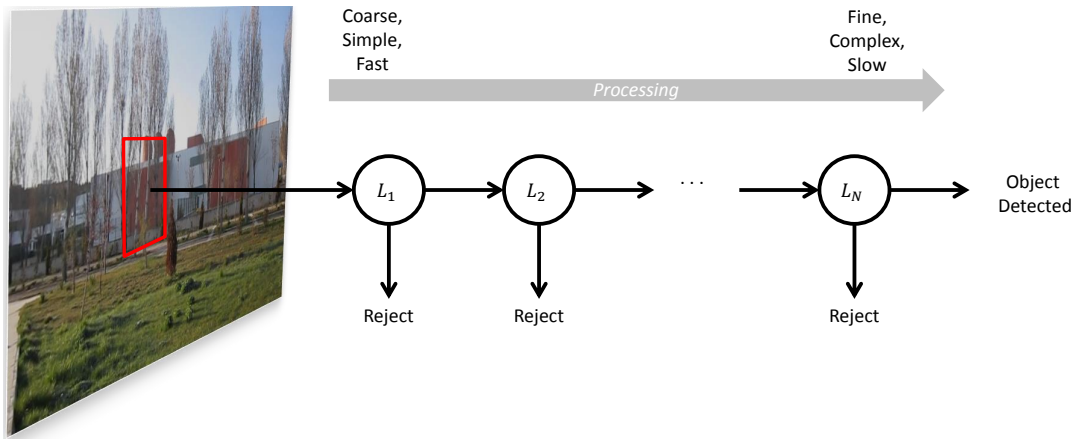



Figure 2.2: The stages of processing in a cascaded approach. At each stage, a decision to reject or to continue processing is made. If all stages pass, then the method declares the detection of the object. Figure is taken from [38].

## 2.2 Review on Relative Localization Systems for mUAVs

In this part of our review, we present studies from robotics literature including a *relative localization* system for mUAVs, since these studies become relevant when we consider detection and distance estimation of mUAVs. We should note that our aim in this thesis is not to develop a full-fledged relative localization system. However, once an mUAV is detected in an image, its relative bearing and elevation can be estimated easily and combination of these bearing and elevation information with the distance estimation results in a relative localization system.

We classify the literature on relative localization systems with respect to underlying main modality employed as (1) radio signals, (2) infrared signals, (3) sound signals, and (4) computer vision. Figure 2.3 provides the classification of these modalities with their sub-categories and also includes relevant references in each category.

As mentioned earlier, the requirement for a relative localization system is three-fold: (1) For intrusion detection purposes around non-public or private territories, (2) For sensing and avoiding purposes of mUAVs or manned aerial vehicles such as airplanes and helicopters, and (3) For using on mUAVs to develop swarms of mUAVs performing complex missions like environmental monitoring, surveillance and exploration. Depending on the application where the relative localization system is used, the expected requirements of the system will change.

In our review, the studies related to only 3D localization are included by excluding the studies on 2D localization of ground robots, since they assume a planar working environment and non-tilting robots which are not valid for mUAVs navigating in 3D space. We preferred also to include mostly the studies employing real mUAVs, however, there are some exceptions. We included these studies, since their eventual objective is to develop systems to be used with mUAVs and their results are potentially useful.

### 2.2.1 Radio Signals

One widely-used approach with radio signals is Global Positioning System (GPS). GPS is the world-wide positioning system enabling a GPS module to locate itself on the earth via receiving radio signals from the GPS satellites orbiting around the world. In a cooperative scenario, each mUAV can be equipped with GPS receivers and share their positions with other agents [40, 93, 105] or with a central control station via wireless communication [14, 44, 75]. However, GPS signals could be affected by weather, nearby hills, buildings, and trees. The service providers may also put limitations on the availability and accuracy of the GPS signals. Moreover, the accuracy of GPS signals is not sufficient for discriminating between close-by neighboring agents unless a Real-Time Kinematic GPS (RTK-GPS) system is used [13, 41].

RTK is a solution to eliminate the errors in standard GPS and to get centimeter level accuracy utilizing carrier-phase measurements. However, RTK needs a fixed base station within 6-10 miles of operating area and a wireless communication link between the base station and rover unit(s). Moreover, the initialization of RTK system requires five common satellites to be tracked by base station and the rover(s) and takes around 30-40 minutes. Once initialization is completed four satellites should be continuously tracked to get an RTK positioning. If the fixation is lost, a new initialization procedure is needed.

Due to high costs of commercial RTK solutions, there are also attempts to develop affordable products. Piksi[2] and Reach[3] are among these efforts. For high-security

[2] http://www.swiftnav.com/piksi.html
[3] http://www.emlid.com/reach/

Figure 2.3: Classification of 3D relative localization systems by the modality used.

demanding applications such as military, development of RTK systems conforming both Selective Availability Anti-Spoof Module (SAASM) feature and Size, Weight, and Power (SWaP) constraints is critical since most of the current RTK products are susceptible to spoofing and jamming. In [13], such a product from Rockwell Collins Inc. is used for relative positioning small unmanned aircraft systems.

Radar is commonly used in planes, but currently no commercial product is available satisfying SWaP constraints of mUAVs. In [68, 69], an X-band radar weighing only 230 gr is developed and tested on mUAVs. It is capable of detecting and identifying mUAVs. Distance measurement is reported as possible with this radar, however no test result is available. Patent rights of this radar is transferred to Integrated Robotics Imaging Systems[4] company. The company is aiming to integrate this radar to its mUAV and than to make it available to the market.

Technologies such as ultra-wide band (UWB) and chirp spread spectrum (CSS) enable relative distance measurement based on time-of-flight measurement of radio signals. Due to beacon units requirement, onboard 3D relative positioning relying fully on these technologies seems to be not possible. However, these technologies are very appropriate to be integrated with other approaches as an aiding method. Although it is in simulation, there is also effort to develop formation control algorithms depending on only relative distances [17] which may eventually lead to real systems by utilizing UWB and CSS technologies.

Received signal strength (RSS) information on receiver side depends on the distance between the transmitter and the receiver. Hence, RSS allows to estimate the distance between the transmitter and the receiver. However this approach has some limitations. RSS does not change linearly with the distance and it is affected from alignment of the antennas and from the objects in the environment. Therefore, an accurate distance estimation is not feasible. Moreover, in order to be able to locate an mUAV, multiple distance measurements obtained from different beacon receivers are required. Requirement of multiple beacons limits the environment where such a system is used.

[4] http://www.uav-alaska.com/

### 2.2.2 Infrared Signals

Infrared (IR) is an electromagnetic radiation whose wavelength is longer than visible light and ranges between $700$ nm and $1$ mm. IR travels at the speed of the light and is invisible to human eyes.

The IR light emitted by IR LEDs at certain wavelengths can be converted to electric current by photodiodes. As an approximation, the amount of electric current generated by a photodiode is inversely proportional to the square of distance between the photodiode and the IR light source. Therefore, it is possible to estimate the distance by measuring the output of photodiode as the received signal strength. A relative localization system could be developed by placing multiple IR LEDs and photodiodes around an mUAV [76, 78, 99]. In such a system, the LEDs emit IR light by flickering at a certain frequency and the output of photodiodes is filtered and amplified. The level of amplified signal is used to estimate the distance. The relative bearing and elevation can be estimated by inspecting which of the receivers on the mUAV get the signal. In order to get an omni-directional and a highly precise coverage, large number of IR LEDs and photodiodes should be placed on the mUAVs. This would increase SWaP of the system. Moreover, to achieve an accurate system, LEDs and photodiodes should be precisely mounted.

Use of modulated light and filtering reduces the interference with signals in the environment. If different frequencies are selected for each mUAV, then the interference between the mUAVs can also be eliminated [99], however this increases the complexity of the hardware. If the same frequency is used for all mUAVs then a time sharing protocol is needed to ensure that only one mUAV is emitting IR light at a certain time [78]. But in this case the update rate of the system will be affected. Considering that the speed of IR light is very high, obtaining a sufficient update rate can be possible.

Environmental reflections can change the intensity of the signals received which causes skews in the accuracy. Therefore, when the environment changes, the accuracy of the system can be affected if the reflectivity of the objects in the environment is different from the ones used in calibration.

Once an mUAV emits IR light, the light can reflect back from obstacles making it possible to measure distances to obstacles. In this way, the system can also be utilized for navigation purposes with modifying only its software [78]. Communication among mUAVs is also possible, however, the bandwidth would be very low [99].

RSS for a signal coming from far will be very low and a large gain amplifier will be needed. However, this high gain would result in saturation of the signals for short distances due to non-linear relation between RSS and distance. For solving this problem and get a larger and more linear dynamic range with increased signal-to-noise ratio and resolution, cascaded filtering technique can be utilized [78]. In this technique, full distance is divided into complementary regions, where for each region, a specific amplifier is used. Even with this enhancement, the distance estimation errors increases with the increasing distance.

All of the studies mentioned above are operating indoors. Due to excessive IR component in the sunlight, photodiodes may get saturated. AC coupling can be utilized to overcome this problem [78]. This technique is very effective indoors even for large ambient light changes, however, no result is available for outdoor. The operation in the night or in dark environments would be much easier.

IR LEDs can also be used in combination with IR cameras where it is possible to detect and localize an mUAV having IR LEDs mounted on it using vision methods. Due to its more close relation to vision, this approach will be presented in Section 2.2.4.1.

### 2.2.3 Sound Signals

Sound signals, audible (20 Hz to 20 kHz) or ultrasound (above 20 kHz) travels at a speed of 340.27 m/s in dry air at $15°$C. This slow propagation speed allows to measure Time of Flight (ToF) or Time Difference of Arrivals (TDoA) among multiple receivers using simple microcontrollers to estimate distance. With single transmitters and receivers mounted on each mUAV, only relative distances can be obtained. Multiple receivers mounted an mUAV can be used to localize other mUAVs carrying a transmitter using (1) trilateration and (2) multilateration.

In trilateration, ToF measurements between the transmitter and each of the receivers

are needed. A ToF measurement between the transmitter and a receiver gives the distance between them. Having at least four such distances, 3D relative position of the transmitter can be found by calculating the intersection of four spheres. To be able to measure ToF of the sound signal, the time when the transmission begins should be known at the receiver side. For this purpose, a wireless communication channel is needed in trilateration through which the transmitter sends a message when it starts the transmission.

Multilateration depends on TDoA measurements. Time differences between the arrivals of the sound signal to different receivers can be measured without knowing the start time of the signal. Therefore, no wireless communication is needed. Each time difference gives an equation of hyperboloid, since for two receivers at known positions and for a certain TDoA, the locus of possible transmitter locations forms a hyperboloid. If there are at least four different TDoA measurements, relative position of the transmitter can be calculated as the intersection of four hyperboloids.

Both trilateration and multilateration work accurately only if ToF and TDoA measurements are accurate. However, it is not possible to measure these times exactly. Moreover, the maximum distances between the receivers on an mUAV are very small resulting in a poor geometry for Geometric Dilution of Precision (GDOP). Combined with the errors in ToF and TDoA measurements, this poor geometry causes large error bound for the possible locations of the transmitter. These problems can be solved by fusing TDoA measurements with inertial sensors and by applying some filtering methods such as particle filtering [6, 7]. In this approach, TDoA measurements are used to estimate relative bearing and elevation of other mUAV. Then, particle filtering is utilized to robustly estimate relative position by fusing erroneous bearing and elevation estimations with the relative motion information of the mUAVs gathered via the inertial measurement sensors and shared between the platforms through a communication network.

Slow speed of sound limits achievable maximum update rate of a localization system. In addition, since only one transmitter can be active at a given time, the update rate of the system will decrease as the number of mUAVs in the swarm increases. Furthermore, sound signals are prone to reflections which complicate to identify incoming

signals correctly.

Audible [6, 7] or ultrasound signals can be used to develop a relative localization system with their own advantages and disadvantages. There are omnidirectional transducers in audible band, however, ultrasound transducers have narrower beam angles which requires lots of sensors to develop an omnidirectional system. The systems working in audible band are more susceptible to noise sources since most of the sound signals in the environment are in audible band. In this respect, the sound of the motors on mUAVs becomes the first and unavoidable disturbance source in the system [6, 7].

### 2.2.4 Computer Vision

Electro-optic cameras capture frames by projecting the light (visible or not) incoming to their lenses onto their imaging sensors. Imaging sensors have a number of pixels and each pixel converts the light to a digital value so that we get digital image frames. Various properties of the captured scene can be understood by processing these frames. The literature reviewed under this section utilizes computer vision, image processing and pattern recognition techniques on digital images to develop localization systems. We divided the literature of this section into four according to the properties of the images utilized: (1) including active markers mounted on the mUAVs, (2) including passive markers mounted on the mUAVs, (3) including the surrounding scenes of the mUAVs to utilize environmental cues, and (4) including mUAVs without any special markers to sense the mUAVs directly via their native appearance.

#### 2.2.4.1 Use of Active Markers

It is possible to attach active markers to pre-defined points on mUAVs and use them for sensing. For being able to use these markers, one should first detect them in the images. For making this detection easier, commonly IR LED markers and cameras with day-light filter [30, 102] or power LEDs emitting visible light and standard cameras with IR filter [24, 64, 101] are used. In this way, a dark image with bright points

indicating the markers can be obtained and a simple thresholding give blobs for the markers. Then using a blob detection algorithm, pixel coordinates of the markers are obtained. One other approach is to use IR LEDs with the camera detached from Wii Remote which tracks up to four LEDs only [22, 29, 100].

Once the markers are located in the image, correspondences between 2D marker positions and actual markers should be determined. Then relative position of the mUAV can be calculated by solving the well known *Perspective-n-Point* (PnP) problem which is the estimation of the camera pose using a 3D-2D corresponding point set. Here $n$ defines the number of markers. At least three markers are required to obtain a solution. It is appropriate to place more than three markers on the mUAVs to increase the accuracy and to overcome any occlusion problem. Solution of PnP provides also the attitude of other mUAVs which could be useful but not a crucial requirement of a relative localization system.

Since the markers are active, the system can operate day and night. The system can work with multiple mUAVs as long as the 3D point patterns on different mUAVs are distinct. The accuracy of the system depends on the resolution of the camera and decreases with the distance.

#### 2.2.4.2 Use of Passive Markers

In this approach, the mUAVs are localized by detecting predefined passive markers mounted on them. Since no power is required to activate markers, this approach is advantageous over active marker usage in terms of power requirements, but sufficient lightning is required for proper operation.

Different types of markers can be utilized. One type is planar markers which enable the localization by detecting only a single marker. These markers can include planar geometric shapes such as circular ring [31, 50, 82] or colored circle [89] or planar augmented reality (AR) markers [71]. Distinctive nature of these markers due to their simple geometric shape, color or pattern allows fast and easy detection. In case of geometric shapes, the size and appearance of the markers enable bearing, elevation and distance estimation, for example, by fitting an ellipse to the contour points of the

detected shape in case of a certain size circular marker. AR markers allow to estimate relative position of the camera with respect to the marker due to their inherent properties. Maximum detection distance depends on the size of markers and the camera resolution, and the accuracy decreases with distance. However, physical size and payload capacity, and the processing power of the mUAV limits the marker size and the camera resolution, respectively. Although detection of a single planar marker is enough for localization, due to their planarity, multiple markers should be mounted around the mUAV to obtain omni-directional coverage.

The problem due to size limitation of a single marker can be overcome by placing multiple relatively small 3D objects like colored balls [47] on pre-defined points on each mUAV. After detecting these objects in the image and finding their correspondences, the problem becomes PnP similar to the case where active markers are used (Section 2.2.4.1). Regarding the scalability to multiple mUAVs, occlusion and accuracy problems stated for active markers hold also for this method.

#### 2.2.4.3 Sensing via Environmental Cues

mUAVs can also locate each other by sharing the visual information they obtain about environment among themselves. For example, with monocular down looking cameras mounted on the mUAVs, they can locate each other by sharing the images among each other and utilizing inertial measurement units (IMU) [2]. In this method, camera images on different mUAVs should have an overlapping region so that transformation between two cameras are calculated using the corresponding feature points in this region. But, this transformation is not in absolute scale. Absolute scale is estimated using IMU data inside an Extended Kalman Filter (EKF) framework.

In another method, each mUAV builds partial map of the environment and localizes itself on this map via Simultaneous Localization and Mapping (SLAM). Each mUAV shares its local map and the image frames captured by its camera with a central global mapping module (GMM). GMM can be either a ground station computer [33, 83] or a computationally more powerful mUAV in the swarm. Although the latter is feasible with current technology, no implementation is available in the literature. Collecting images and local maps from each mUAV requires a communication channel with

high bandwidth. This bandwidth requirement can be overcame by sharing only features in some key frames with relative pose estimates [33]. GMM also demands high processing power due to costly operations required for identifying loop closures and path intersections of mUAVs.

#### 2.2.4.4 Direct Sensing via Native Appearance of mUAVs

Unlike approaches placing predefined markers on the mUAVs and reducing the problem to the localization of those markers, this approach utilizes directly the native appearance of actual mUAV. In this respect and also due to large appearance variations resulting from viewpoint changes, the problem becomes harder. Moreover, if the mUAV is in concave structure, the varying background patterns inside the bounding box encapsulating the mUAV also complicate the problem.

Object detection techniques in the literature can be applied to the problem. However, real-time requirements and processing power available on the platforms limit the applicable techniques. Boosted cascaded classifiers are well suited for this purpose since they achieve real-time performance with high detection rate[59, 95, 108]. These classifiers are composed of multiple stages with increasing complexities. Candidate bounding boxes passing all stages are considered as true detection, and any failure at earlier stages results in pruning of the region. Therefore, most of the regions are pruned with simpler checks performed at earlier stages. Only a small portion of the regions is checked with more complex stages. In this way, these classifiers perform faster than the classifiers having only single complex stage.

Training of cascaded classifiers requires mUAV and background images. Collection of descriptive training image set is critical for the performance of the system. Different feature extraction methods can be used such as Haar-like features (extensions of Haar wavelets to images), Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG)[5]. Via boosting, most descriptive features are selected and used to create classifier stages with increasing complexities.

The first use of cascaded classifiers for mUAV detection appears in [59] by Lin et

[5] To the best of our knowledge, there is no study in the literature using LBP and HOG with cascaded classifiers to sense mUAVs until this study. See Table 2.1.

al. The authors demonstrated a leader-follower formation flight of two quadrotor mUAVs in an outdoor environment. Relative localization is obtained via monocular vision using boosted cascaded classifiers of Haar-like features for detection and Kalman filtering for tracking. In order to estimate distance, they used the width of the leader with the camera model. They tested their vision-based formation algorithm in a simulation and with real mUAVs. The results are provided where the follower tries to keep $6$ m distance from the leader flying up to a speed of $2$ m/s. Their results present only the relative distance of the mUAVs during a flight where the distance information is obtained probably (not mentioned clearly) from GPS. Although the tracking errors were claimed to converge to zero, the results indicate that errors increase while the leader has a forward motion. Only when the leader becomes almost stationary after $35$ s of the total $105$ s flight do the errors start to decrease.

In [108], Zhang et al. studied relative pose estimation problem by extending the approach of Lin et al. [59] without modifying the mUAV detection method. They utilize a set of previously collected images for different view angles whose roll, pitch and yaw angles are recorded via a motion capture system. Once mUAV is detected via a cascaded classifier, its contours are extracted and represented as a shape context [9]. The matching image from the previously collected image set (prototypes) for this contour is found using a shape matching method based on Hungarian algorithm [52]. Once the matching prototype is found, the orientation of the mUAV with respect to the prototype is estimated by computing the best fitting affine transformation via least squares optimization. Their experimental results are not sufficient to deduce the performance of pose estimation. Furthermore, they use the estimated pose to enhance the relative distance estimation method applied in [59]. According to the results given for only $50$ frames, there seems to be an improvement; however, the error is still very high (up to three meters for a $10$ m distance with a variance of $1.01$ m) and GPS is taken as the ground truth whose inherent accuracy is not very appropriate for such an evaluation.

Both studies [59, 108] mentioned above use boosted cascaded classifiers for mUAV detection; however, they provide no analysis about the detection and computational performance of the classifiers. The methods are tested only outdoors, and the results for the tracking and pose estimation are poor for evaluating the performances of the

methods. They use only Haar-like features directly without any investigation. Moreover, no information is available about the camera and processing hardware used. The detection method is reported to run at $5$ Hz.

In an earlier study, Petridis et al. [74] used cascaded classifiers also for detecting aircrafts. Although we include mainly the literature proposed for mUAVs in this section, such studies on aircraft detection are noteworthy, since they are potentially useful for mUAVs, as long as the size, weight and power (SWaP) constraints of mUAVs are complied with. Petridis et al. studied aircraft detection under the presence of heavily cluttered background patterns for collision avoidance purposes. They applied a modified version of boosted cascaded classifiers using Haar-like features for detection. Temporal filtering is also integrated with the system to reduce false positives by checking the previous detections around a detection before accepting it as valid. Their method does not estimate the distance. Experimental results presented on videos recorded via a camera mounted on an aircraft and having a collision course and crossing scenarios indicate a detection rate of around $80\%$ with up to $10$ false positives per frame. No distance information is available between target and host aircrafts. Looking at the images, the distance seems to be on the order of some hundred meters. The performance of the system in close distances is also critical, which is not clearly understood from their experiments. They report that their method has a potential of real time performance; however, no information is available about the frame size of the images and the processing hardware.

In addition to cascaded classifiers, another method utilized in direct sensing approach is morphological filtering. For example, Lai et al. [53] studied collision detection problem for fixed-winged mUAVs using a morphological filter based on close-minus-open (CMO) approach in preprocessing stage. CMO is a combination of top-hat and bottom-hat filters which highlight the regions brighter and darker than background, respectively. CMO merges the properties of these two filters. Since morphological filters assume a contrast difference between the object and the background, once the image is preprocessed, the resulting candidate regions should be further inspected to get the final estimation. This is very crucial, as the morphological filters produce a large amount of false positives, which have to be eliminated. For this purpose, they combined the morphological filtering stage with two different temporal filtering

techniques, namely Viterbi-based and Hidden Markov Model (HMM) based. The impact of image jitter and the performance of target detection are analyzed by off-board processing of video images on a graphical processing unit (GPU). For jitter analysis, videos recorded using a stationary camera are used by adding artificial jitter at three increasing levels, low, moderate and extreme. Both temporal filtering techniques demonstrate poor tracking performances in the case of extreme jitter where inter-frame motion is greater than four pixels per frame. Some failure periods are also observed for the HMM filter in the moderate jitter case. Target detection performance experiments are performed on videos captured during three different flights with an onboard camera mounted on a UAV. Two of these include head-on maneuvers, and in the third one, UAVs fly at right angles to each other. A detection distance between $400$ and $900$ m is reported allowing one to estimate a collision before $8-10$ s to the impact.

In [25, 26], Dey et al. presented another morhological filtering based study for aircraft detection for sensing and avoiding purposes. They propose a detection method without distance estimation consisting of three stages, which are: (1) morphological filtering, (2) Support Vector Machine (SVM) based classification of the areas found by stage 1, and (3) tracking based on the similarity likelihoods of matching candidate detections. They tested the method on videos recorded using stationary cameras of various imaging sensor, lens and resolution options. These videos include aircraft flying only above the horizon; therefore the background patterns are less challenging than the below horizon case, which is not investigated in the study. A detection rate of $98\%$ at five statute miles with one false positive in every $50$ frames is reported with a running time of $0.8$ s for $4$ megapixel frame.

Our approach in thesis thesis also fits into this category. Table 2.1 summarizes the studies in this section in terms of various aspects and compares the studies in the literature with our study. Looking at this comparison table and above explanations, our study fills a void with regard to the comprehensive and systematical analysis of cascaded methods with videos including very complex indoor and outdoor scenes providing also an accurate distance estimation method. Moreover, our study is also remarkable, since it investigates also the use of LBP and HOG, to the best of our knowledge for the first time, with cascaded classifiers for mUAV detection.

Table 2.1: Comparison of the studies on visual detection of aerial vehicles via direct sensing approach.

| **Study** | **Vehicle** | **Detection Method** | **Detection Performance** | **Motion Blur** | **Training Time** | **Testing Time** | **Background Complexity** | **Environment** | **Distance Estimation** |
|---|---|---|---|---|---|---|---|---|---|
| Lin et al., 2014 | mUAV | Boosted cascaded classifiers with Haar-like features | No | No | No | No | Medium | Outdoor | Yes ( low accuracy) |
| Zhang et al., 2014 | mUAV | Boosted cascaded classifiers with Haar-like features | No | No | No | No | Medium | Outdoor | Yes ( low accuracy) |
| Petridis et al., 2008 | Aircraft | Boosted cascaded classifiers with Haar-like features | Yes | No | No | No | High | Outdoor | No |
| Dey et al., 2009; 2011 | Aircraft | Morphological filtering | Yes | No | NA | No | Low | Outdoor | No |
| Lai et al., 2011 | mUAV | Morphological filtering | Yes | Yes | NA | Yes | High | Outdoor | No |
| **Our study** | **mUAV** | **Boosted cascaded classifiers with Haar-like, LBP and HOG features** | **Yes** | **Yes** | **Yes** | **Yes** | **High** | **Indoor and Outdoor** | **Yes** |

# CHAPTER 3

# A CASCADED APPROACH TO MUAV DETECTION

In this chapter, we describe the details of our methods. We will first introduce the three different feature description methods, namely Haar-like features, local binary patterns and histogram of oriented gradients, which will be later utilized to construct three different cascaded classifiers. We will also introduce integral images and integral HOG methods utilized to compute the features in a computationally efficient manner. After that, we will present AdaBoost learning procedure and explain how a strong classifier can be created by selecting the best performing weak classifiers from a large weak classifier set and then linearly combining them. We will later describe the construction procedure of a cascaded classifier. The chapter will end by explaining our distance estimation method.

This chapter is partially published in [38].

## 3.1 Haar-like Features

Haar-like features [58, 73, 94, 95] are similar to Haar wavelet family, which is a rescaled function sequence [96, 97]. Similar to Haar wavelets, Haar-like features are defined for images in the form of various configurations of $+$ and $-$ regions in an image window. Figure 3.1 depicts simplest Haar-like feature prototype with one $+$ region and one $-$ region, and its application on an image. The value of the feature at a given position in the image is calculated by subtracting the sum of intensities in $-$ region from the sum of intensities in $+$ region.

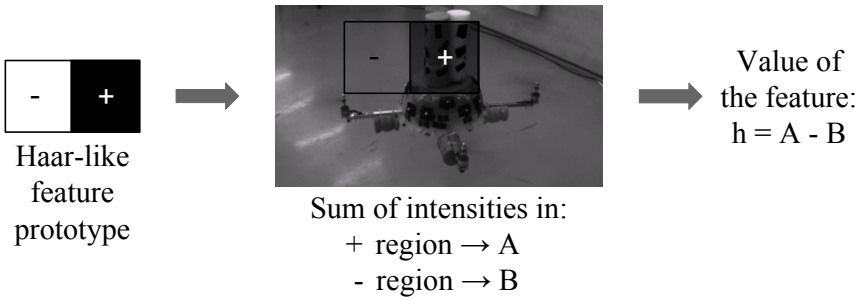



Figure 3.1: Simplest Haar-like feature prototype and its application on an image.

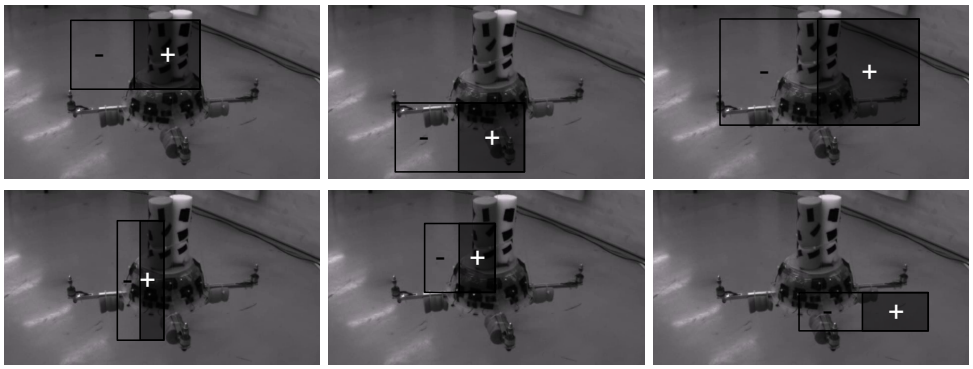


Figure 3.2: A Haar-like feature prototype can be applied to an image by changing location, size and aspect ratio resulting in different features associated with each configuration.

Different features can be calculated by locating the feature prototype inside an image at different locations and also by changing the size and aspect ratio of the prototype, as shown in Figure 3.2.

Various Haar-like feature prototypes can be defined for different configurations of $+$ and $-$ regions. A basic prototype set is defined and used with cascaded classifiers by Viola and Jones [94, 95] which is later extended by Lienhart and Maydt [58]. Figure 3.3 illustrates the feature prototypes used in our study.

If we consider an image window with $40 \times 22$ pixel size (This is the size of our training images as will be presented in Section 4.2.), when we apply our feature prototypes on this image window for all possible locations, sizes and aspect ratios, we get $587408$ different associated features. The number of associated features for each feature prototype are given in Table 3.1 .

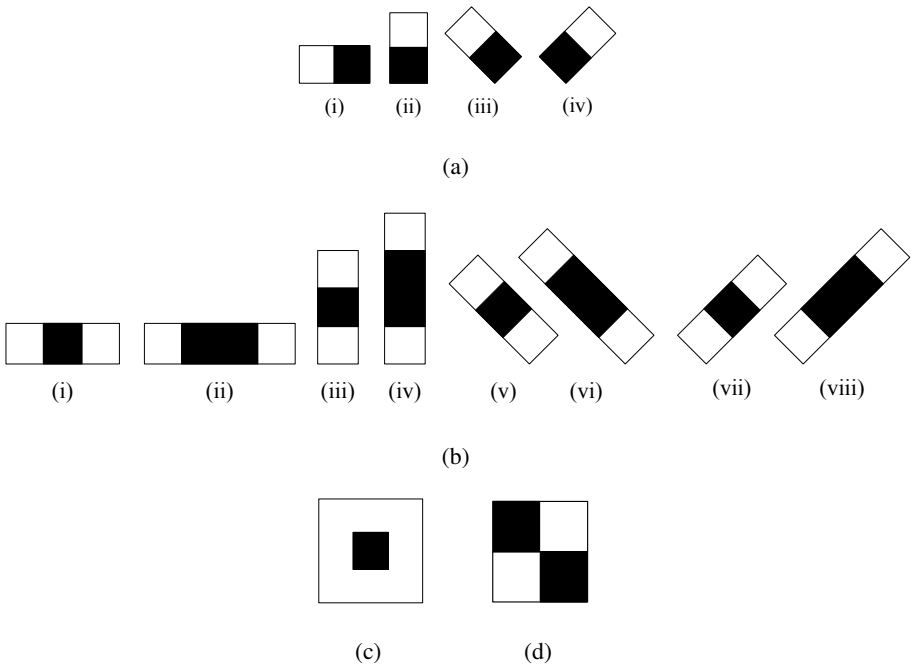



Figure 3.3: Haar-like feature prototypes used: (a) edge features, (b) line features, (c) center surround feature and (d) diagonal feature. Tilted features are rotated $45°$. Edge features have two regions and line features contain three or four regions. Center surround and diagonal features are composed of nine and four regions, respectively. Each region in a feature have the same size and shape. The total areas of $+$ (black) and $-$ (white) regions are equal in a feature except for the features with three and nine regions. In order to compensate the area inequalities in these two features, sum of the intensities in black regions are multiplied by two and eight, respectively for the features with three and nine regions, before subtracting the sum of intensities in white regions. Figures are adapted from [58] © 2002 IEEE.

Table 3.1: Number of associated features for Haar-like feature prototypes given in Figure3.3 when they are applied to an image window of $40 \times 22$ pixel size for all possible locations, size and aspect ratio.

| **Feature type** | | **# of features** |
|---|---|---|
| a | i | 101200 |
| | ii | 99220 |
| | iii | 23705 |
| | iv | 23705 |
| b | i | 65780 |
| | ii | 48070 |
| | iii | 63140 |
| | iv | 45100 |
| | v | 14539 |
| | vi | 9995 |
| | vii | 14539 |
| | viii | 9995 |
| c | | 20020 |
| d | | 48400 |
| **TOTAL** | | **587408** |

### 3.2 Local Binary Patterns

In LBP [72], a window is placed on each pixel in the image, and within which the intensity of the center pixel is compared against the intensities of the neighboring eight pixels. During this comparison, larger intensity values are taken as one and smaller values as zero, and an integer number is calculated by concatenating the binary ones and zeros. To describe it formally, for a window $\Omega(x_c, y_c)$ at pixel $(x_c, y_c)$ in image $I$, LBP pattern $L_p$ is as $L_p(x_c, y_c) = \otimes_{(x,y)\in\Omega(x_c,y_c)}\sigma(I(x, y) - I(x_c, y_c))$, where $\otimes$ is the concatenation operator, and $\sigma(.)$ is the unit step function:

$$\sigma(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.1}$$

The concatenation of $1$'s and $0$'s can be converted to a decimal number, representing the local intensity distribution around the center pixel with a single number:

$$L_2(x_c, y_c) = \sum_{i=0}^{|\Omega(x_c,y_c)|} 2^i \times L_p^i(x_c, y_c). \tag{3.2}$$

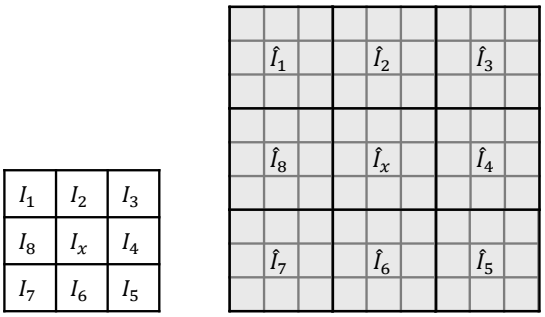



Figure 3.4: In basic LBP, the center pixel is compared to its eight neighbors in a $3 \times 3$ window (left). In the multi-block version, average intensities in the blocks are compared instead (right). Figure is taken from [38].

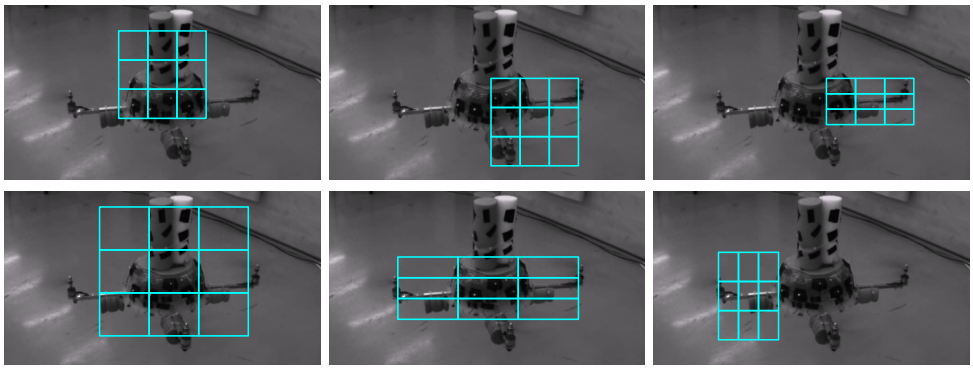


Figure 3.5: Different MB-LBP features can be associated on an image by changing the location size and aspect ratio of the blocks.

The cascaded approach of Viola and Jones [94, 95] has been extended by Zhang et al. [107] to use a *multi-block* version of LBP (MB-LBP) features [57]. In multi-block LBP, instead of comparing the intensities of pixels, the average intensities of blocks in the window are compared to the central block; see Figure 3.4. Although the blocks in Figure 3.4 are square with $3 \times 3$ size, the aspect ratio and size of the blocks can be changed preserving that the nine blocks have the same size and shape. Similar to Haar-like features, various MB-LBP features can associated on an image by changing the location size and aspect ratio of the blocks, as shown in Figure 3.5.

Since our training image size is $40 \times 22$ pixels, as will be explained in Section 4.2, we used MB-LBP features with $3 \times u, 3 \times v$ pixels sizes[1], where $u = 1, ..., 13$ and $v = 1, ..., 7$. When we consider all possible locations for different sized features inside $40 \times 22$ pixels image window, we obtain $20020$ different feature associations.

[1] $u = 1$ and $v = 1$ case corresponds to the original LBP in [72].

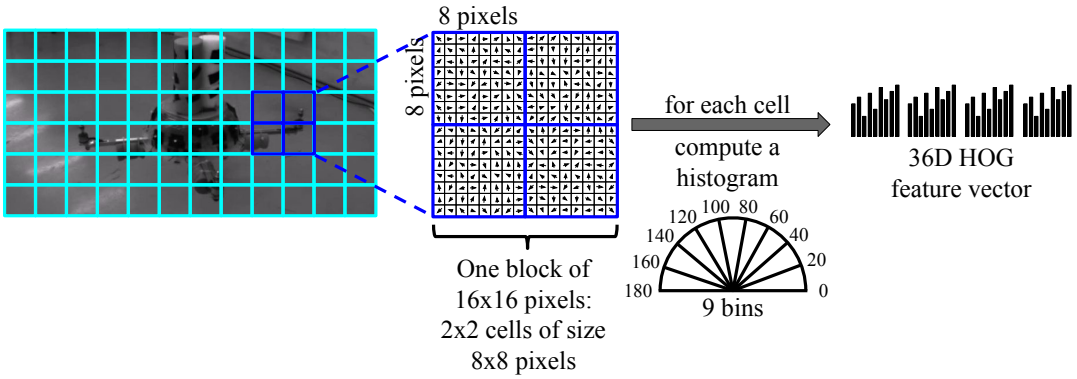



Figure 3.6: Calculation of a HOG feature vector on an image patch. (The direction of the gradients and corresponding histograms are imaginary and for illustrative purposes only.)

### 3.3 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) computes a histogram of gradient occurrences in local grid cells [23]. HOG of an image patch $P$ is defined as follows:

$$HOG(k) = \sum_{p \in P} \delta \left( \left\lceil \frac{\theta^p}{L} \right\rceil - k \right), \tag{3.3}$$

where $\delta(\cdot)$ is the Kronecker delta which evaluates to one if and only if its input is zero, $L$ is a normalizing constant and $\theta^p$ is the image gradient orientation at point $p$. $HOG(k)$ is the value of the $k$-th bin in a $K$-bin histogram. In the experiments, we set $K$ to 9 which makes the value of $L$ equal to $180/K = 20$ [23].

Figure 3.6 illustrates the calculation of a HOG feature vector on an image patch. In original HOG proposed by Dalal and Trigs [23], image patch is divided into cells of $8 \times 8$ pixels. For each cell, a histogram of nine bins according to the gradient angles at each pixel is computed. A HOG feature vector is then calculated for four cells (one block) as shown in Figure 3.6. There is a $50\%$ overlap between the blocks since the step-size between the blocks is eight pixels.

HOG has been demonstrated to be very successful in human detection and tracking.

Zhu et al. [109] extended HOG features so that the features are extracted at multiple sizes of blocks at different locations and aspect ratios. This extension enables the definition of an increased number of blocks on an image patch as compared to the

original HOG. We utilized this extension approach and allowed to define blocks with (1:1), (1:2) and (2:1) aspect ratios and four pixels step-size. However, we kept the smaller dimension of a cell in the blocks constant at eight pixels which resulted in blocks with $16 \times 16$ and $32 \times 16$ pixels for an image patch of $40 \times 22$ pixels. Considering the four pixels step-size, we obtained 20 different feature associations for this image patch size.

### 3.4 Integral Images and Integral HOG

Haar-like and MB-LBP features need the summations of intensities inside various regions in an image patch. If these summations are made for every feature separately, this will require a high amount of processing time. In order to speed up the processing, the computation of each feature in a window is performed utilizing the integral images technique. In this method, for a pixel $(i, j)$, the intensities of all pixels that have a smaller row and column number are accumulated at $(i, j)$:

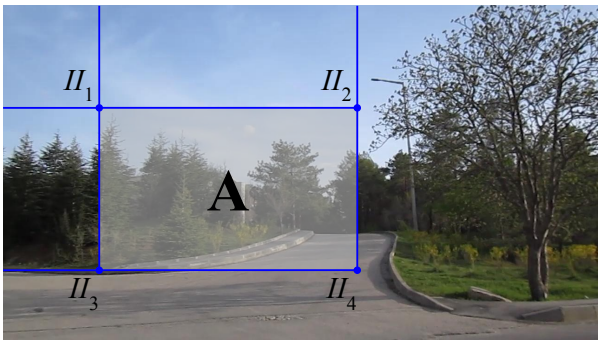
$$II(i,j) = \sum_{c=1}^{i} \sum_{r=1}^{j} I(c,r), \tag{3.4}$$

where $I$ is the original image and $II$ the integral image. Note that $II$ can be calculated incrementally from the $II$ of the neighboring pixels more efficiently.

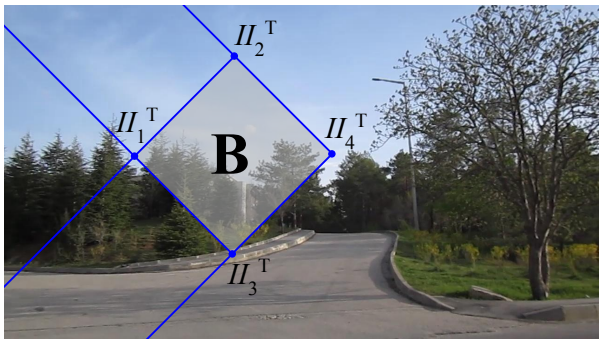
Given such an integral image, the sum of intensities in a rectangular window can be calculated easily by accessing only four values. See Figure 3.7(a) for an example: The sum of intensities in window A can be calculated as $II_4 + II_1 - (II_2 + II_3)$ [94]. With this way, the sum of intensities in different regions of the features are calculated efficiently. Note that the integral image is calculated only once an utilized multiple times to calculate the features.

Equation 3.4 and Figure 3.7(a) assumes a non-tilted window. However, we need also tilted integral image representation for some of the Haar-like features. Similar to non-tilted integral image approach, we can generate a tilted integral image $II^T$ as follows [58]:

$$II^T(i,j) = \sum_{c \leq i, c \leq i - |j-r|} I(c,r). \tag{3.5}$$




(a)




(b)

Figure 3.7: The method of integral images for the efficient computation of sums of intensities in image windows: (a) non-tilted and (b) tilted version. The sum of intensities in window A and B can be calculated as $II_4 + II_1 - (II_2 + II_3)$ and $II_4^T + II_1^T - (II_2^T + II_3^T)$, respectively. Figure in (a) is taken from [38].

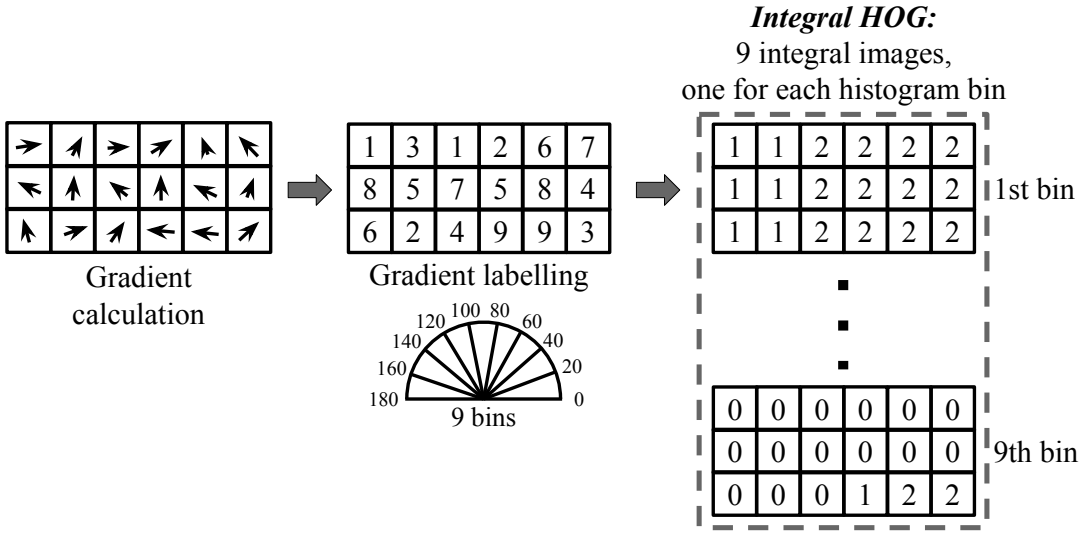



Figure 3.8: Utilization of integral images method for calculating integral HOG on an imaginary $6 \times 3$ pixels image window.

Once we calculate a tilted integral image, we can calculate the sum of intensities in a window as shown in Figure 3.7(b) by accessing only four values as follows: $II_4^T + II_1^T - (II_2^T + II_3^T)$.

Integral image technique can also be utilized to efficiently calculate histograms as required during the calculation of HOG features. This approach is illustrated in Figure 3.8 for a $6 \times 3$ pixels image window. Once the gradients are calculated for each pixel, they are labeled according to the nine bins. For each histogram bin, an integral image is calculated such that only corresponding labels of the current bin are considered during this computation. These nine integral images, which corresponds to integral HOG, can then be utilized to calculate the histogram inside a rectangular window similar to the calculation of intensity summation in Figure 3.7(a). However, the computation of a nine bin histogram will require accessing to $4 \times 9 = 36$ array references.

## 3.5 Feature selection via Adaptive Boosting (AdaBoost)

The features extracted by the feature description methods are utilized to construct weak classifiers. The combination of multiple weak classifiers produces strong classifiers. However, the set of features extracted by the feature description methods,

therefore, the set of weak classifiers are generally overcomplete. For this reason, most meaningful set of the weak classifiers are needed to be selected to create the strong classifiers. In the method proposed by Viola and Jones [94, 95], the AdaBoost learning (see Algorithm 1) is used to select and combine weak classifiers at each stage to capture an aspect of the problem to be learned. A weak classifier, $h_f(\mathbf{x})$, simply learns a linear classification for feature $f$ with a threshold $\theta_f$:

$$h_f(\mathbf{x}) = \begin{cases} 1 & \text{if } pf(\mathbf{x}) < p\theta_f \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

where *p* is the polarity indicating the inequality direction. The best performing weak classifiers are combined linearly to derive a stronger one (one stage of the cascade).

In our study, weak classifiers are implemented as decision trees and Gentle AdaBoost algorithm [35] is used which calculates the error of classification ($\epsilon_f$ in Algorithm 1) as the sum of weighted squared errors and updates the weights with $w_{t+1,i} = \hat{w}_{t,i}e^{-l_i h_f(\mathbf{x}_i)}$ equation where $l_i = 1$ for positive and $l_i = -1$ for negative samples.

### 3.6 Training of a Cascaded Classifier

Cascaded classifiers are composed of multiple stages with different processing complexities [58, 94, 95]. Instead of one highly complex single processing stage, cascaded classifiers incorporate multiple stages with increasing complexities, as presented in Figure 2.2.

Each of the stages in a cascaded classifier is a strong classifier. The early stages of the cascaded classifier have lower computational complexities and are applied to the image to prune most of the search space quickly (*early pruning*). The regions classified as mUAV by one stage of the classifier are passed to the higher stages. As the higher level of stages are applied, the classifier works on a smaller number of regions at each stage to identify them as mUAV or background. At the end of the last stage, the classifier returns the regions classified as mUAV.

In the approach of Viola and Jones [94, 95], the AdaBoost algorithm is used to learn the stages (strong classifiers) in the cascade of classifiers in an iterative manner, as

**Algorithm 1:** AdaBoost Learning (Adapted from [95]).

**input** : The training samples: $\{(\mathbf{x}_i, l_i)\}$, $i = 1, ..., N$, where $l_i = 1$ for positive and $l_i = 0$ for negative samples. $N = m + o$, where $m$ and $o$ are the number of positive and negative samples, respectively.

**output**: Strong classifier, $h(\mathbf{x})$, as a combination of $T$ weak classifiers.

- Initialize the weights for samples:

  $w_{1,i} = \frac{1}{2m}$ for positive samples and $w_{1,i} = \frac{1}{2o}$ for negative samples.

**for** $t = 1$ *to* $T$ **do**

- Normalize weights so that $w_t$ add up to one:

$$\hat{w}_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}. \tag{3.7}$$

**for** *each feature* $f \in \mathcal{F}$*, the set of all features* **do**

- Train a weak classifier $h_f$ for learning from only feature $f$.
- Calculate the error of classification:

$$\epsilon_f = \sum_{i=1}^{n} \hat{w}_{t,i} |h_f(\mathbf{x}_i) - l_i|. \tag{3.8}$$

- Among the weak classifiers, $h_f, \forall f \in \mathcal{F}$, choose the one with the lowest error ($\epsilon_t$):

$$h_t = \arg\min_{f \in \mathcal{F}} \epsilon_f. \tag{3.9}$$

- Update the weights:

$$w_{t+1,i} = \hat{w}_{t,i} \left( \frac{\epsilon_t}{1 - \epsilon_t} \right)^{e_i}, \tag{3.10}$$

where $e_i = 1$ if $\mathbf{x}_i$ is classified correctly and zero if it is not.

- The final classifier is then the combination of all of the weak ones found above:

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}, \tag{3.11}$$

where $\alpha_t = \log \frac{1-\epsilon_t}{\epsilon_t}$.

given in Algorithm 2. The method constructs the cascade by simply training a new strong classifier via AdaBoost algorithm and adding it as a new stage when the current cascade does not yield the desired false positive and detection rates. The method utilizes same positive samples ($\mathcal{P}$) at every iteration, however, a different set of negative samples ($\mathcal{N}$) are used to train each stage. First set of negative samples is randomly selected from negative images. Subsequent sets are generated by running the current (intermediate) cascaded detector on negative images and putting false negative windows, namely the windows falsely classified as positive, into subsequent set of negative samples. With this way, following stages are trained using a harder set of negative samples. This result in simpler stages in earlier ones and more complex stages as the number of stages increases.

In this study, three different cascaded classifiers are trained and used utilizing three different feature description methods described above, namely Haar-like features, multi-block local binary patterns and histogram of oriented gradients. These classifiers will be referred as C-HAAR, C-LBP and C-HOG in order in the rest of the thesis.

### 3.7 Detection with Cascaded Classifiers

A trained cascaded classifier is utilized as a cascaded detector by running at multiple scales and locations on images as illustrated in Figure3.9. Sliding window approach is employed for detecting an mUAV in an image such that a fixed size detection window is slid over the image at its original scale and also over images obtained by downscaling the original image.

The amount of downscale is defined by a scale factor such that, for each image scale, the size of the image is reduced with this factor until getting an image smaller than the size of detection window. Since the detection window size is fixed, features are used without rescaling.

Once a detection occurs at an image scale, the size of detection window at original scale is calculated by multiplying the detection window size with the current scale. Therefore, use of downscaled images with fixed size detection window effectively

**Algorithm 2:** Learning a Cascade of Classifiers (Adapted from [95]).

**input** : Positive and negative training samples: $\mathcal{P} = \{\mathbf{x}_1^+, \mathbf{x}_2^+, ..., \mathbf{x}_L^+\}$,
$\mathcal{N} = \{\mathbf{x}_1^-, \mathbf{x}_2^-, ..., \mathbf{x}_M^-\}$

**output**: The cascade of classifiers

**initialize:**

$i = 0$ : The stage number

$F_i = 1.0$ : False positive rate of the current cascaded classifier

$D_i = 1.0$ : Detection rate of the current cascaded classifier

$\mathcal{N}_i = \mathcal{N}$ : Negative samples for the current cascaded classifier

$f$ : User defined maximum acceptable false positive rate per layer

$d$ : User defined minimum acceptable detection rate per layer

**while** $F_i > F_{target}$ **do**

- $i \leftarrow i + 1$
- $n_i = 0$
- $F_i \leftarrow F_{i-1}$
- **while** $F_i > f \times F_{i-1}$ **do**
  - $n_i \leftarrow n_i + 1$
  - Train a classifier $h_{n_i}$ on $\mathcal{P}$ and $N_i$ with $n_i$ features using AdaBoost (see Algorithm 1)
  - Determine $F_i$ and $D_i$ using the current cascaded detector
  - Decrease threshold $\theta_i$ for $h_{n_i}$ until $D_i > d \times D_{i-1}$
- **if** $F_i > F_{target}$ **then**
  - Run the current cascaded detector with $\theta_i$ on negative images
  - Put any false negative windows into $\mathcal{N}_{i+1}$

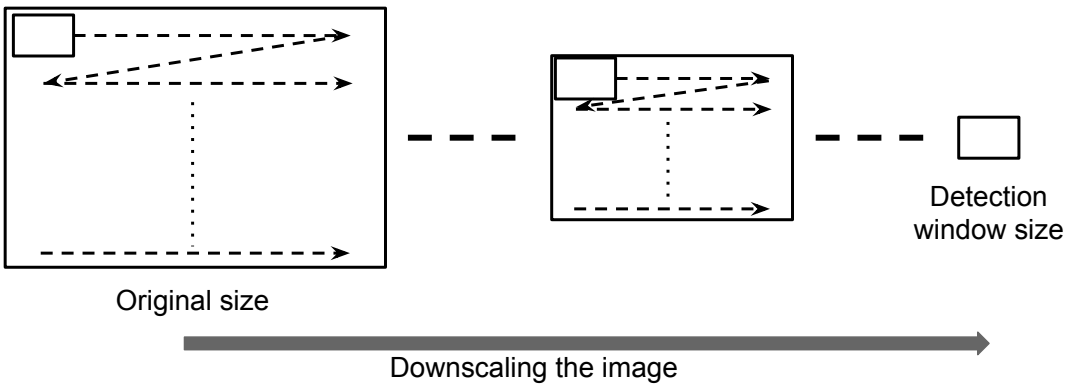



Figure 3.9: The cascaded detectors run in multiple scales and locations on an image. Image is downscaled until the size of detection window. Fixed size detection window is slid over the images.

corresponds to increasing the size of detection window at each scale. The location of the detection window at original scale is also calculated by upscaling.

This sliding window approach running in multiple scales and locations leads to multiple detections in the image for the same object. These detections are merged by looking at the amount of overlap between them, as a post-processing stage.

We should note that due to early pruning property of the cascaded classifiers, most of the detection windows are pruned with simple and fast checks in the earlier stages. Therefore, the number of windows processed through all of the stages are very low. For this reason, sliding window approach does not cause a computational cost problem as it is the case for keypoint-based approaches (Section 2.1.1).

## 3.8 Distance Estimation

Once an mUAV is detected with its bounding box, there are two important information to deduce the distance of the mUAV to the camera. These are width and height of the detection box. Use of camera model with some geometric calculations can be considered at first hand as the approaches presented in [59, 108]. However, since the mUAV tilts and rotates during its motion, its bounding box could be quite different in size for the same distance and such approaches do not provide an accurate estimation. Therefore, a different approach is needed.

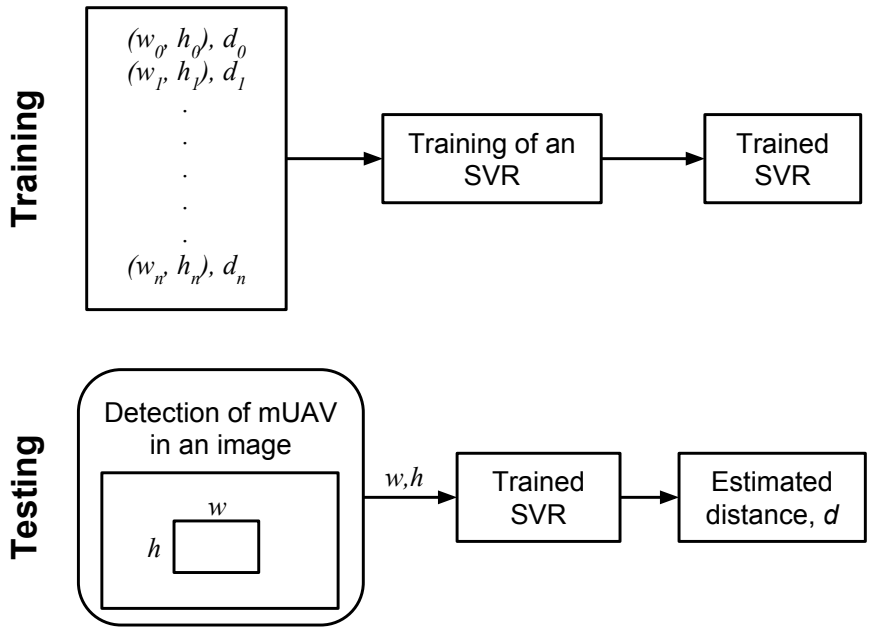



Figure 3.10: Training and testing stages of our distance estimation method.

In this study, we propose to incorporate a Support Vector Regressor (SVR - [84]) for the estimation of mUAV distance, as depicted in Figure 3.10. We collect a training set of $\{(w_i, h_i), d_i\}$, where $w_i, h_i$ are the width and the height of the mUAV bounding box, respectively, and $d_i$ is the known distance of the mUAV. Having such a training set, we train a Support Vector Regressor (SVR - [84]) to estimate the non-linear relation between width and height of the mUAV and its distance. Using the trained SVR, we can estimate the distance of the mUAV once its bounding box is estimated.

# CHAPTER 4

# EXPERIMENTAL SETUP AND DATA COLLECTION

In this chapter, we will introduce our experimental setups for indoor and outdoor environments, the extraction procedures of ground truth data, training and testing datasets and the method we utilized to add artificial motion blur to indoor test videos. This chapter is partially published in [38].

We used the setups shown in Figure 4.1 for collecting systematic data. These setups consist of the following components:

- **mUAV**: We used a quadrotor platform shown in Figure 4.2(a). Open-source Arducopter [1] hardware and software are used as the flight controller. The distance between the motors on the same axis is $60$ cm. The plastic cover of the quadrotor has twelve markers attached to define a rigid body. Figure 4.2(a) illustrates the body coordinate frame of the quadrotor. The forward and right directions of the quadrotor correspond to the $x_Q$-axis and $y_Q$-axis, respectively. The $z_Q$-axis points downwards with respect to the quadrotor.

- **Camera**: We use two different electro-optic cameras for indoors and outdoors due to varying needs in both environments. For indoors, the synchronization property of the camera is vital, since we have to ensure that the 3D position data obtained from the motion capture system and the captured frames are synchronized in time. Complying with this requirement, we use a camera from Basler Scout$^{\text{TM}}$ (capturing $1032 \times 778$ resolution videos at $30$ fps in gray scale) mounted on top of the motion capture system. It weighs about $220$ g, including its lens, whose maximum horizontal and vertical angle of views are $93.6^\circ$ and

$68.9°$, respectively. The power consumption of the camera is about $3$ W, and it outputs the data through a Gigabit Ethernet port. The body coordinate frame of the camera is centered at the projection center. The $x_C$-axis is towards the right side of the camera; the $y_C$-axis points down from the camera; and the $z_C$-axis coincides with the optical axis of the camera lens, as depicted in Figure 4.2(b).

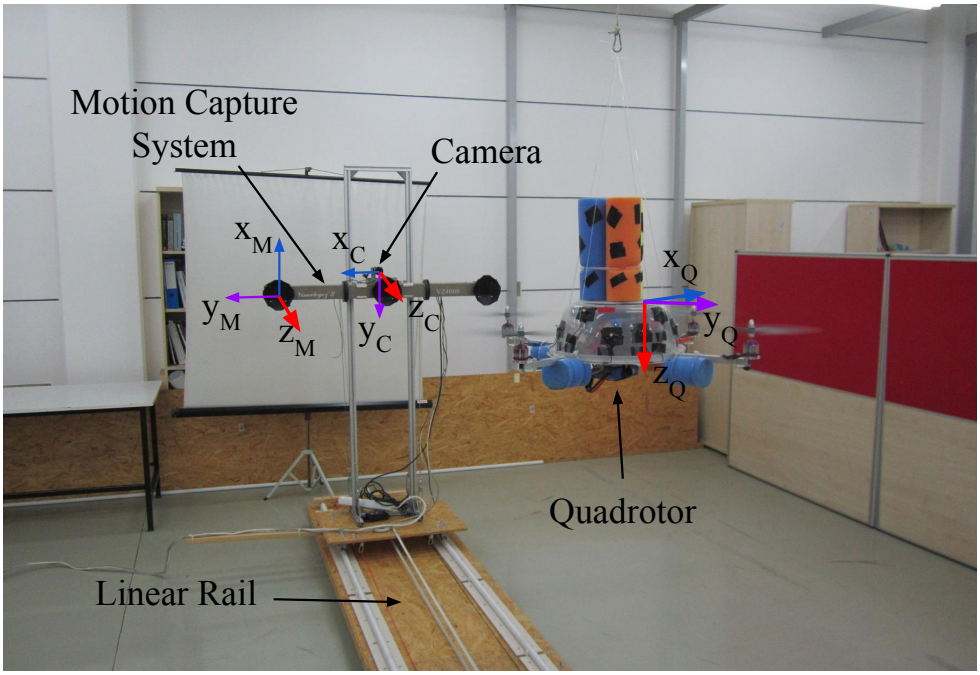
Due to difficulties in powering and recording of the indoor camera outdoors, we used a Canon® PowerShot A2200 HD to capture outdoor videos. This camera is able to record videos at $1280 \times 720$ resolution at $30$ fps in color.

However, we use gray scale versions of the videos in our study.

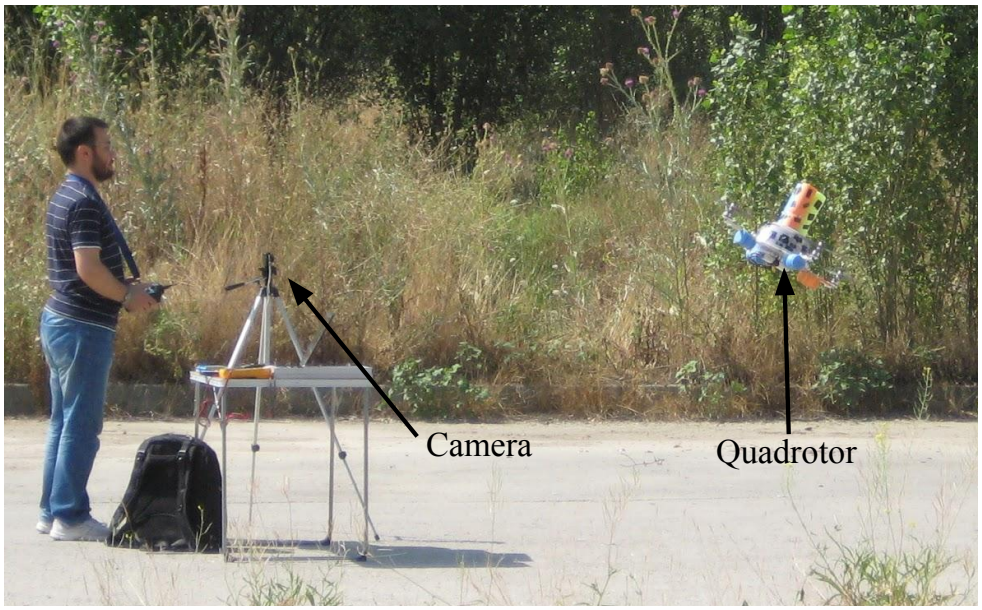
Although we needed to utilize a different camera outdoors due to logistic issues, we should note that our indoor camera is suitable to be placed on mUAVs in terms of SWaP constraints. Moreover, alternative cameras with similar image qualities compared to our cameras are also available in the market, even with less SWaP requirements.

- **Motion capture system (used for indoor analysis)**: We use the Visualeyez™ II VZ4000 3D real-time motion capture system (MOCAP) (PhoeniX Technologies Incorporated) that can sense the 3D positions of active markers up to a rate of $4348$ real-time 3D data points per second with an accuracy of $0.5 \sim 0.7$ mm RMS in $\sim 190$ cubic meters of space. In our setup, the MOCAP provides the ground truth 3D positions of the markers mounted on the quadrotor. The system provides the 3D data as labeled with the unique numbers of the markers. It has an operating angle of $90°, (\pm 45°)$ in both pitch and yaw, and its maximum sensing distance is $7$ m at minimum exposure. The body coordinate frame of the MOCAP is illustrated in Figure 4.2(c).

- **Linear rail platform (used for indoor analysis)**: We constructed a linear motorized rail platform to move the camera and the MOCAP together in a controlled manner to capture videos of the quadrotor only with single motion types, i.e., lateral, up-down, rotational and approach-leave motions. With this platform, we are able to move the camera and MOCAP assembly on a horizontal line of approximately $5$ m up to a $1$ m/s speed.




(a)




(b)

Figure 4.1: (a) The setup used in indoor experiments. The rail was constructed in order to be able to move the camera with respect to the quadrotor in a controlled manner. This allows analyzing the performance of the methods under different motion types. Figure is adapted from [38]. (b) Outdoor experimental setup. The quadrotor is flown manually with a remote control and a fixed camera is used for recording the videos.

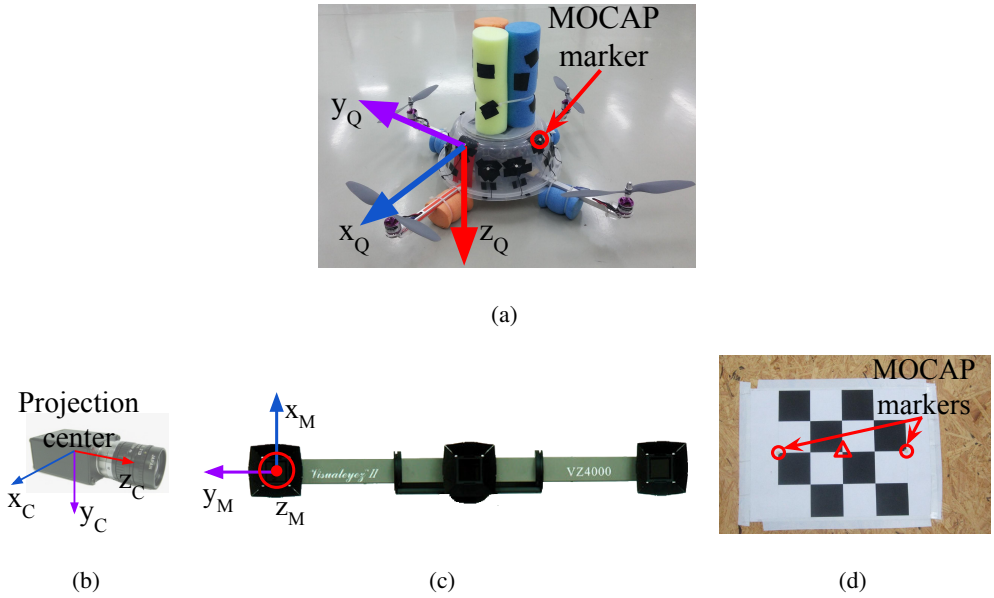



Figure 4.2: (a) The quadrotor used in our study and its body coordinate frame. There are 12 markers mounted roughly $30^\circ$ apart from each other on the plastic cup of the quadrotor. (b) The body coordinate frame of the camera is defined at the projection center. (c) The Visualeyez™ II VZ4000 motion capture system and its body coordinate frame. (d) The calibration tool used to obtain 3D-2D correspondence points needed to estimate the transformation matrix, $T_M^C$, between the motion capture system (MOCAP) and the camera coordinate systems. Circles and the triangle indicate the MOCAP markers and the center of the chess pattern, respectively. Figures are taken from [38].

## 4.1 Ground Truth Extraction

In the indoor experimental setup, the MOCAP captures the motion of active markers mounted on the quadrotor and supplies the ground truth 3D positions of those markers. We synchronized the MOCAP and the camera so that 3D position data and the captured frames belongs to the same time instance. For our purposes, we need the ground truth bounding box of the quadrotor and the distance between the quadrotor and the camera for each frame.

In order to determine a rectangular ground truth bounding box encapsulating the quadrotor in an image, we need to find a set of 2D pixel points ($P'_{Qi}$)[1] on the boundaries of the quadrotor in the image. These 2D points correspond to a set of 3D points ($P_{Qi}$) on the quadrotor. In order to find $P'_{Qi}$, $P_{Qi}$ should first be transformed from the body coordinate frame of the quadrotor to the MOCAP coordinate frame, followed by a transformation to the camera coordinate frame. These two transformations are represented by the transformation matrices $T_Q^M$ and $T_M^C$, respectively, and are applied as follows:

$$P_{Mi} = T_Q^M P_{Qi} \text{ for all } i, \tag{4.1}$$

$$P_{Ci} = T_M^C P_{Mi} \text{ for all } i, \tag{4.2}$$

where $P_{Mi}$ and $P_{Ci}$ are the transformed coordinates in the MOCAP and the camera coordinate frames, respectively. After these transformations, we project the points in $P_{Ci}$ to the image plane as:

$$P'_{Qi} = P_c P_{Ci} \text{ for all } i, \tag{4.3}$$

where $P_c$ is the camera matrix and get $P'_{Qi}$. Then, we can find the bounding box of the quadrotor by calculating the rectangle with the minimum size covering all of the points in $P'_{Qi}$ as follows:

$$x_r = \min(x_i), \tag{4.4}$$

$$y_r = \min(y_i), \tag{4.5}$$

$$w_r = \max(x_i) - \min(x_i), \tag{4.6}$$

$$h_r = \max(y_i) - \min(y_i), \tag{4.7}$$

[1] In our derivations, all points in 2D and 3D sets are represented by homogeneous coordinate vectors.

where $(x_i, y_i) \in P'_{Qi}$, $(x_r, y_r)$ is the upper left pixel position of the rectangle and $w_r$ and $h_r$ are the width and height of the rectangle, respectively.

It is not possible to place a marker on the quadrotor for every point in $P_{Qi}$. Therefore, we define a rigid body, a set of 3D points whose relative positions are fixed and remain unchanged under motion, for $12$ markers on the quadrotor. The points in $P_{Qi}$ are then defined virtually as additional points of the rigid body.

A rigid body can be defined from the positions of all markers obtained at a particular time instant while the quadrotor is stationary. However, we wanted to obtain a more accurate rigid body and used the method presented in [36, 37] with multiple captures of the marker positions. Taking $60$ different samples, we performed the following optimization to minimize the spatial distances between the measured points $M_i$ and the points $R_i$ in the rigid body model.

$$\arg\min_{R_i} \sum_i \|M_i - R_i\|^2, \tag{4.8}$$

where $\|.\|$ denotes the calculation of the Euclidean norm for the given vector.

Once the rigid body is defined for the markers on the quadrotor, if at least four markers are sensed by the MOCAP, $T_Q^M$ can be estimated. Since the MOCAP supplies the 3D position data as labeled and the rigid body is already defined using these labels, there is no correspondence matching problem. Finding such a rigid transformation between two labeled 3D point sets requires the least squares fitting of these two sets and is known as the "*Absolute Orientation Problem*" [43]. We use the method presented in [36, 91] to solve this problem and calculate $T_Q^M$. Note that $T_Q^M$ transformation matrix should be calculated whenever the quadrotor and the camera moves with respect to each other.

There is no direct way of calculating $T_M^C$, since it is not trivial to measure the distances and the angles between the body coordinate frames of the MOCAP and the camera. However, if we know a set of 3D points ($P_{Ti}$) in the MOCAP coordinate frame and a set of 2D points ($P'_{Ti}$) which corresponds to the projected pixel coordinates of the points in $P_{Ti}$, then we can estimate $T_M^C$ as the transformation matrix that minimizes the re-projection error. The re-projection error is given by the sum of squared distances between the pixel points in $P'_{Ti}$ as in the following optimization

criterion:

$$\arg\min_{T_M^C} \sum_i \|P_{Ti}^{'} - T_M^C P_{Ti}\|^2. \tag{4.9}$$

We prepared a simple calibration tool shown in Figure 4.2(d) for collecting the data points in $P_{Ti}$ and $P_{Ti}^{'}$. In this tool, there is a chess pattern and two MOCAP markers mounted on the two edges of the chess pattern. The 3D position of the chess pattern center, shown inside the triangle in Figure 4.2(d), is calculated by finding the geometric center of the marker positions. We obtain the 2D pixel position of the chess pattern center using the camera calibration tools of the Open Source Computer Vision Library (OpenCV) [12]. We collect the data needed for $P_{Ti}$ and $P_{Ti}^{'}$ by moving the tool in front of the camera. Note that, since the MOCAP and the camera are attached to each other rigidly, once $T_M^C$ is estimated, it is valid as long as the MOCAP and the camera assembly remain fixed.

In order to calculate the ground truth distance between the quadrotor and the camera, we use $T_Q^M$ and $T_M^C$ as follows:

$$p_c^{'} = T_M^C T_Q^M p_c, \tag{4.10}$$

where $p_c$ is the 3D position of the quadrotor center in the quadrotor coordinate frame and $p_c^{'}$ is the transformed coordinates of the quadrotor center to the camera coordinate frame. $p_c$ is defined as the geometric center of four points where the motor shafts and the corresponding propellers intersect. Once $p_c^{'}$ is calculated, the distance of the quadrotor to the camera ($d_Q$) is calculated as:

$$d_Q = \|p_c^{'}\|. \tag{4.11}$$

## 4.2 Data Collection for Training

**Indoors:** In order to prepare the quadrotor training image set, we recorded videos of the quadrotor by moving the MOCAP and the camera assembly around the quadrotor manually while the quadrotor is hanged at different heights from the ground and stationary with its motors running. From these videos, we automatically extracted $8876$ image patches, including only the quadrotor using the bounding box extraction method described in Section 4.1 without considering the aspect ratios of the patches.

The windows sizes and aspect ratios of these extracted images are different from each other. The distribution of the aspect ratios for these images is given in Figure 4.3 with a median value of $1.8168$. Since the training of cascaded classifiers requires image windows with a fixed window size, we first equalized the aspect ratios of the extracted images. For this purpose, we enlarged the bounding boxes of these $8876$ images by increasing their width or height only, according to the aspect ratio of the originally extracted image window to ensure they all have a fixed aspect ratio of approximately $1.8168$[2]. We preferred enlargement to fix the aspect ratios, since this approach keeps all relevant data of the quadrotor inside the bounding box. The images were then resized to $40 \times 22$ pixels to be used in training.

For background training image set, we captured videos of the indoor laboratory environment without the quadrotor in the scene. From these videos, we extracted $5731$ frames at a resolution of $1032 \times 778$ pixels as our background training image set. See Figures 4.4(a) and 4.4(b) for sample quadrotor and background images captured indoors.

**Outdoors:** We used a fixed camera to record the quadrotor while it is flying in front of the camera using remote control. Since the MOCAP is not operable outdoors, the ground truth is collected in a labor-extensive manner: By utilizing the background subtraction method presented in [48], we are able to approximate the bounding box of the quadrotor in these videos as long as there are not any moving objects other than the quadrotor. Nevertheless, it is not always possible to get a motionless background. Therefore, the bounding boxes from background subtraction are inspected manually, and only the ones that bound the quadrotor well are selected. Both the number and aspect ratio of the outdoor training images are the same as the indoor images. Similar to indoor quadrotor training images, these images are also resized to $40 \times 22$ pixels.

For outdoor background training images, we have recorded videos at various places on the university campus. These videos include trees, bushes, grass, sky, roads, buildings, cars and pedestrians without the quadrotor. From these videos, we have extracted frames as the same number of indoor background training images at $1280 \times 720$ resolution. See Figures 4.5(a) and 4.5(b) for sample images collected

[2] Due to floating point rounding, aspect ratios may not be exactly $1.8168$.

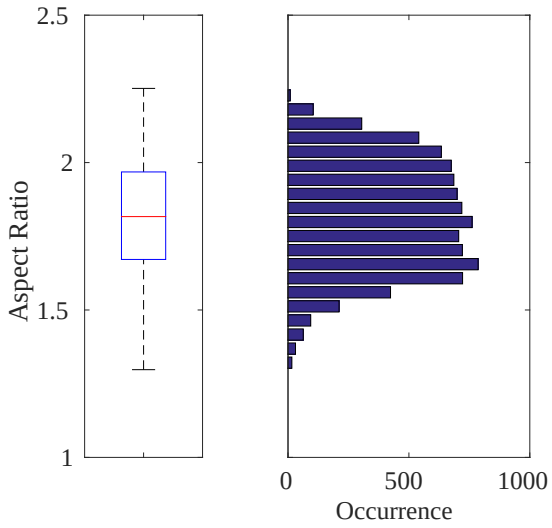
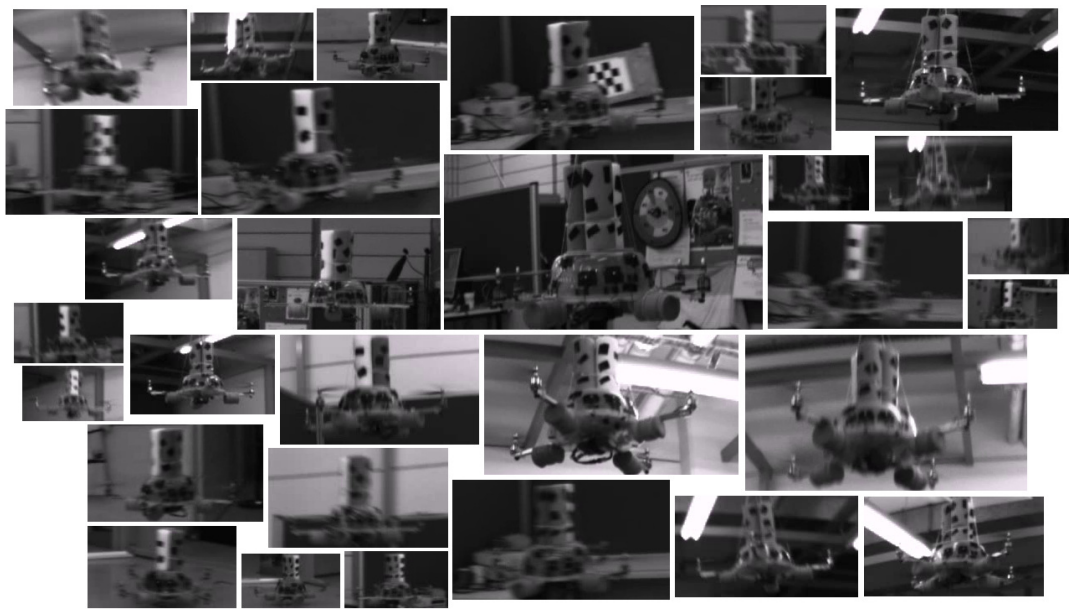



Figure 4.3: Box-plot (Left) and histogram (Right) representation for the aspect ratios of $8876$ quadrotor images automatically extracted from the training videos. In this figure and the subsequent box-plot figures, the top and bottom edges of the box and the line inside the box represent the first and third quartiles and the median value, respectively. The bottom and top whiskers correspond to the smallest and largest non-outlier data, respectively. The data inside the box lie within the $50\%$ confidence interval, while the confidence interval of the data in between the whiskers is $99.3\%$. Here, the median value is $1.8168$, which defines the aspect ratio of the training images used. Figure is taken from [38].

outdoors.

Looking at the training image sets, the following observations can be deduced, which also represent the challenges in our problem: (i) Changes in camera pose or quadrotor pose result in very large differences of in the quadrotor's visual appearance. (ii) The bounding box encapsulating the quadrotor contains a large amount of background patterns due to the structure of the quadrotor. (iii) Vibrations in the camera pose and the agile motions of the quadrotor cause motion blur in the images. (iv) Changes in brightness and the illumination direction yield very different images. (v) Motion in the image can also be induced by the motion of the camera or the motion of background objects (e.g., trees swinging due to wind, etc.).



(a)

(b)

Figure 4.4: Example images from indoor (a) quadrotor and (b) background training image sets. Mostly the challenging examples are provided in the quadrotor images. Figures are taken from [38].

(a)

(b)

Figure 4.5: Example images from outdoor (a) quadrotor and (b) background training image sets. The images are colored; however, their grayscale versions are used in the training. For quadrotor images, mostly the challenging examples are included. Figures are taken from [38].

### 4.3 Data Collection for Testing

Indoor and outdoor environments are significantly different from each other, since controlled experiments can only be performed indoors by means of motion capture systems. On the other hand, outdoor environments provide more space, increasing the maneuverability of the quadrotor and causing many challenges that need to be evaluated. These differences directed us to prepare test videos of different characteristics indoors and outdoors.

In order to investigate the performance of the methods (C-HAAR, C-LBP and C-HOG) systematically, we defined $4$ different motion types, namely lateral, up-down, yaw and approach-leave, for the indoor test videos. Please note that maneuvers in a free flight are combinations of these motions, and use of these primitive motions is for systematic evaluation purposes. The recording procedure of each motion type is depicted in Figure 4.6 for two different views, the top view and the camera view. Each motion type has different characteristics in terms of the amount of changes in the scale and appearance of the quadrotor, as well as the background objects as shown in Table 4.1. The details of each motion type are as follows:

Table 4.1: Properties of motion types in terms of the amount of changes in the *scale* and *appearance* of the quadrotor, and the *background* objects. Table is taken from [38].

| | **Lateral** | **Up-Down** | **Yaw** | **Approach-Leave** |
|---|---|---|---|---|
| Scale | Moderate | Moderate | Small | Large |
| Appearance | Moderate | Large | Large | Large |
| Background | Large | No Change | No Change | Moderate |

- **Lateral:** The camera performs left-to-right or right-to-left maneuvers while the quadrotor is fixed at different positions, as illustrated in Figure 4.6. As seen in the top view, the perpendicular distance of the quadrotor to the camera motion course is changed by $1$ m for each of $5$ distances. For each distance, the height of the quadrotor is adjusted to $3$ different (top, middle and bottom) levels with $1$ m apart, making a total of $15$ different position for lateral videos. Left-to-right and right-to-left videos collected in this manner allow us to test the features' resilience against large background changes.

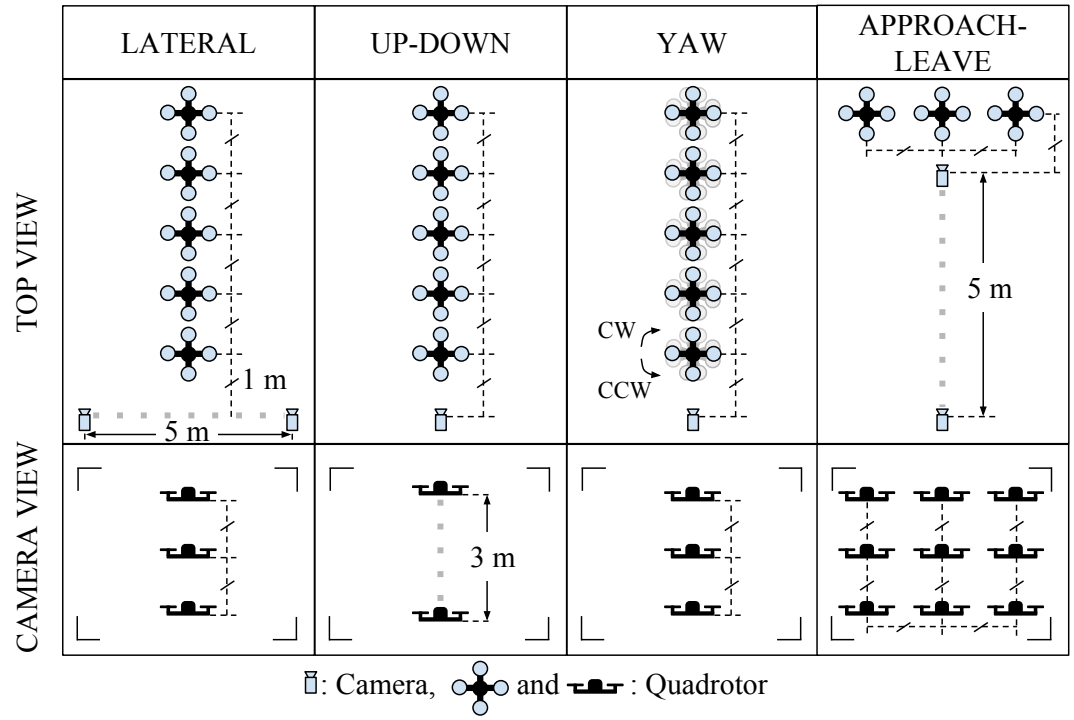



Figure 4.6: Graphical representation for indoor test videos. There are 4 motion types, namely lateral, up-down, yaw and approach-leave. Each of them is illustrated with the top and camera views. Dashed gray thick lines represent the motion of the camera or the quadrotor along the path with the given length. Dashed black thin lines are used to represent dimensions. Figure is taken from [38].

In each video, the camera is moved along an approximately 5 m path. However, when the perpendicular distance is 1 m and 2 m and, the quadrotor is not fully visible in the videos for the top and bottom levels. Therefore, these videos are excluded from the dataset, resulting in 22 videos with a total of 2543 frames.

- **Up-Down:** The quadrotor performs a vertical motion from the floor to the ceiling for the up motion and vice versa for the down motion. The motion of the quadrotor is performed manually with the help of a hanging rope. The change in the height of the quadrotor is approximately 3 m in each video. During the motion of the quadrotor, the camera remains fixed. For each of the 5 different positions shown in Figure 4.6, one up and one down video are recorded, resulting in 10 videos with a total of 1710 frames. These videos are used for testing the features' resilience against large appearance changes.

- **Yaw:** The quadrotor turns around itself in clockwise or counter clockwise directions, while both the camera and the quadrotor are stationary. The quadrotor is positioned at the same $15$ different points used in the lateral videos. Since the quadrotor is not fully present in the videos recorded for the top and bottom levels when the perpendicular distance is $1$ m and $2$ m, these videos are omitted from the dataset. Hence, there are $22$ videos with a total of $8107$ frames in this group. These videos are used for testing the features' resilience against viewpoint changes causing large appearance changes.

- **Approach-Leave:** In these videos, the camera approaches the quadrotor or leaves from it while the quadrotor is stationary. There are $9$ different positions for the quadrotor with a $1$ m distance separation, as illustrated in Figure 4.6. The motion path of the camera is approximately $5$ m. Approach and leave videos are recorded separately and we have $18$ videos with a total of $3574$ frames for this group. These videos are used for testing whether the features are affected by large scale and appearance changes.

We should note that the yaw orientation of the quadrotor is set to random values for each of $50$ videos in the lateral, up-down and approach-leave sets, although the quadrotors in Figure 4.6 are given for a fixed orientation. There are cases where the MOCAP can give the wrong or insufficient data to extract the ground truth for some frames. These frames are not included in the dataset.

For outdoor experiments, we prepared four different videos with distinct characteristics. In all videos, the quadrotor is flown manually in front of a stationary camera. In the first two videos, a stationary background is chosen. These two videos differ in terms of agility, such that in the first video, the quadrotor performs *calm* maneuvers, whereas in the second one, it is flown in an *agile* manner. In the third video, the background includes moving objects, like cars, motorcycles, bicycles and pedestrians, while the quadrotor is flown in a calm manner. The fourth video is recorded to test the maximum detection distances of the methods. In this video, the quadrotor first leaves from the camera and then comes back, flying on an approximately straight $110$ m path. We will call these videos (i) calm, (ii) agile, (iii) moving background and (iv) distance in the rest of the thesis. These videos have $2954$, $3823$, $3900$ and $2468$

frames respectively. The ground truth bounding boxes for each frame of calm, agile and moving background videos are extracted manually. For the distance video, only the ground truth distance of the quadrotor to the camera is calculated.

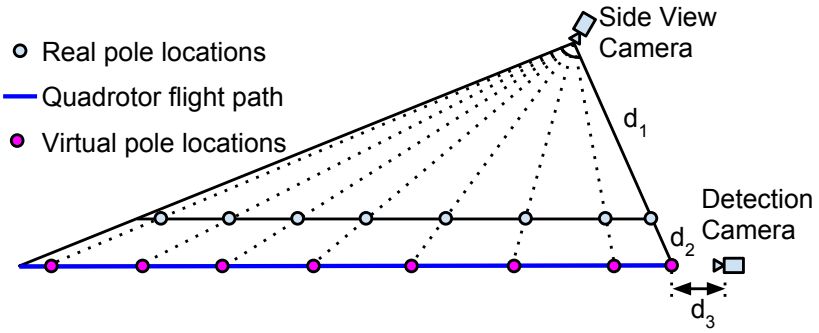
Finding the ground truth distance is a problem outdoors. GPS is not a good solution due to its inherent $3-5$ m accuracy. For this reason, we recorded a simultaneous video of the flight using a side view camera as seen in Figure 4.7(a) at $1080$p resolution. We computed the ground truth distance with some geometrical calculations by utilizing the poles at known locations in the experiment area. We extracted pixel position of the quadrotor center manually for each frame, and also the pixel positions of the poles only once since the camera is static. We measured $d_1$, $d_2$ and $d_3$ shown in Figure 4.7(a) and the distances between real poles. We calculated the distances between virtual poles on the flight path of the quadrotor by using the similarities of the triangles. Here, we assumed that the quadrotor flew on a straight path. Please also note that the pixel positions of real poles and virtual poles are same on the images captured by side view camera. We interpolated a function between the $x$-coordinate of the pixel positions of the virtual poles and the distances between the virtual poles, as illustrated in Figure 4.7(b). Then, we calculated the ground truth distances of the quadrotor to the detection camera by evaluating this function with the pixel locations of the quadrotor, extracted manually as stated earlier, and adding $d_3$ offset.

We should note that the scenes used in testing videos are different from the ones included in the training datasets for both indoors and outdoors.

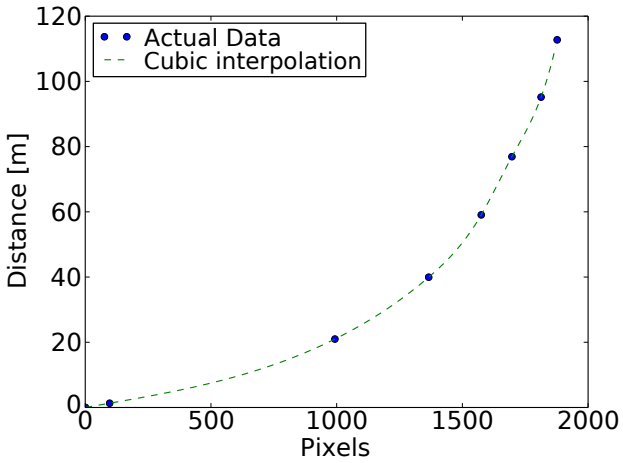
Our dataset is available at the following link: `http://kovan.ceng.metu.edu.tr/~fatih/sensors/`.

## 4.4 Generation of Blurred Videos

For the evaluation of the methods against motion blur in the images, we created test videos by adding artificial motion blur to indoor test videos. Since we do not expect a difference between the effects of motion blur in indoor and outdoors, for the sake of simplicity, blurry videos are generated for indoor dataset only.

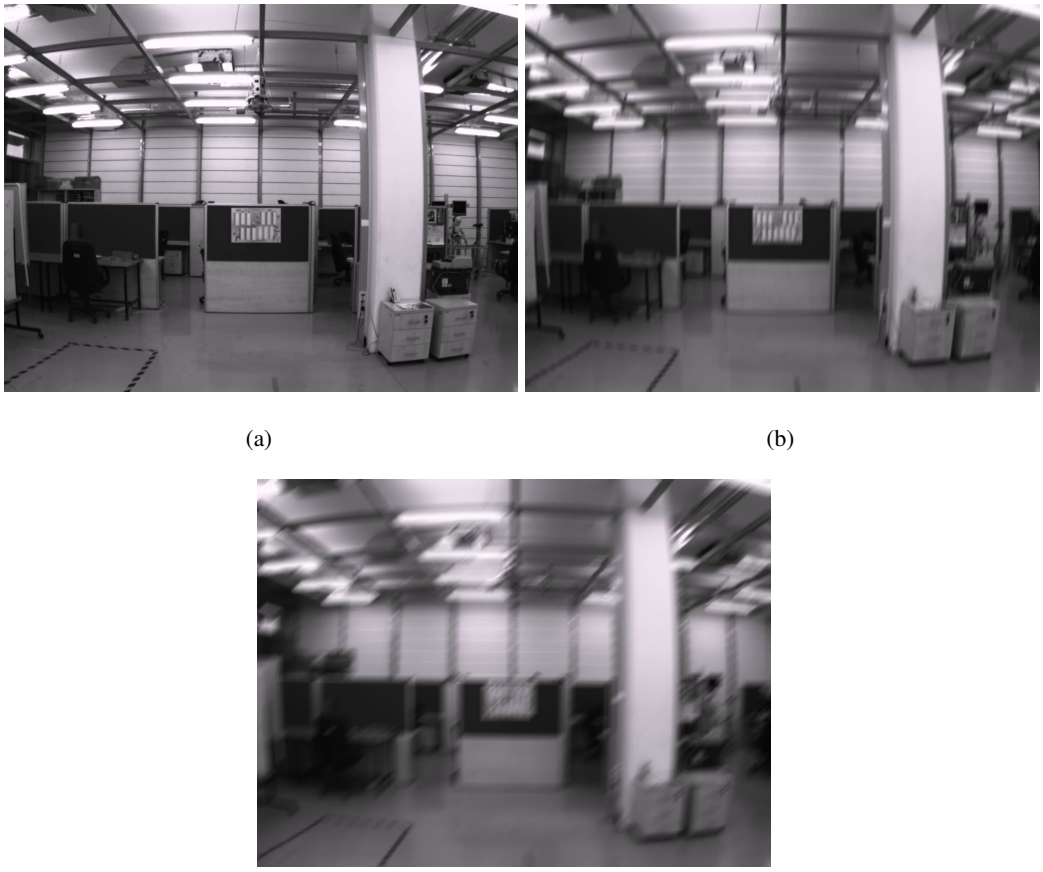



(a)




(b)

Figure 4.7: (a) Top view graphical illustration for the placements of the cameras, flight path of the quadrotor and pole locations during the recording of distance video. (b) Interpolation of a distance function for the distances between virtual poles with respect to the $x$-coordinate of the positions of the virtual poles in the video recorded by the side view camera.



(a) (b)

(c)

Figure 4.8: Example images for blurry images. Same image is applied with three different amounts of motion blur: (a) $\sigma = 0$, (b) $\sigma = 10$ and (c) $\sigma = 25$. The quadrotor is present around the center of the upper half of the images.

We utilized a linear motion blur similar to the one used in [77, 87]. A motion-blurred version of an image $I$ is generated by convolving it with a filter $k$ (i.e., $\tilde{I} = I * k$) which is defined as:

$$k(x, y) = \begin{cases} 1 & \text{if } y = d/2, \\ 0 & \text{otherwise,} \end{cases} \tag{4.12}$$

where $d$ is the dimension of the kernel (*blur length*), determining the amount of motion blur, sampled from a Gaussian distribution $N(\mu = 0, \sigma)$, with $\mu$ and $\sigma$ being the mean and the standard deviation, respectively. We applied this kernel to the video images after a rotation of $\theta$ radian (*blur angle*) chosen from a uniform distribution $U(0, \pi)$. For each frame of a video, a new kernel is generated in this manner, and it is applied to all pixels in that frame. Using this motion blur model, we generated blurred versions of all indoor test videos for five different values of $\sigma$, namely, $5$, $10$, $15$, $20$ and $25$.

Figure 4.8 presents three sample images to show the effect of $\sigma$ on the amount of motion blur. We should note that in a blurry video prepared for a certain value of $\sigma$, different frames may include different amount of motion blur, since dimension of the kernel, $d$ is sampled from a Gaussian distribution. Larger $\sigma$ will increase the chance of sampling larger $d$ values.

# CHAPTER 5

# RESULTS

We implemented the methods introduced in Section 3 using cascaded classifier and detector implementations based on Haar-like features, MB-LBP and HOG, and SVR implementation available in OpenCV [12], and evaluated them on the indoor and outdoor datasets. We trained indoor and outdoor cascade classifiers separately using the corresponding training datasets with the following parameters: For an image window of $40 \times 22$ pixels size, which is also the size of quadrotor training images, C-HAAR extracts $587408$ features, whereas C-LBP and C-HOG yield $20020$ and $20$ features, respectively. $7900$ positive (quadrotor) and $10000$ negative (background) samples were used for indoors and outdoors. We trained the classifiers with $11$, $13$, $15$, $17$ and $19$ stages (the upper limit of $19$ is due to the enormous time required to train C-HAAR classifiers, as will be presented in Section 5.6.1). During our tests, the classifiers performed multi-scale detections for a minimum object size of $80 \times 44$ and enlarging the detection window size by multiplying it with $1.1$ at each scale.

This chapter is published in [38].

## 5.1 Performance Metrics

We use precision-recall (PR) curves to evaluate the detection performance of the classifiers. Precision is defined as:

$$Precision = \frac{tp}{tp + fp}, \tag{5.1}$$

where *tp* is the number of true positives (see below) and *fp* is the number of false positives. The performance increases as the precision approaches to $1$. Recall is defined as:

$$Recall = \frac{tp}{tp + fn}, \tag{5.2}$$

where *fn* is the number of false negatives. Similar to the precision, a closer recall value to $1$ is the indication of a better performance.

A detected bounding box ($B_D$) is regarded as a true positive if its Jaccard index (J) [46], calculated as follows, is greater than $60\%$:

$$J(B_D, B_G) = \frac{|B_D \cap B_G|}{|B_D \cup B_G|}, \tag{5.3}$$

where $B_G$ is the ground truth bounding box. Otherwise, $B_D$ is regarded as a false positive. If there are multiple detections in a frame, each $B_D$ is evaluated separately as a *tp* or *fp*. If no $B_D$ is found for an image frame by the classifier, then *fn* is incremented by one.

The PR curves are drawn by changing the threshold of the classifiers' last stages from $-100$ to $+100$, as performed by [94, 95]. For each threshold, a precision and recall pair is calculated and represented as a point in a PR curve. The precision and recall pairs for all thresholds constitute the PR curve. Note that each stage of the cascaded classifiers has its own threshold determined during the training, and that decreasing the threshold of a stage $S$ to a low value such as $-100$ results in a classifier with $S-1$ many stages at the default threshold.

A widely-used measure with PR curves is the normalized area under the curve. If a PR curve, $p(x)$, is defined at the interval $[r_{min}, r_{max}]$, where $r_{min}$ and $r_{max}$ are the minimum and maximum recall values, respectively, the normalized area $A_p$ under curve $p(x)$ is defined as:

$$A_p = \frac{1}{r_{max} - r_{min}} \int_{r_{min}}^{r_{max}} p(x)\, dx. \tag{5.4}$$

We use also F-Score in our evaluations calculated as follows:

$$F\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{5.5}$$

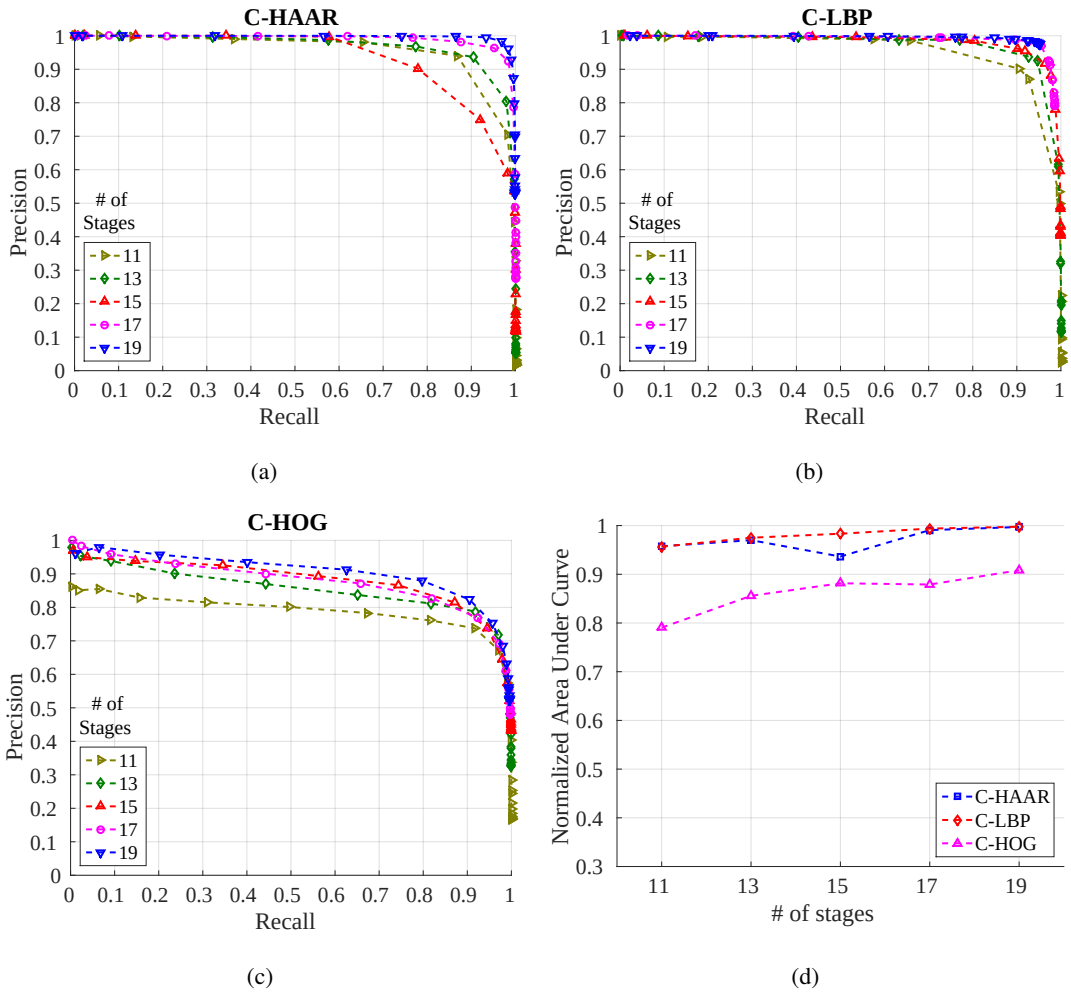



Figure 5.1: Precision-recall (PR) curves showing the performance of (a) C-HAAR, (b) C-LBP and (c) C-HOG for different numbers of stages on all indoor test videos. (d) Normalized areas under the PR curves in (a), (b) and (c). Figures are taken from [38].

## 5.2 Indoor Evaluation

We tested the classifiers trained with the indoor training dataset on indoor test videos having 15934 frames in total with four different motion types, namely lateral, up-down, yaw and approach-leave, as presented in Section 4.3. We evaluated the classifiers for five different numbers of stages to understand how they perform while their complexity increases. Figure 5.1 shows the PR curves, as well as the normalized area under the PR curves for each method and for different numbers of stages. In Table 5.1, the maximum F-Score values and the values at default thresholds are listed.

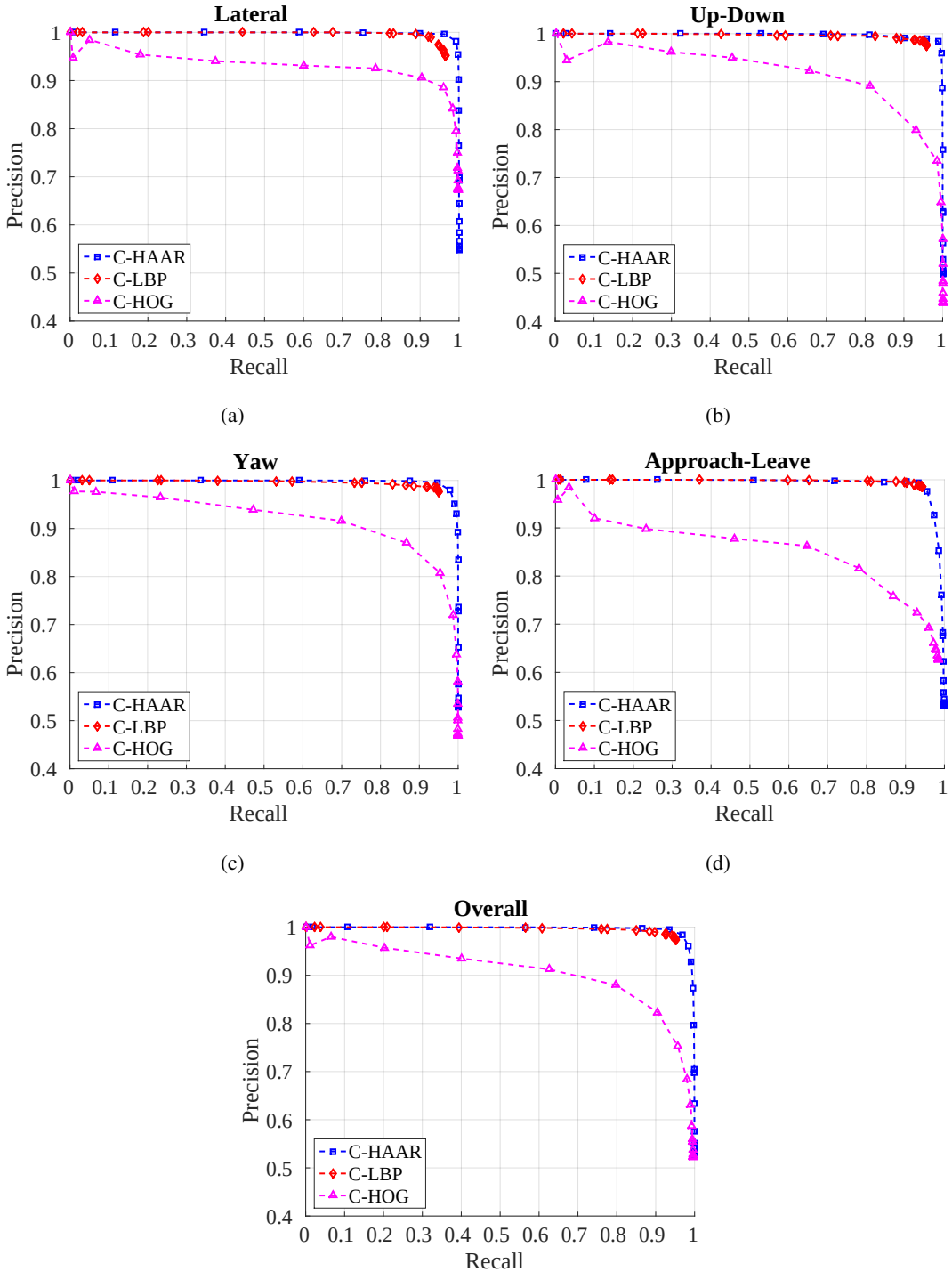



Figure 5.2: PR curves for (a) lateral left-to-right and right-to-left, (b) up and down, (c) yaw clockwise and counter-clockwise, (d) approach and leave, and (e) all motion types. Figures are taken from [38].

The performances of C-HAAR and C-LBP are close to each other in terms of maximum F-Scores (Table 5.1) and the normalized area under the curve (Figure 5.1(d)), except for a decrease on stage $15$ of C-HAAR, and they both perform better than C-HOG in all aspects. The lower performance of C-HOG is due to the low number of features it extracts from a training window. Even with the extension of Zhu et al. [109], only $20$ features are extracted from a $40 \times 22$ pixels training image. For AdaBoost to estimate a better decision boundary, more features are required. The difference between the number of features used by C-HAAR and C-LBP, however, does not result in a considerable performance difference.

We observe a slight difference between C-HAAR and C-LBP in terms of the lowest points that PR curves (Figure 5.1) reach. This is related to the performance differences between the methods at their default threshold. As mentioned earlier, decreasing the threshold of a classifier's latest stage, $S$, to a very low value results in a classifier with a stage number of $S - 1$. Therefore, since the performances of C-LBP classifiers at their default thresholds are greater than the default performances of C-HAAR classifiers, we observe PR curves ending at higher points in the case of C-LBP.

For all methods, training with $19$ stages outperforms training with less stages. Therefore, taking $19$ as the best stage number for all methods, we present their performances on different motion types in Figure 5.2 with their overall performances on all motion types. The performance of C-HAAR is slightly better than C-LBP on lateral, up-down and yaw motions, since it has PR curves closer to the rightmost top corner of the figures. C-HOG gives the worst performance in all motion types.

When we look at the performances of each method individually for each motion type, C-HAAR performs similar on lateral, up-down and yaw motions; however its performance diminishes on approach-leave, which is the most challenging motion in the indoor dataset. C-LBP has a performance degradation on lateral motion, showing that it is slightly affected by the large background changes. Other than this, the performance of C-LBP is almost equal for other motion types. C-HOG performs better on lateral than other motions. Notable performance degradation is observed for the approach-leave motion.

## 5.3 Outdoor Evaluation

We evaluated the classifiers trained with the outdoor training dataset using all outdoor motion types, namely calm, agile and moving background. For each motion type and for overall performance, we present the resulting PR curves and the normalized area under the curves in Figure 5.3 and Figure 5.4, respectively. The F-Score performances are listed in Table 5.2.

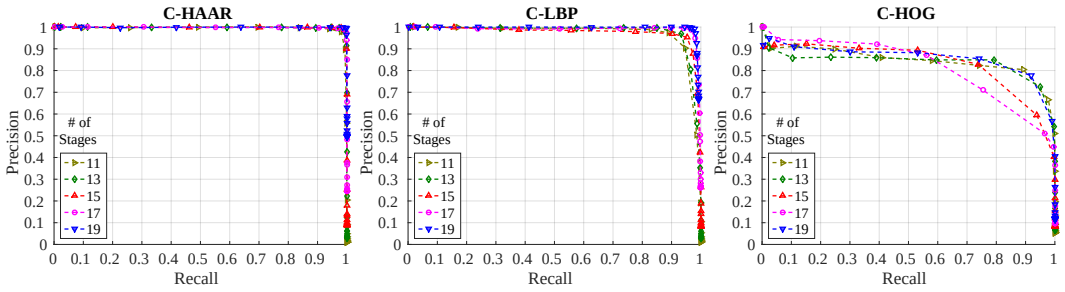
We notice that the performances of C-HAAR and C-LBP are remarkably better than C-HOG in all experiments. When comparing C-HAAR and C-LBP, C-HAAR gives slightly better results in terms of all measures. Under the agile maneuvers of the quadrotor, C-LBP and C-HOG display a performance degradation, while C-HAAR's performance is hardly affected. This suggests that C-HAAR is more robust against appearance changes due to the rotation of the quadrotor. Slight performance decreases are observed in moving background video for C-HAAR and C-LBP.

When compared to the indoor evaluation, C-HAAR classifiers with low stage numbers perform better outdoors. The performance of C-HOG decreases in outdoor tests. In terms of the F-Score, the best performing stage numbers differ for C-HAAR and C-HOG. Unlike indoors, the performances of the C-LBP and C-HAAR classifiers at their default thresholds are close to each other, resulting in PR curves reaching to closer end points when compared to indoor results.

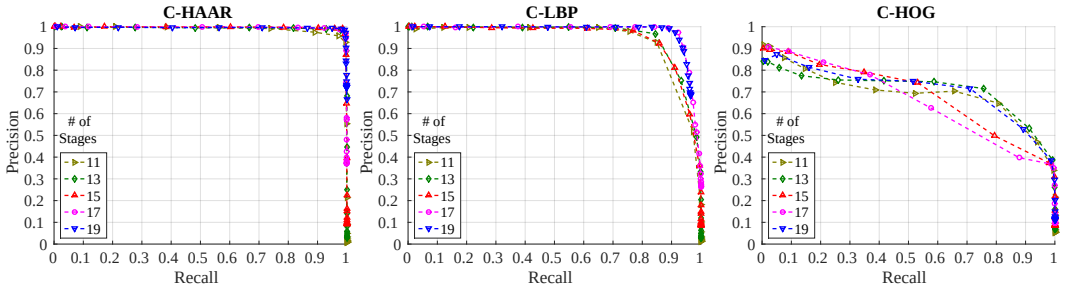
In order to determine the maximum distances at which the classifiers can detect the quadrotor successfully, an experiment is conducted with distance test video using the best performing classifiers on the overall according to the F-Scores in Table 5.2. In this experiment, the minimum object size is set to $20 \times 11$. The resulting maximum detection distances are $25.71$ m, $15.73$ m and $24.19$ m, respectively, for C-HAAR, C-LBP and C-HOG.

## 5.4 Performance under Motion Blur

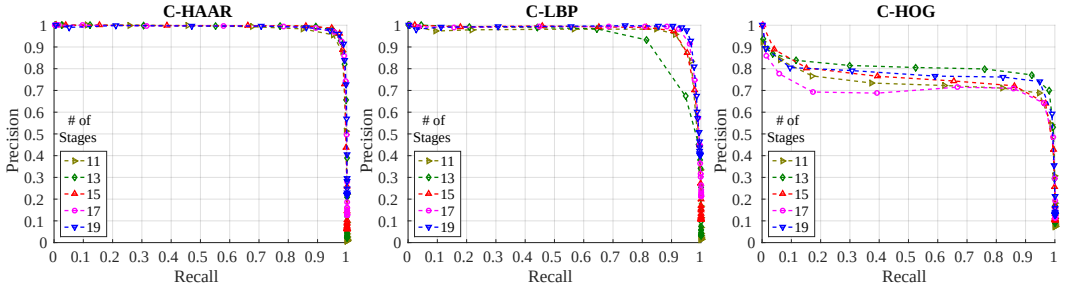
We tested the best performing classifiers having $19$ stages and giving the maximum F-Scores in Table 5.1 on the blurred and original videos. The results depicting the



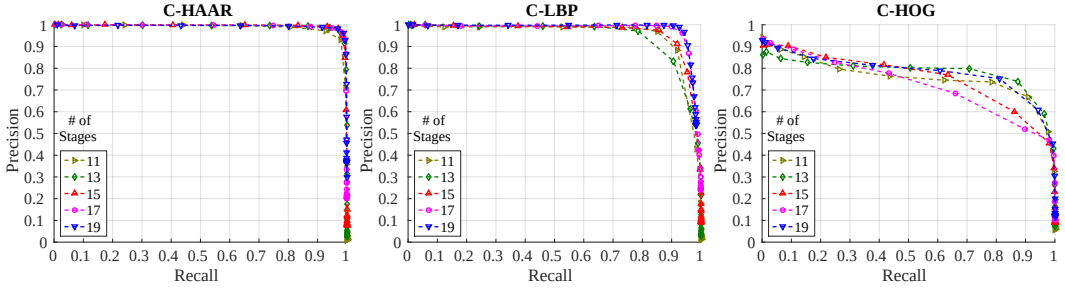

(a) Performances for calm test video.




(b) Performances for agile test video.

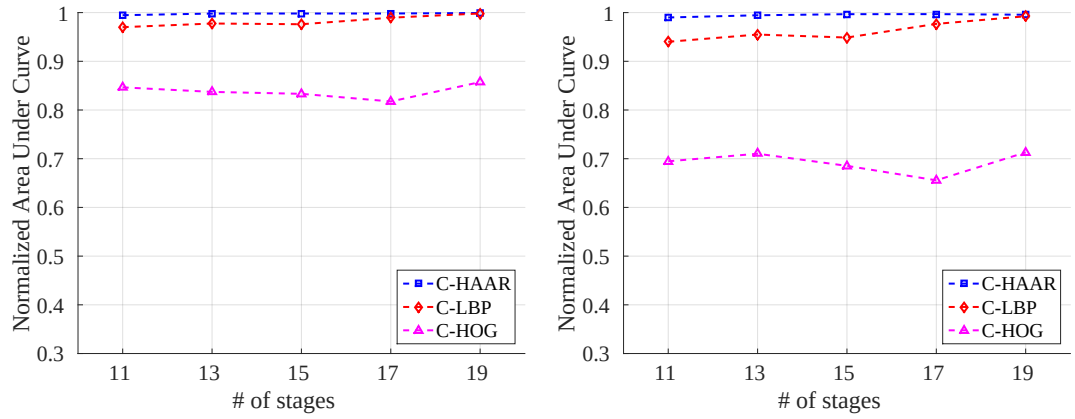



(c) Performances for moving background test video.



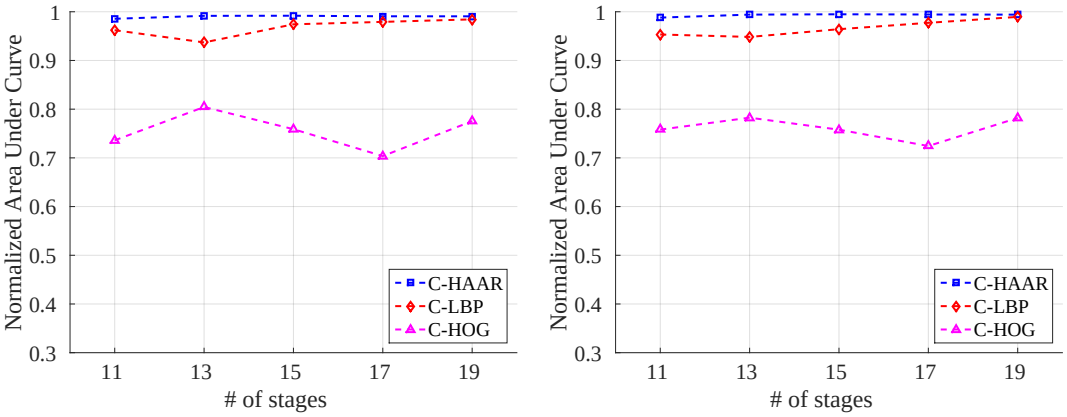

(d) Overall performances

Figure 5.3: PR curves for outdoor evaluation (Best viewed in color). Figures are taken from [38].




(a) Stationary background calm flight

(b) Stationary background agile flight




(c) Moving background calm flight

(d) All outdoor flights combined

Figure 5.4: Normalized area under curves for outdoor evaluation. Figures are taken from [38].

Table 5.1: F-Score values of the methods on **indoor** test videos. Two different F-Scores are given: (1) Maximum achievable F-Score by changing the threshold of the last stage, and (2) F-Score at the default threshold. Bold indicates best performances. Table is taken from [38].

| **Feature Type** | **C-HAAR** | | | | | **C-LBP** | | | | | **C-HOG** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of Stages | 11 | 13 | 15 | 17 | 19 | 11 | 13 | 15 | 17 | 19 | 11 | 13 | 15 | 17 | 19 |
| Maximum F-Score | 0.903 | 0.920 | 0.836 | 0.958 | **0.976** | 0.904 | 0.936 | 0.940 | 0.962 | **0.964** | 0.818 | 0.848 | 0.842 | 0.839 | **0.862** |
| F-Score at Default Threshold | 0.058 | 0.143 | 0.286 | 0.570 | **0.822** | 0.104 | 0.345 | 0.774 | 0.943 | **0.954** | 0.404 | 0.550 | 0.627 | 0.664 | **0.716** |

Table 5.2: F-Score values of the methods on **outdoor** test videos. Two different F-Scores are given: (1) Maximum achievable F-Score by changing the threshold of the last stage, and (2) F-Score at the default threshold. Bold indicates best performances. Table is taken from [38].

| **Feature Type** | | **C-HAAR** | | | | | **C-LBP** | | | | | **C-HOG** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of Stages | 11 | 13 | 15 | 17 | 19 | 11 | 13 | 15 | 17 | 19 | 11 | 13 | 15 | 17 | 19 |
| CALM | Maximum F-Score | 0.979 | 0.987 | 0.991 | 0.991 | **0.997** | 0.930 | 0.951 | 0.953 | 0.977 | **0.985** | **0.846** | 0.822 | 0.781 | 0.732 | 0.842 |
| | F-Score at Default Threshold | 0.036 | 0.112 | 0.248 | 0.536 | **0.734** | 0.040 | 0.095 | 0.266 | 0.670 | **0.930** | **0.118** | 0.144 | 0.168 | 0.189 | 0.216 |
| AGILE | Maximum F-Score | 0.965 | 0.983 | 0.988 | 0.987 | **0.989** | 0.887 | 0.902 | 0.890 | **0.947** | 0.942 | 0.719 | **0.735** | 0.619 | 0.600 | 0.713 |
| | F-Score at Default Threshold | 0.034 | 0.108 | 0.282 | 0.727 | **0.906** | 0.041 | 0.094 | 0.260 | **0.704** | 0.920 | 0.121 | **0.146** | 0.168 | 0.188 | 0.211 |
| MOVING BACKGROUND | Maximum F-Score | 0.955 | 0.965 | **0.969** | 0.963 | 0.967 | 0.935 | 0.870 | 0.940 | 0.954 | **0.964** | 0.797 | **0.840** | 0.785 | 0.777 | 0.832 |
| | F-Score at Default Threshold | 0.030 | 0.084 | **0.169** | 0.274 | 0.441 | 0.043 | 0.111 | 0.269 | 0.480 | **0.747** | 0.158 | **0.180** | 0.199 | 0.216 | 0.234 |
| **OVERALL** | Maximum F-Score | 0.955 | 0.972 | **0.977** | 0.973 | 0.975 | 0.906 | 0.869 | 0.915 | 0.949 | **0.957** | 0.770 | **0.801** | 0.707 | 0.672 | 0.781 |
| | F-Score at Default Threshold | 0.033 | 0.099 | **0.221** | 0.429 | 0.627 | 0.042 | 0.100 | 0.265 | 0.594 | **0.850** | 0.132 | **0.157** | 0.178 | 0.198 | 0.221 |

changes in F-Score, precision and recall against the amount of motion blur are given in Figure 5.5.

We see that C-HAAR and C-LBP display a more robust behavior compared to C-HOG, since the decreasing trend in their F-Score and recall values is slower than C-HOG. C-LBP performs better than C-HAAR in terms of F-Score and recall. However, the precision of C-HAAR and C-HOG increases slightly with the increasing amount of motion blur. The reason for this increase is the decrease in the number of false positives, since they start to be identified as background by C-HAAR and C-HOG when there is more noise. However, this trend has a limit, since, at some point, the noise causes a major decrease in the number of true positives. Here, $\sigma = 25$ is the point where the precision of C-HAAR and C-HOG starts to decrease.

In the case of C-LBP, precision values are continuously decreasing due to an increasing number of false positives. However, this degradation in precision is not so rapid. Moreover, the decreasing trend in the recall of C-LBP is slower than other methods. This slow decline rate in the recall is resulting from a high number of correct detections and a low number of incorrect rejections.

## 5.5 Distance Estimation

This section presents the experimental evaluation of the methods in terms of distance estimation and time to collision estimation.

In order to train the distance estimator described in Section 3.8, we prepared a training set of $35570$ pairs of $\{(w_i, h_i), d_i\}$, where $w_i, h_i$ are the width and the height of the mUAV bounding box, respectively, and $d_i$ is its known distance, acquired using the motion capture system (see Chapter 4 for the details).

Support Vector Regressor (SVR - [84]) inside the distance estimator has been trained on the training set with the Radial Basis Function (RBF) kernel. The values of the parameters are optimized using a grid-search and five-fold cross-validation, yielding the following values: $\nu = 0.09, C = 0.1$ and $\gamma = 0.00225$. With these values, a training error of $6.44$ cm as the median is obtained. The distribution of distance

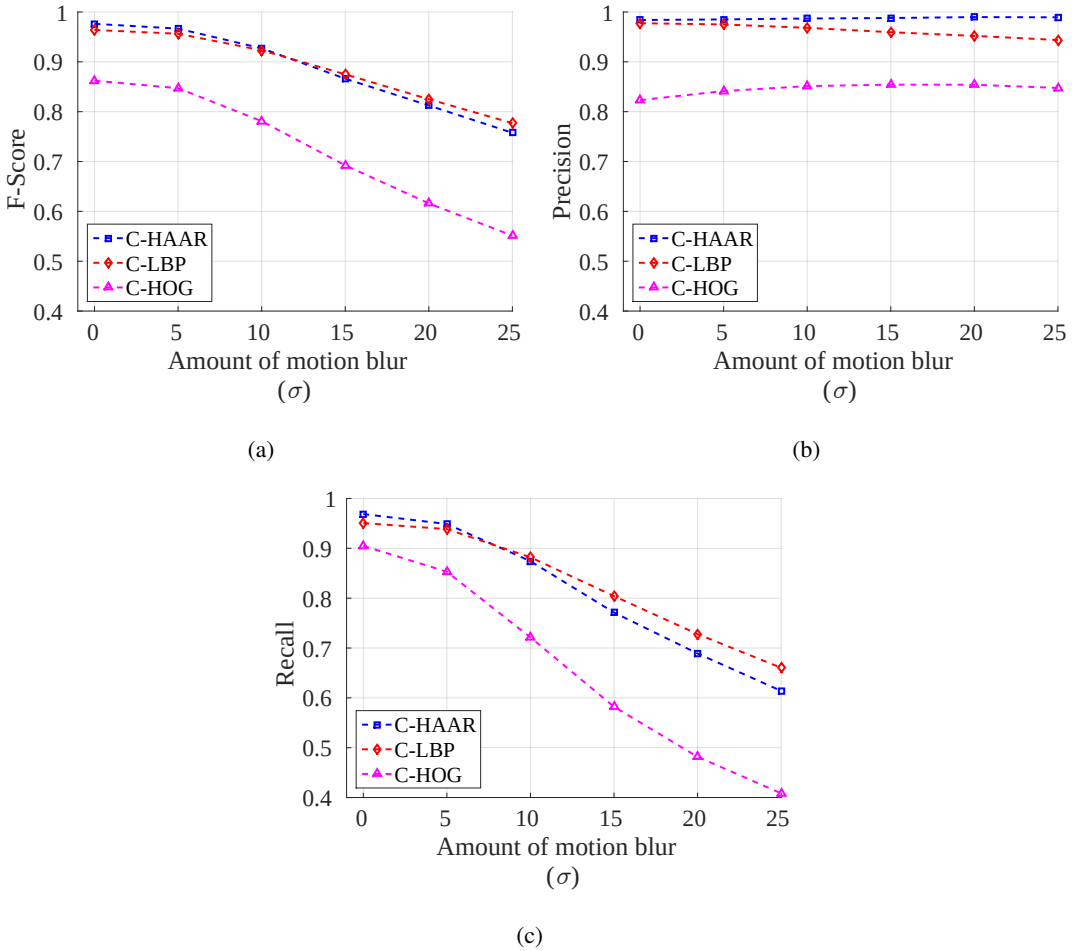
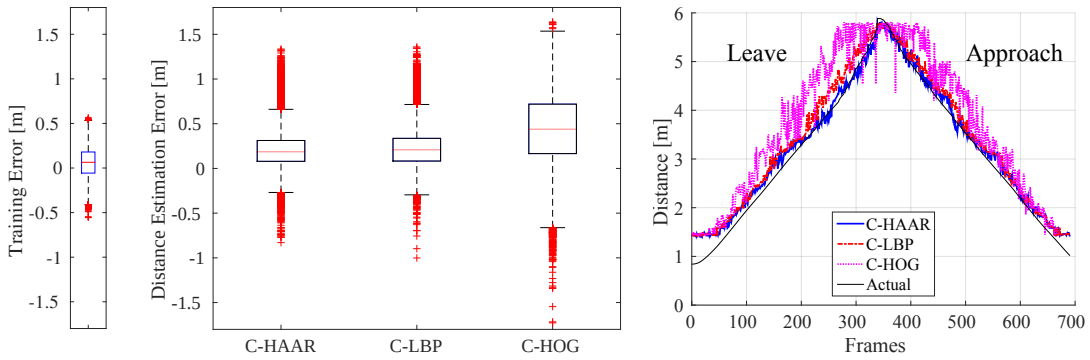



Figure 5.5: Performance of methods under motion blur. (a) F-Score, (b) Precision, and (c) Recall. To better illustrate the unexpected changes in precision and recall, they are plotted separately. $\sigma = 0$ corresponds to original videos without motion blur. Figures are taken from [38].




(a) (b) (c)

Figure 5.6: (a) Training error distribution for distance estimation. (b) Distribution of distance estimation error for each method. (c) Distance estimations during a leave motion followed by an approach. Figures are taken from [38].

estimation errors over the training set is shown in Figure 5.6(a).

Since there is no ground truth distance information at hand for the outdoor dataset the distance estimation has been evaluated by means of indoor videos only.

As in motion-blur analysis, we tested the best performing classifiers having 19 stages resulting in maximum F-Scores tabulated in Table 5.1. The resulting distance estimation distributions are displayed in Figure 5.6(b).

We see that the performance of C-HAAR is slightly better than C-LBP. The medians of the error for C-HAAR and C-LBP are 18.6 cm and 20.83 cm, respectively. The performance of C-HOG is worse than the other two methods with a median error of 43.89 cm and with errors distributed over a larger span.

In Figure 5.6(c), we plot estimated and actual distances for a leave motion followed by an approach. These plots are consistent with the results provided with Figure 5.6(b) such that the performances of C-HAAR and C-LBP are close to each other and better than C-HOG.

### 5.5.1 Time to Collision Estimation Analysis

We have analyzed the performance of the methods in the estimation of time to collision (*TTC*). In order to estimate *TTC*, the current speed ($v_c$) is estimated first:

$$v_c = \frac{d_c - d_p}{\Delta t}, \tag{5.6}$$

where $d_c$ is current distance estimation, $d_p$ is a previous distance estimation and $\Delta t$ is the time difference between two distance estimations. $d_p$ is arbitrarily selected as the $90 - th$ previous distance estimation to ensure a reliable speed estimation. Once $v_c$ is calculated, *TTC* can be estimated as:

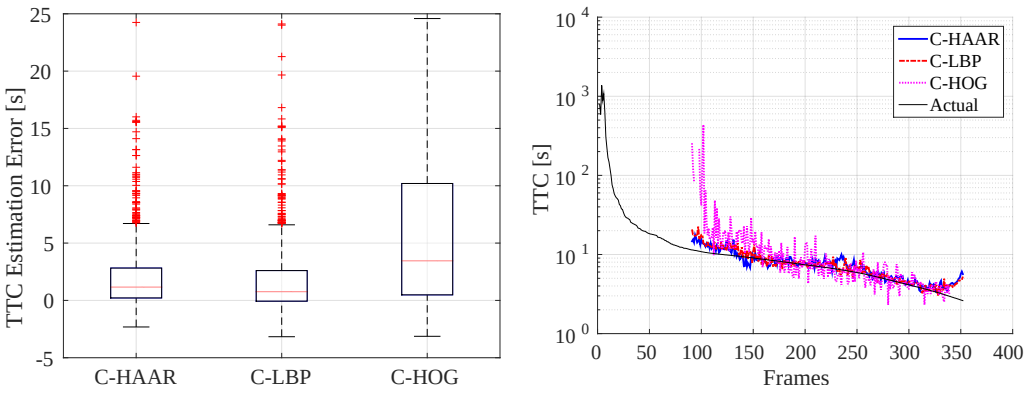
$$TTC = \frac{d_c}{v_c}. \tag{5.7}$$

Note that depending on the values of $d_c$ and $d_p$, it is possible that $v_c$ also may take negative values or be zero which makes *TTC* negative or $\infty$, respectively. $\textit{TTC} < 0$ means that the quadrotor is leaving away. $\textit{TTC} = \infty$ corresponds to a case where the quadrotor is stationary. In both cases, no collision is in question.

Using this approach, we have evaluated the methods on indoor approach videos. Figure 5.7(a) shows the resulting box-plots for errors in estimating *TTC*. Figure 5.7(b) illustrates the estimated and actual *TTC*'s for a single approach video. The performances of C-HAAR and C-LBP are close to each other with a smaller median error for C-LBP. C-HOG performs worse than C-HAAR and C-LBP as a result of its low performance in distance estimation.

## 5.6 Time Analysis

The training and testing time of the methods are analyzed in detail for the indoor and outdoor datasets on a computer with an Intel® Core™ i7-860 processor clocked at $2.80$-GHz and 8 GB DDR3-1333MHz memory, running Ubuntu 14.04. Currently, processors with similar computational power are available for mUAVs [5, 45].




(a) (b)

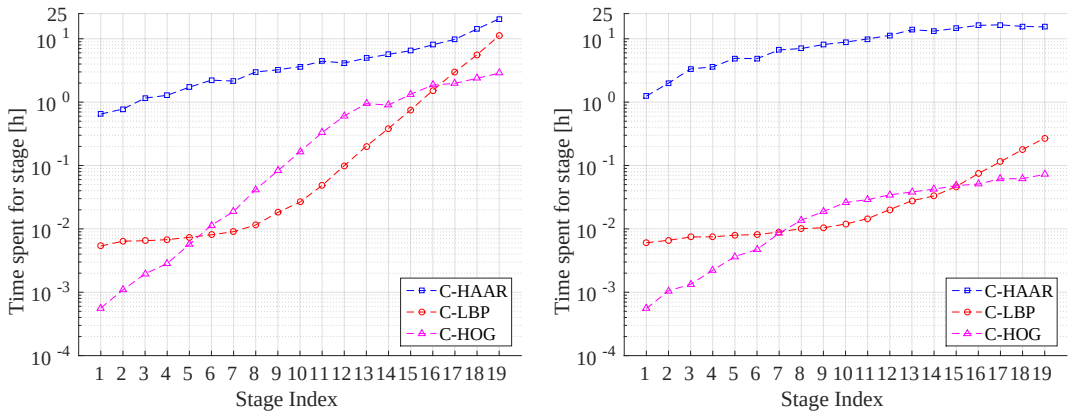
Figure 5.7: Indoor time to collision predictions of the methods for (a) all approach motions and (b) a single approach motion. In (a), there are outliers also outside the limits of the y-axis. However, in order to make differences between the methods observable, y-axis is limited between $-5$ and $25$. In (b), the y-axis is in $log$-scale, and no estimation is available until the $90^{th}$ frame. The missing points after the $90^{th}$ frame are due to negative or infinite time to collision estimations. Figures are taken from [38].

### 5.6.1 Training Time Analysis

Figure 5.8 shows the amount of time required to train *each stage of the classifiers*, and Table 5.3 lists the total training times needed for the training of all $19$ stages (the upper limit of $19$ has been imposed due to the excessive time required for training C-HAAR). We observe that C-HAAR is the most time consuming method, which is succeeded by C-LBP and C-HOG. It is observed that C-HAAR requires on the order of days for training, whereas C-LBP and C-HOG finish in even less than an hour.

The main reason behind the differences in the training times of the methods is the number of features extracted by each method from an image window. As mentioned previously, the ordering among the methods is C-HAAR, C-LBP and C-HOG, with the decreasing number of associated features with an image window of $40 \times 22$ pixels. The increase in the number of features amounts to an increase in training the cascaded classifier to select the subset of good features via boosting.

We also observe a significant difference between indoor and outdoor training times




(a) (b)

Figure 5.8: (a) Indoor and (b) outdoor training times consumed for each stage in the cascaded classifier. The y-axes are in $log$-scale. Figures are taken from [38].

Table 5.3: Training times for the cascaded classifiers having 19 stages in hours. Table is taken from [38].

| Feature Type | C-HAAR | C-LBP | C-HOG |
|---|---|---|---|
| Indoor | 98.31 | 22.94 | 13.53 |
| Outdoor | 177.59 | 0.87 | 0.52 |

for each method. For the outdoor dataset, C-HAAR is twice slower than on the indoor dataset, where C-LBP and C-HOG are 26-times faster. The reason for this is the fact that the outdoor background images are more distinct, enabling C-LBP and C-HOG to find the best classifier in each stage more quickly. However, this effect is not observed in C-HAAR, since Haar-like features are adversely affected by the illumination changes, which are observed substantially in our outdoor dataset.

### 5.6.2 Testing Time Analysis

We have measured and analyzed the computation time of each method in two different aspects: i) on a subset of the indoor videos, we measured the computation time by changing the distance of the quadrotor to understand the effect of the distance; and (ii) we analyzed the average running times needed to process indoor and outdoor frames, with respect to the number of stages and the thresholds.

For the first experiment, we have selected five videos from the yaw motion type for 1-,

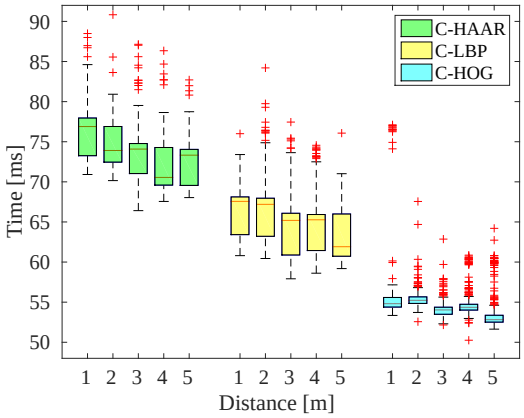


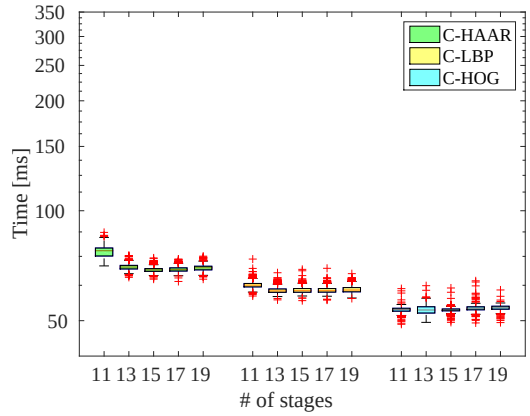

Figure 5.9: Change of computation time required to process one video frame with respect to the distance of the quadrotor. Figure is taken from [38].

2-, 3-, 4- and 5-m distances for middle level height. In total, there were 1938 frames in these videos. We tested the performance of the classifiers having 19 stages at their default thresholds, as shown in Figure 5.9, with respect to the distance between the quadrotor and the camera. Although there are fluctuations, the time required to process a single frame shows an inverse correlation. This is so because as a quadrotor gets further away, its footprint in the image will decrease, and hence, the bigger scale detectors will reject the candidate windows faster, which will yield a speed up in the overall detection.

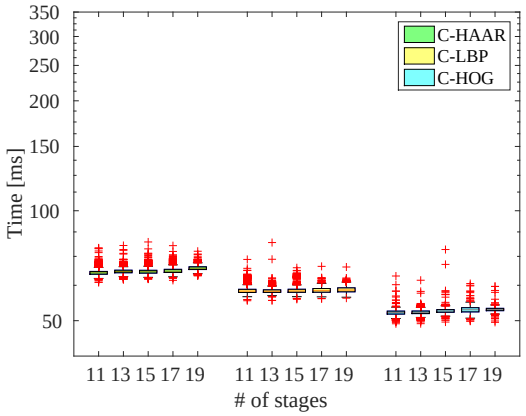
In our second experiment, we tested the running time performance of the classifiers with respect to the number of stages. This has been performed both for the classifiers at their default threshold, as well as with thresholds giving the maximum F-Score (See Table 5.1 and Table 5.2).

For indoor experiments, a subset of the indoor dataset consisting of videos from approach, down, lateral left-to-right and yaw-clockwise motion types containing 1366 frames in total was used. For the outdoor experiments, a total of 1500 frames from all motion types, namely calm, agile and moving background, were used. Figure 5.10 displays the resulting time performance distributions.

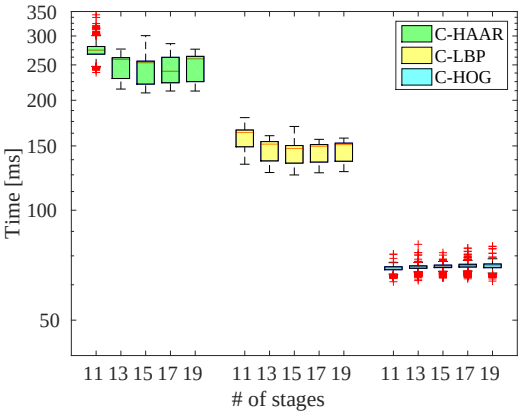
When we compare indoor and outdoor results, we observe that all three methods require more time to process outdoor frames. This increase reaches up to three times for C-HAAR and C-LBP. Outdoor frames are bigger than indoor frames by a factor of 1.15. This accounts partially for the increase in the processing time. However, the



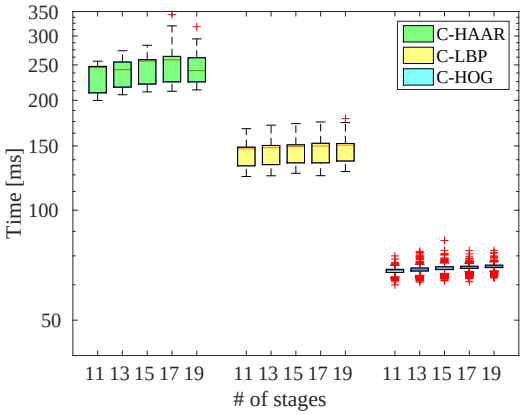

(a) Indoor tests with default thresholds




(b) Indoor tests with maximum F-Score thresholds




(c) Outdoor tests with default thresholds




(d) Outdoor tests with maximum F-Score thresholds

Figure 5.10: Analysis of time required to process one frame of (a-b) indoor and (c-d) outdoor videos. In (a) and (c), the classifiers are tested with their default thresholds, whereas in (b) and (d) the thresholds yielding maximum F-Score are used. Figures are taken from [38].

main reason is the higher complexity of outdoor background patterns, which manage to pass the early simple processing stages of the cascades more; thus, they consume more time before being identified as background.

When the results at the default thresholds and the maximum F-Score thresholds are compared, we observe an increase in the time spent on the lower stages of C-HAAR and C-LBP. This is due to the increasing number of candidate bounding boxes that are later merged into the resulting bounding boxes. Both detection and merging of these high number of candidate bounding boxes causes the processing time to increase.

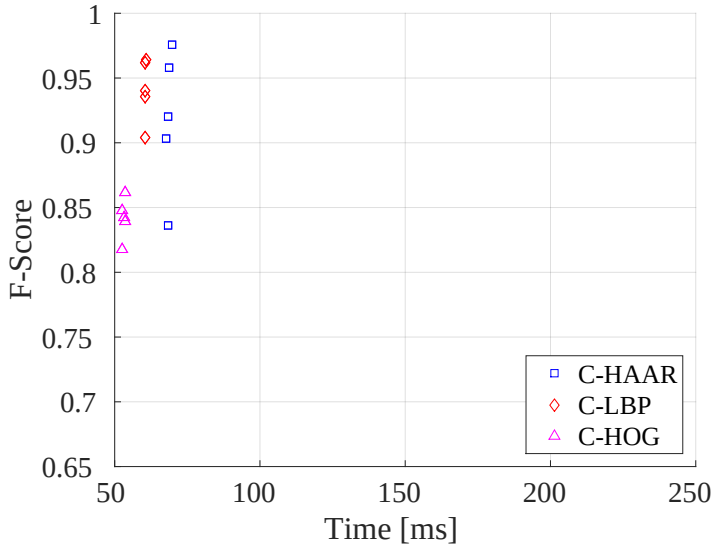
For the maximum F-Score thresholds, processing time increases with the number of stages. This is an inherent result due to the increase in the number of stages.

The scatter plots in Figure 5.11 display the distribution of F-Scores with respect to the mean running times both for indoors and outdoors. The classifiers used in these plots are the ones giving maximum F-Scores. The F-Score values for C-HAAR and C-LBP are close to each other and higher than C-HOG. For C-HAAR, the F-Score values are spread over a larger range for indoors, while the deviations in its mean time requirement increase for outdoors. The distributions observed for C-LBP for indoors and outdoors are similar to each other. The F-Score values of C-HOG decrease and disperse over a wide range for outdoors, but the spread of its mean time requirement is very similar for indoors and outdoors.

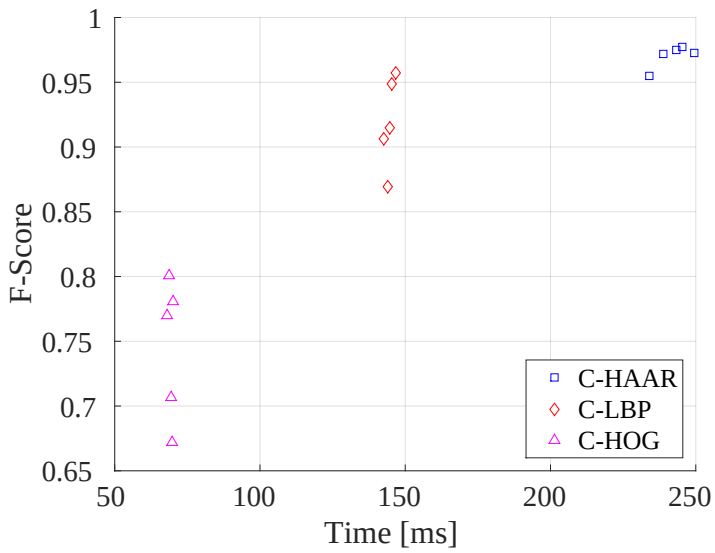
## 5.7 Sample Visual Results

In Figure 5.12, we present samples of successful detection and failure cases. These images are obtained using only the best performing C-LBP classifiers for the sake of space. C-LBP is remarkable among the three methods, since its detection and distance estimation performance is very high and close to that of C-HAAR. Furthermore, it is computationally more efficient than C-HAAR, both in training and testing. Three videos[1] are also available showing the detection performance of C-LBP on video sequences from the indoor and outdoor test datasets.

[1] Available at: `http://www.kovan.ceng.metu.edu.tr/~fatih/sensors/`

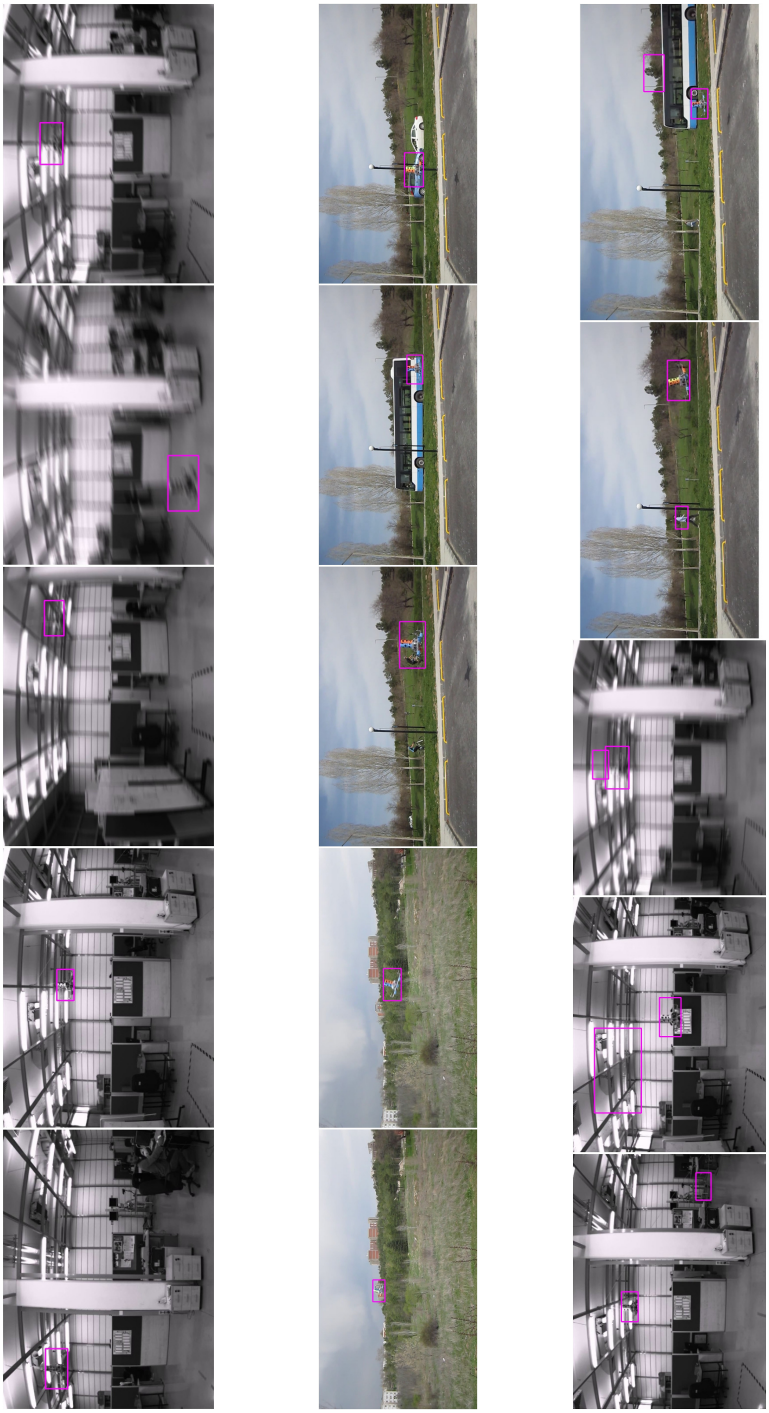



(a)




(b)

Figure 5.11: (a) Indoor and (b) outdoor scatter plots for F-Score and mean running times. Each F-Score value corresponds to a different classifier with different number of stages at the threshold resulting in maximum F-Score. Figures are taken from [38].



(a) Successful detections from indoor experiments.

(b) Successful detections from outdoor experiments.

(c) Failures from indoor and outdoor experiments.

Figure 5.12: Successful detection and failure examples from indoor and outdoor experiments obtained using best performing classifiers of C-LBP (only C-LBP results are provided for the sake of space). Figures are taken from [38].

The images in Figure 5.12(a) display the performance of the detector in an indoor environment that has extensive T junctions and horizontal patterns. The performance of the detector under motion blur is also displayed. Outdoor images in Figure 5.12(b) exemplify the outdoor performance of the detector where there are very complex textures, including also moving background patterns (pedestrians and various types of vehicles). When we look at the failures in Figure 5.12(c), we observe that the regions including T junctions, horizontal patterns and silhouettes very similar to the quadrotor's are the confusing areas for the algorithms.

# CHAPTER 6

# CONCLUSIONS

In this thesis, we have studied whether an mUAV can be detected and its distance can be estimated with a camera through cascaded classifiers using different feature types[1]. In order to demonstrate this in a systematic manner, we performed several experiments indoors and outdoors. For indoor evaluations, a motion platform was built to analyze the performance of the methods in controlled motions, namely in lateral, up-down, rotational and approach-leave motions. For outdoor evaluations, on the other hand, the methods were evaluated for the cases where the mUAV was flown in a calm manner, an agile manner or with other moving objects in the background. The maximum detection distances of the methods are also analyzed with an outdoor experiment.

We evaluated the detection performance of three methods, namely C-HAAR, C-LBP and C-HOG, where, in each method, a different feature extraction approach is combined with the boosted cascaded classifiers and with a distance estimator utilizing SVR. Our experiments showed that near real-time detection and accurate distance estimation of mUAVs are possible. C-LBP becomes prominent among the three methods due to its: (1) high performance in detection and distance and time to collusion estimation; (2) moderate computation time; (3) reasonable training time; and (4) more robustness to the motion blur. When it comes to distance estimation, C-HAAR performs better, since it positions the bounding boxes more accurately compared to the other methods. On the other hand, our time analysis reveals that C-HOG is the fastest, both in training and testing.

[1] This chapter is partially published in [38].

We have demonstrated that an mUAV can be detected in about $60$ ms indoors and $150$ ms outdoors in images with $1032 \times 778$ and $1280 \times 720$ resolutions, respectively, with a detection rate of $0.96$ for the F-score, both indoors and outdoors. Although this cannot be considered real time, a real-time performance with cascaded classifiers is reachable, especially considering that the implementations are not optimized. We also showed that distance estimation of mUAVs is possible using simple geometric cues and the SVR; even the change in the pose of the quadrotor or the camera results in different bounding boxes for the same distance between mUAV and the camera.

The performance of detection can be improved significantly when combined with tracking methods. Such methods limit the search space of the detector in the next frame(s) by using the properties of the current and previous detections. This can improve both the detection performance and the running time substantially. Various tracking methods are studied in the literature for object tracking (See [56, 63, 104] for reviews.). Since we have a detector, the tracking problem here fits into detection based tracking category [63]. Once mUAV is detected, visual properties inside the detection window such as intensity and color (if available) could be utilized to determine most prominent locations for the detection window in a next frame. However, since mUAVs tilt and rotate during their motions, the appearance of them could change in subsequent frames and this should be considered along with the non-convex structure of the mUAVs. Optical flow can be also utilized for the same purpose. The motion of the mUAV can be estimated from previous detections by also integration the distance estimation. Kalman filter [49, 98] or particle filtering [60, 67] approaches can be employed for this purpose. In a swarm study, if there is a communication link between the mUAVs, they can share their inertial navigation information with their neighbors. This information can be used to enhance their motion estimations.

Cascaded approaches are known to generalize rather well with the increase of the number of objects. By looking at simple, fast, yet effective features at multiple stages to minimize false positives and to maximize detection rates, successful applications on complex and challenging datasets with many exemplars of the same class have been reported [28, 107, 109]. These indicate that, for mUAV detection, cascaded approaches are very suitable, even if many mUAV variants with appearance characteristics are included.

# REFERENCES


[1] 3DRobotics. Arducopter: Full-featured, open-source multicopter uav controller. http://copter.ardupilot.com/ [Last accessed: 19 August 2015].

[2] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart. Collaborative stereo. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2242–2248, Sept 2011.

[3] E. Ackerman. When drone delivery makes sense. *IEEE Spectrum*, 25 Sep 2014. Available: http://spectrum.ieee.org/automaton/robotics/aerial-robots/when-drone-delivery-makes-sense [Last accessed: 19 August 2015].

[4] A. Andreopoulos and J. K. Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891, 2013.

[5] AscendingTechnologies. Asctec mastermind. http://www.asctec.de/en/asctec-mastermind/ [Last accessed: 19 August 2015].

[6] M. Basiri, F. Schill, D. Floreano, and P. Lima. Audio-based Relative Positioning System for Multiple Micro Air Vehicle Systems. In *Robotics: Science and Systems (RSS)*, 2013.

[7] M. Basiri, F. Schill, D. Floreano, and P. Lima. Audio-based localization for swarms of micro air vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4729–4734, May 2014.

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

[9] S. Belongie and J. Malik. Matching with shape contexts. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 20–26, 2000.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[11] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[12] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[13] E. Brewer, G. Haentjens, V. Gavrilets, and G. McGraw. A low swap implementation of high integrity relative navigation for small uas. In *Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION*, pages 1183–1187, May 2014.

[14] A. Bürkle, F. Segor, and M. Kollmann. Towards autonomous micro uav swarms. *Journal of Intelligent & Robotic Systems*, 61(1-4):339–353, 2011.

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. *European Conference on Computer Vision (ECCV)*, 6314:778–792, 2010.

[16] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[17] Y.-C. Choi and H.-S. Ahn. Formation control of quad-rotors in three dimension based on euclidean distance dynamics matrix. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, pages 1168–1173, Oct 2011.

[18] I. Colomina and P. Molina. Unmanned aerial systems for photogrammetry and remote sensing: A review. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 92(0):79 – 97, 2014.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[20] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. In E. Şahin and W. Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin Heidelberg, 2005.

[21] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[22] M. Cutler, B. Michini, and J. How. Lightweight infrared sensing for relative navigation of quadrotors. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 1156–1164, May 2013.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:886–893, 2005.

[24] M. B. Darling. Autonomous close formation flight of small uavs using vision-based localization. Master's thesis, California Polytechnic State University, San Luis Obispo, 2014.

[25] D. Dey, C. Geyer, S. Singh, and M. Digioia. Passive, long-range detection of aircraft: Towards a field deployable sense and avoid system. In *In Proceedings of Field and Service Robotics*. Cambridge, MA, 2009.

[26] D. Dey, C. Geyer, S. Singh, and M. Digioia. A cascaded method to detect aircraft in video imagery. *International Journal of Robotics Research*, 30(12):1527 – 1540, October 2011.

[27] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[28] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[29] W. Etter, P. Martin, and R. Mangharam. Cooperative flight guidance of autonomous unmanned aerial vehicles. In *CPS Week Workshop on Networks of Cooperating Objects (CONET)*, CPS Week 2011, Chicago, 2011.

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

[31] J. Faigl, T. Krajnik, J. Chudoba, L. Preucil, and M. Saska. Low-cost embedded system for relative localization in robotic swarms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 993–998, May 2013.

[32] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531 vol. 2, June 2005.

[33] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular slam with multiple micro aerial vehicles. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3962–3970, Nov 2013.

[34] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[35] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[36] A. Gaschler. Real-time marker-based motion tracking: Application to kinematic model estimation of a humanoid robot. Master's thesis, Technische Universität München, Germany, 2011.

[37] A. Gaschler, M. Springer, M. Rickert, and A. Knoll. Intuitive robot tasks with augmented reality and virtual obstacles. In *IEEE International Conference on Robotics and Automation (ICRA)*, June 2014.

[38] F. Gökçe, G. Üçoluk, E. Şahin, and S. Kalkan. Vision-based detection and distance estimation of micro unmanned aerial vehicles. *Sensors*, 15(9):23805–23846, 2015.

[39] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[40] S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J.-C. Zufferey, and D. Floreano. Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5015–5020, Sept 2011.

[41] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control Engineering Practice*, 19(9):1023 – 1036, 2011. Special Section: DCDS'09 - The 2nd {IFAC} Workshop on Dependable Control of Discrete Systems.

[42] K. Holmes. Man detained outside white house for trying to fly drone. *CNN*, 15 May 2015. Available: http://edition.cnn.com/2015/05/14/politics/white-house-drone-arrest/ [Last accessed: 19 August 2015].

[43] B. K. P. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1988.

[44] H. Hörtner, F. Berger, C. Bruckmayr, P. H. A. Jalsovec, M. Mayr, M. Mörth, P. Müller, K. Obernhumer, B. Olsen, and M. Platz. Arselectronica spaxels. http://www.aec.at/spaxels/, 2012. Accessed: 2015-07-08.

[45] D. Hulens, J. Verbeke, and T. Goedeme. How to choose the best embedded processing platform for on-board uav image processing? In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, pages 377–386, 2015.

[46] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[47] J. Jimenez Lugo, A. Masselli, and A. Zell. Following a quadrotor with another quadrotor using onboard vision. In *European Conference on Mobile Robots (ECMR)*, pages 26–31, Sept 2013.

[48] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In P. Remagnino,

G. Jones, N. Paragios, and C. Regazzoni, editors, *Video-Based Surveillance Systems*, pages 135–144. Springer US, 2002.

[49] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[50] T. Krajnik, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail. A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 76(3-4):539–562, 2014.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[52] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[53] J. Lai, L. Mejias, and J. J. Ford. Airborne vision-based collision-detection system. *Journal of Field Robotics*, 28(2):137–157, 2011.

[54] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 5 2015.

[55] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. *International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011.

[56] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.*, 4(4):58:1–58:48, Oct. 2013.

[57] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics*, pages 828–837. Springer, 2007.

[58] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing*, volume 1, pages I–900–I–903 vol.1, 2002.

[59] F. Lin, K. Peng, X. Dong, S. Zhao, and B. M. Chen. Vision-based formation for uavs. In *IEEE International Conference on Control Automation (ICCA)*, pages 1375–1380, June 2014.

[60] J. S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[61] D. G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision (ICCV)*, 2:1150–1157, 1999.

[62] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[63] W. Luo, X. Zhao, and T. Kim. Multiple object tracking: A review. *CoRR*, abs/1409.7618, 2014.

[64] Z. Mahboubi, Z. Kolter, T. Wang, and G. Bower. Camera based localization for autonomous uav formation flight. In *AIAA Infotech@Aerospace Conference*. American Institute of Aeronautics and Astronautics, 2011.

[65] M. Martinez, P. Vercammen, and B. Brumfield. Above spectacular wildfire on freeway rises new scourge: drones. *CNN*, 19 July 2015. Available: http://edition.cnn.com/2015/07/18/us/california-freeway-fire/ [Last accessed: 19 August 2015].

[66] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 36.1–36.10, 2002.

[67] P. D. Moral. Nonlinear filtering: Interacting particle resolution. *Markov Processes and Related Fields*, 2(4):555–580, 1996.

[68] A. Moses, M. Rutherford, and K. Valavanis. Radar-based detection and identification for miniature air vehicles. In *Control Applications (CCA), 2011 IEEE International Conference on*, pages 933–940, Sept 2011.

[69] A. Moses, M. J. Rutherford, M. Kontitsis, and K. P. Valavanis. Uav-borne x-band radar for collision avoidance. *Robotica*, 32:97–114, 1 2014.

[70] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, pages 382–400. Springer, 2006.

[71] T. Naegeli, C. C. andA. Domahidi, M. Morari, and O. Hilliges. Environment-independent formation flight for micro aerial vehicles. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[72] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *12th IAPR Int. Conf. on Pattern Recognition*, 1:582–585, 1994.

[73] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer vision*, pages 555–562. IEEE, 1998.

[74] S. Petridis, C. Geyer, and S. Singh. Learning to detect aircraft at low resolutions. In A. Gasteratos, M. Vincze, and J. Tsotsos, editors, *Computer Vision Systems*, volume 5008 of *Lecture Notes in Computer Science*, pages 474–483. Springer Berlin Heidelberg, 2008.

[75] S. Quintero, G. Collins, and J. Hespanha. Flocking with fixed-wing uavs for distributed sensing: A stochastic optimal control approach. In *American Control Conference (ACC)*, pages 2025–2031, June 2013.

[76] T. Raharijaona, P. Mignon, R. Juston, L. Kerhuel, and S. Viollet. Hypercube: A small lensless position sensing device for the tracking of flickering infrared leds. *Sensors*, 15(7):16484, 2015.

[77] I. M. Rekleitis. Visual motion estimation based on motion blur interpretation. Master's thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada, 1995.

[78] J. Roberts, T. Stirling, J. Zufferey, and D. Floreano. 3-d relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1-2):5–20, 2012.

[79] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *European Conference on Computer Vision (ECCV)*, 3951:430–443, 2006.

[80] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[81] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[82] M. Saska, J. Chudoba, L. Precil, J. Thomas, G. Loianno, A. Tresnak, V. Vonasek, and V. Kumar. Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 584–595, May 2014.

[83] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Fraundorfer, E. B. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss. Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments. *IEEE Robotics and Automation Magazine*, 21(3):26–40, September 2014.

[84] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[85] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

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

[87] A. K. Soe and X. Zhang. A simple psf parameters estimation method for the de-blurring of linear motion blurred images using wiener filter in opencv. In *International Conference on Systems and Informatics (ICSAI)*, pages 1855–1860, May 2012.

[88] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.

[89] R. Tron, J. Thomas, G. Loianno, J. Polin, V. Kumar, and K. Daniilidis. Vision-based formation control of aerial vehicles. In *Workshop on Distributed Control and Estimation for Robotic Vehicle Networks*, 2014.

[90] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[91] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, Apr 1991.

[92] P. Vandergheynst, R. Ortiz, and A. Alahi. FREAK: Fast Retina Keypoint. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 0:510–517, 2012.

[93] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3866–3873, 2014.

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

[95] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[96] E. W. Weisstein. Haar function. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/HaarFunction.html [Last accessed: 19 August 2015].

[97] E. W. Weisstein. Wavelet. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Wavelet.html [Last accessed: 19 August 2015].

[98] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[99] J. Welsby, C. Melhuish, C. Lane, and B. Qy. Autonomous minimalist following in three dimensions: A study with small-scale dirigibles. In *Proceedings of Towards Intelligent Mobile Robots*, 2001.

[100] K. Wenzel, A. Masselli, and A. Zell. Visual tracking and following of a quadrocopter by another quadrocopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4993–4998, Oct 2012.

[101] D. Wilson, A. Goktogan, and S. Sukkarieh. A vision based relative navigation framework for formation flight. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4988–4995, May 2014.

[102] D. B. Wilson, M. Schwarzbach, A. H. Göktoğan, and S. Sukkarieh. An infrared vision system for uav close formation flight. In *Australian International Aerospace Congress*, February 2015.

[103] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1800–1807. IEEE, 2005.

[104] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomput.*, 74(18):3823–3831, Nov. 2011.

[105] B. Yu, X. Dong, Z. Shi, and Y. Zhong. Formation control for quadrotor swarm systems: Algorithms and experiments. In *Control Conference (CCC), 2013 32nd Chinese*, pages 7099–7104, July 2013.

[106] C. Yuan, Y. Zhang, and Z. Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, 45(7):783–792, 2015.

[107] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li. Face detection based on multi-block lbp representation. In S.-W. Lee and S. Li, editors, *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 11–18. Springer Berlin Heidelberg, 2007.

[108] M. Zhang, F. Lin, and B. M. Chen. Vision-based detection and pose estimation for formation of micro aerial vehicles. In *International Conference on Automation Robotics Vision (ICARCV)*, pages 1473–1478, Dec 2014.

[109] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:1491–1498, 2006.

# CURRICULUM VITAE

## PERSONAL INFORMATION

**Surname, Name:** Gökçe, Fatih
**Nationality:** Turkish (TC)
**Date and Place of Birth:** 17.03.1982, Antalya
**Marital Status:** Married
**Phone:** 0 312 2105545

## EDUCATION

| Degree | Institution | Year of Graduation |
|---|---|---|
| Ph.D. | Department of Computer Engineering, METU | 2015 |
| M.Sc. | Department of Computer Engineering, METU | 2008 |
| B.S. | Department of Electrical and Electronics Engineering, Selçuk University | 2004 |
| High School | Antalya İmam-Hatip High School | 1999 |

## PROFESSIONAL EXPERIENCE

| Year | Place | Enrollment |
|---|---|---|
| 2005-Present | Department of Computer Engineering, METU | Teaching Assistant |
| 2005-Present | KOVAN Research Laboratory, Department of Computer Engineering, METU | Researcher |
| 2006-2008 | TÜBITAK Project 104E066 | Researcher |
| 2011-2012 | TAI MRC Project | Researcher |

## PUBLICATIONS

### Journal Publications

- **F. Gökçe**, G. Üçoluk, E. Şahin, and S. Kalkan. Vision-Based Detection and Distance Estimation of Micro Unmanned Aerial Vehicles. *Sensors*, 15(9):23805-23846, 2015.
- **F. Gökçe** and E. Şahin. The pros and cons of flocking in the long-range "migration" of mobile robot swarms. *Theoretical Computer Science*, 411(21):2140-2154, 2010.
- A. E. Turgut, H. Çelikkanat, **F. Gökçe**, and E. Şahin. Self-Organized Flocking in Mobile Robot Swarms. *Swarm Intelligence*, 2(2-4):97-120, 2008.

### International Conference Publications

- **F. Gökçe** and E. Şahin. To flock or not to flock: Pros and cons of flocking in long-range "migration" of mobile robot swarms. In *Proceedings of The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 65-72, 2009.
- A. E. Turgut, C. Huepe, H. Çelikkanat, **F. Gökçe**, and E. Şahin. Modelling Phase Transition in Self-Organized Mobile Robot Flocks. In *Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS)*, LNCS volume 5217, pages 108-119, 2008.
- A. E. Turgut, H. Çelikkanat, **F. Gökçe**, and E. Şahin. Self- Organized Flocking with a Mobile Robot Swarm. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 39-46, 2008.

### National Conference Publications

- **F. Gökçe**, S. Olgunsoylu, G. Üçoluk, E. Şahin, and S. Kalkan. Görüntü İşleme ile Mikro İnsansız Hava Araçlarının Algılanması. In *Türkiye Otonom Robotlar*

*Konferansi (TORK)*, ODTÜ, Ankara, 6-7 Kasim 2014.

- A. E. Turgut, **F. Gökçe**, H. Çelikkanat, L. Bayındır, and E. Şahin. Kobot: Sürü Robot Çalışmaları için Tasarlanmış Gezgin Robot Platformu (*Eng.* Kobot: A Mobile Robot Platform Developed for Swarm Robotics Research). In *Türkiye Otomatik Kontrol Ulusal Toplantısı (TOK)*, pages 259-264, 2007

**Technical Reports**

- A. E. Turgut, **F. Gökçe**, H. Çelikkanat, L. Bayındır, and E. Şahin. Kobot: A mobile robot designed specifically for swarm robotics research. Tech. Rep. METU-CENG-TR-2007-05, Dept. of Computer Eng., Middle East Tech. Univ., Ankara, Turkey, 2007.