Smart video surveillance of pedestrians: fixed, aerial, and multi-camera methods

Author: Pau Climent-Pérez

First Supervisor: Prof. Paolo Remagnino

Second and Third Supervisors: Prof. Dorothy N. Monekosso, Prof. Sarah A. Barman

Robot Vision Team (RoViT)

Faculty of Science, Engineering & Computing

Kingston University

Penrhyn Road, Kingston-Upon-Thames

KT1 2EE, London, U.K.

This Thesis is being submitted in partial fulfilment of the requirements of

Kingston University for the award of Doctor of Philosophy (Ph.D.)

3rd Oct, 2016

Some images have been removed from this thesis for copyright reasons.

Declaration

This thesis is submitted as requirement for the Ph.D. Degree in Artificial Intelligence, in the Digital Imaging Research Centre of the School of Computing and Information Systems, Faculty of Science, Engineering and Computing at Kingston University. It is substantially the result of my own work, except where explicitly indicated in the text.

This thesis does not contain any material that has been previously submitted for a comparable academic award at an institute of Higher Education either in the UK or overseas.

Kingston-upon-Thames, London, United Kingdom.

3rd Oct, 2016

Dedication

A Aida...

...la meua co-autora en la vida,

t'estime, t'estimo, t'estim-ø

Als meus pares...

...pel seu suport.

Abstract

Crowd analysis from video footage is an active research topic in the field of computer vision. Crowds can be analysed using different approaches, depending on their characteristics. Furthermore, analysis can be performed from footage obtained through different sources. Fixed CCTV cameras can be used, as well as cameras mounted on moving vehicles. To begin, a literature review is provided, where research works in the fields of crowd analysis, as well as object and people tracking, occlusion handling, multi-view and sensor fusion, and multi-target tracking are analysed and compared, and their advantages and limitations are highlighted. Following that, the three contributions of this thesis are presented: in a first study, crowds will be classified based on various cues (i.e. density, entropy), so that the best approaches to further analyse behaviour can be selected; then, some of the challenges of individual target tracking from aerial video footage will be tackled; finally, a study on the analysis of groups of people from multiple cameras is proposed. The analysis entails the movements of people and objects in the scene. The idea is to track as many people as possible within the crowd, and to be able to obtain knowledge from their movements, as a group, and to classify different types of scenes. An additional contribution of this thesis, are two novel datasets: on the one hand, a first set to test the proposed aerial video analysis methods; on the other, a second to validate the third study, that is, with groups of people recorded from multiple overlapping cameras performing different actions.

Resum

L'anàlisi de multituds a partir de vídeo és un tema de recerca que resulta d'interès en el camp de la visió per computador. Aquesta anàlisi es pot fer des de diversos enfocaments, depenent de les característiques de la multitud. A més, pot realitzar-se amb vídeos obtinguts de diverses fonts. Per exemple, hi ha càmeres de vigilància fixes, i n'hi ha de muntades sobre vehicles en moviment. Per començar, s'hi inclou una revisió de la bibliografia, en què s'hi presenten els avantatges i limitacions i s'hi comparen treballs relacionats amb els camps de l'anàlisi de multituds, així com de seguiment de trajectòria de persones i objectes; maneig de les oclusions, fusió de dades provinents de sensors diversos o múltiples vistes; així com seguiment de trajectòria amb múltiples objectius. A continuació, es presenten les tres contribucions d'aquesta tesi: en un primer estudi, es classificaran les multituds depenent de diversos factors, com ara la densitat i l'entropia, de forma que es podrà seleccionar automàticament el millor enfocament per realitzar les tasques d'anàlisi subsegüents. Després d'això, un segon estudi presentarà solucions novedoses a alguns dels reptes actuals per a l'anàlisi de trajectòries d'individus amb seqüències preses des de vehicles aeris. Finalment, s'ofereix un estudi sobre l'anàlisi de grups de gent. Tenint en compte els moviments de les persones i els objectes presents a l'escena, la idea és d'intentar seguir la trajectòria de tanta gent del grup com siga possible, i obtindre'n coneixement a nivell de grup, classificant els diferents tipus d'escenes. Com a contribució addicional, aquesta tesi presenta dos conjunts de test de referència: per un costat, un primer per validar els mètodes d'anàlisi de vídeos aeris; per un altre costat, un segon per validar el tercer estudi, això és amb grups de persones realitzant accions de grup enregistrades des de diverses càmeres amb camps de visió sobreposats.

Acknowledgments

I would like to start by thanking Dr. Francisco Flórez-Revuelta for having inspired in me the interest in research, and for encouraging me to take this journey. I would also like to thank all my colleagues of the DAI research group in the University of Alicante, but very specially those who have become good friends: Alexandros Chaaraoui, and José Padilla. Congratulations on your wedding, Alex and Mela ♡! This thesis would not have been possible without my supervisor, Prof. Paolo Remagnino, and the rest of the supervisory team: Prof. Ndedi Monekosso, and Prof. Sarah Barman. Thank you for the great insights and your yummy pumpkin and pecan pies. Many thanks also to *my* co-authors and collaborators, specially Mei Kuan Lim and Giounona Tzanidou. Special thanks to the examiners, Prof. Tim Ellis and Prof. Mark Nixon. I would also like to thank all the academic and non-academic staff at Kingston University in London, and especially my interviewing panel and for eventually awarding me a fully funded Ph.D. studentship, allowing for this research to be conducted. Also Jackie Deacon and Rosalind Percival, for their prompt response, professionalism, and patience with my many questions. A very warm thank you to all the academics, Ph.D. students, and visitors at DIRC, the "DIRC lab", and some members of the Maths department, some have become very good friends during these years, they know who they are. Special thanks to Spyros, Matthaios, Reyhaneh, Vic and Raphael (whose template, modified by Spyros, was used in this thesis), and to J.C. Nebel, Dimitris Makris, and Gordon Hunter. If you are unhappy because you could not find your name here, browse pages v–x of Bakas [25], your name must *certainly* be there ;). Many thanks to all the partners in FP7 project PROACTIVE, it has been a pleasure to meet and collaborate with all of you. This work has been supported by the European Commission's Seventh Framework Programme (FP7-SEC-2011-1) under grant agreement Nº 285320 (PROACTIVE project). Finally I want to thank my family, friends, and my girlfriend, Aida, for all the patience they have had with me during the harder times of this Ph.D. *Gràcies!*

Contents

List of Publications

Book chapters

- M. Thida, Y. L. Yong, P. Climent-Pérez, H.-l. Eng, and P. Remagnino. A Literature Review on Video Analytics of Crowded Scenes. In A. Cavallaro and P. K. Atrey, editors, *Intelligent Multimedia Surveillance: Current Trends and Research*, pages 17–36. Springer Berlin Heidelberg, 2013

Conferences

- P. Climent-Pérez, A. Mauduit, D. N. Monekosso, and P. Remagnino. Detecting events in crowded scenes using tracklet plots. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, volume 2, pages 174–181, 2014

- P. Climent-Pérez, D. N. Monekosso, and P. Remagnino. Multi-view event detection in crowded scenes using tracklet plots. In *22nd International Conference on Pattern Recognition*, pages 4370–4375, 2014

- P. Climent-Pérez, G. Lazaridis, G. Hummel, M. Russ, D. N. Monekosso, and P. Remagnino. Telemetry-based search window correction for airborne tracking. In *International Symposium on Visual Computing*, pages 457–466, 2014

- G. Tzanidou, P. Climent-Perez, G. Hummel, M. Schmitt, P. Stutz, D. Monekosso, and P. Remagnino. Telemetry assisted frame registration and background subtraction in low-altitude UAV videos. In *Advanced Video and Signal Based*

Surveillance (AVSS), 2015 12th IEEE International Conference on, pages 1–6, Aug 2015

- P. Stütz, G. Hummel, M. Kaiser, A. Schulte, N. Theissing, P. Climent-Pérez, P. Remagnino, D. Lund, A. Magzoub, Y. Tsado, J. Gozdecki, and K. Loziak. UAV integration aspects within the PROACTIVE network. In *Proceedings of the International Conference on World of UAV (Unmanned Aerial Vehicles)*. [in press], 2015

List of Tables

List of Figures

Acronyms

8-DOF - Eight degrees of freedom

ASL - Above sea level

AUC - Area under the curve

BL, BR - Bottom left, bottom right

BTF - Bayesian tracking formulation

BoF, BoW - Bag-of-features, or words

CCTV - Closed-circuit television

DFT - Discrete Fourier transform

EMD - Earth mover's distance

FN, FP - False negatives, false positives

FOV - Field of view

GMM - Gaussian mixture model

HOG - Histograms of oriented gradients

HSV - Hue, saturation, value

IMS, IMU - Inertial magnetic sensors, or units

INS - Inertial navigation system

KF, EKF - Kalman filter, and extended Kalman filter.

KL - Kullback-Leibler

k-NN - k-nearest neighbour

LBP - Local binary patterns

LLE - Locally linear embedding

LMedS - Least median of squares

LOOCV - Leave one out cross-validation

OF - Optical flow

MCMC - Markov-chain Monte Carlo

MI - Mutual information

MSE - Mean squared error

MSSIM - Mean structural similarity

MV-TPH - Multi-view tracklet plot histogram

MVU - Maximum variance unfolding

PCA - Principal component analysis

PDF - Probability distribution function

PSNR - Peak signal-to-noise ratio

RANSAC - Random sample consensus

RFID - Radio frequency identification

RGB - Red, green, blue

ROI - Rectangle of interest

SDE - Semi-definite embedding

SLAM - Simultaneous localization and mapping

SVD - Singular value decomposition

TL, TR - Top left, top right

TP, TPH - Tracklet plot (histogram)

UAV - Uninhabited (or unmanned) aerial vehicle

UTM - Universal transverse Mercator

VTOL - Vertical take-off and landing

WGS-84 - World geodetic system 1984

Chapter 1

Introduction

1.1 Context

Video surveillance of individuals, small groups and crowds is of importance for today's societies in which the overpopulation of urban spaces is growing, and overcrowding is likely to happen more frequently. Big hubs such as airports, train stations, and underground networks, but also concert halls and big demonstrations, need vigilant supervision to avoid incidents –deliberate or otherwise– that might cause hundreds of deaths and serious injuries. As a consequence, security operators all over the world are demanding systems capable of dealing with these situations, and able to provide flagging of suspicious events and inference of advanced knowledge from, potentially multiple, video sources.

In recent years, many developed countries have seen an increase in the installation of closed-circuit television (CCTV) cameras for these purposes (i.e. public safety, asset security, crime reduction), to the point that these have become ubiquitous. However, this large amount of data is seldom processed by computer vision algorithms, but rather, used as a deterrent for offenders, and for forensics once an incident has happened. Automated solutions have been proposed in the past using single camera systems, and, to a lesser extent, with multiple fixed camera networks. Using multiple cameras is an effective way to mitigate or counter the effects of occlusions among people and objects, which are a limiting factor in single-view approaches. Furthermore,

with the recent advent and reduction in price of civilian off-the-shelf uninhabited[1] aerial vehicles (UAVs), it is possible to deploy video surveillance in remote areas where fixed cameras are not or cannot be installed.

Regarding the nature of the analysis performed by the algorithms, when dealing with video surveillance of environments where multiple people are present, analysis can be performed using different approaches, depending on the density (and other cues) of the crowd. With sparser scenarios, people can be tracked individually with a multi-target visual tracker, whereas in densely packed crowds, approaches dealing with the crowd as a whole are preferred. Therefore, different levels of crowdedness translate to different approaches: i.e. *microscopic* and *macroscopic*, respectively. At an intermediate level between macroscopic and microscopic analysis, there is *mesoscopic* analysis, that is, the use of microscopic cues (e.g. tracks from a visual target tracker), that can be used to obtain information of all the individuals forming the crowd, and thus, infer knowledge from the crowd as a whole.

With all this given context, the focus of this thesis will lie on video surveillance methods, introducing a novel crowd granularity assessment method as a first step (i.e. to select the best-performing methodology depending on the case). Once the granularity has been established, and avoiding single-view procedures, given their stated limitations, two additional approaches will be presented, using multiple fixed views, and cameras mounted on UAVs, respectively.

1.2 Motivation

In this section, the limitations and current challenges in the fields related to this thesis will briefly be presented. The reader is referred to Chapter 2 for an in-depth analysis of the related literature.

[1] Also referred to as *unmanned*.

1.2.1 Crowd granularity evaluation

As stated, it is necessary to first assess the crowd *granularity*, that is, whether the scene can be analysed by investigating each individual in the scene separately (i.e. *fine* granularity, with more detail of each individual motion), or the crowd method to be used can only rely on information of the crowd as a whole (i.e. *coarser* grain). Previous works, do not directly address crowd granularity assessment, but instead focus on evaluating the level of danger in a crowd. Additionally, these methods have been based mostly on density estimation only. Despite this, early works suggest that more than one cue would be necessary to better assess how dangerous a crowd is. Therefore, using density as the sole means for assessment seems unreasonable, since using additional cues could contribute to a better understanding of the situation. The same is applicable to the assessment of the best tools to further analyse the scene. Cues can be obtained from the analysis of the crowd as a whole, by analysing the whole video frame and determining density and entropy via specific estimators.

However, when measuring density or entropy in a video sequence, prior knowledge on the areas of the image where people can stand needs to be known, since this allows for normalisation of the values over the *possible area*. Furthermore, using two cues for assessment (density, entropy) leads to having two different scores, making linear ($1D$) ordering of different scenarios no longer possible. Merging both scores into a single figure, could be possible but would require engineering a weighting mechanism for each score. A way to overcome this is to plot the $2D$ point, given by the two scores, in a $2D$ curve, and to label points falling inside marked areas in that $2D$ space as having certain properties. Quantisation of the $2D$ space to delimit those areas could be a possibility.

1.2.2 Video surveillance from UAVs

Microscopic analysis, as said, entails the tracking of individuals. This can be done from fixed cameras or from cameras mounted on moving vehicles. Fixed CCTV camera

networks do not always reach all the regions of interest where events might occur. For this reason, cameras mounted on UAVs can be useful. Nevertheless, tracking from such cameras leads to a series of difficulties, since many of the methods for background modelling (i.e. for the segmentation of moving objects), human detection, and tracking, assume that the camera is fixed. Many existing methods counter the motion of the camera (ego-motion), however, these methods heavily rely on interest point (i.e. corner) detection and matching, although, point detectors are highly dependent on good texture of the background (i.e. the ground in this case, since the camera is pointed downwards). In cases were the terrain has poor texture (such as when the events happen on grass or tarmac surfaces), the only detected corners are those of moving objects, which causes a failure in the matching process, since the points corresponding to these objects would in other cases be detected as *outliers* by the method (thus ignored by the matching). For this reason, telemetry can be of interest for the improvement of matching, or for the elimination of matching altogether, by calculating the position of targets on the ground in the next video frame based on the transformation undergone by the vehicle. Good matching is very important: if ego-motion is corrected successfully, the cues obtained from the analysis of airborne video streams (e.g. tracks from a visual tracker), can be used in the same way as is done with fixed cameras, for knowledge inference at the group level (i.e. in a mesoscopic approach).

Despite all that, methods that rely on telemetry have a disadvantage in cases where a UAV flies within a covered area, such as below dense tree branches, or near tall buildings or similar objects that might limit the number of satellites available for geo-positioning. Also, telemetry data has a non-negligible accuracy error in the measurements, which limits its application for precise measurements required for instance in background modelling. Yet, this can be corrected using global refinement techniques and/or more precise instrumentation. As stated, telemetry-based techniques can be useful in cases where texture of the ground (background) is poor, and could potentially be used as complementary methods to well-established 'interest point'-

based techniques, since they might work well together in situations in which either of them fails (i.e. poor texture, bad GPS signal).

1.2.3 Event detection in groups from multiple views

Different types of events occurring in a large group of people can be detected using a mesoscopic approach, if using the tracks from all the individuals forming the group, and aggregating these in a scene descriptor. This is opposed to methods based on macroscopic approaches (e.g. techniques based on 'optical flow'), in which only a dominant motion can be inferred, and the problem is cast as an outlier detection, that is, only deviations from the inferred pattern are detected as abnormalities. Furthermore, only anomalies comprising the whole crowd can be detected, whereas a method tracking all individuals can detect events involving a minority or a single individual in the scene. However, existing methods are limited to single views in most cases, and therefore have poor performance with occlusions.

Extending these systems to combine information from multiple views adds a computational overhead. Besides, information fusion can be performed at different levels (decision-, model-, and feature-level), each having its own advantages and drawbacks, therefore requiring the introduction of mitigating mechanisms to overcome the drawbacks of the selected fusion level. Feature level fusion, for instance, entails creating a concatenated or averaged feature vector, yet, it requires the different information sources to be synchronised. Additionally, when the feature is concatenated, it grows linearly with the number of information sources, thus reaching high dimensionality. This can be overcome with the use of dimensionality reduction techniques.

1.3 Aim and Objectives

Having evaluated current challenges and limitations of existing techniques, and as a summary of what has been said, this thesis will cover methods for video surveillance from three different perspectives: crowd granularity assessment, individual detection

and tracking from aerial footage, and small crowd event detection combining cues from multiple views. As will be seen, all these methods are interrelated in a common theme: crowd analytics, yet involving different scene density levels (individuals, small groups, and crowds), as well as different analysis modalities (micro-, meso-, and macroscopic), from different video sources (single-view, multi-view and aerial).

Therefore, the **aim of this thesis** is to explore smart video surveillance methods from single-, aerial- and multi-view footage, and specifically to contribute to the fields of human tracking from aerial video, crowd granularity analysis, and group event detection. The following is a list of the objectives of this work:

1. **Objective 1.** Given the limitation of crowd assessment based only on density estimation, to explore how additional cues (i.e. orderliness, entropy), can help determine the granularity of a crowd, and subsequently decide on the approach to use (i.e. microscopic or macroscopic analysis).

2. **Objective 2.** Since purely video-based methods perform badly with poorly textured scenarios, to explore how telemetry data can be used for background modelling as well for improved and on-line tracking from airborne video cameras.

3. **Objective 3.** Observing the limitations of single-view analysis, to study how tracking of individuals in small-to-medium crowds from multiple views can be used to detect and classify different events and abnormalities, while alleviating the computational overhead added by multi-view fusion.

1.4 Contributions

There are three contributions to this thesis, which naturally follow from the objectives listed above. As stated, video surveillance techniques can be classified according to several *dimensions*, that is: as single-view, or multi-view approaches; using airborne, i.e. moving, or fixed cameras; and performing an analysis at different levels, i.e. micro-, meso-, or macroscopic. Of course, many combinations are possible, however, a selection

has been made including relevant (i.e. unresolved) ones based on the analysis of the existing literature. Tackling all possible combinations would be unrealistic given the time constraints of a Ph.D. programme. It is worth mentioning here that two of the contributions were inspired by the work developed by the candidate during his participation in the PROACTIVE[2] project. Specifically, these are contributions 2 and 3 in the following list showing the contributions of this thesis, with labels according to the different axes defined.

- **Contribution 1.** A novel density–entropy signature for crowd classification is introduced, that allows determining levels of danger in a crowd using other cues apart from density, i.e. by adding *orderliness* (as entropy) as a cue. This additional *entropy* cue adds more information to the assessment of the level of danger of a crowd, since it measures how orderly the crowd is. The idea behind including more cues is that a highly dense crowd can be safe as long as orderly (e.g. think of a crowded marathon in an urban setting). Yet, a very dense scenario with people walking or running in different directions might be much more unsafe.

  single-view, fixed-camera, macroscopic analysis.

- **Contribution 2.** Two telemetry-based methods for the analysis of cameras mounted on aerial vehicles are presented. As opposed to most works in the literature, which are based on properties of the texture of the terrain (i.e. the ground), using interest point detectors for matching, the proposed methods use reliable data from global positioning system and inertial magnetic sensors (GPS/IMS). The methods presented are: a background modelling technique for the detection of moving targets on the ground; and, a method for the correction of a visual tracker's search window for fast, on-line tracking of ground targets.

  single-view, airborne camera, microscopic analysis.

[2]This was a project of the seventh framework programme of the European Comission (EC FP7), finished in May 2015. It entailed, among many others, computer vision analysis, as a module in a multi-sensor fusion framework to predict and detect terrorist attacks.

- **Contribution 3.** Finally, a method for the classification of small crowd events from multiple views, using a novel scene descriptor called *tracklet plots* will be introduced. Several of these scene descriptors, one from each view, are combined into a single multi-view feature, that is then used for scene classification. Combining descriptors from several views allows the system to always perform as the best available view, which might not be known beforehand, and therefore gives advantage over single-view approaches.

  multi-view, fixed cameras, mesoscopic analysis.

Furthermore, in Fig. 1.1, the contributions are shown according to their execution time, ranging from real-time[3] to off-line methods. As it can be observed, contributions 1 and 2 are closer to real-time in performance, whereas tracklet plots are closer to 'off-line'. In the corresponding chapter, however, tracklet plots are presented as a scene descriptor with real-time capabilities, it is only that in the way it is used in that chapter, events are detected after a sequence has been seen.

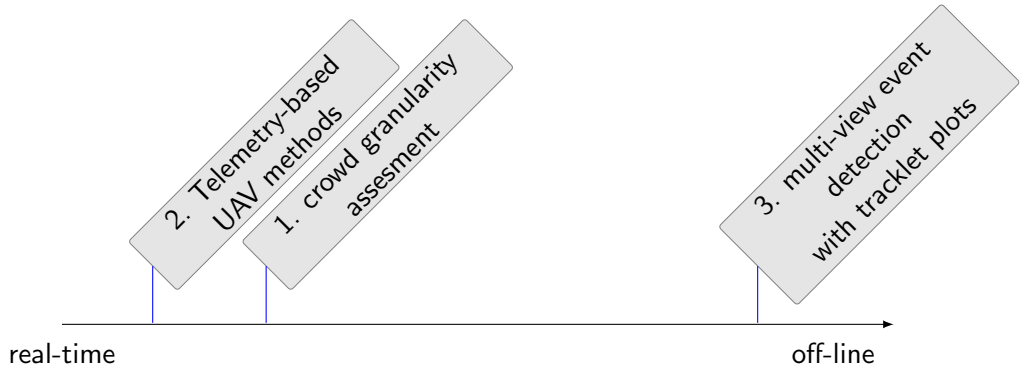


Figure 1.1: The contributions of this thesis, ordered according to their response time.

1.5 Thesis Structure

This thesis is organised as follows: Chapter 2 will introduce the state-of-the-art techniques in the several relevant fields, as well as will clarify the concepts behind the

[3]the term 'real-time' is used **in this thesis** to mean "without a significant delay" or at "interactive rate", that is, one that allows live interaction, rather than as used in real-time computing (*i.e.* systems subject to time constraints or *deadlines*, [231]).

different approaches for modelling crowd dynamics, which have already been mentioned in this chapter, i.e. microscopic and macroscopic analysis, with a special focus on the former, where topics like target tracking, occlusion handling, multi-target tracking, fusion of multiple sensors, and tracking from aerial vehicles will be included. Chapters 3 through 5 will cover each of the contributions listed in Sec. 1.4, respectively. Finally, in Chapter 6, a summary of the highlights of the proposed methods, including their advantages, disadvantages and future work will be stated. Moreover, some conclusions will be drawn, and some final remarks presented.

Chapter 2

Background

Overview

In this chapter, the reader is familiarised with the topics of research involved in this thesis, namely: crowd analytics from video sources, visual tracking of individuals, multiple target tracking, multi-view (multi-device) analysis, and tracking from aerial platforms. First, a topology is presented, to divide tasks depending on the density of the crowd: macroscopic analysis is then presented as a means to analyse crowds as a single entity, whereas microscopic analysis is shown to be better for cases in which the crowd might be a bit sparser, and involves tracking individuals separately. These levels are shown to be able to interact, and interaction among techniques at the different levels is also reviewed. A summary table of identified gaps and how these have been addressed in this thesis is also provided, as a summary.

Publications

- A book chapter [241], which was published in the book "Intelligent Multimedia Surveillance: Current Trends and Research".

2.1 Introduction

Automatic crowd analysis appears as a need to reduce costs and improve people's safety while reducing the burden of manual video analysis [40, 63]. Crowd analysis in public environments has received attention in the last decade [74], and it is of interest to a very wide range of fields, as described in [105, 282]: from the identification of anomalous behaviours to avoid crime [67], or to avoid stampedes and congestion in large events [33] or traffic sites [113]; going through the design of buildings that are easy to evacuate, or the management of public transport systems [40, 146, 147]; to the design of intelligent cars or moving robots that identify pedestrians and act consequently [8, 197, 200, 219, 242, 271]. Systems for crowd simulation [21] are also relevant for several of the mentioned fields, as it allows testing different crowd control strategies without any actual danger, and at a reduced cost.

Video analytics involves various steps present in most existing Computer Vision systems [169], as depicted by Figure 2.1: first, an optional step of background segmentation is performed; then, a set of features need to be extracted from the video (segmentation tends to ease it); following that, tracking is performed using such features; later, at a training stage, models of different behaviour are learnt from either the features or the tracks; finally, events are inferred from the input using the model induced. Even when some algorithms can be used in common, the tracking of crowds poses its own specific problems. This classical workflow is used for single camera systems, however, more sophisticated systems using multiple cameras need to deal with other issues that arise, such as the problem of trajectory association [161], or object selection throughout the capture devices [67, 69, 160, 272]; camera topology discovery is also studied [73]. There are also proposals for multi-modal fusion, that is, to complement vision (camera sensors) with other types of sensors; a review on this field is presented in [18]. Multi-modal fusion includes many kinds of sensory devices such as RFID tags and readers [94, 108, 220], thermal/infrared (IR) sensors [242], pressure mats [108], Bluetooth-enabled devices [246], etc.; furthermore, some robotic

systems also present other means of data fusion such as interaction with or supervision by humans [28, 138, 229] to enrich vision-acquired knowledge.

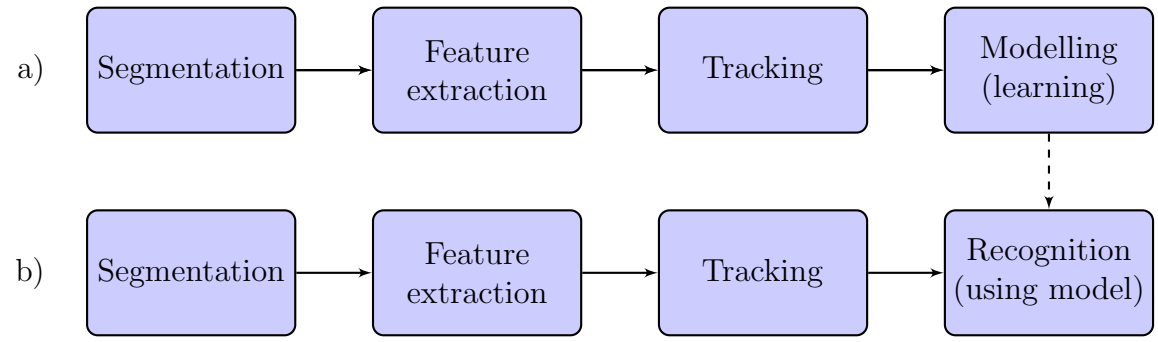


Figure 2.1: Classical workflow in video analytics: a) modelling/training step; b) recognition step.

2.2 Approaches or levels for modelling crowd dynamics

Methods for scene analysis which consider each individual in isolation, that is, via people detection, subsequent tracking, and activity analysis based on the obtained tracks, often face challenging situations due to occlusions among the pedestrians, or due to complex interactions among the members of the crowd. This is why, methods that analyse the crowd as a whole use global estimations of density, or find overall motion patterns, etc. The status of the crowd is reported as being normal or abnormal based solely on the dynamics shown by the whole crowd. However, this might not be always the case, in some situations, crowds will not show such a constrained set of motions, but instead, individuals will be able to move freely. In such cases, the analysis of single individuals, rather than the crowd as a single entity, might be able to capture richer information.

Treating these sparser scenes in the same way as denser ones will fail to identify abnormal events which only affect a single individual. For instance, a running person in a crowd can indicate an abnormal event if the rest of the crowd is

walking. Thus, considering the crowd as one entity can cause false detections in such cases, but useful as a general trend indicator.

Thus, modelling of crowd dynamics can be tackled in different ways. In spite of there being a continuum from sparse scenes of few people to crowds of individuals in mass gatherings, a discretisation of this continuum into several classes will lead to the creation of a topology or categorisation of the scenes, where various levels are defined, depending on the kind of analysis that is performed on the crowd.

One proposal to assign levels to these different approaches for crowd modelling and crowd feature extraction divides approaches into: micro-, meso- and macroscopic [282]; these are roughly equivalent to individual-, group- or crowd-level analysis. For instance, at the pedestrian level, tracking of individuals by means of local features, using Particle filter- or CONDENSATION-based, mean-shift-based, or similar approaches, is taken into account [109, 110, 115, 117, 201, 217]; while, at the crowd level, density and counting are studied [116, 158], as well as motion patterns and behaviours [67, 162, 212]. At the group level, interactions among individuals forming the crowd are studied [78]; other works use mixed approaches both for people counting and individual tracking [68]. The different approaches can be used depending on the aim of the system, and more specifically, based on the density of the crowd [65], and/or other similar features.

2.2.1 Interaction among models of different levels

These levels of analysis are not necessarily exclusive, neither they need to work in isolation [241]; that is, cues extracted using a microscopic analysis (such as individuals' tracks in a scene) can be used in upper layers of abstraction to infer knowledge about the existing groups or crowds. Feedback and feed-forward techniques are possible, thus closing the loop among different analysis levels.

As an example, this thesis will cover several of such interactions. For instance, Chapter 3 proposes a method using macroscopic analysis for the determination of the best methodology to use next (which could be microscopic approaches for cases where the crowd is sparser). In the method proposed in Chapter 5, the cues from

microscopic analysis are aggregated among individuals, and views, and used to detect events at the group level. The work in Chapter 4 in contrast, is focused purely microscopic analysis, although, the cues obtained from this analysis could be used, subsequently for further analysis at another level (as done in Chapter 5). Examples in the literature also exist, for instance: a process by which tracking of people is improved by crowd-level (macroscopic) analysis of dominant motion in a crowd (top-down approach) [112, 161]; or another where the short tracks, or *tracklets*, extracted from individuals at the microscopic level can be helpful to determine the existence of groups of people (mesoscopic) from a single view [36, 78, 79], or even help determine crowd-level general terms or anomalies (bottom-up approach to crowd analysis). Additionally, due to the size and extent of crowds, their behaviour might need to be analysed from more than one camera, since they might span through multiple views [111].

But what is a *tracklet*? How are they useful?

A tracklet is the short track of an individual (i.e. target) which has been tracked for a short time interval using a visual tracking algorithm. The idea comes from the fact that most visual trackers, perform better over short periods, since the appearance of non-rigid objects (e.g. people) deviate from the initial pattern as time goes by, regardless of recent advances. After that given period, the tracking algorithms can be restarted, so that new tracklets are obtained. More details about *tracklets* and the way they can be exploited, as well as a further review on related topics, can be found on Chapter 5, where a descriptor for crowd scenes is introduced and used to detect different group events, both from single and multiple views.

2.3 Macroscopic modelling

As said, macroscopic modelling techniques are helpful when the crowds analysed present a constrained set of motions, and the focus is on the detection of motion

abnormalities, which are understood as deviations from the *normal* behaviour. To do this, two main methods are used: the first one is the spatio-temporal gradient features, in which cuboids observing gradient/texture change are used as a feature to describe the motions of the crowd [119, 150, 154, 194]; the second one, and very widely used, is optical flow (OF), which obtains the instantaneous motion field between consecutive frames. The information obtained by optical flow can be further exploited in different ways:

- To find sinks and sources of people (and therefore determine common paths between pairs of source–sink). This is achieved by merging flow vectors along the video frames and finding its originating and ending positions [5, 6, 95, 96].

- Another way of exploiting this information is for optical flow clustering [9, 10, 207]. In [9, 10] crowd behaviour is represented by using unsupervised feature extraction on the optical flow, which applies spectral clustering to find the optimal number of models to represent a normal motion. In [207], mixtures of Gaussians are used to model the normal behaviour, instead.

- Additionally, the vectors obtained can also be used to model the interaction forces of a crowd, and then use the inferred model to determine the stability of the crowd. For instance, social force models could be used [90, 162, 165], where the motions of the pedestrians are modelled with two forces: a personal *desire* force, that determines the goal the individual would like to achieve (maybe an identified sink); and an *interaction* force, that determines the attraction or avoidance between pedestrians.

- Finally, optical flow fields can also be used in local spatio-temporal motion variation modelling, in which sample patches are collected from videos. Some of these patches *observe* a similar motion, and can be clustered accordingly [114, 118, 119, 150, 154, 268, 275]. For instance, in [275] the patches are clustered to find cluster representatives or 'visual words' in a bag-of-words approach. In this

way, any video can then be described by its bag (histogram of word appearance frequencies).

Figure 2.2 shows a summary of the different techniques presented in this section, used to exploit information of crowds extracted at the macroscopic level.

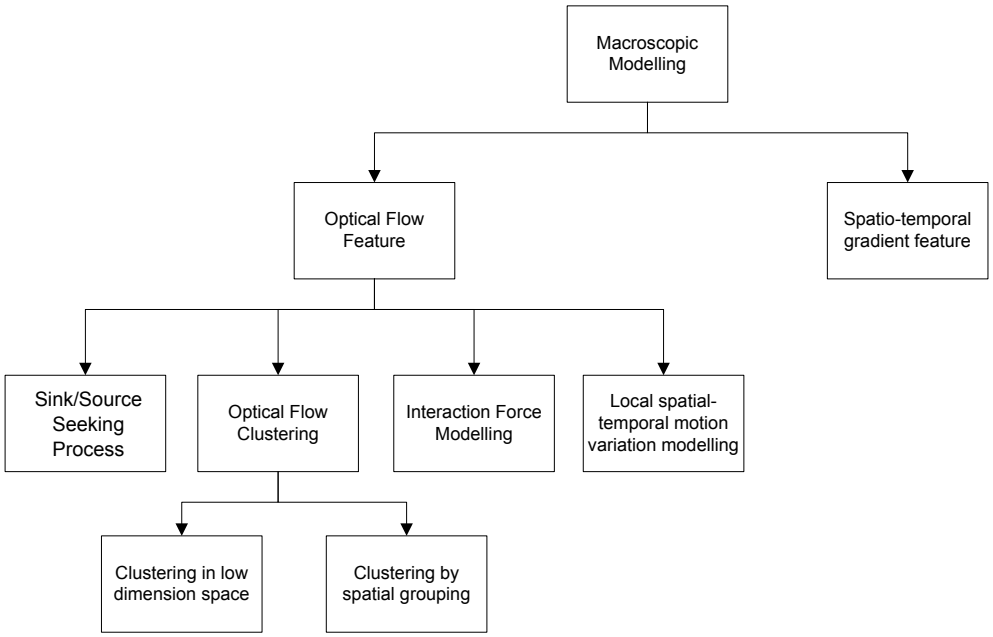


Figure 2.2: A schematic illustration of the topics involved in macroscopic crowd video analysis.

2.4 Microscopic modelling

As mentioned in [241], the microscopic analysis of crowd dynamics and its modelling rely on the analysis of video trajectories of moving entities (either cars, people, animals, etc.). This approach, in general, is performed in various phases:

- First, moving targets present in the scene are detected (using segmentation via background modelling methods, as in [238]); this could also be done by object detectors (e.g. Histograms of Oriented Gradients or HOGs, as in [223]; or the Viola-Jones detectors [248, 249]).
- Tracking of the detected targets; and

- Analysis of the trajectories to detect dominant flows, and to model typical motion patterns.

The complexity of tracking algorithms in microscopic modelling of crowd behaviour depends on the context and environment in which the tracking is performed [241]. As the density of people increases, tracking becomes more difficult: a higher density introduces additional complexity due to the interactions and occlusions between people in the crowd (subsection 2.4.2 is dedicated to this particular problem). A number of tracking methods (Sec. 2.4.1) have been proposed to overcome the challenges encountered in a crowded scene. Figure 3 shows the different topics covered by this section.

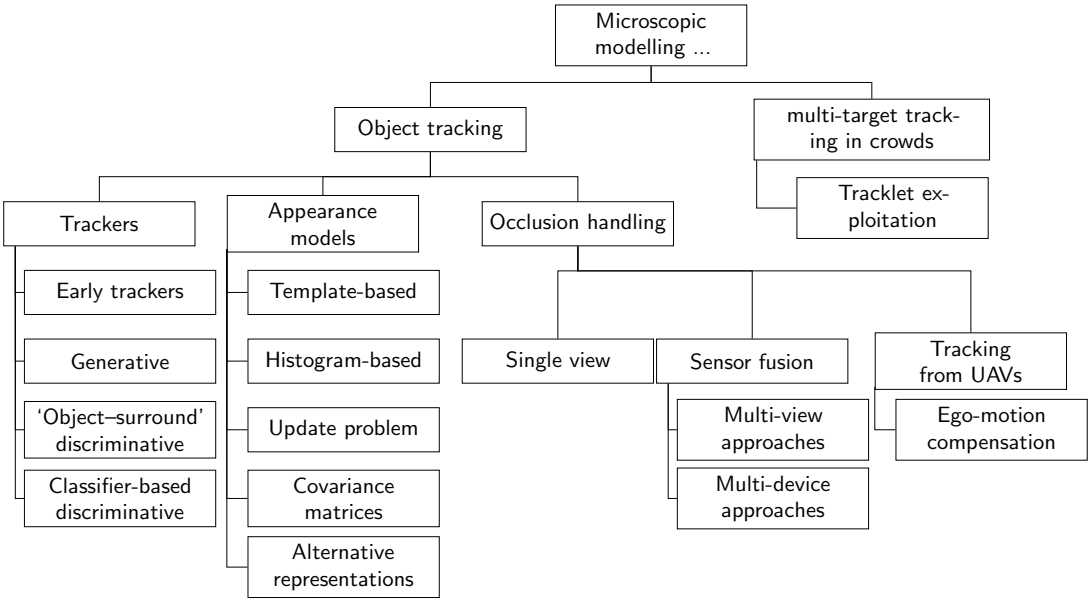


Figure 2.3: Topics covered under 'Microscopic analysis', in this section.

2.4.1 Person and object tracking

Visual tracking of moving targets (humans, cars, and others) is of great relevance for various tasks in Computer Vision. Given the rapidity at which the field of visual tracking evolves, it is necessary to introduce new concepts and advancements that have seen the light in the years past since the publication of previous reviews [274, 276]. In

the particular case of human tracking (also referred to as pedestrian tracking), the trajectories obtained can be used for further analysis at higher layers of abstraction (i.e. at a group or crowd level), which can help determine group and crowd behaviours, as reviewed in [171].

Of particular interest is the emergence and evolution of new evaluation frameworks. New benchmarks and challenges appear (or are updated) frequently. These challenges, such as the Visual Object Tracking (VOT) challenge, attract much attention, as can be observed from the growing number of participants in every edition [120–122].

As said, various previous reviews have been published on the topic. Yilmaz et al. [276] is a comprehensive review from 2006, to which the reader is referred to for detail on earlier methods. Some years later, in 2011, Yang et al. [274], published another review, in which they included up-to-date advances and trends in the field. Table 2.1 shows a summary of the different aspects analysed by these reviews, as well as this section. The earliest review in the table [276], covers a very wide spectrum of target representations (for shape and appearance modelling), which were popular at the time of publication. However, the works covered in this review, for the most part, use only one particular type of shape model, i.e. the so-called 'rectangular patch' or bounding box. Nonetheless, most of the proposed appearance models are still widely used, i.e. 'probability densities of object appearance', 'templates' and 'multi-view appearance models', which are nowadays known for its use in *discriminative* trackers, see Sec. 2.4.1.3.

Regarding [274], it introduces the concepts of *model update*, as 'online learning methods', yet, this term does not seem accurate, as the word 'learning' seems to limit these strategies to trackers that are equipped with a some sort of classifier, which is not the case. In their review, the authors present a series of feature descriptors which are used in object detection, and assert that those could easily be used for tracking. Nevertheless, this seems to ignore the main difference between those two fields, namely, that in object detection *intra-class* object differences are not important (i.e. a person detector should detect all people in the scene), but rather, they focus on *inter-class*

Review	Aspects analysed
Yilmaz et al. 2006 [276]	1) Object representations 2) Image features for appearance modelling 3) Object detection 4) Categorisation of existing methods into: – Point tracking – Kernel tracking – Silhouette tracking
Yang et al. 2011 [274]	1) Feature descriptors 2) Online learning methods 3) Context exploitation 4) Monte Carlo sampling
This review (Sec. 2.4.1)	1) Historical analysis *(see Table 2.2)* 2) Classification according to: – Motion models – Appearance models – Update strategies – Detection and response combination 3) Tracking evaluation standards

Table 2.1: Recent reviews on the field with aspects analysed

differences (i.e. a person detector should not detect vehicles as people). This is very different to the case of a visual tracker, where not only an object class is sought, but a particular instance of that class, i.e. the target needs to be found, and differentiated, at every frame. Their work also analyses context-aware methods, as well as different motion models, under the 'Monte Carlo sampling' epigraph.

The aim of the review in this section (Sec. 2.4.1) is to introduce recent advances with visual trackers and their model update strategies, since the publication of [276] in 2006, but with a special focus on the period 2011–present, which is not covered by [274]. The structure followed is derived from the historical challenges faced by visual tracking, which are introduced (almost) chronologically in Table 2.2. In that same table, a series of methods proposed to overcome the limitations are presented, with references to relevant papers and sections of this review.

2.4.1.1 Early trackers

Initially, the challenge of tracking a target through a video can be seen as a problem where one tries to find a given pattern at every single frame ("object detection", or re-identification approach to tracking, also known as "tracking-by-detection") [11]. That is, given the image patch of the target to track, and using a method such as correlation or a variation of it, the target can be found in the next frame.

From patterns to histograms: mean-shift

The method presented above is very limited, since pose variations and out-of-plane rotations yield target appearances that are very dissimilar to the pattern learnt. Pose variations are due to the fact that targets are not always rigid (e.g. cars, planes), but have a dynamic motion pattern, that is, they have moving limbs or parts (e.g. pedestrians, animals, helicopters). Out-of-plane rotations are those that do not happen on the 2D plane of the image, which can cause a drastic change in the appearance of the target. To some extent, this can be alleviated if the target is represented as a histogram or other distribution, and tracking is understood as a task where the

Year	Limitations/challenges found	Methods proposed to solve or alleviate the limitations
ca. 2002	Pattern search too limited	Use of histograms, introduction of mean-shift (Sec. 2.4.1.1)
ca. 2003	Inefficient state-space search	Condensation, Particle Filter, Monte Carlo Sampling (Sec. 2.4.1.1)
2004–2005	The template update problem [159]	Early model update approaches [190, 244] (Sec. 2.4.1.2)
–	The plasticity–stability dilemma [83]	improved methods: generative and discriminative (Sec. 2.4.1.3)
2005	Need for Object–surround discrimination	Augmented Variance ratio (AVR) [58]
2006	Multi-feature histograms too high-dimensional	Region covariance descriptor [190, 244]
2008–2010	Using a single positive example (discrim. trackers)	Use of n positive examples [82], multiple instance learning [22] (Sec. 2.4.1.3)
2006, 2010	Use of single models for the whole target	Parts- and patch-based modelling (Sec. 2.4.1.4)
2010	Use of a single and/or monolithic algorithm	Collaboration and tracker decomposition (Sec. 2.4.1.5)
2011–2012	Pixel-wise matching / effective image representations	Super-pixel matching (Sec. 2.4.1.6)
2013–2014	Smoothness assumption too limiting	Abrupt motion models and trackers [126, 140] (Sec. 2.4.1.7)
2015	Limitations of model-based discriminative trackers	Colour-based model-free tracking [191], somehow similar to [58] (Sec. 2.4.1.6)
2010–present	Limitations of discriminative trackers (drifting)	Sparse signal representation, correlation, deep learning (Sec. 2.4.1.6)
2010–present	Standard tracking evaluation criteria	Several standardisation efforts (Sec. 2.4.1.8).

Table 2.2: Identified challenges in visual tracking

next state is the one with the most similar distribution to the one learnt at the initial moment (to do this, a probability map, based on the back-projection of the distribution is constructed). Many algorithms have been used for this, but the most used one is mean-shift [59] or cam-shift (which is an adaptation that can deal with scale changes and rotation) [7, 15, 52, 128, 145, 179, 234]. In these approaches, a given search window around the current position of the target is defined. By sliding a sample window the size of the target over that search window, and evaluating the distance of that sample's histogram to that of the pattern learnt in the first frame, a probability distribution function (PDF) can be created. Mean-shift is a gradient-descent (or ascent, depending on the metric used) that finds the mode of the distribution using first-order moments, and therefore, using the PDF can estimate the new position in the window. Additionally, once the mean-shift process is finished (to find the location of the target), an additional step can estimate the size and angle of the target, based on second-order moments (this would then be cam-shift).

More efficient state-space exploitation

Despite all this, mean-shift and cam-shift do not integrate any means to estimate the new location, that is, they do not use any information of the movement of the target in order to better search the state space (i.e. samples are equally taken from all around the target). Furthermore, as the histograms used to represent the target get more complex (i.e. have more dimensions), generating the back-projection maps needed for mean-shift is rendered impractical.

In order to exploit the target's motion pattern, the Kalman Filter (KF) can be integrated into trackers, specially in the case of tracking-by-detection, with data association [168, 240]. Kalman filters have existed for a long time, and are useful in many fields [48]. The basic version of KF is limited to a linear model, and can find the exact conditional probability estimate when all errors are assumed to be Gaussian-distributed. The latter assumption should not be a problem in most cases, but the former assumption on linearity is problematic, as many situations do not necessarily

fall into this case. To overcome this limitation, more sophisticated versions appeared, such as the Extended Kalman Filter (EKF) that projects the non-linear system into a linear system by the use of a kernel. The EKF, however, has other limitations [16], and therefore techniques were introduced which obtain the likelihood for a subset (a number of samples) of the whole search space: CONDENSATION [104], also known –with some variations– as sequential Monte Carlo (SMC) or particle filtering (PF) [29, 100, 134, 177, 180, 188, 236, 251, 261, 267] and extensions to it [97, 288]. There are also a number of trackers that are mixtures or fusions of those tracking algorithms; such as cam-shift-based solutions that rely on or use particle filtering to some extent [209, 257, 277]; or those based on particle filtering that use principles of mean- or cam-shift [23, 136, 153, 236].

2.4.1.2 The appearance model update problem

Some years after these algorithms were first introduced, the main drawback of non-adaptive trackers was made evident: as the appearance of the tracked target changed over time, a single initialization seed was shown to be insufficient, since most trackers would lose track after a while. This problem is referred to as "the template update problem" [159], and is dealt with by "adaptive" trackers, that can keep an updated model.

An early example of such an adaptive tracker is shown in [190, 244]. In these works, the authors state that most histogram-based approaches are either colour-based or based on gradient information of some sort. Nonetheless, trying to integrate all this information into a single multi-dimensional histogram can be impracticable due to its high dimensionality. An alternative approach is to have separate histograms for different features, but having n 1-dimensional histograms does not take full advantage of all available information (a single n-dimensional histogram would be much richer in terms of descriptive power). Furthermore, using separate single-feature histograms would require some sort of fusion at a later stage [129, 230]. Porikli et al. [190] propose a tracker in which covariance matrices are used as the descriptor to represent the

target's appearance, the descriptor itself is presented in [244]. To obtain it, they create image tensors, in which each pixel has not only colour information but other useful information of additional features, such as gradient and intensity information. Matrices representing the covariance of these pixel-vector values over a given region of interest are constructed to represent the target. The construction of these matrices, takes advantage of integral image structures (integral tensors, in this case) that make it possible to calculate the mean and covariance values much more rapidly. Yet, this descriptor has a main disadvantage: covariance matrices do not lie on the Euclidean space. This has two major negative consequences: it requires a special distance metric; and model update based on a running mean of the appearance requires complex mathematical operations. Regarding the first, trackers require a distance metric to determine the best proposed state based on the known target model. Distance metrics for histograms and PDFs derived from them have been studied in detail, as is shown in [42, 205]. This is not true for covariance matrices, that due to their mathematical properties (i.e. they are symmetric positive definite), require special distance metrics, as they have Lie group structure. The proposed distance metric requires the calculation of the generalised eigenvalues, which can be slow. Regarding the second, the model update strategy proposed by the authors is based on calculating the *intrinsic mean* covariance matrix that needs be based on Riemannian geometry, since as occurs with distance metrics, a mean based on Euclidean distance cannot be used for covariance matrices, unless the mean is calculated using all previous image patch values directly, but this requires that the patches are kept, and assumed to be of the same size (no changes in scale) and equally influential in the mean, which is not convenient or practical. As a consequence, the proposed model update strategy based on Lie algebra requires the calculation of matrix exponentials, which slows the process down. Both of these problems are addressed by approximate distance calculations [50], or by a model update strategy based on incremental covariance tensor learning [269].

Nevertheless, the approach taken in [190, 244] where the appearance model is updated at every frame based on the new state of the tracker, has its own disadvantages:

first, when the template is found on the new frame, small errors are introduced (i.e. the new image does not match the original template exactly, and the estimated location might be off by some pixels), these small errors can accumulate over time, leading to *drifting* of the tracker; second, bounding boxes can contain pixels which are part of the background, and therefore, when updating the model, if the portion of background is more prominent than that of the foreground, the newly trained model can make the tracker *drift* towards the background. This second problem can be solved if it is known which pixels are part of the target, since update could then be applied only on selected pixels.

An example of a more advanced approach to alleviate the model update problem is shown in [203], where the authors introduce the concept of *incremental learning*. Under this paradigm, taking advantage of the similarity in appearance among the samples obtained from the tracked target, a low-dimensional subspace representation is kept. In this case, the appearance is modelled as an *eigenbasis*, that is, the eigenvectors U of the sample covariance matrix (that is a matrix representing the covariance of the samples to the mean sample). Equivalently U can be obtained from the singular value decomposition (SVD) of a matrix with columns equal to each sample minus their mean. Therefore, adapting to new samples is equivalent to re-training the eigenbasis with additional images. The authors also present an efficient way of re-training U on-line [139], as opposed to classical off-line methods. Their model is integrated into a particle filter sampling method. Each particle (image patch) is assigned a probability of being some variation of the learnt representation. At every step, new appearance samples are obtained, and used to update the model. Older samples are given less weight in the model, therefore allowing the method to adapt faster to changes in appearance.

Despite that, the approach used in [203] can also be problematic, since the rate at which new samples are incorporated into the model, and the portion of the initial model that is kept (if any), is important. This other problem is known as the "plasticity–stability dilemma" [83], as mentioned in [211]. When these two variables are not well

balanced, an appearance that has not been seen for long could have been removed from the tracker, leading to poor results. It has to be noted, that the appearance is in many cases periodic or cyclic (e.g. as is the walking pattern of a person), and therefore, a good update strategy should keep appearances at different time scales. Finally, a good update strategy should also take occlusions into account. If unaware of them, the model might be updated with images of the objects (or people) occluding the target.

2.4.1.3 Discriminative trackers

With the two challenges presented in the previous subsection (2.4.1.2) in mind (namely the "template update problem" and the "plasticity–stability dilemma"), several trackers have been devised. Some of them are *generative*, since they update the model or pattern with new instances (or examples) of the target, while another type of them are called *discriminative* as they also keep a model of negative examples (background patches), and therefore, tracking is interpreted as a classification problem, as opposed to a simple object detection as was the case for previously introduced approaches.

Based on the premise that "features that best discriminate between object and background are also the best for tracking an object", Collins et al. [58] select the most discriminative features to track based on an evaluation of the "augmented variance ratio" (AVR), particularly the VR of the log likelihood ratio between the distributions of the foreground (target) and the background, which are sampled from the target's most recent location and a ring around it, respectively. A bank of different colour features (linear combinations of the RGB channels) are evaluated in this fashion, and then ranked. mean-shift tracking is applied to each of the back-projections generated by each feature, and then the median is employed to combine the trackers' local estimates into a final global estimate. This work is cited as being one of the first attempts to treat tracking as a binary classification problem, yet it does not use a classifier for the task.

In contrast, Avidan [20] presents an "ensemble tracking" algorithm, where weak

classifiers are trained on the feature space to distinguish target and background pixels. The weak classifiers are combined using AdaBoost, and the resulting strong classifier then finds the target on the next frame. Finally, the weak classifiers are evaluated (ranked), and the best performing ones are kept, while the rest to complete the maximum number of weak classifiers is filled with newly trained classifiers. This allows the tracker to be adaptive to appearance changes, via keeping multiple classifiers, and introducing new ones that could potentially be better at distinguishing the target from the background.

Furthermore, in the "co-tracking" algorithm [237] an on-line support vector machine (SVM) is built for each feature (e.g. RGB histogram, and histogram of oriented gradients –HOG–). Each then generates a confidence map, which is combined via a weighting system based on the classification error of each classifier. The target is located by finding the global maximum on the combined confidence map, the authors state this is more general, as it does not introduce a spatial constraint as a gradient ascent algorithm (e.g. mean-shift or similar) would. As the most notable novelty, this method uses an update strategy, where "co-training" is used. The process starts with new samples being extracted from the processed frame. To find the most significant negative examples, the highest peaks from the confidence maps that do not overlap with the new target location (state). To avoid bias towards the negative examples, the new positive example is given a weight equivalent to the sum of the weights of negative examples added. Then, the samples generated (extracted) from one feature confidence map, are fed into the classifier of the other feature. The rationale behind this is that the classifier passing the samples to the other classifier is finding those difficult to classify correctly, and chances are, the receiving classifier might be able to do a better job given those new samples in the future. A given classifier will not perform better on the negative samples in the future, since no local feature extraction parameters change from frame to frame, but the other classifier might be able to improve based on these new samples. However, for the initialisation phase of the algorithm, another tracker needs to be used, since several samples are needed to start, and it would be impossible

to collect them from a single frame. Furthermore, for the update of the model, a threshold needs to be manually tuned based on confidence of the target detection.

Yu et al. [278], also use co-training, but in this instance, they use a hybrid approach, that is, a generative model (based on intensity patterns) represents the global target appearance, by keeping all appearance variations that have been observed, compactly. It is known that such variations lie on a low-dimensional manifold, which might be globally non-linear, but local appearance variations might still be approximated as a linear subspace. The more samples are collected, the higher the descriptive power of the generative model. Yet, the discriminative classifier (based on an on-line version of SVM, using HOG as a feature) cannot deal with large amounts of new samples, as this would lead to too many support vectors, and very slow performance. Instead, a temporal sliding window is used, which bounds the number of samples used, to focus on recently observed appearances. Additionally, this method does not use mean-shift or any other means of restricting the motion to a "smooth motion assumption", and therefore can be used for reacquisition after full occlusions of the target: since the method employs a Bayesian formulation of the tracking problem, the covariance matrix that defines the area to search can be increased or decreased based on the confidence (acceptance or rejection) of each individual model.

Discriminative with multiple positive examples

Basing the update on one positive example only as done in, for instance Tang et al. [237], presented above, can also lead to drift, specially since this single positive example is weighted as much as the many negative samples collected. That is putting too much confidence on a single positive example, which as said earlier, might have small errors in the estimation that add up with time [159]. To overcome this, Grabner et al. [82] propose a tracking method that is based on semi-supervised on-line boosting [130]. The difference to previous works is that, instead of adding new samples to the on-line classifier as either positive or negative from the tracker's results (normally a positive sample drawn from the new state, and negative samples from the surroundings), they

add new samples that do not have 'rigid/hard' labels assigned, but give them 'soft' labels by using a prior classifier, and on-line semi-boosting. With the same goal, tracking via online multiple instance learning (MIL) is introduced in [22] to avoid the problem of basing the update on one positive example only, as well as would do using multiple positive examples around the new state which would confuse the tracker and decrease discriminative power. This seems logical for two reasons: firstly, since more positive examples are accepted, several maxima could be found in the next frame, leading to a 'flatter peak' in the response; secondly, since background pixels that these examples include could be modelled as part of the target, as there is a reinforcement of this by several positive examples containing that same background information. Instead, MIL is specifically designed to deal with such problems by not learning on single samples, but instead bags of them. These bags of samples are then labelled as either positive or negative, but not the individual samples contained. With this method, the authors claim to achieve better performance, compared to [1, 82], among others.

Nevertheless, in discriminative trackers, the classifier, which yields the label prediction (positive or negative) and *actual* objective of the tracker (accurate localisation) are decoupled. The classifier is trained only on binary labels (regardless of whether there is a single or multiple positive labels) and has no information about transformations (i.e. translation or scale changes undergone by the target). Hare et al. [87] propose to incorporate location information, by learning a prediction function that directly estimates the object transformation between frames. The discriminant function used includes the transformation explicitly, meaning it can be incorporated into the learning algorithm. The discriminant function measures the compatibility between sample and translation pairs, and gives a high score to those which are well matched. A series of SVMs are used for classification, and a budget is used in order to cap the number of SVMs that are maintained, thus eliminating less discriminative ones, and adjusting the weights of the rest to counter the negative effects of the removal.

2.4.1.4 Parts-based and patch-based modelling

Up to this point, the presented trackers work on representations of the target as a whole; some other trackers [1, 115, 273], use several models based on parts of the target, thus creating "sub-models" that fixate on specific features of the target (e.g. these could be salient or discriminative). As a first example, Adam et al. [1] propose a "fragments-based" tracker, in which a target is represented by multiple image fragments or patches. The patches are arbitrary in contrast to parts-based trackers or object detectors, which are based on assumptions and pre-defined knowledge about the targets (i.e. they are based on the detection of limbs and torso for humans, and other *engineered* structures with spatial constraints). More exactly, the patches they obtain are based on non-overlapping grids of patches. As opposed to equally sampling using such a structured approach, Klein et al. [115] present a classifier-based approach that trains (weak) threshold classifiers on randomly spatially-distributed Haar-like centre-surround features which are boosted to select and combine the most discriminative ones into a strong classifier, using AdaBoost. The confidence of the final classifier is converted into a likelihood function of the target state that is then used as the observation model within a CONDENSATION-based tracker, with a motion model based on first-order auto-regression (as used by [188]). As opposed to previous discriminative approaches, the classifier is not used to sample positive and negative samples of the target (as a whole), but instead, an ensemble of classifiers is used as the observation model. Each weak classifier focuses on a small portion of the target, and somehow *specialises* on recognising some particular feature of the target. The boosting technique then selects those weak classifiers (assigns them a higher weight), that perform better, that is, that are specialised on a very salient (distinctive) characteristic of the target. Each candidate in the particle filter updates its classifier (i.e. its observation model), based on the new state (positive example), and the remainder of the frame (negative examples). Similarly, Yang et al. [273] use a bag-of-features (BoF) model with two codebooks (one per feature: for RGB, and local binary pattern –LBP– feature vectors, respectively), obtained from samples that are randomly picked from within the defined

rectangle where the target is. Yet, in order to have enough samples to train the BoF, they need to run another tracker for a few frames (five in their experiments). This is similar to what was done by the "co-tracking" algorithm [237], mentioned above. In each frame, N image patches are collected. An RGB histogram and an LBP descriptor are extracted for each. Then all these features are gathered into clusters, and cluster representatives (centres) are obtained to form the codebooks. After that, training images can be represented by bags (occurrence frequency histograms of each codeword). As a new frame arrives, using a particle filter approach, T candidate targets are picked, and from each, N patches are extracted, the closest codeword is found (Euclidean distance), and a new bag is created which is compared to existing (trained) bags. The distance to the closest trained bag is found (by Chi-square test). A "patch similarity measure" is also obtained, which is based on the distances of the patches to the cluster centres. This process is done for both codebooks, therefore two bag similarities and two patch similarities are found. In the final similarity for one candidate, bag similarities are used to weight each feature's patch similarities. The k-means algorithm is updated with the best patches collected over a given number of frames, thus updating the observation model of the particle.

2.4.1.5 Decomposition and collaboration

Following to the co-training and co-tracking ideas, other works have explored the collaboration among different tracking modalities. An example of this is PROST [211] which is a method where three different trackers are run in parallel, and interact among them to achieve better overall performance. The selected trackers act on different temporal scales, that is the information they use updates differently. For instance, the first of the trackers is a mean-shift based optical flow (FLOW), which is considered the most dynamic: it does not remember any previous information, and therefore relies on new information on every frame. In the mid-range, there is an adaptive tracker based on on-line random forests (ORF). The reason for using ORF is that, as opposed to boosting, used in several works presented so far [20, 82, 115], it is

much less sensitive to noise in the labelling of the data, which happens when using rectangles to initialise positive examples. Finally, the static (as in temporally invariant) tracker is a normalised cross-correlation (NCC) tracker, that is a simple "template" finding via correlation, which fails when the appearance of the target changes. Tracker combination is achieved via a fall-back cascade: the optical-flow based tracker is the main tracker, since it can easily lose the target, it can be overruled by ORF. Finally, NCC is used to avoid the ORF tracker to update too often (and on wrong instances). They apply simple rules to know when a tracker should take over: 1) FLOW is overruled by ORF if they do not overlap, and ORF has a confidence above a certain threshold; 2) ORF will only be updated when its proposed new state overlaps with that of either FLOW or NCC (this avoids model updates when occlusions occur, for instance). In similar terms, visual tracking decomposition (VTD, [125]) uses several appearance and motion models, having $r \times s$ trackers (for r motion and s appearance models), that are then integrated into a compound tracker using interactive Markov chain Monte Carlo (IMCMC) framework. In this algorithm, the basic trackers communicate with one another, implicitly helping calculate the weight of each other, in order to improve the overall performance, as achieved by boosting.

Another approach where tracking is "decomposed" and its components separated, while keeping the interaction among them is tracking-learning-detection (TLD, [107]) where the authors' main goal is to achieve long-term tracking. To achieve it, the problem is decomposed into *tracking*, *learning* and *detection* as separate components of an interactive framework. The *tracker*'s sole purpose is to follow the target from frame to frame, based on an optical-flow-like method. The *detector* localises all the appearances observed so far, and uses its knowledge to correct the tracker's decisions. The *learning* component finds out when the detector fails, and uses two "experts" that focus on false negatives (missed detections) and false positives (false alarms), respectively, so that this valuable information can be added to the detector, and better estimates can be obtained in the future. As it can be seen, this is somehow similar to PROST, in that it uses an OF-based tracker, that can be corrected by the detector

(similar to the NCC and the ORF tracker in the case of PROST), but with the novelty of the positive and negative (P/N) experts that can improve the detector.

In contrast to tracking algorithms like PROST, which avoid updates based on certain criteria to avoid contamination of the model, Zhang et al. [284] propose a multi-expert tracking framework, where a discriminative tracker and its instances in past frames (referred to as snapshots) constitute an expert ensemble. The best expert is selected based on a minimum loss criterion to restore the tracker in case of disagreement among the experts (due to an occlusion, and the introduction of bad updates to the model at any given time). Traditional loss functions rely on supervised learning environments, but the authors overcome this limitation by introducing an optimization function that is regularized by entropy, as the criterion for expert selection.

2.4.1.6 Alternative representations

Superpixel matching

There are other approaches, that, similarly to, for instance, PROST, keep both a rigid and deformable model of the target. An example is the locally orderless tracking (LOT, [183]), based on a novel matching: the locally orderless matching (LOM), which is a probabilistic interpretation of the earth mover's distance (EMD) [205]. This matching, expresses the likelihood of a given patch (P) being a noisy replica of patch Q. The reason behind using this method is that, it can adapt well to both rigid and deformable (non-rigid) targets, since it keeps the spatial information as in template matching (valid for rigid targets), and in case the target is deformable, it can act as a histogram matching, given the properties of the matching used. The authors report better results on a number of well-known (i.e. benchmark) video sequences used by previous methods, as compared to IVT, MIL, VTD and OAB (on-line AdaBoost, based on a previous version of [82]). However, since it uses superpixels, even using a rapid implementation such as TurboPixels [131], it takes approximately 5 seconds for a full-resolution image, and the authors report 1 second per frame for a bounding box of 50×50 pixels. Also, on-line parameter update is based on the distance from the

current signature to the initial signature, which does not seem to be updated, which could lead to non-adaptivity when the appearance changes significantly.

Instead of using superpixels for reducing the computational cost of pixel-wise matching, as in [183] above, Wang et al. [256] start on the premise that there is a lack of effective image representations that account for appearance variations. The authors state that existing trackers use either high-level appearance structures, or low-level cues. As opposed to those, they propose the use of superpixels [199] as a "mid-level" cue which can capture structural information. The tracking task is then formulated by computing a target-background confidence map, and obtaining the best candidate by maximum posterior estimate. On-line update is achieved by retraining the model on a set of retained frames; the process is carried out every W number of frames. Furthermore, the authors also introduce a means to detect occlusions, so that update can behave accordingly during those frames. The occlusion detection relies on the retained frames used for the update process.

Sparse signal representation

Sparse signal representation [155, 264] is based on the idea that signals such as audio and images can have sparse representations based on transformations (i.e. Fourier, wavelet, curvelets, and concatenations thereof). In computer vision, rather than being able to recover a high-fidelity image from a sparse representation, the idea is to be able to use these representations as a summary of the semantics of the image or patch. Images are very high dimensional, yet, in many applications, images that belong to the same class lie close together within a manifold.

If a collection of representative samples is found for the distribution, a typical sample should be expected to have a very sparse representation with respect to such a (possibly learned) "basis" (or prototypes). That is, each signal is approximated by a sparse linear combination of prototypes called dictionary elements, resulting in simple and compact models [156]. Still, choosing the basis for the representation of the data becomes a crucial challenge to solve, in order to apply sparse representation

successfully, as extensively covered in [155].

Motivated by [264], in [163] tracking is cast as finding a sparse approximation in a template subspace. A target candidate is represented as a linear combination of the template set composed of both target templates (from previous frames) and so called "trivial templates". The sparse representation is achieved by solving an ℓ_1-regularised least squares problem. Nonetheless, as mentioned, solving this problem is slow, and therefore in [164], they present the bounded particle re-sampling ℓ_1 Tracker (BPR-L1) which uses a modified particle filter (PF) algorithm, in which particles are evaluated via a "minimum error bound" (MEB). The whole idea relies in the fact that the reconstruction error from the target templates in ℓ_2 norm is bounded by a minimum error that can be calculated much faster than solving an ℓ_1 minimisation function. In the proposed PF algorithm, there is a two-stage re-sampling step. To avoid expensive ℓ_1 minimisation calculation on all samples, the much faster MEB allows to re-order the list of samples in the first step of the re-sampling method. Then, the ℓ_1-minimisation is calculated only for a subset of the samples.

Furthermore, the work presents an occlusion detection module. In this case occlusions are detected by using the "trivial coefficients", which indicate pixel-wise contamination (i.e. occlusion) in a given sample. Therefore, samples showing contamination are not used to update the model, thus maintaining a clean set of samples.

Liu et al. [142–144] have also researched in the field of sparse representation tracking. In [142], they propose performing an on-line feature selection. A Bayesian framework with joint optimisation is presented. The method uses minimum error reconstruction while selecting those features with better discriminative power. That is, it performs feature selection on the feature vectors, the final result being the one that minimises *both* the reconstruction error and the proper binary selection of features. Their method outperforms the compared methods, which are: the ℓ_1 tracker [163], MIL [22], and IVT [203]. In [143, 144], they present SPT, a sparse representation tracker, based on their previous work. The dictionary (i.e. the basis set) is learnt only once, and not updated, to avoid drifting while keeping the flexibility. Yet, the sparse

coding histogram is updated on-line (i.e. how the target is described using the fixed dictionary). In order to find the best candidate, a reconstruction error regularised mean-shift algorithm is also introduced, in order to find a more accurate position estimate of the target.

Correlation-based trackers

Most modern trackers use a discriminative classifier, typically trained with sample patches that have been translated and scaled. Information in that type of sample sets is very redundant. In opposition to that approach, Henriques et al. [91] propose to use an analytic model that can consider thousands of translated patches. The authors prove that the resulting data matrix is circular, and therefore can be diagonalised with the Discrete Fourier Transform (DFT), reducing the storage and computation requirements by several orders of magnitude. For linear regression, the formulation is equivalent to a correlation filter which is very fast. For kernel regression, they derive a new kernelised correlation filter (KCF), that has the exact same complexity as its linear counterpart, they also propose a dual correlation filter (DCF). Both outperform Struck [87] and TLD [107] on a 50 videos benchmark, running at hundreds of frames per second, with a very simple implementation.

Danelljan et al. [62] criticise, on the other hand, the lack of good scale estimation mechanisms in existing trackers, and propose a learning correlation filter based on a scale pyramid representation. Their method is a joint translation-scale tracking based on learning 3-dimensional scale space correlation filter, and outperforms exhaustive search on the whole space.

Zhang et al. [285], however, focus on the importance of context learning, as the surrounding of an occluded object remains almost unchanged and can be exploited for improved tracking. A spatial context model is learnt based on the spatial correlations between the object and its surround. Tracking in the next frame is formulated by computing a confidence map as a convolution problem that integrates the dense spatio-temporal context information. The authors state that it has the “merits of

both generative and discriminative": discriminative because it includes not only the target but also the immediate background, and generative because both are treated as a single model.

Back to colour-based modelling

As opposed to the trend in many discriminative trackers to ignore colour information and rely solely on greyscale images, Liang et al. [137] present a benchmark in which they encode 10 chromatic models into 16 state-of-the-art colour-based trackers. Their results clearly show the benefit of colour encoding for visual tracking. They further analyse the different tracker and chromatic-model pairs, the degree of difficulty of some sequences, as well as how the performance can be impaired to different extents depending on the challenges present in the video sequences.

An example of this is the work by Possegger et al. [191], in which the authors, contrary to the shift towards classifier-based methods (many presented so far), follow the original idea of generative trackers, but instead suggest the use of better appearance models and/or better object–surround and object–distractor discrimination (similar to [285], above). They advocate for the return to colour-based and model-free tracking. That is, to provide mechanisms that can overcome the problems of generative trackers (e.g. drifting), as opposed to substituting them for *pure* discriminative, classifier-based ones. They argue that trackers based on standard colour representations can still achieve state-of-the-art performance. To avoid the drifting problem, they propose two models for the target: an object–surrounding model; as well as an object–distractors model (so that drifting towards objects, or subjects, showing similar appearance can be controlled and reduced). It is worth noting that this is similar to the likelihood ratios presented in Collins et al. [58], but using a Bayesian formulation instead. Somehow, this work closes a loop, in that it goes back to the origin and formulates a solution that is more similar to an early approach, instead of following the same trend as contemporary solutions, while breaking away from the idea that discriminative, classifier-based approaches will perform always better. The provided results confirm this, since the

authors are capable of providing the best results for the visual object tracking (VOT) challenges of both 2013 and 2014.

Deep learning for feature selection

Inspired by advances in deep learning, Wang et al. [253] propose learning a deep compact image representation. That is, deep learning is used to automatically find and select the features in the image that are most representative of the target. To this end, they employ offline training over a large dataset of auxiliary natural images, followed by knowledge transfer to the online tracking process. A later work by the same authors [254] proposed to use a better dataset for the auxiliary images, since the dataset used in the previous work was formed of full images, rather than object-related patches. Following a similar idea, Li et al. [133] present an online tracking algorithm using a single CNN for learning feature representations of the target over time. Similarly to Struck, higher performance is achieved by 'structurising' the binary classifier, using a loss function that employs the 'structural loss', including information of the target localisation (transformation) as well. Simple cues are used as additional 'channels' that can be used to train the lowest layers of the CNN and then are, in a higher layer, combined with one another.

Instead of treating CNN as a black-box feature extractor, Wang et al. [252] analyse the properties for CNN features offline after training with massive image datasets. As a consequence, they propose to exploit layers for different objectives. Top layers are better as a category detector, lower layers more discriminative and can separate target from distractors better. Using the top layers, a general detector can be built, which detects both the target and similar distractors. Using the lower layer, a specific target detector can be built. A distractor detection scheme decides on the heat map to be used at each frame. Additionally, they also found that for tracking a target, only a subset of neurons are relevant. A feature map selection method is developed to remove noisy and irrelevant feature maps. Similarly, Ma et al. [149] also identify the differences between earlier and later layers in the neural network. For this reason

they criticise earlier works by the authors of [133, 253], since the algorithms presented by those works draw positive and negative samples online to incrementally learn a classifier over features extracted from a CNN. This presents two issues: the use as of a CNN as an online classifier, as is done for object recognition, using output from last layer, is not accurate when intra-class variations have to be taken into account or when precise localisation is needed (as is the case for tracking); the second issue, is that extracting enough training samples is impossible online, as done in visual tracking. Therefore, the proposed solution is to Learn an "adaptive correlation filter", as in [91], over the features extracted from each CNN layer and use these multi-level correlation response maps to collaboratively infer target location.

2.4.1.7 Motion modelling: smooth versus abrupt

Recent works have started to focus on the problem of abrupt scale change and abrupt motion of the targets [126, 140]. A first approach would be a simple loosening of some constraints to search for the tracked target in the vicinity of the target in the previous frame, but this leads to a more extensive search of the state space, which is more computationally expensive. As with the appearance model update, a trade-off is needed between exploration and exploitation of the state space to find the best next state while covering as much of the state space as possible. To that end, the method proposed in [126], keeps track of the target in the near vicinity, but also updates a map of probabilities where the tracked target might jump unexpectedly. This is especially interesting for the case of moving cameras, or cameras mounted on moving vehicles; but also, it is interesting for scenarios in which there is frame dropping and sudden frame rate variations. In [140], the problem is addressed by formulating tracking as an optimisation problem, where abrupt motions are dealt with by a particle swarm optimisation (PSO) where the spatial distribution of the particles is such that the candidate states are sampled from all over the image. Furthermore, dynamic acceleration parameters (DAP) are introduced, to determine the best mean and variance of the distribution used for sampling based on the averaged velocity

information of the particles, which leads to more accurate model, and therefore better performance.

2.4.1.8 Evaluation Frameworks

Historically, there has been a lack of uniformity in the evaluation of trackers, as opposed to other fields, such as disparity estimation, optical flow computation and video coding, where commonly accepted evaluation procedures are used by their respective research communities. Some earlier efforts appeared in the form of surveillance datasets, (CAVIAR[1], i-LIDS[2], PETS[3]), but there was a lack of specific measures for tracking assessment. For instance, as stated in [211], the MIL tracker [22] is evaluated using a score representing the mean centre location error in pixels. Another similar measure is used in [190], where a 9×9 pixel neighbourhood is taken around the ground truth centre, and tracking is considered correct if the centroid of the bounding box yielded by the tracker lies within this neighbouring area. Either measure cannot be considered a good choice, since none takes the size of the bounding box into consideration (and therefore the accuracy in estimating the size of the target), as explained in [175, 211]. As an alternative to centre-to-centre distance-based measures, overlap measures can also be used [175], which is commonly represented by O [174], and is given as:

$$O_k = \frac{|TP_k|}{|TP_k| + |FP_k| + |FN_k|} \quad , \tag{2.1}$$

where TP, FP and FN are the areas in pixels for the true positive, false positive and false negative values, and k is the frame number. That is, the larger the O_k the better the result is (closer to one). Figure 2.4 depicts how this formula translates to an actual comparison between the ground truth and the proposed tracking system's outcome. Shaded in red are the areas that the measure *penalises*, that is, the areas that should be as small as possible for the final overlap value to be high. On the contrary, the area shaded in green should be as big as possible for the overlap value to be high.

[1]Context Aware Vision using Image-based Active Recognition
[2]Imagery Library for Intelligent Detection Systems
[3]Performance Evaluation of Tracking and Surveillance

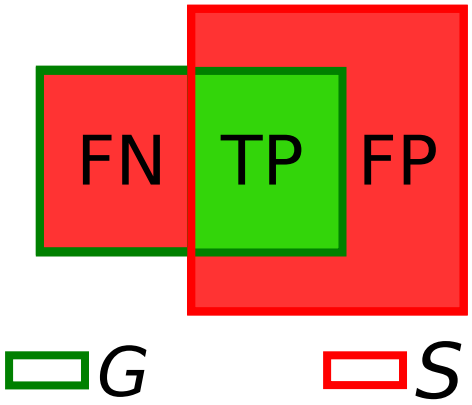


Figure 2.4: The overlap measure is one of the most restrictive measures used. Depicted are the Ground truth (G) and the system's outcome (S). Areas in red, are penalised by the overlap measure (part of the denominator), areas in green, are *encouraged* (numerator).

This measure has become very popular in recent years, and used either as is, or in measures derived from it, as, for instance, when binarised through the use of a fixed threshold to consider which frames have been correctly tracked. One such example is the PASCAL overlap criterion, which was first introduced in the visual object challenge (VOC), an object detection challenge introduced in [71], which is now commonplace in tracking evaluation (e.g. as proposed by [211]) . Under this criterion, an object is correctly detected (accurately detected) if the overlap value is above a fixed threshold of 0.5. For any given sequence, this translates to:

$$P = \frac{1}{K} \sum_{k=0}^{K} \delta(O_k > 0.5) \quad , \tag{2.2}$$

where K is the total number of frames, and $\delta(\cdot)$ is a function returning 1 if the condition is met, or 0 otherwise.

Besides, to avoid having to decide on a specific threshold, in [174], the authors, propose a novel evaluation criterion. Motivated by the lack of uniformity in tracking evaluation, they propose a protocol that unifies some other existing datasets and tools. Their protocol entails a series of videos aimed at different applications: face tracking, rigid/vehicle tracking, and articulated/people tracking. Furthermore, the protocol also introduces testing the trackers on the selected videos under challenging conditions

(referred to as *trials* in their paper). Trial 0 has no perturbations or modifications. In trials 1–3 the initialisation seeds (initial bounding boxes) are perturbed in different ways to analyse the tracker's behaviour. In trial 4, trackers are tested in the presence of noise. In trial 5, frame skipping is simulated. Lastly, in trial 6, trackers are tested against illumination changes (also simulated). To evaluate the videos, they introduce a novel evaluation criterion, which is called the "area under the lost-track ratio curve" (AUC_λ). To calculate it, the overlap measure (O_k) is calculated for each frame k, as depicted already in eq. (2.1).

Furthermore, the *lost-track ratio* (λ) is computed based on the number of frames where the track is lost over the total number of frames. To determine when the track is lost, instead of using a fixed threshold value (as in the PASCAL measure where it is set to $\tau = 0.5$), the threshold (τ) is changed in increments of $\Delta_\tau = 0.01$, taking values in the range $[0, 1]$. Finally the AUC_λ is calculated as:

$$AUC_\lambda = \Delta_\tau \sum_{\tau=0}^{1} \lambda(\tau) \quad , \tag{2.3}$$

with $0 \leq AUC_\lambda \leq 1$, and where lower values represent better trackers (those that lose the track on fewer occasions, *in general*, for all values of τ).

In [175], the authors go further and propose a criterion, called the *combined tracking performance score* (CoTPS), in which both the accuracy and the robustness are combined. Values of $CoTPS$ are in the range of $[0, 1]$, and smaller values are better, since both its components can be seen as *penalty* scores. The accuracy is expressed in very similar terms of the AUC_λ in their previous work, however, in this occasion it is renamed to Ω. Robustness is expressed as the tracking failure score (λ_0), that is, the number of times in which the overlap falls to zero. Their combined measure uses a self-adaptive parameter β, which is used to weight the contribution of each independent measure. Therefore the final score is calculated as:

$$CoTPS = \beta\Omega + (1 - \beta)\lambda_0 \quad . \tag{2.4}$$

Since β is calculated as the proportion of frames for which there is some overlap ($O_k > 0$), the CoTPS score can be interpreted as: "the penalty in the accuracy of the tracker, taken into account only to the extent where the tracker did well; plus the penalty of failures, taken into account only to the extent where the tracker failed completely".

Similarly to the AUC_λ (or Ω) [174, 175], in [270] the authors propose an area under the curve measure for overlap. This is plotted as a *success plot*. Furthermore, they also propose to use, *precision plots* to represent centre location error as the Euclidean distance between tracker's output and ground truth. To avoid the problem when this distance gets too large due to an error, the plot shows the number of frames for which the distance is within a certain boundary threshold. Nonetheless, as already explained many authors criticise such centre-distance based measures, as they disregard the variations in size of the target [41, 175, 211].

Seeing the many different measures used in the field, in [41], the authors disagree with the arbitrary selection of features, since they might not be independent, as is the case of [270]. Therefore, they perform a correlation analysis between different popular measures, and decide to use the ones that are the least correlated to each other (as it happens, the measures of [175]: Ω and λ_0, happen to be good selections after all). According to their results these are the *averaged overlap* ($\bar{O}$, or $\bar{\Phi}$ using their notation). This study has determined measures used for the accuracy and robustness ranks (ρ_A, ρ_R) used in accuracy–robustness plots in the visual object tracking (VOT) challenge evaluation framework [120–122]. Under this scheme, trackers' accuracy is estimated as the mean overlap value over several runs, which are necessary due to the stochastic nature of some trackers. Furthermore, the robustness is calculated as the mean number of resets needed by the tracker over the runs. Nonetheless, they disagree that this can be brought into a single measure such as the CoTPS of [175], since, they argue, is somehow cryptic and not sufficiently justified. However, the VOT challenges have become very popular, and its measures are becoming the *de facto* standards by which new trackers are assessed.

Nevertheless, regarding the size of the datasets which are used in evaluation of trackers, many papers resort to a number of no more than 50 benchmark sequences, which in most cases is closer to 10 or 20. Based on this observation, large-scale datasets for comparison have been compiled [132, 225]. In both cases, more than 300 videos (315 [225] and 365 [132]) are used, and evaluation shown for more than 20 trackers (many presented in the current section).

Large datasets can be helpful in avoiding over-fitting for particular benchmark sequences, which are very commonly used, but another factor when it comes to the evaluation of newly proposed trackers, is subjective bias towards the proposed method, as compared against other trackers. In their review, Pang et al. [186], establish that this is unavoidable, and not necessarily intentional, since many proposed algorithms have a number of parameters that need tuning for best performance. Therefore, they propose the comparison of 'second best' results, that is, all other trackers to which the newly introduced tracker (in a hypothetical paper) are compared. They generate a database of this results, which are more likely to be unbiased, and perform a rank based on the many results provided by several papers that compare 'overlapping' sets of trackers.

2.4.1.9 Summary of models and strategies used

To summarise the features of the visual tracking methods reviewed in this chapter, the following tables group these methods based on different functional aspects, namely: the motion models used to determine the position in the next frame (shown in Table 2.3); the appearance traits used to model the target (see Table 2.4); the update strategies to refresh the appearance model based on information from new frames, to deal with appearance variations (summarised in Table 2.5); and, the detection or response combination (shown in Table 2.6).

Regarding the motion models used, Table 2.3 shows four clearly defined groups of algorithms. First there are the simple approaches, such as global maximum and a simple search window or pre-defined radius, here mean-shift related algorithms are

also included. Then, there are also particle filters and Monte Carlo simulations (and derived). There is also optical flow based displacement estimation. Some of these consider abrupt motions. Finally, there are other methods for tracking through abrupt motions such as a PSO-based algorithm.

Method	Examples
No model used, just global maximum	[190, 237].
Only a radius around the previous position, or . . . a search window defined by an enlargement of the current target region, or . . . integrated in a mean-shift algorithm	[22], [82, 191], [20, 58, 144].
Optical-flow based displacement estimation	[107, 211].
Bayesian tracking formulation (BTF), either as is, or . . . integrated into particle filter (PF), or . . . as Monte Carlo simulations (e.g. Markov chain Monte Carlo, MCMC, or derived).	[142, 256, 278], [115, 163, 164, 183, 203, 273], [125, 126].
Particle swarm optimisation (PSO) framework with adaptive mean/variance via dynamic acceleration parameter estimation (based on averaged velocity information) for abrupt motion	[140].

Table 2.3: Motion models used in reviewed works

Table 2.4, on the other hand, classifies methods according to the appearance models used for modelling the target. Here, an evolution can be seen, as depicted in Table 2.2: from pattern-based modelling, to histogram-based (the most abundant), to approaches using local features (e.g. HOG, Haar-like features, LBP). It is important to mention methods where pixel values are used as feature vectors, which can be used as is, or in covariance matrices and *eigenbasis* representations. Superpixels also appear here, as do hybrid approaches, and complex, ad-hoc models.

Model update strategies are shown in Table 2.5. Again, four main blocks emerge. First, methods that do not use model update, the list is not complete since this would include most pre-2005 tracking algorithms. Then, there are simple or straightforward approaches, such as those using the new target position to update the model, or

Method	Examples
Pattern-based models (e.g. intensity, edge, colour channels):	
– Normalised image patches	[107, 142, 144, 163, 164],
– Selection of best performing linear combination of RGB channels	[58],
– Sparse PCA applied over feature templates extracted from: hue, saturation, intensity and edge templates	[125].
Histogram-based models (simple or multi-dimensional)	
– Based on colour	[126, 140, 191, 237, 256],
– or HOG	[237].
Pixel values as feature vectors (e.g. fed to classifiers) using one or several feature layers (apart from the colour or intensity information) to create image tensors used in covariance matrices or *eigenbasis* representations	[20, 190, 203].
Haar-like features	[22, 82],
. . . and spatially distributed Haar-like features	[115].
Superpixels represented by their location within the target region and its average appearance (average HSV values)	[183].
Hybrid, more complex approaches, or ad-hoc models:	
– Combination of intensity patterns (for subspace learning) and local HOGs (used by SVM classifier)	[278],
– Local LBP features and RGB histograms independently used by two codebooks in a baf-of-words framework	[273],
– In PROST, three different trackers use: 1) a mean-shift procedure over the estimated flow field; 2) pattern matching; and 3) pixel value information.	[211].

Table 2.4: Appearance models presented in this review

retaining some frames, based on different rules, or by a linear interpolation of old and new model (i.e. similar to a running mean). More complex approaches are presented in the third block, that is, early discriminative trackers (where poor-performing classifiers are replaced with new ones), as well as co-training trackers and subspace modelling techniques. Finally, the fourth block represents further discriminative trackers that are strictly classifier-based. In this block, positive as well as negative examples are used, in different ways, either using a single positive example, or several, or bagging them before being fed to the classifier (as in MIL, [22]).

Finally, Table 2.6 shows final target detection, or precisely response combination methods, for those algorithms in which several cues are used. Again, four main response combination techniques appear. First, methods based on global maximum (e.g. in co-trained tracker [278], where a product of the likelihoods is used). Next, there are a series of methods in which the response is integrated into other frameworks such as mean-shift, particle filters, MCMC, PSO, etc. Other methods rely on weighted linear combinations, either via boosting using AdaBoost or an equivalent method (e.g. interactive MCMC, or IMCMC). Finally, there are several combination methods which are specific to some works. For instance, in TLD, there is an *integrator* module, which is in charge specifically of integrating the different responses from the tracker and the detection modules.

2.4.1.10 Concluding remarks

In this section, a chronological review of the most relevant papers in the field of visual tracking of objects has been presented. This review has covered the period from 2006 to the present, with a special focus on the period of 2011 to this day.

If stopping to analyse trends, it can be seen that there has been an enormous focus on discriminative trackers in the period of 2005–2012, specifically methods that use classifiers that are re-trained on-line with new samples. Yet, in the more recent past (2012–present), there has been a shift towards classifier-less trackers and, precisely towards an old idea of improving discriminative power of the features used by using

Method	Examples
None used, or, specifically: – Uses a new combination of features in the new frames, which can counteract the otherwise non-adaptive model – Appearance changes are dealt with by using three different approaches to tracking in PROST	[126, 140], [58], [211].
Simple approaches: – New target signature based on new target's position – Periodical update based on some retained frames – Based on initial image and a few recent instances – By linear interpolation of old and new model (with a given learning rate)	 [183], [190], [256], [125], [191].
More complex approaches: – Best-performing weak classifiers in a boosting framework are kept, new ones replace poor-performing ones – Best-performing templates in a sparse representation framework are kept, new ones replace poor-performing ones – Static sparse representation dictionary with dynamic basis representation – Codebook updating scheme, where patches with highest similarity are added to the codebook via retraining – Co-tracking approaches in which trackers based on different features exchange failure cases – Online subspace model update (manifold learning) – As above, but combined with a sliding window for the selection of the SVM samples to use	 [20], [142, 163, 164], [144] [273], [237], [203], [278].
Classifier-based approach with online retraining based on positive and/or negative examples: – Positive examples only, from around the new target location – Based on positive/negative examples – Positive/Negative examples sampled from the new state and surroundings and fed as sample bags to the MIL classifier – Positive and negative experts update the way the detector works in TLD	 [82], [115], [22], [107].

Table 2.5: Update strategies employed by the methods reviewed

Method	Examples
Global maximum, 'search window' or 'radius' maximum:	
– Global	[190],
– Local	[22, 82, 191, 256, 278].
Integrated into:	
– Mean-shift	[20, 58, 144],
– pure Bayesian formulation, or Particle filter	[115, 142, 163, 164, 183, 203, 273],
– Markov chain Monte Carlo (MCMC)	[126],
– or particle swarm optimisation	[140].
Weighted linear combination of different responses:	
– Either via boosting, e.g. AdaBoost	[237],
– or an equivalent, e.g. the interactive MCMC	[125].
Hybrid or ad-hoc approaches:	
– Combination of responses based on the individual discriminative and generative models/trackers	[278],
– Cascaded decision, with manually-set take-over rules (PROST)	[211],
– Using an 'integrator' module that takes into account the tracker and the detector responses (TLD)	[107],
– Distractor-aware detection using a Bayes classifier	[191].

Table 2.6: Target detection, or response combination methods

trackers that improve the object–surround and object–distractors variance ratios (this idea dates back to [58], from 2005).

Another trend in recent years, that was not covered by the previous reviews analysed ([274, 276]), is that of sparse coding and sparse signal representation, as well as alternative representations such as 'superpixel matching'-based trackers. Regarding sparse representation, it has widely been used for many applications in recent years, not just tracking as covered in [264]. However, obtaining real-time trackers using this approaches has only been possible by using approximations to the calculation of the problem so that early pruning of target candidates is possible, as presented in [164]. Otherwise, ℓ_1-minimisation problems are solved at the expense of a high computational cost.

Furthermore, in recent years standardisation of evaluation measures for tracking assessment has been made possible thanks to a shift towards measures that use overlap of bounding boxes rather than just centre-to-centre distance-based measures. Besides, the analysis of many existing measures and the selection of those with the least correlation has led to better evaluation criteria for tracker ranking. Alas, these new criteria have become popular thanks to the VOT challeges in recent years, which is a promising horizon towards better tracker comparison techniques.

Nonetheless, visual tracking for long periods of time is still a very challenging task, specially in crowded environments, due to distractions (i.e. similar targets in the vicinity), and abrupt appearance changes. These challenges can be partially alleviated by novelties introduced progressively in the tracking methods presented in this section, but no general solutions exist so far that work "in the wild".

2.4.2 The occlusion problem in tracking

Occlusions[4] during tracking pose a major challenge for most existing tracking algorithms, since generalised models for them are not straightforward [124]. According to the survey in [276], occlusion can be classified into three categories: self-occlusion,

[4]This is an excerpt from [241], from a section written by the author of this thesis.

which occurs while tracking articulated objects; inter-object occlusion (or dynamic occlusion [247]), which arises when two tracked objects occlude each other; and occlusion by the background (or scene occlusion [247]), which occurs when structures in the scene (e.g. tree branches, pillars, etc.) occlude the object/s being tracked [238, 289]. Yilmaz et al. [276] deal with occlusion handling from the lens of the tracking technique in use. A series of different tracker families are presented (point, 'geometric model'-based and silhouette); each tracking technique is then classified according to whether or not it can handle occlusions, and in the case it does, whether these can be full or only partial. Following this idea, trackers that respond well when occlusions are present, can be used for occlusion handling. In [283], the Kanade-Lucas-Tomasi (KLT) tracker is employed to resolve occlusions, while a particle filter is used as the main tracker. Similarly, a technique based on mean-shift is used in [47]. These solutions can be applied to sparse crowd situations, but their performance is poorer in densely crowded scenarios.

2.4.2.1 Handling occlusions explicitly from a single camera

Apart from exploiting the features of "occlusion-friendly" trackers, a series of occlusion handling techniques have also been devised, which can be found throughout the literature. Wang et al. [255], present a good historical review of such methods, which rely on the person's or object's motion model, and keep predicting its location until it reappears. The authors state that severe long-term occlusions cannot be dealt with by this kind of techniques, since observations cannot be obtained while the person is occluded for a long period. Vezzani et al. [247] propose what they call the non-visible regions model, which deals with partial and full occlusions, whether these are inter-object or due to the scene. The person/object model is updated differently in a pixel-wise fashion: the appearance is updated only for the visible pixels; the probabilities associated with those are reinforced, while they remain unchanged for invisible pixels; furthermore, in pixels with no correspondence due to changes in the shape of the person or object (called appearance occlusions) probabilities are smoothed.

Wang et al. [255], on the other hand, propose a means of modelling the occluder; once modelled, when targets disappear due to occlusion, a search is performed around the occluder in order to find the occluded object as it reappears. In [111], the authors present a series of monocular approaches to occlusion handling, although this is only to conclude that single-view systems are intrinsically unable to handle occlusions correctly.

2.4.2.2 Fusing multiple evidence as a solution for the occlusion problem

As suggested in [111], having multiple evidence will reduce the amount of hidden regions, thus reducing uncertainty. Many works follow this assumption, which will be discussed in more detail next.

> Another approach to occlusion handling is avoiding them in the first place. Occlusions can be reduced by placing the camera appropriately, as suggested by [276] (i.e. by placing a bird-eye view camera, no occlusions occur between the objects on the ground, assuming outdoor scenes with no tree crowns blocking the view).

In the next subsection, 2.4.3, multiple, simultaneous person or object tracking methods will be introduced. Subsection 2.4.4 will present methods for fusing multiple evidence; either from multiple homogeneous cameras, or diverse heterogeneous sensors. Finally, subsection 2.4.5, will deal with tracking from aerial vehicles, which can partly overcome the problem of occlusion, although this approach will also introduce new challenges.

2.4.3 Multi-target tracking in large crowds

The particle filter approach has been extended for tracking multiple targets [3, 39, 81, 112, 182]. For example, Okuma et al. [182] extend a particle framework by incorporating a cascaded AdaBoost algorithm for the detection and tracking of multiple hockey players in a video. The AdaBoost algorithm is used to generate detection hypotheses of hockey players. Once the detection hypotheses are available, each hockey player is modelled with an individual particle filter that forms a component of a mixture particle

filter. Similarly, Ali and Dailey [3] combine an 'AdaBoost cascade classifier'-based head detection algorithm and the particle filtering method for tracking multiple persons in high density crowds. The performance is further improved by a confirmation-by-classification method to estimate confidence in a tracked trajectory. Choi et al. [51] propose tracking and detecting activities at the same time, since they hypothesise there is a link between a person's motion, their activity and the activity of neighbouring individuals.

On a completely different way, Oxtoby et al. [184] propose a "myriad" target tracking. Their application consists of tracking thousands of particles in a dusty plasma. They employ extended Kalman Filters (EKF), along with a Bayesian inference step that uses the particles' dynamics equations to assist the tracker. To be able to track thousands of particles in the dust, multiple trackers are employed, each of them tracking the movement of six neighbouring particles.

Regardless of the tracker being employed, maintaining the stability of the tracks on multiple targets arises as a new challenge. Data association is also considered for multi-target tracking on single camera views. In this case, the multiple tracks are linked to their new states, in the presence of clutter [173]. Huang et al. [98] state that most data association multi-target trackers have two basic elements, namely a tracklet affinity model —which determines how affine two tracklets are—, and an association optimization framework, which determines which tracklets should be linked given the affinity between them. Previous works used parametric models based on the measurement of tracklet affinities, based on human knowledge. In contrast, they propose to use non-parametric models which are inferred from training data.

Song et al. [227, 228] propose a solution based on the hypothesis that trackers can obtain fairly good tracks in the short run. Then, they analyse these short tracks, or tracklets, and develop associations between them, in order to obtain longer tracks, both in single and multiple camera systems.

2.4.4 Fusion of multiple sensors

As it has been previously stated in Sec. 2.4.2, the acquisition of data from multiple sensors is a good means to reduce the problem of occlusions and resolve uncertainty; either from a system employing only cameras, or a variety of cameras and other sensory devices. However, since sensors are noisy, having more sensors also implies having more noise to filter. Furthermore, it also increases the complexity of the algorithms, due to the communication and coordination. The next two subsections will explore the fusion from multiple sensors. First, multi-view or multi-camera systems will be introduced. After that, multi-device or multi-sensor systems will be discussed.

2.4.4.1 Homogeneous multi-view approaches

Systems utilising multiple cameras, or multiple views, can reduce the amount of uncertainty, and handle occlusions; but with such approaches, a number of issues or new challenges need to be addressed. These problems include camera calibration, and ground plane estimation [88, 110, 111, 177] (in the case of the cameras sharing a plane); trajectory association, or person re-identification [35, 86, 161] of an individual object along the multiple views [67, 69, 160, 272]; finally, camera topology discovery is also studied [73]. Khan and Shah [110, 111] obtain the ground plane estimation, by fusing the foreground likelihood information (foreground segmentation detection) from different views. Haselhoff et al. [88] use multiple oblique-view cameras to handle occlusions appropriately, and devise a common plane reconstruction, using communication among cameras.

The drawback of the systems that rely on overlapping views is the fact that most existing camera networks were not initially devised for the reconstruction of the ground plane, and so, multiple views of the same area are not available [241]. Song et al. [228] are able to track people from multiple non-overlapping views by reducing the problem of camera hand-off (person re-identification) into a data association problem, in which tracklets from different cameras are merged into bigger tracks across cameras. A very similar approach had been used by the same authors in [227], to associate tracklets of

multiple objects in a single camera. In [161], the authors provide an extensive review of the state of the art in this kind of algorithms, and propose a method which models the probabilities of people's trajectories based on a series of landmarks of interest, which draw individuals towards (or away) from them.

Nevertheless, in very crowded scenarios, tracking individuals in full is impossible given the amount of partial occlusions [34]. For such situations, some authors propose using "head and shoulders" detectors, or upper torso (or Omega-shape [135, 161]) which is more likely to be visible from the camera perspective, given they are installed most commonly over the heads of people. However, this is only true for people tracking, and general solutions for other types of object would be desirable (e.g. a crowd of animals, traffic jams, etc.). Using multiple cameras installed in a high vantage point can be beneficial as a general solution in the presence of occlusions.

2.4.4.2 Heterogeneous multi-device approaches

There are proposals that employ UAVs such as the one in [210], which rely on the fusion of data captured by multiple sensors, such as the data from the autopilot, to reliably track moving objects or people in the scene. These proposals for multi-modal fusion, that is, that assist and enrich vision systems with other devices, such as sensors or data sources (like in the cited case of the autopilot mechanism), can improve the performance of computer vision systems. Atrey et al. [18] present a review on the matter.

Multi-modal fusion includes many kinds of sensory devices such as RFID tags and readers [94, 108, 220, 266]. For instance, in [94, 266], object use is evaluated to determine whether an action is taking place or not. The occurrence of an action is evaluated both by RFID data and other sensors. Knowledge can be combined into rule-based decision schemes, that yield a probability of an action taking place [43, 93]. The major drawback of this kind of approaches is the fact that the RFID readers employed can read the labels in short range distances only, and because of that, the

described works were always carried out in indoor environments, for the recognition of activities of daily living (ADLs) in the context of ambient-assisted living (AAL). Furthermore, the accuracy of such technologies is poor.

For wide-area crowd dynamics analysis, Bluetooth-enabled devices can be useful. In [246], a system is described which was deployed in the city centre of Ghent, in Belgium. Several Bluetooth antennae readers were installed in different locations of an open-air venue. Those would estimate crowd densities in several points, based on the number of devices that could be detected by a particular antenna; furthermore flow maps of the visitors' trajectories could be generated.

Unfortunately, the work in [246] does not include the utilisation of vision as a means for detection. It could be interesting to further investigate hybrid systems in very crowded events, since it could facilitate many currently challenging situations, such as people re-identification among non-overlapping views. Similarly to [266], the data from non-vision sensors can be employed as a ground truth during the training stage.

2.4.5 Tracking from unmanned aerial vehicles

There are two types of unmanned aerial vehicles (UAVs), those similar to planes (fixed-wing platforms) or those similar to copters (rotorcrafts, also called RUAVs [178]). These two different types of UAV platform define several operational aspects, such as for instance, the way video footage is captured and further analysed. Fixed-wing UAV platforms need to fly in circular or similar patterns to be able to retrieve data from a specific spot of interest, which causes a reduced frame rate for that particular area. This will have an impact on the analysis methods that are required for human tracking or crowd analysis. Examples of common flight patterns are shown in Fig. 2.5.

This particular limitation makes fixed-wing UAVs unsuitable for low altitudes. Flying in a pattern might be dangerous in some scenarios. Also, the lower the altitude, the more the field of view changes, and therefore it might be impractical to apply further image processing techniques. In contrast, rotorcraft UAVs can fly safely at

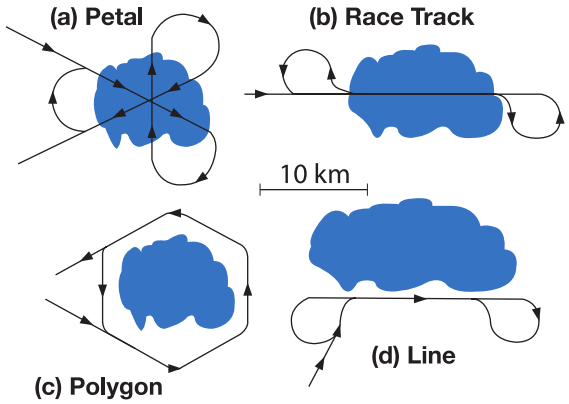


Figure 2.5: Flying patterns for a fixed-wing UAV over an area of interest (source: `http://www.nasa.gov/vision/earth/environment/30oct_lightning_prt.htm`, accessed 04/04/2016.)

much lower altitudes. They can be equipped with as many helices as necessary for improved stability, quad- or octo-copters being the most common configurations. Since they can hover over the same spot for a period, image registration is facilitated.

Additionally, most existing tracking methods rely on videos that are stable and the "smoothness constraint" applies [126]. Under this assumption, the position and scale of a particular object being tracked does not change abruptly between contiguous frames. This is true when the movements of objects in the scene are not abrupt, the frame rate is fixed, and there are no interruptions in the video stream.

However, UAVs are, by definition, flying moving objects with vibrations (due to the motors used) and sensitive to weather conditions, particularly wind gusts. All those make recording of noise-free videos from UAVs particularly challenging. Furthermore, the UAV is continuously moving (either vibrating or actually flying in a pattern over an area), and the ego-motion, that is, the motion of the UAV, needs to be estimated, so that it can be accounted for, and subtracted from the actual movement of the objects in the scene or area of interest [178].

2.4.5.1 Ego-motion correction and the Smoothness Assumption

Ego-motion compensation techniques can be purely based on video inputs, or use some additional cues from telemetry sensors (global positioning and inertial navigation

systems —GPS/INS). Ego-motion estimation is also referred to as 'homography estimation' and has been extensively used for several applications in the field of computer vision, and specifically, it has been used in moving vehicles, both terrestrial and aerial, as well as robots, for the compensation of the motion of the vehicle. This pre-processing step allows for frame differencing to be calculated, and as such, it allows many fixed-camera methods (and assumptions) to be employed (e.g. background modelling for moving object detection, tracking with 'smooth motion constraint', etcetera).

A very common approach among the studied works is to perform an 'image registration' or 'image correspondence', in order to calculate the homography between consecutive frames [189, 198, 280], specially using the extraction of features from both images, via an interest point detection algorithm (e.g. Kanade-Lucas-Tomasi's 'good features', or others). Once the points have been extracted, a correspondence among them is found, and the most common step after that is to apply a random sample consensus (RANSAC) algorithm to eliminate outliers. By doing so, a homography matrix can be estimated, which encodes the translation and rotation of the aerial vehicle between two frames. In some works this idea is extended, and maps of the explored area are created, via 'image stitching' or 'mosaicking' techniques [103, 176] or simultaneously locating the vehicle and mapping the scene (UAV SLAM) [38]. Another means for obtaining image correspondences are correlation-based methods as shown in [214], where a spectral image registration (the improved Fourier-Melling Invariant, iFMI) method is employed. Other methods based on visual inputs only, are based on optical flow segmentation [279], or similarly a dense motion field estimation [178].

However, video-based homography-based techniques have a major drawback, since they require highly textured images, specifically, images where the background texture is rich, so that points or other features can be extracted from them. An ill-posed case occurs when the only available texture is that of the moving objects (due to a homogeneous background), the homography matrix estimation procedure will take into consideration the motion of the ground objects, and as such, the obtained matrix

will not be representative of the vehicle's motion.

Poorly textured backgrounds are common when the UAV is flying close to the ground over an individual (e.g. over a grassy patch or a paved surface). In such cases, the repetitive nature of the background texture causes mismatches in purely video-based approaches. Texture mismatches are also common when flying over a large group of people or a crowd (e.g. in a demonstration or rally) where static elements of background might not be visible, or might constitute the minority of the matched points, thus wrongly being classified as outliers.

With the improvement of GPS/INS sensors, the problem of the calculation of the homography between frames can be addressed using the data provided by those devices [210], especially since all UAVs are equipped with such systems, and thus the only requirement is for their data to be available. Applications, include among others: mapping or SLAM [92], and target geo-location [27, 66].

For geo-location, image coordinates can be translated into geographical coordinates, and therefore, the target can be localised using a coordinate frame that is different from that used by the sensor or camera (image coordinates). The computational overhead introduced by image transformations, once the homography matrix is estimated, is not negligible, and this raises the question of the real necessity for such image warping [221].

As explained in Sec. 2.4.1, many tracking algorithms incorporate the idea of a motion model, many of which assume a 'smooth motion constraint', which makes it difficult to use these algorithms for the task of tracking from UAVs. A naive approach, when using Bayesian tracking formulations, would be increasing the variance of the Gaussian that is employed to sample candidate states for the new frame, however, this increases the computational cost, since there is a larger area to cover. The same occurs when using mean-shift or derivatives, and the search window size is increased. Some works presented in the aforementioned section propose to use no modelling of the motion at all, since they can re-detect the target even over the whole frame (global maximum, e.g. [237]). In that same section, recent developments in motion models for abrupt motion (e.g. [125, 126, 140]) have also been introduced, which would allow

tracking objects undergoing abrupt/unforeseen movements which could be used for tracking from UAVs. This is especially interesting for the case of moving cameras, or cameras mounted on moving vehicles; but also, it is interesting for scenarios in which there is frame dropping and sudden frame rate changes; all those being characteristics of a video stream captured by a UAV. Please refer to the aforementioned section for more details on tracking methods and their motion models, and how these could be utilised for the purpose of tracking from moving aerial platforms.

> However, a method that keeps a model of the motion undergone by the camera would have an advantage when there are several people with similar appearance in the scene. By exploiting the camera motion, and not simply using a global maximum (estimated as either a simple probability map or by using abrupt motion models), these types of maxima conflicts could be better solved.

2.5 Summary

After having reviewed the relevant literature, the limitations in current methods can be identified. This will help draft the research lines of the present thesis. For instance, macroscopic approaches for crowd analysis are seen as too *coarse*, and limited in scenarios where the abnormalities have multiple classes and/or entail a single individual. This thesis addresses that particular problem by first analysing the crowd granularity (Chapter 3), to assess whether a *finer* microscopic approach can be applied, and in such case, proceed with the proposed *mesoscopic* method (Chapter 5). The following table shows additional examples (see Table 2.7), and includes unresolved research opportunities that were identified in the literature review, along with the ways in which these have been addressed in the thesis.

Section of this literature review	Conclusions or identified research niches	How this thesis addresses them
2.2 and 2.3 Macroscopic modelling	• Tracking sparser scenes in the same way as denser ones will fail to identify abnormalities affecting a single individual. • Macroscopic analysis can be useful as an indicator of general trends.	• Assessment of crowd granularity to determine the best analysis approach to use (Ch. 3). • Aggregation of cues from microscopic analysis in a *mesoscopic* approach for event detection in large groups of people forming sparse crowds (Ch. 5).
2.4.1 Person and object tracking	• In crowded environments, distractions and abrupt appearance changes lead to failure in long-term, robust tracking.	• In tracking from UAVs, use of a corrected search window can lead to reduced distractions (Ch. 4). • Aggregation of *tracklets* into a scene descriptor to detect events in groups (Ch. 5).
2.4.2 The occlusion problem in tracking	• Occlusion can be handled from single view cameras to some extent, but multiple view systems can avoid the problem.	• Multi-view fusion for crowd event analysis (Ch. 5). • Detection and tracking of targets from UAV (Ch. 4).
2.4.3 Multi-target tracking in large crowds	• Tracking thousands of elements is studied, but only for very short periods of time and from a single view. • Data association for multi-target tracking is expensive but allows re-identification and longer-term tracking.	• Use of short periods of time to collect *tracklets* from each view and aggregate them into scene descriptors that are then combined among views (Ch. 5).
2.4.4 Fusion of multiple sensors	• In the case of moving cameras, data fusion from other sensors can facilitate ego-motion compensation. • Bird-eye views, and cameras placed in high vantage points can also be used to avoid occlusions.	• Background modelling and tracking window correction from UAV aided by telemetry sensors (Ch. 4). • Analysis of groups of people from multiple high vantage points (Ch. 5).
2.4.5 Tracking from unmanned aerial vehicles	• Ego-motion compensation is paramount but can fail if purely video-based. • Using a search window might be beneficial to avoid distractions in tracking from UAVs.	• Telemetry-aided ego-motion compensation with applications in background modelling and search window correction (Ch. 4).

Table 2.7: Identified gaps, and corresponding chapters of this thesis were these are addressed.

Chapter 3

Crowd classification using a density-entropy signature

Chapter highlights: A density–entropy signature for the assessment of the level of danger in a crowd, as well as determining the best-performing analysis to be applied.

Overview

Population growth in urban settings can lead to overcrowding, which can in turn rapidly become dangerous in the presence of panic or agitation. Different contexts call for different methods of analysis. In heavily cluttered and crowded scenes, a classic pedestrian tracker is likely to fail. However, a macroscopic approach, analysing the crowd as a whole might be more appropriate in such cases, at the cost of losing fine granularity of individual behaviours. In this chapter, a novel classification method for crowded scenes is presented, based on density ρ and entropy $\mathcal{E}$ estimators. The two are then combined into a signature, used to categorise scenes. The presented results show the potential of this method, compared with ground truth obtained with an innovative manual labelling of the employed test data.

Main contributions, outcomes, and publications

The main contribution of this chapter is the density–entropy signature mentioned above for crowd granularity assessment.

3.1 Introduction

The overcrowding phenomenon in urban areas poses a challenge for current crowd management and monitoring systems. Overcrowding can lead to dangerous situations, specifically when incidents or accidents happen in publicly-managed spaces. Monitoring systems able to detect and predict such scenarios are desirable and automation is key to avoiding human errors caused by tiredness and monotony. Dangerous events tend to happen very sporadically. Therefore, most of the observed scenes are normal, leading to long and tedious periods of time spent in front of a monitor.

Several authors agree that crowds can be analysed with different techniques, depending on the application (e.g. crowd counting versus density estimation), as well as other factors, such as the density of the crowd which causes occlusions and impacts performance negatively [75]. For instance, according to [65], when a crowd is sparse, pedestrian tracking methods (microscopic analysis) can be applied to track individually each person. With heavily crowded scenes, optical flow methods are more appropriate, to analyse a crowd as a whole (macroscopic analysis).

Alternatively to this nomenclature (*microscopic* or *macroscopic*), methods can be classified into *direct* or *indirect*, depending on whether they rely on object/pedestrian detection or, instead are based on local or pixel-based features [208]. In an earlier review [105], crowd analysis methods are classified into pixel-, texture- or object- based, naming the latter as appropriate only in cases where crowds are sufficiently sparse, and leaving the former two for dense crowds, at the cost of coarser results. In that same review, crowd behaviour analysis algorithms are also divided into object-based, relying on object/human detection or holistic approaches, in which the crowd is analysed as a whole due to a lack of gaps between pedestrians, caused by occlusions.

In general, it is desirable to have a method able to discern such different scenarios. That is, an automatic crowd estimator that can classify crowded scenes based on different cues. Density, which has already been mentioned, is extensively used as a means to determine the "level of danger" of a crowd [75]. however, early projects were concerned with measuring both "motion and density and hence potentially dangerous situations" [63]. On the other hand, in [208] it is stated that crowd size can be seen as an important indicator for dangerous situations. However, the authors do not explore other types of indicators. In this chapter, density is used along a novel crowd entropy score, which is used as an indicator for crowd "orderliness".

The reason entropy was picked for the measure of the orderliness of a crowd is that entropy is by definition a way to measure chaos (or lack of it), and therefore it is *intuitively* a natural choice for the task at hand. Nonetheless, in a more formal way, to measure the orderliness of a crowd it is necessary to find a measure that is minimised when the crowd follows (mostly) the same direction(s), and that is maximised when people are moving in many directions. Additionally, it is desirable that when the observed directions of the people change, the measure does so continuously. Entropy has all the desired properties: it is the unique continuous function that is maximised by the uniform distribution, and minimised by the point distribution (peak) [185].

Depending on the density of the analysed crowd, and its orderliness (or lack of it), analysis at different levels can be recommended (microscopic or macroscopic analysis). Such a system can also help determine if the entropy of a scene is too high (people or vehicles, for instance, moving rapidly in different directions), and therefore there is a risk for people present in the area. In such cases additional safety measures should be taken (e.g. an underground station might be closed due to overcrowding, or an act of violence). The method proposed in this chapter is aimed at helping in these situations.

This chapter presents a novel method to discern situations based on their density-entropy $(\rho, \mathcal{E})$ signature. There appear to be no works combining density and entropy together (as orderliness or *excitedness* of the crowd, akin to the concepts of violence presented, or similar) to obtain a signature that classifies the current scene. This

is important, since using density alone might not be enough to determine the level of danger, given that a densely crowded scene could still be safe if the orderliness of the crowd is maintained. This chapter is distributed as follows: Section 3.2 will review some existing previous work. Then, Section 3.3 introduces the methodology employed to obtain the proposed signature. Next, in Section 3.4, the experimental set-up and results are presented. Following that, a discussion of the results is carried out in Section 3.5. Finally, concluding remarks are summarised in Section 3.6.

3.2 Previous work

Density is used extensively as a means to assess the danger of a crowd (e.g. to determine how likely it would be for a human avalanche to happen), as seen in earlier reviews on crowd analysis [105, 282]. Both of these surveys include sections on crowd density estimation and/or people counting. Some recent examples of methods that measure density of crowds exist [75, 76], as well as a survey [208]. For instance, in [75] the authors propose an approach for crowd density measure based on local information at pixel level, as an alternative to methods based on people counting, that heavily rely on object or human detection. The method consists in generating density maps using local features as an observation of a probabilistic density function. The local features used were based on features from accelerated segment test (FAST). In their work, the proposed density measure is presented as able to provide additional information to other video surveillance tasks, in order to improve the otherwise limited success of methods such as tracking and detection. Similarly, in [76] a metric for density estimation called gradient magnitude entropy (GME) is calculated, for this the entropy of a probability distribution function (PDF) based on the sum-one normalisation of a proposed histogram of oriented gradient magnitudes (HOGM), which in turn is a variation of the commonly used histograms of oriented gradients –or HOG– [61]. To obtain the gradient magnitudes they apply $1D$ gradient kernels to the greyscale images. Similarly, in [193] an estimation of density is achieved by calculating the entropy of several descriptors related to crowd texture.

It is worth noting that, in this chapter, entropy is understood as the degree of order of the crowd (i.e. related to the lack of orderliness in the directions of motion of a crowd), as opposed to the works by [76, 193], in which an entropy-based measure is introduced for density calculation. However, methods that calculate the entropy for this purpose are not so common. An exception to this is [84], where entropy is used as a means to calculate the spatial distribution of the crowd (how scattered or gathered people are). For this, the authors refer to the definition of entropy and propose an algorithm that can represent the crowd distribution information. The authors employ individual entropies obtained from the spatial distribution of moving particles in both coordinates of the image. The final entropy score is then calculated as the product of the individual entropies obtained for each axis. Particle motion is calculated as follows: first, particles are placed on a grid, then motion of the pixel represented by the particle is calculated via optical flow, following that, the particles whose motion is above a certain threshold are marked as moving. The spatial distribution of these on both axes are then used to calculate their entropy score, and combined with speed information, used to estimate the parameters of a Gaussian mixture model (GMM) over the normal crowd behaviour.

There are very few works using the concept of entropy as defined here for the purpose of measuring crowd orderliness. Even so, some other works exist that assess the level of violence of crowds [89, 287]. For instance, Hassner et al. [89] present a violent flows descriptor (ViF), which is based on the magnitude variations of a dense (pixel-by-pixel) optical flow. The reason to use magnitude variations among frames rather than the magnitude values is that magnitudes represent arbitrary quantities, which depend on frame resolution and are affected by where the motion is located. However, variations convey meaningful measures of the observed motion magnitudes in a frame, compared to the previous one. Similarly, in [287] a way to localise violence in videos is presented. In their two-step approach, a proposed Gaussian model of optical flow (GMOF) is used to detect candidate regions, which employs the magnitude of the optical flow (OF) as a means to determine the areas with high motion, similar

to classical Gaussian mixture models used for background modelling. For the flagged areas, they introduce an orientation histogram of the optical flow (OHOF) descriptor, which is used in a support vector machine (SVM) classifier [45], to determine whether violence is indeed occurring.

Table 3.1 summarises the works analysed in this and the previous section.

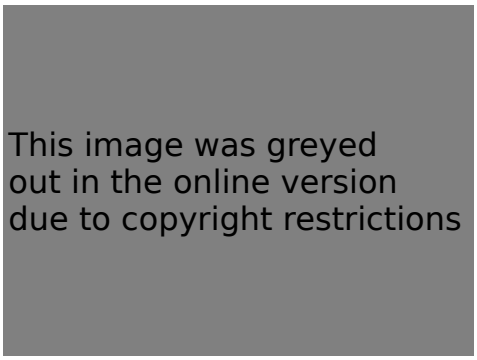
Field or scope	Papers
Density estimation	[75], [76] [a], [208] (survey)
Entropy[b] estimation	[84], [89] †, [287] †
Granularity[c] assessment	[65], [208], [105]

[a)] uses entropy concept, but applied to density estimation.
[b)] understood as *orderliness*, includes related concepts.
†) crowd violence estimation methods.
[c)] or best approach selection: micro- or macroscopic analysis.

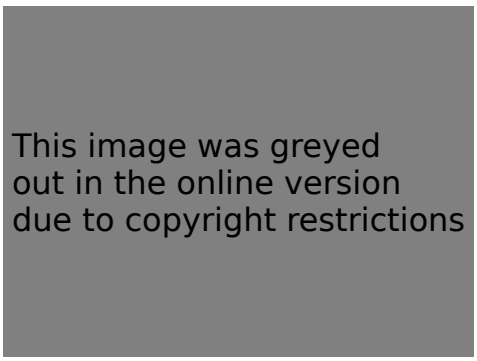
Table 3.1: Classification of previous works analysed

To conclude, Figure 3.1 provides a qualitative comparison of the analysis performed by other methods in the literature, and the differences with the proposed approach. With regards to density, the work in [75] calculates a density map for the whole image (Fig. 3.1(b)). In contrast, the work presented here will calculate a density score for the whole image (Fig. 3.1(d)), based on the foreground mask. With regards to entropy, the work by Gu et al. [84] proposes to calculate the entropy of a crowd in terms of the distribution of the moving particles in the image axes (Fig. 3.1(c)). What is intended by their measure is to describe how scattered (in space) it is, as it uses the distribution of the particles' positions, rather than their direction of motion. As a consequence, their method outperforms others in the literature on the UMN dataset, which contains only *rapid scatterings* as its *abnormal* behaviour class. However, it would not be suitable to explore other types of abnormalities related to incoherent motions (people following different directions of motion). These incoherent motions, nonetheless, are indicative of the likelihood of inter-target occlusions in the scene, and can be helpful for the task of selecting the best analysis approach. The same can be

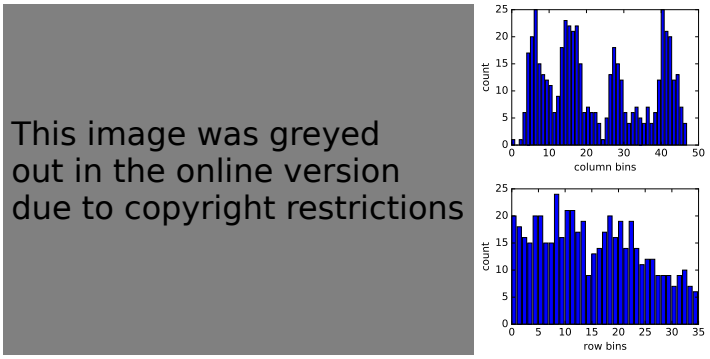
said about violence detection techniques, as violence per se might be representative of the "level of danger" in the crowd, but is not directly representative of the difficulty for tracking inter-occluding targets. Therefore, in this work the entropy (Fig. 3.1(e)) is calculated using the directions of motion obtained from a dense optical flow (described in the methodology, Sec. 3.3), which accounts for the number of different directions in which the crowd moves. That is, it acts as an indicative measure of how (in)coherent the crowd's movements are.



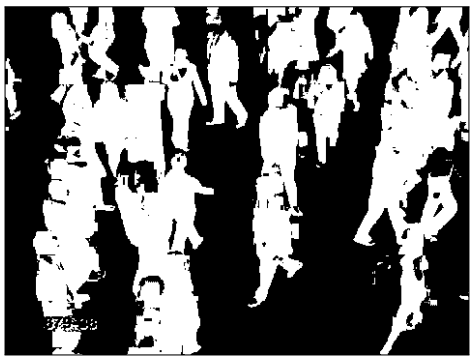
(a)



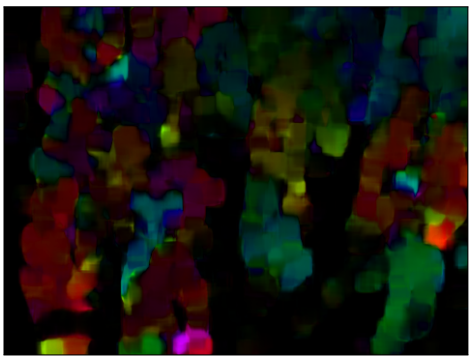
(b) © 2015 Elsevier[†]



(c)



(d) $\rho = 0.68$



(e) $\mathcal{E} = 0.46$

Figure 3.1: An example video frame (a) along with the estimation of *density* and *entropy* as understood by this and other works in the literature: (b) Crowd density map, as estimated by Fradi et al. (reproduced from [75]); (c) distribution of particles (simulated) with histograms of particle distribution in the X and Y axes used for 'particle entropy' calculation by Gu et al. [84]; (d) and (e) density and entropy values $(\rho, \mathcal{E})$ calculated in this work from the foreground mask and dense optical flow.

[†] Reprinted from Information Fusion, Vol. 24 (July 2015), Hajer Fradi and Jean-Luc Dugelay, "Towards crowd density-aware video surveillance applications", 3–15, Copyright 2015, with permission from Elsevier.

3.3 Method

A $(\rho, \mathcal{E})$ signature is a point in the $2D$ $[\rho\ \mathcal{E}]$ space (or curve). Each signature can be employed to characterise a scene (see examples in Fig. 3.2), and is obtained by calculating each individual score separately, namely the density and entropy of the scene.

To obtain these scores, the density and entropy of the crowd need to be estimated. Using segmentation via background modelling (accounting for presence, thus used for density) and optical flow (accounting for directions of motion, thus used for entropy), two maps will be generated, named D and E, respectively. These density and motion maps will be then used to calculate the final scores ρ and $\mathcal{E}$ that define a point in the proposed $2D$ space. Each of these maps are effectively stochastic signals which are easy to represent as probability distribution functions. Each PDF is then compared to a uniformly distributed PDF, to be able to measure how irregular they are: the closer the sample PDF $X \mid X := \{D, E\}$ is to a uniformly distributed profile U, the more irregular dynamics are and the higher the entropy is. To this end, the mutual information (MI) is employed, defined as:

$$I(X;U) = H(X) + H(U) - H(X,U)\ , \tag{3.1}$$

where $H(\cdot)$ is the entropy of a PDF, and $H(\cdot,\cdot)$ represents the joint entropy of the compared PDFs. In turn, the entropy of each PDF is calculated as:

$$H(X) = -\sum_i P(x_i)\ \log P(x_i)\ , \tag{3.2}$$

where $P(\cdot)$ is the probability mass function. Similarly, the joint entropy is defined as:

$$H(X,U) = -\sum_i \sum_j P(x_i,u_j)\ \log[P(x_i,u_j)]\ . \tag{3.3}$$

However, since the mutual information has no upper boundary (i.e. $[0,\infty)$), it is important for our scores to be bounded above, and normalised to the range

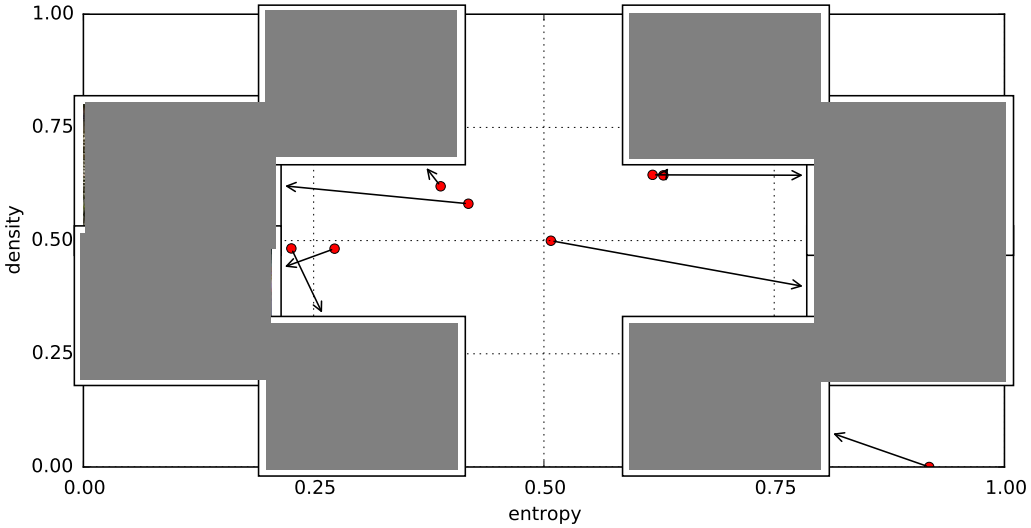


Figure 3.2: Qualitative results for the classification of crowded scenes based on their $(\rho, \mathcal{E})$ signature; showing 8 selected examples (2 per quadrant). As can be observed, the top-right quadrant shows scenes with high density and entropy whereas the rest show much sparser scenes in either or both dimensions analysed.

$[0, 1]$, so that 1.0 represents the highest density possible, and the least orderly crowd (chaotic), respectively. For that reason, a normalised mutual information measure is introduced [70], using the *redundancy* [19]:

$$R = \frac{I(X;U)}{H(X) + H(U)} \,, \tag{3.4}$$

as well as the maximum value that can be achieved (total redundancy, $R_{\max}$), which serves as a normalisation term:

$$R_{\max} = \frac{\min(H(X), H(U))}{H(X) + H(U)} \,. \tag{3.5}$$

Therefore, the **normalised** mutual information, can be expressed as:

$$I' = \frac{R}{R_{\max}} = \frac{I(X;U)}{\min(H(X), H(Y))} \,. \tag{3.6}$$

To obtain the density measure ρ of the $(\rho, \mathcal{E})$ signature, a foreground mask is first obtained by using a standard method, such as an adaptive Gaussian mixture model [291]. Once the foreground mask is obtained, and taking into account the active

area, the process continues as follows: First, a sampling is performed over the active area. This leads to a need to introduce the concept of *active area*: when analysing surveillance videos, only a portion of the camera view is of interest; this is because of the camera vantage point, also capturing portions of the scene where no activity occurs (for instance, walls, the sky or other similar inactive areas). For simplicity, a mask is manually marked, once for all, to highlight only the area of interest.

A number of samples are taken at fixed intervals δ in both directions ($\delta_x = \delta_y$). The value of this parameter is set in the experimental section. Then, for each sample, a square window of size $L \times L$ pixels is obtained, as depicted in Fig. 3.3. The density D in each window w is then evaluated as the number of pixels that are in the foreground mask and are part of the *active area*, divided by the number of total pixels in the area. That is, each element of the density map will be given by:

$$D_w = \frac{\sum_{i,j \in w} f(i,j) \cdot a(i,j)}{\sum_{i,j \in w} a(i,j)} \ , \tag{3.7}$$

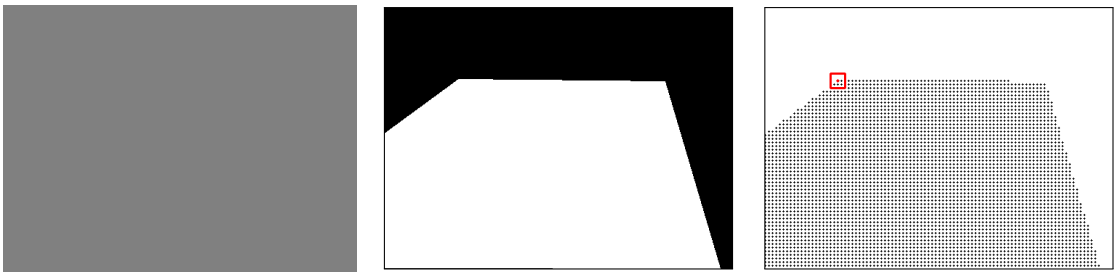
where $f(\cdot,\cdot)$ is one if pixel at position i, j within the window is part of the foreground mask, and zero otherwise. The same goes for $a(\cdot,\cdot)$, which returns one if the evaluated pixel is part of the active area. Following that step, the aforementioned *density map* (D) is obtained. As explained previously, this signal is compared to a uniformly distributed (i.e. random) signal (U_D) of the same size as D, using equation (3.6). The final density score is therefore given as:

$$\rho = 1 - I'(D; U_D) \ . \tag{3.8}$$

Similarly, to obtain the entropy score ($\mathcal{E}$), a Farnebäck's dense optical flow [72] of two consecutive frames is calculated. Once obtained, the flow vectors (F) are split into magnitudes (M) and angles (Θ):

$$F = \{\Theta, M\} \ . \tag{3.9}$$

Using the magnitudes as a threshold to filter out motion vectors that are due to



(a) Example frame (b) Binary mask (c) Samples used for density

Figure 3.3: A video frame from the 'Airport' sequence, with a manually annotated mask marked as a green polygon (a); the binary mask obtained (b); and, the sampling used for density estimation: points in black are separated by δ pixels, the red box is a sampling window D_w, of size $L \times L$.

noise, the angles are collected into a directions-of-motion (E) map. Furthermore, those that are not part of the active area can be filtered out.

$$E = \theta_{i,j} \in \Theta \mid m_{i,j} > \tau_m \ \vee \ a(i,j) \quad \forall \theta_{i,j} \in \Theta, m_{i,j} \in M \tag{3.10}$$

where $\theta_{i,j}$ and $m_{i,j}$ are elements of the angles map Θ, and magnitudes map M, respectively; and τ_m is a motion magnitude threshold set experimentally. The set of all vector angles within these constraints comprises the direction-of-motion (E) map. As with the density, this map (E) is compared to a random signal of the same size (U_E) using mutual information, from equation (3.6), and the final entropy estimate is given as:

$$\mathcal{E} = 1 - I'(E; U_E) \ . \tag{3.11}$$

3.4 Experimentation and Results

To test the proposed method, benchmark sequences of the publicly available UCF crowds dataset [5] have been used. As for the parameters, δ is set to be 5 pixels and L is set to be 20 pixels (windows are 20×20 in size). These values have been chosen experimentally (see Table A.1 in the materials Appendix, p. 170). As explained, *active*

area masks are manually annotated as polygons, and used in the process as already described.

A number of sequences from the dataset were selected for the experiments on the basis that the set-up of the camera needs to be such that the apparent size of people closer to the camera or further away from it is not very different. This is achieved by placing the camera on a high vantage point and tilted towards the floor.. Unfortunately not all videos in the dataset conform to these constraints, therefore a subset of video sequences showing these characteristics was selected. The initial intention was to have a large number of videos, labelled by a large number of volunteers via a *crowd-sourced* labelling platform. However, a previous stage to this would be to test it in a representative subset of videos. Therefore, the selection of videos was made in such a manner that the videos would contain a variety of scenarios. For instance, the 'Running' sequence contains motions in one single direction (low entropy) and a high density (it shows an urban marathon), the 'Motorway' sequence has similar characteristics, but featuring cars. On the other hand, the 'Crossroad' sequence shows cars moving in several directions (but with medium-low entropy) and a medium-low density. As a different example, the 'Street' sequence shows a crowd with very diverse directions of motion (high entropy), and high density. Example frames for the selected sequences are shown in Figures 3.6 and 3.7 in the Discussion section below (additionally, in the materials Appendix, p. 171). Human labelling was provided by five volunteers, who labelled each video continuously. That is, users had to provide labelling while the video was playing at normal speed, and not on a frame-by-frame basis.

To be more specific, the volunteers used a purpose-made application for ground truth labelling, a snapshot of which can be seen in Fig. 3.4. Once the user loaded the video sequence to label (1), they could see the progress of their labelling on the progress bars (2). After that, they would follow the instructions in the text box to the left (3). Then, the user would watch each video sequence twice. On the first pass, they would be asked to label the density of the crowd using the top slider from the set of sliders at the bottom of the screen (4). Ground truth was collected in real-time, that

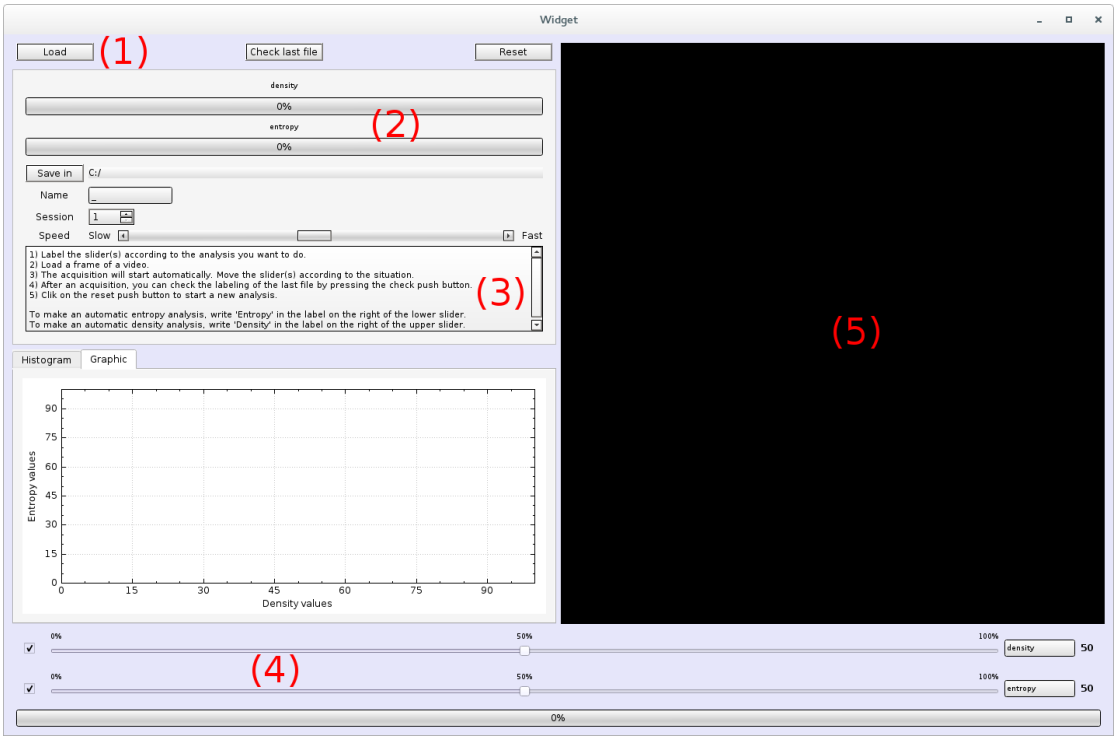


Figure 3.4: An image of the graphical user interface used for ground truth collection (a bigger version can be found in Appendix A). The users could load videos (1) and see progress of their labelling (2). The sliders at the bottom (4) were used for manual ground truth annotation of the video shown to the right (5). Instructions were given to the user in the text box to the left (3). All other widgets were used to visualise the output of the presented method (i.e. the automatic response).

is, as the video continued to play on the black box marked (5). On a second pass, they would be asked to label the entropy (orderliness or lack thereof) of the crowd, using the bottom slider of the set (4). Regarding what was to be understood as 'orderliness', the volunteers were instructed to look for the coherence in motion of the people in the scene, that is, whether objects were moving in the same or different directions.

Although a larger number of participants is always desirable, the number of volunteers is justified by the fact that what is assessed is a physical measure (e.g. density), and, since variability and error are generally small in such cases, the *effect size* can be considered to be large, and therefore there are no objections on the use of a small sample size [263].

The labelling information gathered from all the volunteers is shown in Fig. 3.5 for four of the employed sequences (left-hand plots in each sub-figure, with density on

the top, in blue, and entropy at the bottom, in green). The result obtained by the proposed method can be observed on the right column of the sub-figures.

The $[\rho\ \mathcal{E}]$ space is split into four quadrants, to allow for a discrete labelling of a scene (as low or highly crowded, and low or highly orderly). In order to evaluate the proposed method, each $(\rho, \mathcal{E})$ signature component is binarised into $Q_\rho(t)$ and $Q_\mathcal{E}(t)$, respectively, to obtain the correct quadrant Q where each $(\rho, \mathcal{E})$ signature falls into. The quadrant is calculated as follows:

$$Q(t) = \big(Q_\rho(t), Q_\mathcal{E}(t)\big) \tag{3.12}$$

$$Q_x(t) = \begin{cases} 1, & \text{if } x\,(t) > 0.5 \\ 0, & \text{otherwise} \end{cases}, \quad x := \{\rho \,|\, \mathcal{E}\}\ , \tag{3.13}$$

where x in (3.13) refers to either density (ρ) or entropy ($\mathcal{E}$). Similarly, for the human labelled sequences a binarisation into quadrants is also applied:

$$H_x(t) = \begin{cases} 1, & \mu_x(t) + \sigma_x(t) > 0.5 \\ 0, & \mu_x(t) - \sigma_x(t) \leq 0.5 \end{cases}, \quad x := \{\rho \,|\, \mathcal{E}\}\ , \tag{3.14}$$

where $\mu_x(t)$ denotes the mean ground truth value (averaged over participant response) for a particular frame t, with $\sigma_x(t)$ standard deviation. The final success rate s for each sequence is then calculated as:

$$s = \frac{1}{N} \sum_{t=0}^{N} \delta(H_\rho(t), Q_\rho(t))\ \cdot\ \delta(H_\mathcal{E}(t), Q_\mathcal{E}(t)) \tag{3.15}$$

where $\delta(\cdot, \cdot)$ denotes a function that returns 1 if both values are the same, and 0 otherwise; and N is the number of frames in the sequence.

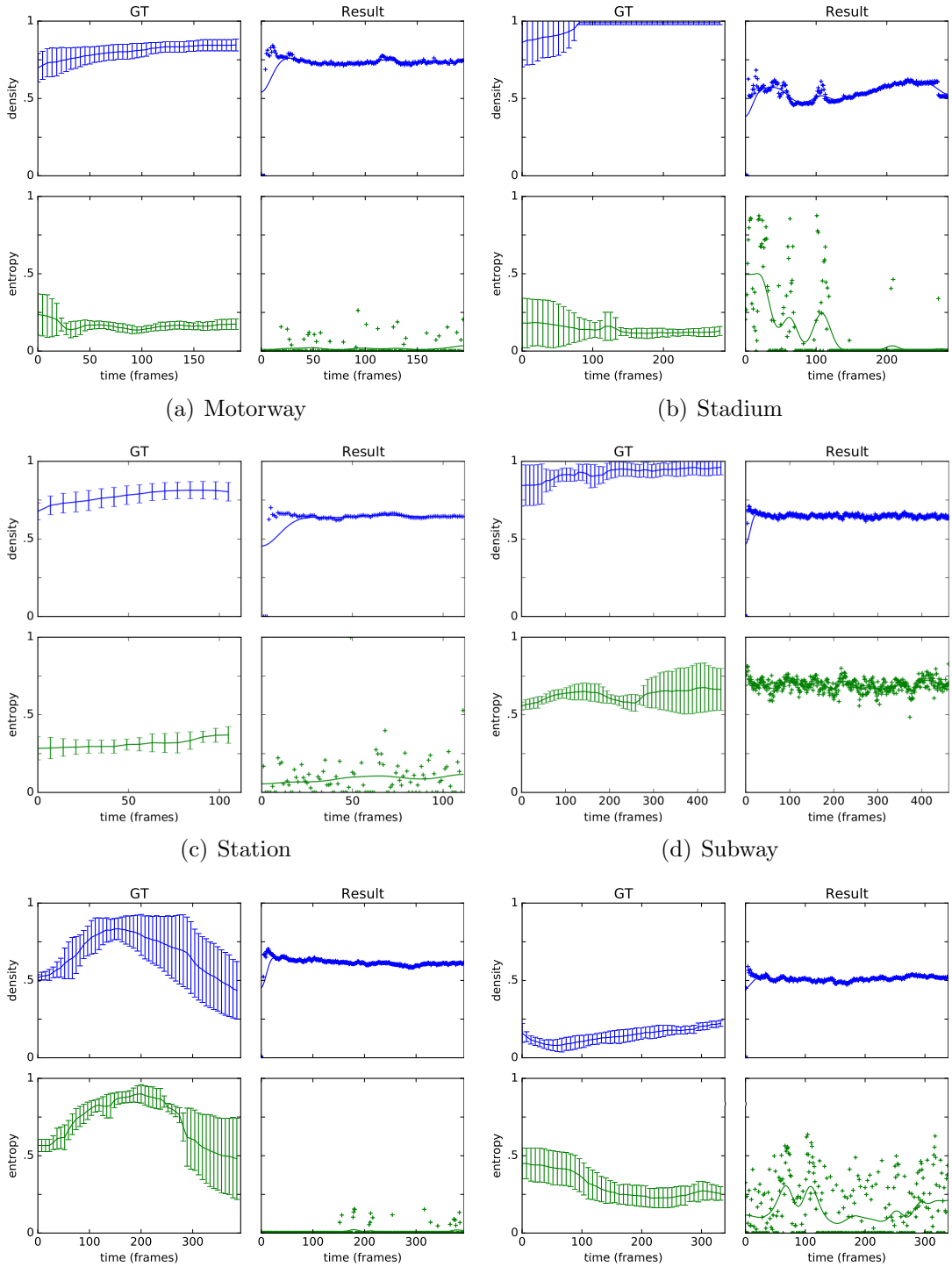
Table 3.2 shows the quantitative results for the method, for all ten evaluated sequences. The first two results columns show the marginal (i.e. total) estimator results, that is, the average percentages of successfully classified values for one estimator. This will allow us to determine how good the estimations for density and entropy were.

The other four columns show the number of estimations that were correctly classified for *both* dimensions (i.e. the estimation was in the same quadrant as the human label), and the number of instances that were misclassified in one dimension, but not the other (partial failures as 'ρ-only' and '$\mathcal{E}$-only' columns), or fully misclassified (*fail*). The last two blocks of rows show averaged results over all sequences: mean and standard deviation; followed by the median and the median absolute deviation (MAD).

Additionally, the right-hand side plots in each subfigure in Fig. 3.5 show the output results of our method for the four selected sequences for which the human labelling is given. Successfully classified instances are 98%, 67%, 88% and 98% for the 'Motorway', 'Stadium', 'Station' and 'Subway' sequences, respectively (shown in boldface in Table 3.2).

Sequence	Estimator results		Failure cases			Success
	ρ	$\mathcal{E}$	*fail*	ρ-only	$\mathcal{E}$-only	*both (s)*
Escalator	0.37	0.40	*0.52*	0.11	0.08	0.29
Running	0.95	0.70	0.00	0.05	0.30	0.65
Motorway	0.98	1.00	0.00	0.02	0.00	**0.98**
Airport	1.00	0.26	0.00	0.00	*0.74*	0.26
Station	0.94	0.94	0.00	0.06	0.06	**0.88**
Subway	0.98	0.98	0.02	0.00	0.00	**0.98**
Stadium	0.77	0.90	0.00	0.23	0.10	**0.67**
Crossroad	0.98	0.60	0.00	0.02	0.40	0.57
Market	0.26	0.93	0.04	*0.70*	0.02	0.24
Street	0.80	0.62	0.01	0.19	0.37	0.43
Mean	0.80	0.73	0.06	0.21	0.14	0.59
Std. dev.	0.26	0.25	0.15	0.23	0.20	0.27
Median	0.94	0.80	0.00	0.09	0.06	0.61
Med. abs. dev.	0.05	0.18	0.00	0.09	0.06	0.29

Table 3.2: Crowd classification results for the analysed sequences



(a) Motorway (b) Stadium

(c) Station (d) Subway

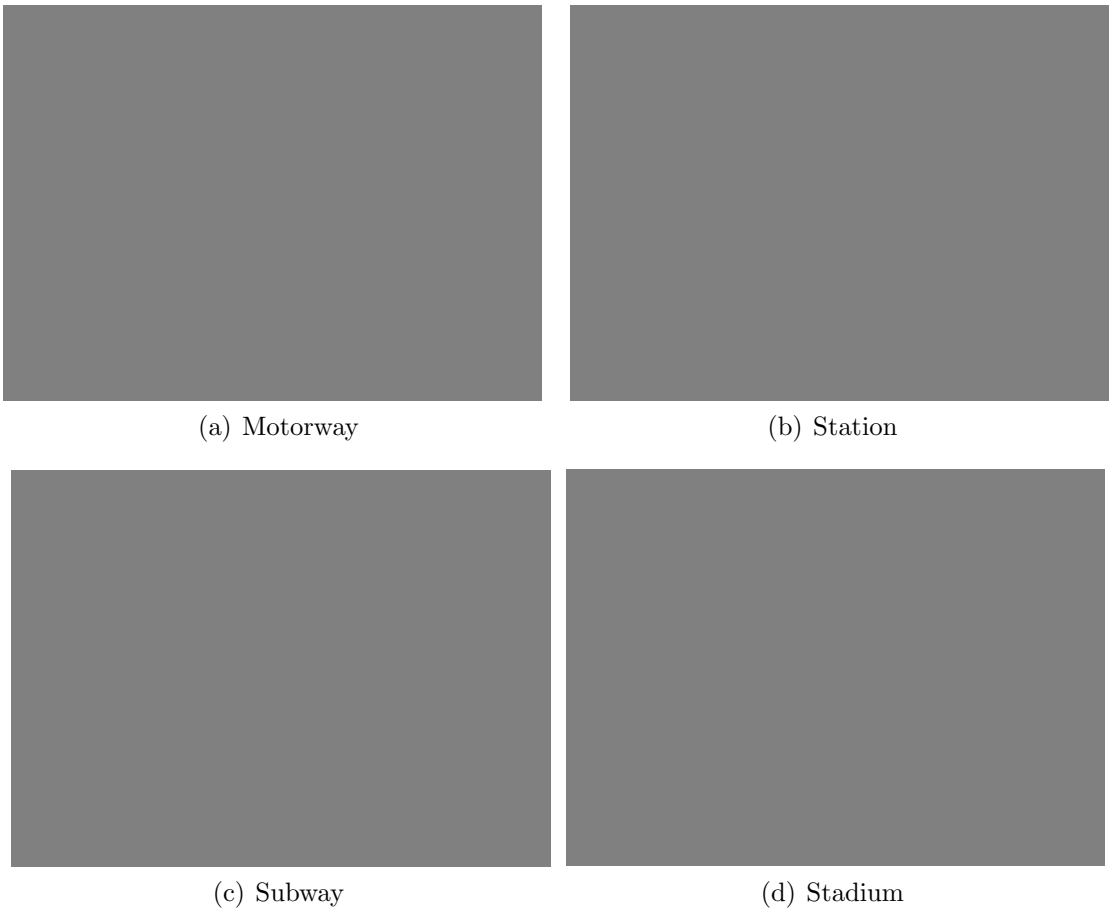
(e) Airport (f) Market

Figure 3.5: Analysis results for the best-performing sequences (a–d), and worse-performing ones (e–f) each sub-figure shows the human-labelled ground truth average and standard deviations (left column) and estimations of the presented algorithm (right column) for density (top row, in blue) and entropy (bottom row, in green). Results for other sequences can be found in Appendix A.

3.5 Discussion

The reader will observe from Table 3.2, that for a majority of the evaluated sequences, the number of instances that were classified correctly on both dimensions (i.e. shown in the '*both*' column) are greater than the sum of the miss-classifications of any nature by at least 10 percentage points. The average success rate is close to 60% (median and mean up and down by one point each, respectively). However, for the 'Escalator', 'Airport', and 'Market' sequences, miss-classifications are much larger than the average values. This negatively impacts the mean values for the failure cases, as can be observed by the large standard deviations as well as the differences between the mean and the median, which is known to be more robust to outliers.

Figures 3.6 and 3.7 show frames for the ten sequences used for evaluation, grouped by performance. Figure 3.6 shows correctly classified sequences, and Fig. 3.7 shows example frames of sequences with intermediate and lower results. As can be seen, the nature of the videos themselves is not very different, however, several factors could help explain the oddly large misclassification results on the mentioned sequences (values in italics in Table 3.2). Observing Fig. 3.5, it can be seen that in the 'Airport' sequence the entropy fails in most cases, since the underlying optical flow seems to have problems in matching the motion of pixels between frames, and therefore the entropy estimations are zero in most cases. This seems to be related to the fact that this video was edited so motion appears very smooth. Furthermore, the density score for the 'Airport' sequence is high due to the discretisation into quadrants, but the density signal over time does not vary as the human ground truth does. This is likely due to the fact that the illumination conditions of the video are not optimal for the employed background subtraction algorithm. On the other hand, in the 'Market' sequence, density estimations fail. It is worth noting that this scene is very cluttered, with pillars and banners preventing density estimation. Finally, in the 'Escalator' sequence, both detectors fail, this could be caused by video editing and a very large appearance change in the size of the targets due to the camera perspective.

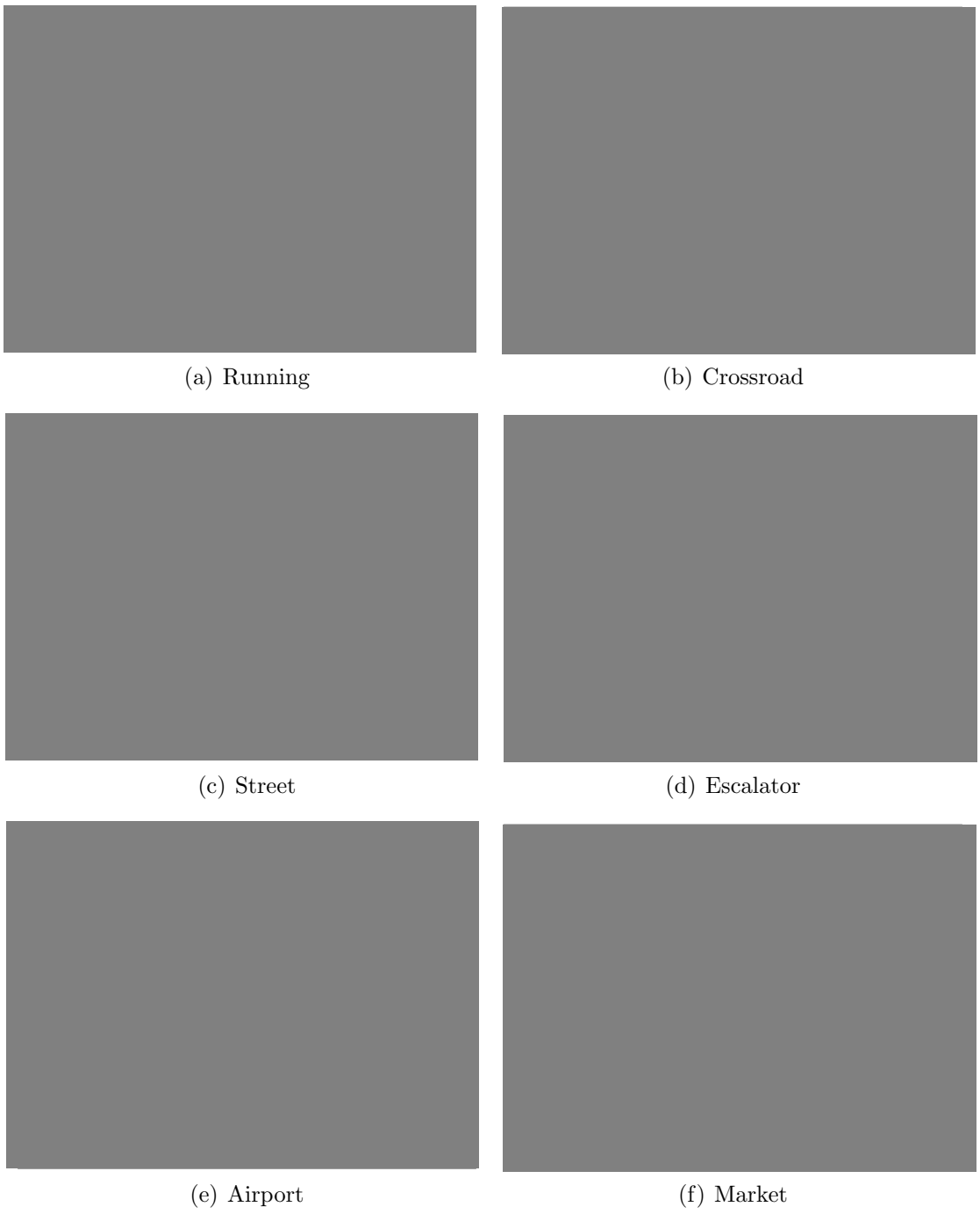


(a) Motorway (b) Station

(c) Subway (d) Stadium

Figure 3.6: Example frames of correctly classified video sequences.

Furthermore, if looking at Fig. 3.5, and in general for all evaluated sequences, it can be seen that density is overall closer to the human labelling average, if somewhat lower or higher; whereas the entropy seems to be less correlated to the human labelling data. This is also shown by the totalled 'estimator results' shown in the first two columns of Table 3.2. These totals (marginals) have been calculated as the sum of all correctly classified instances for that estimator (regardless of the result of the other estimator). Additionally, Fig. 3.8 shows how density and entropy perform, as an 'error tolerance' is increased. That is, a point in the curve represents how many estimations are within the boundaries of the mean value (i.e. using the actual values, not the quadrants) for the human-established ground truth, given that the system *tolerates* a certain error margin. It can be seen that density performs generally better than entropy. For instance, with an accepted error of 0.2 in the score, slightly more than



(a) Running (b) Crossroad

(c) Street (d) Escalator

(e) Airport (f) Market

Figure 3.7: Example frames of sequences with intermediate results (a–c), and misclassified sequences (d–f).

50% of the estimations are correct for the density value, whereas correctly estimated entropy values are around the 40% mark. This indicates that efforts to improve success rates should be aimed at improving entropy estimation.

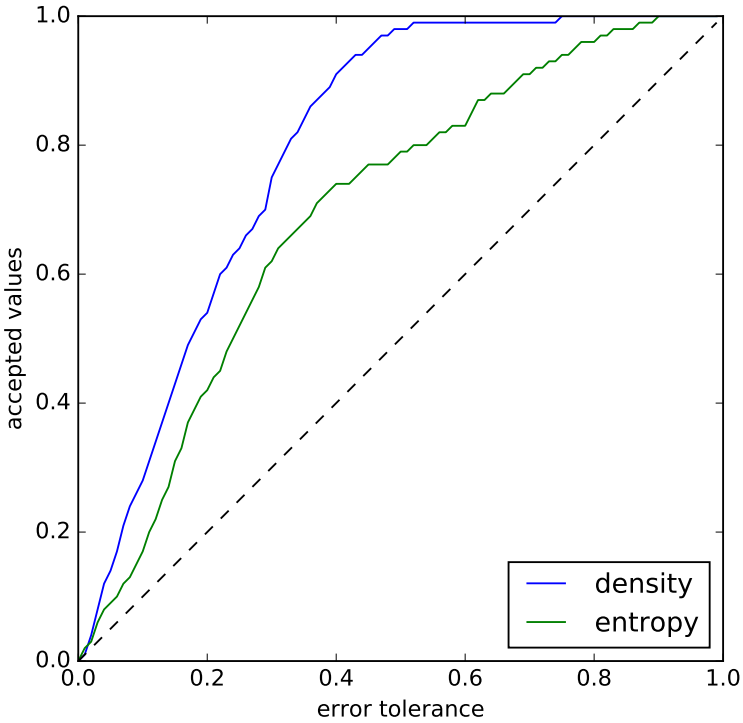
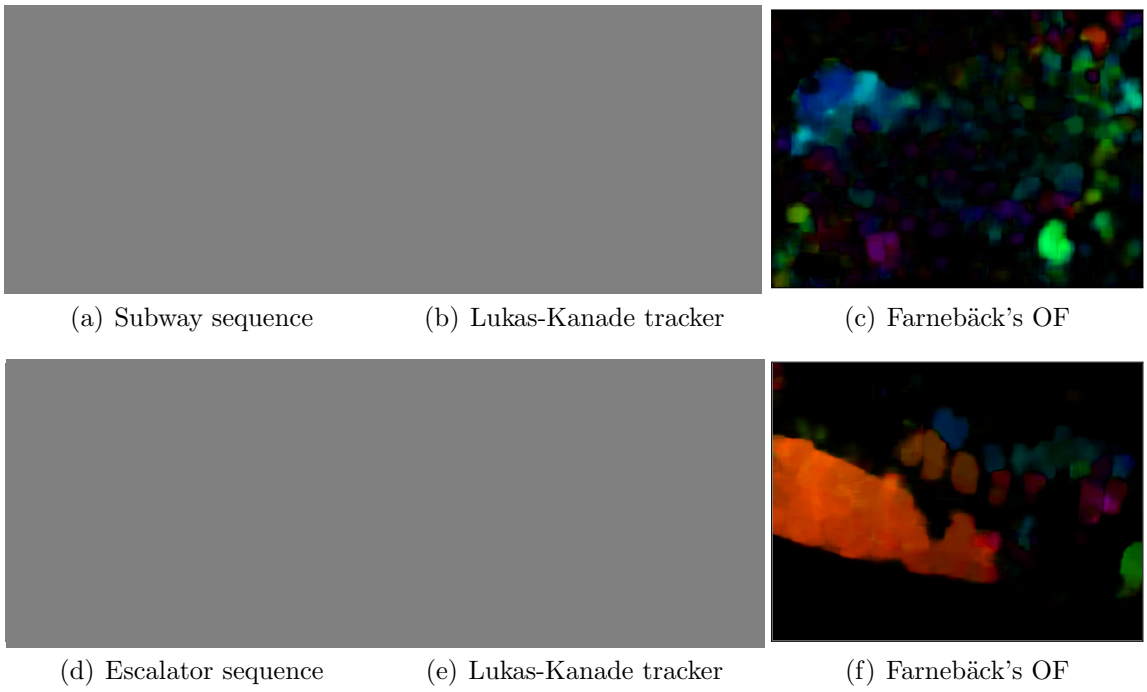


Figure 3.8: Error tolerance plot, showing how fast the number of accepted estimations increases as accepted error tolerance is increased. The reader will observe the density estimation (blue) is generally better than entropy estimation (green). Increasing entropy estimation accuracy would increase the area-under-the curve (AUC), and would therefore benefit the final combined response.

Regardless of this, low performance in density estimation is, in most cases, linked to the fact that the foreground detection algorithm used has a restrictive threshold set that could be lowered in order to allow more pixels to be part of the foreground mask. Yet, these values can be dependent on the dataset or even the sequence, and therefore it is out of the scope of this work to dynamically adapt that threshold. Regarding entropy estimation, in a first approach a sparse optical flow (Lukas-Kanade, LK) tracking algorithm was used [148, 218], but due to its sparseness, and lack of correspondence in some situations, it was impossible to determine the values of entropy correctly (Fig. 3.9, top). Using a dense Farnebäck optical flow [72], other problems

arise, such as a more noisy correspondence, which leads to false detections of motion, and therefore wrong estimations of entropy values (Fig. 3.9, bottom). Nevertheless, Farnebäck's algorithm is only one of the multiple alternatives for dense optical flow calculation, some more recent works exist [211, 262], which claim to obtain a less noisy angle and magnitude estimation for the flow, as well as better border preservation. However, as much as it could benefit the performance, since this is an exploratory work, it is not of critical importance to find the optimal flow estimation method.



(a) Subway sequence (b) Lukas-Kanade tracker (c) Farnebäck's OF

(d) Escalator sequence (e) Lukas-Kanade tracker (f) Farnebäck's OF

Figure 3.9: Issues encountered with different optical flow algorithms for the estimation of entropy. Top row: (a) 'Subway' sequence, (b) LK algorithm is not able to track due to lack of correspondence, and (c) Dense OF is able to find correspondence. Bottom row: (d) 'Escalator' sequence, (e) LK has no trouble finding correspondence, but (f) a dense OF in this case is noisy and does not preserve borders.

3.6 Conclusion

In this chapter a density-entropy signature was introduced as a way to classify crowded scenes. By combining these two cues, each frame of a series of crowd video sequences could be given a $2D$ point in the density–entropy space. The results shown look

promising, illustrating the potential of the proposed method. To show this, some qualitative results were presented, as well as some quantitative results for a selection of sequences from a known dataset. Further exploration of this and similar methods seem a good idea, given their potential. These methods could lead to benefit society both regarding its safety and efficient organisation of crowds in urban and other settings.

Chapter 4

Telemetry-based airborne video surveillance methods

Chapter highlights: Ego-motion of an UAV is compensated using telemetry information and used in two different applications: visual tracking and background modelling.

Overview

In this chapter, the objective is to develop novel methods to perform specific video surveillance tasks from an airborne camera. Classically, pre-processing of the video is normally carried out to detect moving objects in the scene. Typically, this would entail segmentation via background modelling, or some other means of detection (e.g. a histogram of oriented gradients –HOG– detector [61]). Once the video has been pre-processed, other algorithms can be applied, for instance, to track the detected moving objects, alternatively further processing can help classify detected objects into categories (i.e. humans, vehicles, etc.) and subsequently work only on relevant targets (e.g. it might not be important to track vehicles, but only humans).

Nevertheless, in this chapter, and following a chronological line of how the presented contributions were explored, a first contribution will show how to avoid frame

registration for tracking, and instead find the homography to only correct a tracker's search window. This is much faster than the classical workflow, where homography is calculated and then full frame registration or mosaicking is applied. The necessity for full frame registration, however, is dependent on the application. As will be seen, it is unnecessary for tracking, but unavoidable for background modelling. Precisely, that is the second contribution of this chapter, an algorithm for background modelling using telemetry-based homography estimation with a *global registration* refinement step for frame registration and subsequent background modelling.

Specifically, in the first contribution a novel ego-motion compensation approach is presented, that transforms the local search window of the visual tracker. This is much more computationally efficient, and can be applied regardless of the amount of texture in the background. This is justified by the fact that tracking from airborne cameras is very challenging, since most assumptions made for fixed cameras do not hold. Therefore, compensation of platform ego-motion is seen as a necessary pre-processing step. Most existing methods perform image registration or matching, which involves costly image transformations, and have a restricted operational range. Experiments with ground truth and tracker output data are conducted and show the validity of the approach.

In the second contribution, an approach to detect moving objects from Unmanned Aerial Vehicles (UAV) is presented. A common framework for most of the existing techniques is using image registration to warp consecutive frames as an ego-motion compensation step and applying frame differencing to detect moving objects. Under the assumption of a planar scene, it is proposed to exploit telemetry information available from Global Positioning and Inertial Navigation Systems (GPS/INS) to estimate a similarity transformation matrix that would map the image points from one frame to another. It is shown that telemetry-based image registration, combined with global registration methods, produces more accurate results than the traditional image registration techniques in case of a scene with poor or no texture. To segment moving objects, a probabilistic background modelling method with mixture of Gaussian

distributions is employed.

Main contributions, outcomes, and publications

To summarise, there are two main contributions to this chapter, which translated to two publications, as presented in the next list. Both are connected by the fact that both were devised using the same type of UAV in mind, that is, an octo-copter, and taking advantage of telemetry data provided by the vehicle. These contributions are:

- A search window correction method for airborne tracking [54], and
- a frame registration and background modelling technique [245].

This chapter also introduces a dataset (Sec. 4.4.1), formed of a series of video sequences collected using the prototype vehicle used in the project, with two different camera and telemetry sensor set-ups (please see Sec. 4.2 for details).

No previous work introduces the use of telemetry and video data in combination to transform the local search window of an object tracker as a more efficient ego-motion compensation method that does not require image transformations. Also, no techniques appear to combine SIFT point homography estimation with global registration refinement for background modelling (used for comparison to the proposed method), or use telemetry combined with a global registration refinement (as in the proposed method).

4.1 Introduction

Although UAVs were primarily designed for military purposes, they have gained considerable popularity with the recent production of small UAVs for civil and commercial applications. Decreasing costs due to developments achieved in embedded technologies have led to an increasing use of video analysis using UAVs equipped with cameras for applications such as agriculture and natural preservation [243], traffic monitoring [216], or emergency response [12].

One of the most common applications of UAVs is video surveillance in remote or inaccessible areas where stationary surveillance cameras are absent. In this case, the primary goal of the UAV is the detection and tracking of moving targets. For specific terminology associated with aerial video surveillance and a general framework, the reader is referred to the work by Kumar et al. [123].

Using an aircraft platform introduces noise such as vibrations, rocking, locomotion making it difficult for tracking algorithms relying on the *smoothness assumption* (seen in Chapter 2, Sec. 2.4.5) to work properly. It also leads to a highly dynamic and constantly changing image background. Therefore compensation of platform ego-motion is necessary before commencing with the actual image processing techniques. Alternatively, background-independent methods would need to be used, although they do not abound.

4.1.1 Tracking from airborne cameras

Most existing techniques for ego-motion compensation tackle the problem by applying image stabilisation [216], camera pose estimation [215] or image matching (or registration) [101, 196]. These techniques are costly, as compared to the proposed approach, and have a restricted operational range (i.e. will not work with backgrounds showing poor texture). In the following, video-based, hybrid, and telemetry-based methods for ego-motion compensation will be further discussed.

Homography estimation has been extensively used for many applications in the field of computer vision, and specifically, it has been used in moving vehicles, both terrestrial and aerial, as well as robots, for the compensation of the motion of the used vehicle (or ego-motion). This pre-processing step allows for frame differencing to be calculated, and as such, it allows many fixed-camera methods (and assumptions) to be employed.

In the Background chapter (Sec. 2.4.5), it was seen that the most common approach to perform ego-motion compensation is through image registration or image correspondence, achieved by corner or interest point detection, and the random

sample consensus (RANSAC) algorithm to find the homography between consecutive frames [148, 170, 187, 189, 198, 279, 280] or to create a map of the explored area [38, 176]. However, corner-based techniques cannot work on homogeneous (i.e. poorly textured) backgrounds, or when the only available texture is that of moving objects (e.g. when flying over a very crowded scene). Therefore, other methods propose to use global positioning and inertial navigation systems (GPS/INS) [27, 66, 92]. Even so, given the computational overhead of image warping, it might be possible to avoid it in some applications where frame differencing is not necessary (e.g. tracking), as is done in one of the methods presented here (Sec. 4.3.1).

4.1.1.1 Planarity and orthogonality assumptions

Several of the studied works make an assumption about the orthogonality of the axis of the camera to the ground plane, where the UAV is hovering over the plane, and thus the roll (ψ) and pitch (θ) angles are near-to-zero all the time [12, 240]; as well as the assumption that at enough distance from the ground, the surface inside the field of view (FOV) of the camera, is planar, i.e. 'planarity assumption' [37, 92, 189, 286]. This allows the $3D$ problem to be constrained to a plane ($2D$), and also avoids having to use complex $3D$ models of the ground. By doing so, the calculations can be simplified. Besides, in the reviewed works, the camera is assumed to be co-axial with the centre of mass of the UAV, and the offset in position between the GPS/INS devices and the camera is negligible, and the FOV angles are known, or have been calculated.

4.1.2 Background modelling

As opposed to what is said above, in order to apply the common background modelling techniques for a video captured from a UAV, it is necessary to register consecutive frames. The most popular frame registration methods are based on feature point detection and matching to calculate the homography that describes the correspondence between frames. The advantage of this method is that the large number of correspond-

ing points allows the estimation of an 8-DOF [1] homography that can describe any kind of camera motion. On the other hand, a major drawback of this method is its dependence on the texture and structural information of the scene. Imagine a scene with poor texture like a tarmac, or a scene with a repetitive texture like vegetation. In the first case, the feature points would most probably be located on the moving object; while in the second case, the matching of feature points would be inaccurate. For this reason, the need for more robust techniques becomes prominent. In this section some of these techniques will be reviewed.

To detect independent motion in an airborne video, one can use consecutive frame registration and motion segmentation by optical flow (OF) [57]. Although this method is general and applicable to many situations, it fails when the target and the camera have the same motion pattern. Alternatively, feature detection and matching between two consecutive frames can be employed to determine the homography that describes the image transformation. It is assumed that, while calculating the homography, the feature points rejected as outliers by means of LMedS (Least Median of Squares) [222] or RANSAC (RANdom SAmple Consensus) [250] belong to objects exhibiting independent motion. The outliers are clustered [222, 250] to form regions that enclose the moving objects. Otherwise, the homography is estimated to register the consecutive frames as a camera motion compensation step to apply common background subtraction or frame differencing techniques [2, 4, 30, 141, 172].

The proposed background modelling method (Sec 4.3.2) is related to the latter of the mentioned approaches, therefore, it is important to give an insight into the related techniques. Alignment of consecutive frames can be achieved either by employing methods for global registration, or feature point detection and matching, or a combination of both. Global registration methods are restricted to detecting only translational and rotational motion, while feature-based methods are capable of producing an affine, or even projective, transformation matrix. Global registration methods used for aerial image registration include mutual information [172] which corrects only the translation,

[1] degrees of freedom

and region phase correlation [213] which estimates 8-DOF homography. Feature-based methods include a wide gamut of approaches as those using Harris corners [99] [4], SURF features [2], SIFT features [141, 195] and Shi & Tomasi corners [49]. Approaches found to combine both techniques use SIFT features with mutual information [141], Harris corners with efficient second order minimization [192] and Harris corners with gradient-based alignment [4]. The COCOA system [4] has attracted special attention, as well as its successor COCOALIGHT [30], since they both use a gradient-based registration first introduced by Mann et al. in [157]. Their detailed experiments prove its superiority over other above-mentioned feature-based registration techniques.

Once the camera motion compensation step is performed, moving objects can be detected by several methods, such as: frame differencing [2, 13, 49], accumulative frame differencing [4, 30, 99], median background subtraction, statistical mode background subtraction [141], or normal optical flow [172]. Since frame differencing techniques do not segment the whole object, image segmentation techniques are used to improve the results [2, 99]. Probabilistic background modelling, such as Gaussian Mixture Model (GMM) are avoided according to [30] given that: first, the frame rate is not high enough to learn the rapidly changing background; also, there is an accumulated alignment error because of consecutive homography computations; finally, there are errors produced due to parallax. However, in the proposed method it is shown how GMM can be effectively used for moving object detection.

4.2 Context: the 'OctoXL' UAV platform

Before delving into the methodology, it is worth introducing the employed UAV platform a bit more. For the PROACTIVE project, the team working at the Institute for Flight Systems (IFS) of the Universität der Bundeswehr in Munich designed and built two prototypes. A self-constructed vertical take-off and landing (VTOL) platform with eight electric propelled motors was employed in both cases (see Fig. 4.1). The OctoXL is based on a construction Kit from HiSystems GmbH. It is worth noting, however, that due to the evolution of the prototype, the works that will be introduced

Figure 4.1: An image of the employed UAV platform (showing hardware of the first configuration).

next (in Sec. 4.3) used two different set-ups or configurations of the employed vehicle. Additional details for both set-ups are provided in Russ et al. [206] and Stütz et al. [235]. These configurations were used for tracking correction (Sec. 4.3.1) and background modelling (Sec. 4.3.2), respectively. In both cases the video is transmitted wirelessly and in real time to a server where the processing (i.e. the presented algorithms) will run. Furthermore, also in both configurations, the camera is mounted orthogonal to the plane defined by the propellers.

4.2.1 First configuration

This configuration was used to capture the video sequences used for the method presented in Sec. 4.3.1. In this case, the UAV is equipped with an embedded computer board with Intel Core i7 processor, a solid-state drive (SSD) and a VRMagic camera with a resolution of 752×480 pixels capturing video at 15 fps. The utilised Lensagon lens has 3.5 mm focal length. The UAV flies at an altitude of 10 to 15 m above ground. The aircraft is equipped with a Xsens MTI-G inertial measurement unit (IMU) with 2.5 m position accuracy and 0.25° angular accuracy, providing inertial data at 120 Hz. GPS data is provided at 4 Hz.

4.2.2 Second configuration

In this case, the UAV is equipped with a different camera; the resolution of the acquired video is 512×512 pixels at 30 fps[2] and the utilized lens has 16 mm focal length. The UAV flies at an altitude of 10 to 15 m above ground. The aircraft is equipped with a GPS with a 2.5 m position accuracy that is updated at a rate of 5 Hz and 0.5° angular accuracy for the inertial data, which is provided at a rate of 100 Hz. The UAV and the hardware used in this occasion are described in [235].

4.3 Methodology

The methods that will be presented in this section, as stated, rely on the telemetry information provided by the UAV sensors. Using this information, and taking into account some assumptions, the tasks can subsequently be performed. Since the UAV is an octo-copter (as seen in Sec. 4.2), the planarity and orthogonality assumptions (Sec. 4.1.1.1) can easily be made. This, along with knowledge of camera parameters (such as FOV angles, see Fig. 4.2), constrains the problem to a $2D$ plane which facilitates the math.

Therefore, taking advantage of the telemetry information, the proposed algorithms obtain all current camera (vehicle) pose parameters for each video frame, in the form of a tuple:

$$d_{telemetry} = [(\varphi, \lambda),\ (\psi, \theta, \phi),\ h] \,, \tag{4.1}$$

where the pair (φ, λ) represents the latitude and longitude in degrees from Equator and Greenwich meridian, respectively; ψ, θ, and ϕ represent the roll, pitch and yaw angles in degrees, respectively (all with relation to the upright, north-facing position); and h represents the current altitude (in meters) from the ground. In the first OctoXL configuration (seen in Sec. 4.2.1), these had to be manually calculated as:

[2]The video is acquired at 30 fps, but then down-sampled to 8 fps for project-related reasons.

$$h = h_{(\mathrm{ASL},t)} - h_{(\mathrm{ASL},0)} \text{ [m]}, \tag{4.2}$$

where $h_{(\mathrm{ASL},0)}$ is the initial ground altitude above the sea level (ASL) of the UAV before take-off, and $h_{(\mathrm{ASL},t)}$ is the current ASL from the telemetry reads. For the second configuration (seen in Sec. 4.2.2) the h is calculated by the on-board computer. Some intrinsic camera parameters such as focal length f, sensor active area width W_{sensor}, and optical centre are known *a priori*. Another important parameter that should be calculated is the camera field of view angles (FOVs) which are defined as:

$$FOV_W = \mathrm{atan}\frac{W_{sensor}}{2f} \quad , \quad FOV_H = \mathrm{atan}\frac{H_{sensor}}{2f} \text{ [rad]} \tag{4.3}$$

To convert the pixel positions, the information about the altitude of the vehicle and the FOV angles of its camera are employed. This allows estimation of the width and height in meters (W_m, H_m) of the area covered by the camera via trigonometric rules (see Fig. 4.2). Furthermore, a ratio r_c can establish the conversion between the image pixels and meters of the *real* area covered (i.e. the field of view):

$$\begin{aligned} (W_m,\ H_m) &= (2h \cdot \tan \mathrm{FOV}_W,\ 2h \cdot \tan \mathrm{FOV}_H) \text{ [m]} \ , \\ r_c &= W_m / W_{img} \text{ [m/pixels]} \ . \end{aligned} \tag{4.4}$$

where W_{img} is the width of video frame in pixels (from the camera's resolution), alternatively, the height of the image could be used (the ratio should be the same).

It can also be observed that there is a discrepancy both in units (degrees, meters, pixels) as well as in coordinate systems (GPS, UAV and image) employed. For this reason, a common framework is introduced, expressing all geo-location data in meters, except for the yaw (φ) which is expressed in radians (the other two angles will not be used in the calculations, as they are assumed to be zero or negligible).

For the conversions of the latitude and longitude data (WGS-84 standard) provided by the inertial measurement unit (IMU), a simplified version of the Universal Transverse

Mercator (UTM) conformal projection coordinate system is used. UTM is a cylindrical projection separating the surface of the earth in 6 degree-wide zones. The position of an object is given in a zone, a band (or an hemisphere), the *northing*, and the *easting* value. Within a zone, a Cartesian coordinate system is used with the northing and easting values expressed in meters. The easting is given from the zone's initial easting and the northing is given from the Equator.

The translation of the UAV over the surface of the world is calculated as the difference in northing and easting values $(\Delta N, \Delta E)$:

$$\Delta N = N_t - N_{t-1} \quad \text{and} \quad \Delta E = E_t - E_{t-1} \tag{4.5}$$

In subsequent calculations, the change in yaw of the vehicle $\Delta\phi$ will be needed. This is the difference between the current and the previous values. In the first configuration (Sec. 4.2.1), the yaw angle is given in degrees and is positive towards starboard and negative towards port: yet, in the second configuration (Sec. 4.2.2) the yaw is always positive and increasing towards the starboard (clockwise), therefore it needs to be normalised to the range $[-180°, +180°)$ first:

$$\Delta\phi = \frac{\pi}{180}\,\mathrm{sgn}(\phi_t - \phi_{t-1}) \cdot \min(|\phi_t - \phi_{t-1}|,\ 360 - |\phi_t - \phi_{t-1}|)\ [\mathrm{rad}] \quad , \tag{4.6}$$

where $\mathrm{sgn}(\cdot)$ is the *signum* function.

Knowing these parameters in the world coordinates; the translation, rotation and scaling in the image domain can be computed. Since the camera coordinate system is not necessarily aligned with the world coordinate system, to calculate the displacement in the image domain it is important to rotate the world coordinate systems clockwise to align it with the camera's. Consequently, if the yaw angle ϕ is expressed in radians, then the displacement Δx and Δy along the x and y axis in the image domain will be described by the following equations:

$$\begin{aligned} \Delta x &= (-\Delta N \cdot r_c^{-1} \cdot \sin(\phi_{t-1}) + \Delta E \cdot r_c^{-1} \cdot \cos(\phi_{t-1})) \ , \\ \Delta y &= -(\Delta N \cdot r_c^{-1} \cdot \cos(\phi_{t-1}) + \Delta E \cdot r_c^{-1} \cdot \sin(\phi_{t-1})) \ . \end{aligned} \tag{4.7}$$

Please note the difference in the calculation of Δx and Δy: an additional minus sign is used for Δy due to the change in coordinate systems (i.e. the x, y coordinates of an image pixel are counted from the top-left corner of the image, whereas northing N and easting E start from the equator, and the corresponding UTM zone start, respectively).

The last thing to take into account is the scaling effect produced by change in altitude. Therefore, the ratio between the previous (h_{t-1}) and current height (h_t) measurement will be used as an image scaling factor and is defined as:

$$r_h = h_{t-1}/h_t \ . \tag{4.8}$$

4.3.1 Method 1: 'Search window' correction for tracking

With all the information gathered from equations (4.1) to (4.8), in this section, a novel search window correction method to facilitate tracking from UAVs, based on the motion of an aerial vehicle is presented. Full image registration is shown to be unnecessary in this particular case, because of its computational overhead. In the proposed method transformation operations are applied on the 'search window' of the used tracker directly from one frame to another.

The visual tracker employed is a covariance tracker [190, 244], which was presented in the literature review (Chapter 2, Sec. 2.4.1.2), as an adaptive tracker. Apart from its adaptiveness to target appearance variations, this tracker was selected since the feature it employs, the region covariance matrix [244], can be used not only for tracking but also for re-identification [24, 202], which implies it is a very discriminative feature. The authors of the original work state that the method does not require a search window, as search can be performed on a reduced search-space by using an image with a quarter

of the original resolution. This, however, introduces a prediction inaccuracy. For this reason, in this work, it is used with the full resolution, but instead, using a search window to limit the computational cost and have a faster approach. Once the UAV is flying, a human operator can see the camera output, and decide on the rectangle of interest (ROI) enclosing a target of choice. The local search window (win) is then defined as a rectangle, enclosing the ROI, with an allowance or margin where the target might be re-detected in subsequent frames (for specifics about how the margin is set, the reader is referred to Sec. 4.4). At this point, the presented method recalculates the position of the local search window in the next frame, based on the movement of the camera mounted on the UAV. For every frame in the video feed, the tracker provides a ROI enclosing the tracked person, and a wider local search window (win) is calculated around it. The search window is expressed in pixels, with a coordinate pair that represents its upper-left corner (win_x, win_y), and its size (win_w, win_h). Once this information has been gathered, it is important to analyse which changes in the pose of the camera have the most influence on the apparent motion of the search window, taking into consideration the platform type (copter). Three different aspects are found to have the greatest effect on the window's apparent motion:

- The translation of the UAV along the X and Y axes (related to (φ, λ), because of the assumption of orthogonality introduced earlier),
- the translation of the UAV along the Z axis (changes in its altitudes, or h), and
- the rotation of the UAV about the Z axis (changes in its yaw, or ϕ).

Here the X, Y, Z axes are in the vehicle frame, that is, the X axis crosses the UAV from back to front, the Y axis crosses the UAV from left to right, and the Z axis crosses the vehicle from top to bottom (as depicted in Fig. 4.2). As it can be observed, the roll and pitch angles (ψ and θ) are not employed, because of the assumptions introduced earlier (Sec. 4.1.1.1). With all the data collected previously, and taking into account the aspects affecting the apparent motion of the search window, a correction for each of these aspects is proposed next.

4.3.1.1 Correction due to XY translation.

The first operation to obtain the corrected search window (win'), is to counteract for the motion in the X, Y axes, which are correlated to (φ, λ) and therefore to northing and easting (N, E), that is, translation of the UAV over the surface of the world. For this, the northing and easting value differences $(\Delta N, \Delta E)$ are needed from eq. (4.5), and then used to calculate $(\Delta x, \Delta y)$ respectively –in eq. (4.7)–, which are then used for the correction of the rectangle in the image space, calculated as:

$$win'_x := win_x + \Delta x \quad \text{and} \quad win'_y := win_y + \Delta y \ . \tag{4.9}$$

4.3.1.2 Correction due to altitude changes.

The second operation that is performed on the local search window, is related to its size. Due to the changes in altitude (h) of the UAV between frames, the apparent size of the target in the image changes, and as such, the local search window around the target must grow or shrink accordingly, so that an optimal size is maintained. To proceed, the ratio r_h, from eq. (4.8) among the altitude (h) values in the current and previous frame is used as a factor to resize the local search window:

$$(win'_w,\ win'_h) := (win_w \cdot r_h,\ win_h \cdot r_h) \ . \tag{4.10}$$

4.3.1.3 Correction due to the yaw changes.

In this last operation, the local search window is corrected to compensate for variations of the rotation on the Z axis of the UAV coordinate system (yaw or ϕ). Changes in yaw occur when the vehicle steers either when hovering over an area, or in conjunction with a translation in the XY axes. To apply this correction, the position of the new window is calculated based on the difference between the current and previous yaw values calculated as Δ_ϕ, as shown in eq. (4.6). First, the central position of the window is needed $(w_x,\ w_y)$; then, this point is expressed as a vector $\mathbf{c}$ from the centre of the video frame (o_x, o_y); after that, the rotation over Δ_ϕ is applied over that vector,

to counteract the rotation undergone by the vehicle, thus becoming $\mathbf{c}'$, as shown in eq. (4.11); finally, the coordinates are translated back to have their reference back to the top-left corner of the image, as originally:

$$\begin{aligned}
(w_x,\ w_y) &= (win_x + win_w/2,\ win_y + win_h/2) \\
(o_x, o_y) &= (W_{img}/2,\ H_{img}/2) \\
\mathbf{c} = (c_x,\ c_y) &= (w_x - o_x,\ w_y - o_y) \\
\mathbf{c}' = (c'_x,\ c'_y) &= (-c_x \cdot \cos\Delta_\phi + c_y \cdot \sin\Delta_\phi,\ -c_y \cdot \sin\Delta_\phi - c_y \cdot \cos\Delta_\phi) \\
(w'_x, w'_y) &= \left(o_x - c'_x,\ o_y - c'_y\right) \\
(win_x,\ win_y) &= \left(w'_x - win_w/2,\ w'_y - win_h/2\right)
\end{aligned} \tag{4.11}$$

Please note the inverted signs in the calculation of $\mathbf{c}'$ in eq. (4.11), since what is intended is to revert or counteract the effect of the ego-motion, and Δ_ϕ represents its magnitude.

4.3.2 Method 2: Background modelling

Taking into account the assumptions introduced in Sec. 4.3, the calculation of a projective transformation matrix is redundant and the camera motion can be described by a similarity transformation matrix. By combining all the information in equations (4.1) to (4.8) a similarity matrix can be constructed, as shown in this section.

But before that, it is worth introducing the need for a global registration method, that will improve the results obtained from using telemetry-only homography estimation, since this is much more important for background modelling methods than it is for visual tracking. Although it is safe to assume that the measurements provided by the IMU regarding the altitude and rotation are very accurate, the same assumption does not hold for the GPS data. For a GPS with an accuracy of 2.5 m, the predicted location is 95% of the time within 2.5 m of the real one. This of course leads to the conclusion that the found transformation matrix will not be accurate. To improve

the translational accuracy, a global registration method based on the Discrete Fourier Transform (DFT) is employed, as proposed by Guizar-Sicairos [85]. The advantage of this method is its efficiency and robustness to noise and occlusions. The usual FFT-based approach to finding the cross-correlation peak to within a fraction of a pixel entails several steps: first, computing the DFT of each image, then embedding the result of the product by the conjugate into a larger array of zeros the size of the image, followed by a computation of the inverse FFT to obtain an up-sampled cross-correlation, where the peak is finally found. Instead, the algorithm in [85] obtains an initial estimate of the cross-correlation peak, by a fast Fourier transform (FFT) and then refines the shift estimation by up-sampling the DFT only in a small neighbourhood of that estimate by means of a matrix-multiply DFT.

To improve the accuracy of the proposed method and make it invariant to brightness changes, the DFT registration is applied to the gradient image. The calculation of gradient eliminates redundant information such as fine texture and illumination keeping the higher level structures in images such as shapes. The translation (dx, dy) calculated by this algorithm is compared with the amount of translation expected due to GPS inaccuracy which is calculated as $\Delta x_{expected} = \Delta y_{expected} = 2.5/r_c$. The translation predicted by the DFT registration should not be higher than the expected which would mean that there is a significant error in registration. In the unlikely case that this happens, the translation correction will solely rely on the GPS provided geo-location. Given the additional data the final similarity matrix will be:

$$S = \begin{bmatrix} r_h \cos\Delta\phi & -r_h \sin\Delta\phi & \Delta x - dx \\ r_h \sin\Delta\phi & r_h \cos\Delta\phi & \Delta y - dy \\ 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

To model the background, the well known algorithm proposed by Stauffer and Grimson [232] is employed. Geo-registration and mosaicking are avoided, as opposed to what is usually done in the literature (i.e. [141]). Instead, all the transformations are applied directly to the background model so that it matches the current frame.

If a point $P(x_t, y_t)$ at time t belongs to the current frame, then the location of the corresponding point $P(x_{t-1}, y_{t-1})$ on the background model at time $t-1$ can be found by applying the following transformation:

$$\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{bmatrix} = S \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \tag{4.13}$$

Since the coordinates (x_{t-1}, y_{t-1}) are in general non-integer, nearest-neighbor interpolation algorithm is used to obtain smooth results.

Aerial video changes almost continuously, except when the UAV hovers steadily. This means that with every frame a small portion of pixels is added to the background or foreground distributions. It is assumed that the newly introduced pixels belong to the foreground and their mean is initiated with the current pixel value, the variance takes the maximum variance of the closest pixel and the maximum weight of the closest pixel. In this way the newly introduced intensities most probably belong to the portion of distributions that represent the background. A high learning rate and high variance are used to update the background. An example of segmented foreground is shown in Figure 4.3.

The main advantage of the proposed method is its real-time performance capability and high accuracy. In addition, the alignment error does not accumulate, since the background model is constantly updated. Noise induced by parallax can be mitigated by gradient suppression, as explained by [195].

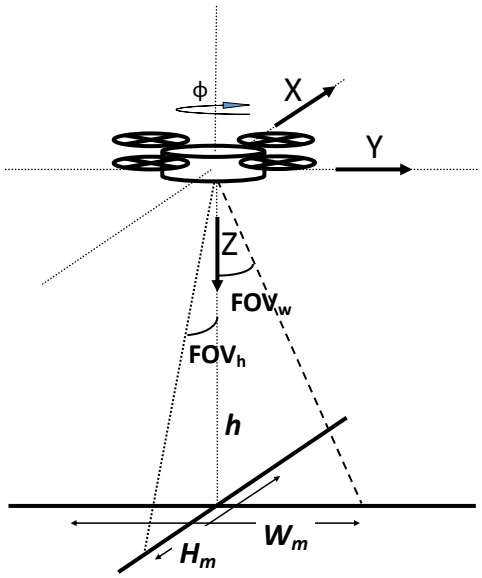


Figure 4.2: A schematic representation of the UAV hovering over the ground plane. The UAV coordinate system, the two FOV angles, and the W_m and H_m ground dimensions are depicted.

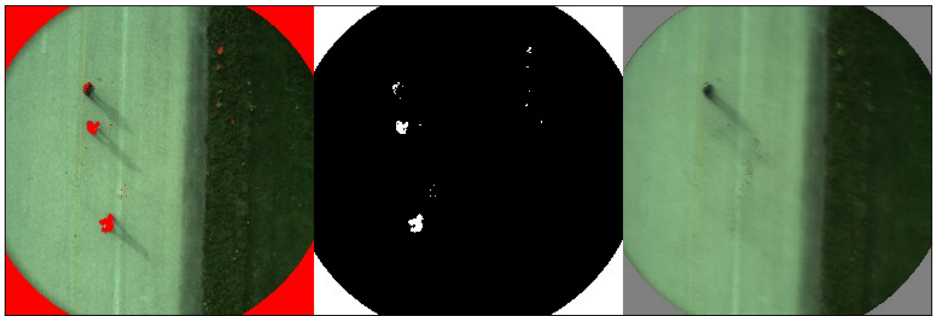


Figure 4.3: Example of segmented foreground after 43 frames from the initialization. The image to the left is the labelled foreground, in the centre is the foreground mask, and right is the averaged background model.

4.4 Experiments and analysis

This section will introduce the experiments that were conducted to validate and test each of the presented methods. The first method is validated against ground truth and then compared to a baseline method, whereas the second method is tested against well-established purely video-based methods in the literature. Before that, however, it is necessary to introduce the datasets used.

4.4.1 Acquisition and definition of datasets

Since the utilised vehicle was used with two different camera set-ups, two different datasets were collected using the different configurations, for each of the presented methods, respectively. In both cases, the videos were recorded at the Institute for Flight Systems (IFS) at the Universität der Bundeswehr in Munich, Germany (a partner in the PROACTIVE project). The resulting sequences were provided along with synchronised telemetry data (GPS/INS signals). As seen in Figs. 4.4 and 4.5, the frames have a different size, as a consequence of the different camera resolutions (i.e. 752×480 versus 512×512). Also, the rates at which new data from the inertial magnetic units, the GPS sensor, and the camera (i.e. image) are dispararate, and differ depending on the configuration. To overcome this issue, and synchronise the image with the IMU and GPS data, all streams are timestamped. This allows to proceed as follows for synchronisation: all streams are played simultaneously, and when a new image is available from the camera, the latest IMU sensor data is attached to it. For GPS data, however, the framerate is much lower than it is for other telemetry information, therefore, GPS positioning data is Kalman filtered to interpolate the values that are missing between frames. The whole process is described in more detail in [32]. This method is applied regardless of the configuration, and since inertial sensor data is captured in much higher rate than the images, it guarantees that the accuracy will always be bounded to a few milliseconds. For instance, with 400 Hz inertial data this leads to a maximum deviation of 2.5 ms. For the first configuration inertial data

arrives at 120 Hz, which leads to an accuracy of 8.3 ms. For the second, it is 10 ms (for 100 Hz IMU data).

For the first method, the recorded data, acquired with the first configuration of the OctoXL (Sec. 4.2.1), has been divided into several sequences to form a dataset. From the original capture, 6144 frames long, several sequences have been selected, most around 300 frames (with a mean of 338 frames, as shown in Table 4.1 in Sec. 4.4.2 below). The selection of the sequences was done taking two considerations into account. One the one hand, since the intention is to observe the improvement of the proposed window correction method, sequences were selected showing different amounts of variation in yaw. This is shown in Fig. 4.7 also in Sec. 4.4.2. For instance, different degrees and speeds in rotation can be observed in the selected sequences, e.g. from lower to higher: `red`, `blue`, `blue1`, `black`, `white`, `white_s`, and `blue2`. On the other, some sequences had almost no yaw rotation but posed a challenge due to the properties of the tracked object (e.g. `black`). The sequences have been named after the most prominent colour of the clothing of the person to track. Manual annotations on the position of that person are given for all frames in all sequences. Figure 4.4 shows some examples of captured frames.

(a) Altitude of 12.35 m (b) Altitude of 13.46 m

Figure 4.4: Example frames from the **first dataset**, captured with the first set-up of the OctoXL vehicle used for the tracking correction method (enhanced contrast for better visibility). Please note the difference in people's appearance due to the changes in altitude from (a) to (b).

For the second method, the collected videos depict two different types of scenes: one has limited texture information and the background (ground) is mostly vegetation

and tarmac, this sequence is referred to as `green` hereafter; the second one is on a snowy landscape with rich texture in the form of ruts and footprints created by vehicles and people named `snow` (see Figure 4.5). The `green` sequence has frames with brightness changes, which are very useful to test the algorithm under challenging conditions.

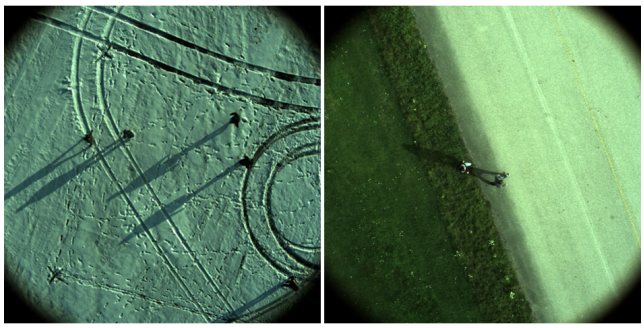


Figure 4.5: Example frames from the **second dataset**: the `snow` and `green` sequences.

4.4.2 Method 1: 'Search window' correction for tracking

The presented approach for 'search window' correction is validated by three different experiments using the first dataset. The first one is a validation method that uses ground truth data from the tracking bounding boxes in order to determine the overlap of local search windows with the actual tracking target at any given time. The second experiment tests the presented approach in conjunction with the visual tracker employed, i.e. the covariance tracker [190, 244]. Finally, the third experiment is conducted using the tracker without any correction for comparison purposes (i.e. *baseline* results). In all three cases, the search window sides are set to be twice as big as those of the tracked object ROI.

In the first experiment, the aim is to test how the local search window correction method performs by itself. To do so, ground truth data is used. The ground truth has been manually annotated for all sequences in the dataset, and is used to provide the 'real' ROI (win_{gt}), that is, it is as using a *perfect* tracker. For any frame, its

corresponding window is then calculated to compare against the window estimated by the correction algorithm (win_e). After that, the estimated window (for frame t) and the window generated from the ground truth (frame $t+1$) are compared using the overlap measure or Jaccard index:

$$J = \frac{win_e \cap win_{gt}}{win_e \cup win_{gt}} \, . \tag{4.14}$$

A novel measure, named the *C-measure* or *C-value*, is also employed. This measure is similar to the overlap, but as opposed to it, it is used to tell *how well contained* (therefore the C) within the local search window the tracked object ROI is. Its definition is as follows:

$$C = \frac{\text{ROI} \cap win_e}{\text{ROI}} \, . \tag{4.15}$$

The logic behind the C-measure is that if the ROI is fully contained within the local search window, the tracker will have a much better chance to find it than if it is partially outside its scope (the local search window). The top part of the fraction will be the full size of the box if the box is fully contained within the window; the denominator is used to normalise the measure to the range of $[0, 1]$.

In the second experiment, the goal is to test the proposed method in a real situation. For this, a covariance tracker with a local search window is employed. The search window is corrected at each frame using the proposed method. In this case, the overlap measure between the detected object and the ground truth is estimated, and used as a measure of tracking quality.

Table 4.1 shows quantitative results of the first experiment, i.e. the validation with ground truth data, using the overlap and C-measure introduced in eq. (4.15). Table 4.2 shows quantitative results for the two other experiments introduced in Section 4.4.2. The overlap measure is given for both, as well as the PASCAL overlap criterion [71]. This is a very common criterion used for the evaluation of trackers, as stated in Sec. 2.4.1.8 (Equation 2.2) on tracker evaluation frameworks. With this criterion, a

Sequence	Length	Validation with GT	
		Overlap$^{\%}$	C-value$^{\%}$
white	668	85.1 ± 7.7	99.3 ± 3.2
white_s	161	77.1 ± 7.9	98.7 ± 3.5
black	391	82.9 ± 8.8	99.8 ± 1.6
red	390	88.9 ± 6.4	100.0 ± 0.5
blue	303	86.6 ± 7.3	99.9 ± 1.3
white1	365	84.6 ± 4.8	100.0 ± 0.4
black1	102	89.5 ± 4.6	99.9 ± 0.7
blue1	316	85.5 ± 6.6	100.0 ± 0.4
blue2	354	87.5 ± 7.6	100.0 ± 0.8
mean	338.8	85.3 ± 6.8	99.7 ± 1.4

$^{\%}$ denotes values are expressed in percent.

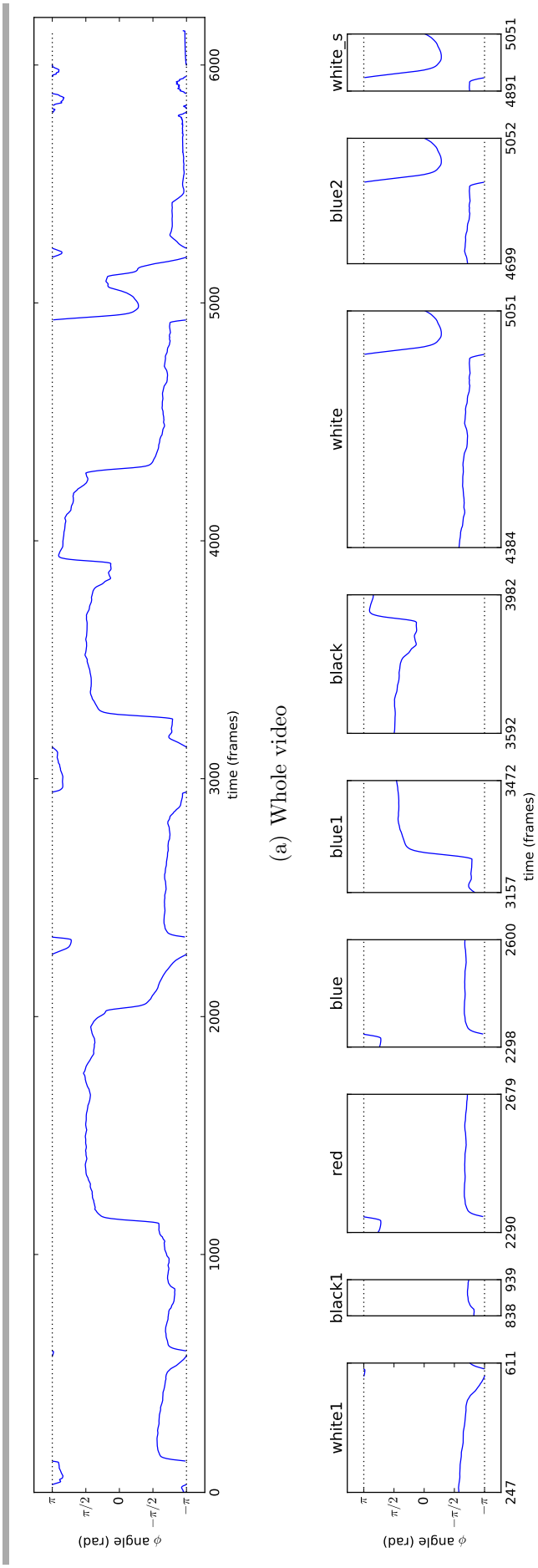
Table 4.1: Sequences of the first dataset and validation results ($\bar{x} \pm \sigma$).

match is said to be such only if the overlap is greater than 50%. The presented results are an average over the whole sequence of the accomplishment of this criterion at each frame. Figure 4.6 shows some qualitative results.

Analysing the results from the first experiment, it can be seen that the C-measure is next to a 100% in most cases, with very low deviations. This means that the proposed method successfully keeps the object within the local search window, and therefore it fulfils its main goal, that is, independently of the tracking method used.

With regards to the comparison between the baseline and the proposed correction method (Table 4.2), several aspects need to be noted. First, the generally low values for the overlap are due to the strictness of this measure, which heavily penalises false negatives and false positives (as shown in Fig. 2.4, Sec. 2.4.1.8 of Chapter 2). However, in the original works where the covariance tracker was introduced, the authors used a much more relaxed measure for the evaluation [190, 244]: any detection within a window of 9×9 pixels of the ground truth centroid was considered a *match*. Also, the window size was not taken into account, only the distance from the estimated point to the ground truth point. Therefore, lower values when using this stricter measure should be expected, and values as low as 50% are normally accepted as a fair amount

Figure 4.6: Frames 2220, 2240, and 2260 from the '`blue`' sequence. In 40 frames, the UAV undergoes a severe rotation about the Z axis. The proposed method successfully tracks the target (bright green) using a reduced search window size (dark red). Ground truth (dark green) is also provided for comparison, along with the search window that will be used in the next step (bright red) calculated by centring a window around the current result.



(a) Whole video

(b) Selected sequences

Figure 4.7: Yaw values over time showing the rate of rotation in the video sequences: (a) for the whole captured video, and (b) for each selected video sequence separately. Additionally, the reader is referred to Fig. A.12 in the materials Appendix, that shows the trajectory of the UAV.

Sequence	Tracker w/ correction		Baseline (no correction)		$\Delta^{\dagger}$
	Overlap$^{\%}$	PASCAL$^{\%}$	Overlap$^{\%}$	PASCAL$^{\%}$	
`white`	48.2 ± 15.4	$68.3 \pm 17.1^{*}$	15.8 ± 12.9	20.6 ± 23.8	**3.31**
`white_s`	61.6 ± 11.8	$82.1 \pm 24.7^{*}$	21.1 ± 16.3	27.2 ± 20.4	**3.02**
`black`	64.6 ± 8.4	$88.8 \pm 24.5^{*}$	53.1 ± 17.0	72.4 ± 20.1	**1.23**
`red`	44.7 ± 5.8	18.7 ± 24.4	44.8 ± 5.8	$19.7 \pm 24.1^{*}$	0.95
`blue`	41.4 ± 8.6	14.5 ± 25.0	44.2 ± 8.1	$26.6 \pm 20.7^{*}$	0.54
`white1`	29.6 ± 10.4	9.6 ± 23.8	18.6 ± 12.8	$9.8 \pm 24.0^{*}$	0.97
`black1`	61.4 ± 7.4	80.6 ± 24.3	61.6 ± 7.6	$85.4 \pm 25.1^{*}$	0.94
`blue1`	27.1 ± 9.5	$10.4 \pm 24.2^{*}$	27.7 ± 9.3	10.4 ± 24.2	**1.00**
`blue2`	63.0 ± 8.9	$80.6 \pm 24.2^{*}$	47.6 ± 14.9	65.1 ± 14.4	**1.24**
mean	49.1 ± 9.6	50.4 ± 23.6	37.2 ± 11.6	37.5 ± 21.9	1.50

† Δ-factor denotes the improvement ratio between PASCAL values ('corrected' over 'baseline').
$^{\%}$ denotes values are expressed in percent.
* denotes best PASCAL value.

Table 4.2: Results for the conducted experiments, compared to baseline (both as $\bar{x} \pm \sigma$).

of overlap for tracking, as is done with the PASCAL overlap criterion [71].

From the 'tracker with correction' experiment, it can be observed that in general, the correction is beneficial or works as well as the baseline method. In the best cases, the improvement factor is greater than 3 (3.31 for the `white` sequence, for instance), with an average factor of 1.50 (that is a 50% improvement on average over the baseline results). There are also some other sequences where the baseline tracking performs on a par with the corrected tracking (factor is ≈ 1.00; for instance in the `white1`, `red`, or `blue1` sequences). The reason for this can be explained by the nature of the sequences, where the UAV's movements are smoother or slower than in other videos, that is, in these cases the correction does not do much, because the tracker search window, itself, contains the target on the next frame, since the UAV motion was not fast-paced. This can be seen in Fig 4.7(b), where the rate of rotation of the vehicle around the yaw axis is plotted against time. It can be observed that, most of the sequences that perform on a par, might have some degree of rotation present (e.g. `red, blue1, black`), but it is not as fast-paced or of such magnitude as other sequences (e.g. `white, blue2`). On

a single case (`blue` sequence), the proposed correction method actually disadvantages the tracker with respect to the baseline. This can be attributed to a tracker issue, since the validation results for that same sequence are among the highest ($99.9 \pm 1.3\,\%$, as shown on Table 4.1). That is, the target is well contained in the expected search window, and re-detection should not be problematic.

4.4.3 Method 2: Background modelling

To prove the validity of the proposed method, it is compared with 'interest point'-based registration with SIFT features (which has been proved to be the most effective, compared to other features [30, 250]), multi-scale Harris corners [167], and a combination of them with global registration methods as in [141]. SIFT features undergo median filtering as proposed in [250] to smooth the estimated motion. RANSAC fitting of matched features is used to find the homography matrix.

To compare image registration techniques, an image similarity measure has to be employed. Instead of selecting a traditional image similarity metric, such as the mean-squared error (MSE) or one of its variants such as peak signal-to-noise ratio (PSNR), the mean structural similarity (MSSIM) index was chosen, since it is widely used for image quality assessment, and also for what is described as its main *drawback* [258]: "its sensitivity to relative translations, scaling and rotations of images" which makes it ideal for evaluating image alignment methods. Furthermore, PSNR produces irregular results with high variance, therefore MSSIM is adopted to evaluate the image registration algorithms presented here.

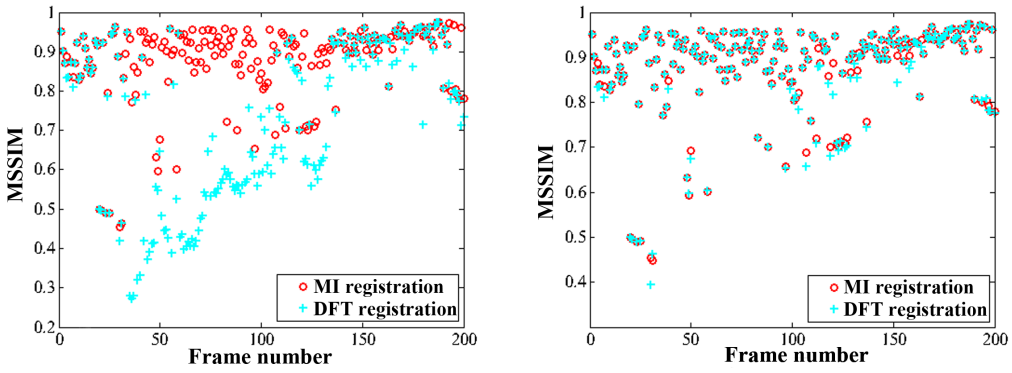
Two experiments were conducted: the first one is focused on determining which is the best method for global registration, whereas the second is used to show the results of the proposed method compared to 'interest point'-based methods.

In the first experiment, mutual information (MI) registration and the discrete Fourier transform (DFT) registration are compared as methods that refine the crude alignment achieved by 'key point'-based registration, or the proposed telemetry-based registration. To make an objective assessment possible, it is assumed that the crude

alignment is implemented by telemetry-based registration in both cases. The results presented in Table 4.3 show that, for the frames with brightness change (BC), DFT registration on the gradient image outperforms the DFT applied on the colour image, while MI registration is shown to be invariant to brightness changes. For the frames from the `snow` sequence, DFT on gradient images performs significantly better than MI with the mean MSSIM = 0.7731. A visual representation of the results for the `green` sequence in Fig. 4.8, confirms that DFT registration applied on gradient images is equivalent to MI. As MI algorithm is computationally expensive, the faster DFT registration is chosen, but using gradient images to obtain the best of both methods.

Sequence	Colour images		Gradient images	
	DFT	MI	DFT	MI
BC	0.5319	0.7409	0.7489	0.7405
`green`	0.7313	0.8757	0.8730	0.8763
`snow`	0.7765	0.5398	0.7731	0.5291
mean	0.6799	0.7188	0.7983	0.7153

Table 4.3: Evaluation of DFT and MI registration methods based on the mean MSSIM metric obtained from 200 frames for each case.

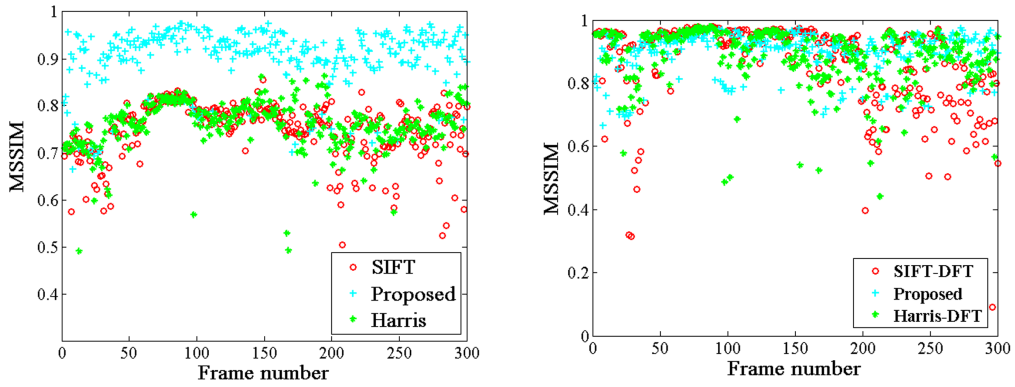


(a) Colour images (b) Gradient images

Figure 4.8: MSSIM metric obtained for 200 frames of the `green` sequence comparing the MI and DFT registration on (a) colour images, or (b) gradient images. DFT can perform as well as MI when using gradient images.

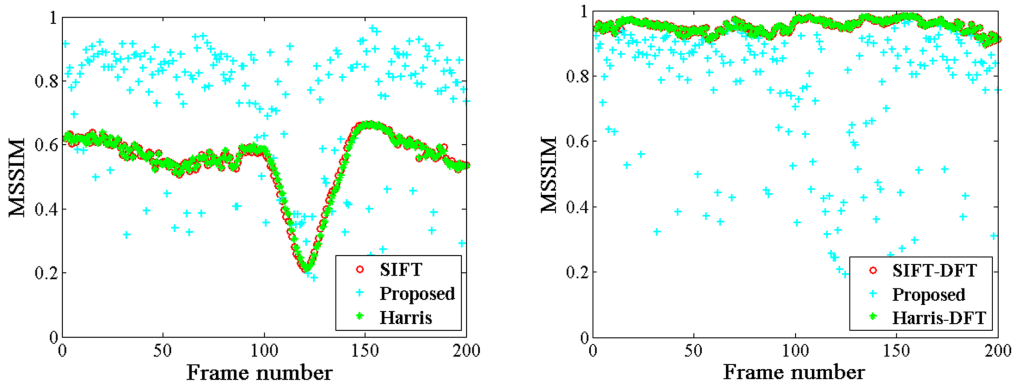
The MSSIM metrics for the proposed algorithm and for SIFT and Harris feature-based registration are compared in figures 4.9 and 4.10. The acceptance threshold of Harris has been decreased so that it detects the maximum number of feature points. Figure 4.9 refers to the `green` sequence where the presence of tarmac and vegetation is dominant, while Fig. 4.10 refers to the `snow` sequence where the texture is richer. By observing Fig. 4.9(a) it can be inferred that the proposed algorithm performs better than the SIFT and Harris registration which seem to degrade at the last 100 frames where the background scene is dominantly tarmac (See Fig. A.13, in the materials Appendix, p. 183). Figure 4.9(b) displays the results for the above mentioned methods refined by the proposed gradient-based DFT. It is easy to see that the refinement by gradient-based DFT has enormously increased the MSSIM metric for SIFT and Harris registration. However, even in this case, the proposed method seems to display a more consistent pattern than the other two, which have larger standard deviation. In Fig. 4.10(a) it can be clearly seen that the proposed method performs better than SIFT and Harris on the `snow` dataset, which are improved significantly with the refinement by DFT algorithm and the benefit of having rich texture, as seen in Fig. 4.10(b). The huge decline in the MSSIM observed in frames 100 to 150 in Fig. 4.10(a) is due to the rotational component of the motion undergone by the UAV. The combination of SIFT features and multi-scale Harris with gradient-based DFT registration has not been seen in the literature yet, and from the conducted experiments it is proved to be a reliable method.

In general, the proposed telemetry and gradient-based DFT technique shows a robust performance for the `green` sequence but it is less accurate for the `snow` one. Since the `snow` sequence is rich in texture, the MSSIM metric penalises small mismatches more, i.e. its sensitivity is increased, which explains the high variations in Fig. 4.10. The main advantage of the proposed algorithm is its robustness in feature-less scenes and its computational efficiency as the calculation of initial transformation matrix based on telemetry data happens in constant time with complexity $O(1)$ and the subsequent DFT registration algorithm, with pixel accuracy, has complexity $O(W_{img}H_{img})$ [85] .



(a) No DFT on SIFT/Harris (b) All DFT-refined

Figure 4.9: MSSIM metric obtained for 300 frames (400 to 700) of the poorly textured `green` sequence comparing the proposed method to: (a) pure SIFT and Harris, and (b) DFT-refined SIFT and Harris.



(a) No DFT on SIFT/Harris (b) All DFT-refined

Figure 4.10: MSSIM metric obtained for 200 frames (600 to 800) of the richly textured `snow` sequence comparing the proposed method to: (a) pure SIFT and Harris, and (b) DFT-refined SIFT and Harris.

4.5 Conclusions

In this chapter it has been demonstrated that telemetry data can be very useful as an additional cue for video surveillance tasks from aerial video footage. Important improvements in performance have been achieved with the proposed methods, as compared to baseline results, or to comparable methods, respectively.

On the one hand, a novel method for the correction of a local search window has been proposed. Validation with ground truth data showed the validity of the method. Furthermore, when using a real visual tracker, important improvements in performance can be achieved (up to three-fold, 50% on average). However, other factors need to be taken into account, such as the accuracy of the tracker at calculating the size of the detected object and location of the centre point, or the loss of track due to sudden changes in the appearance model, rather than evolving changes, which are controlled by the internal mechanisms of the tracker. On the other hand, a telemetry-based aerial video frame registration as ego-motion compensation step has been presented, that, opposed to existing works in the literature, introduces the novelty of applying all the transformations directly to the background model, so that it matches the current frame. Existing research focuses on feature point-based registration, which is computationally expensive and far from real time. In contrast, the proposed approach is computationally efficient and has real-time capabilities (as it uses a fast DFT and other computationally inexpensive approaches), it is robust in scenes with poor texture, where the only detectable feature points are located on the moving object rather than on the scene. This is a major advantage, avoiding severe deformations of the warped image resulting in huge accumulated alignment error. Moreover, the probabilistic background model compensates for the accumulated alignment error, as the background model is constantly and rapidly updated. The experiments showed that the algorithm is robust to illumination changes and GPS location inaccuracies. However, the disadvantage lies in the fact that stationary foreground objects are quickly absorbed in the background. This issue can be resolved if the detection process

is combined with a robust visual tracker. Another disadvantage is the dependence on the GPS/INS, which means that faulty equipment or bad weather conditions can hinder the accuracy of the system.

To summarise, some conclusions can be drawn: tracking can be improved greatly, without the computational expense of full image registration, by simply correcting the location of the search window. Besides, background modelling can be performed using telemetry information for a crude alignment, and refined using a global registration method. It has also been shown that even point-based matching methods can benefit from such refinement. Finally, the proposed method for background modelling can work cooperatively with point-based methods, as they specialise in texture-less and textured scenes, respectively, thus always capturing the best result (for instance, one could pick the method with the best MSSIM correspondence).

Chapter 5

Analysis of crowd behaviour from microscopic analysis

Chapter highlights: Individual's *tracklets* are used in a novel *mesoscopic* scene descriptor to infer group-level knowledge and detect events.

Overview

This chapter is dedicated to the analysis of behaviour of crowds and small groups of people in scenes captured with multiple cameras. A dataset is introduced (see Section 5.3), since a thorough study of the literature has shown none of the existing datasets to date would be suitable for the task at hand. Here, the concept of *tracklet plots* is introduced: short tracks (i.e. tracklets) obtained from a multi-target visual tracker (such as those presented in Chapter 2, specifically sections 2.4.1 and 2.4.3) are aggregated into a feature vector that can describe the whole scene from a single viewpoint. Features from multiple cameras can be combined, and then be used to classify a scene into one of the several predefined categories or classes. A bag-of-words model is employed to characterise the sequences using the available feature vectors as words, and creating bags for each sequence, that can subsequently be recognised using a nearest-neighbour approach. Both the single-view and the multi-view work flow will

be presented, and compared.

On the usage of the 'tracklets' term.

As introduced in the literature review (Sections 2.2.1 and 2.4.3), other authors refer to tracklets meaning partial tracks, and perform data association on several of these tracklets to form longer tracks, as part of a long-standing tracking algorithm. However, in this chapter, the concept is used to refer to intentionally short tracks, i.e. that have been captured for short periods of time deliberately, as tracking for long periods of time has not still been fully achieved and leads in most cases to loss of track. In this sense, the *tracklets* presented here could have used a different name, such as *pathlets*, or any other appropriate term. Nonetheless, the name of *tracklets* is kept throughout the wording, as this chapter introduces some published works that used the term.

Main contributions, outcomes, and publications

There are two main contributions presented in this chapter, leading to the following publications:

- **Tracklet plots** as a scene descriptor (or feature) [55], and
- a tracklet plot **fusion scheme** for multiple views [56].

Another outcome, or minor contribution of this chapter is the **"Penrhyn Road campus dataset"** for small crowd event detection, recorded from multiple overlapping viewpoints, which serves to the purpose of testing the proposed algorithms, since no other datasets with the desirable features (i.e. multi-camera and with multiple abnormality classes) existed.

5.1 Introduction

The detection of groups of people and events can be valuable in a number of different situations: from urban environments and events or large gatherings, to targeted

marketing in commerce and shopping centres, to security in airport terminals or other similar spaces [40, 105, 226, 282]. Furthermore, automation in these cases can help cut down costs and improve public safety, as well as reduce error-prone manual surveillance [40, 63].

Crowds can differ in their density and extent, from sparse and small groups of people, to big crowds, all forming a continuum [233, Ch. 2][181]. However, when looking for the best tools for crowd analysis, it is suggested that there could be a topology with different levels [171, 282]. For instance, the authors in [282] recognise three: micro-, meso-, and macroscopic; this would roughly deal with individuals, groups, or crowds, respectively. Furthermore, these levels of analysis are not necessarily exclusive, neither do they need to work in isolation [241]; that is, the types of cues or features extracted using microscopic analysis (such as individuals' tracks in a scene) can be used as input for analysis at higher abstraction layers to infer knowledge about the existing groups or crowds. Interaction among algorithms at these different levels allows feed-back and feed-forward (from microscopic to macroscopic and vice versa). Moreover, due to the nature of crowds, their behaviour might need to be analysed from more than one camera, since they might span through multiple views [111].

Based on these ideas, in this chapter, two contributions are presented. Firstly, a scene descriptor called *tracklet plot* (TP) will be described. Next, a method to fuse information from multiple tracklet plots is presented. Experiments carried out on the presented dataset will validate the descriptor, and will reveal the benefits of fusion from multiple views.

Tracklet plots and the algorithm involved in their generation will be described in the Methodology section (Sec. 5.2). However, as this concept is at the core of this chapter, the idea behind this will be introduced here briefly. Figure 5.1 shows an overview of the concept. To generate a *tracklet plot*, the tracklets from individuals in the scene (Fig. 5.1, left) are used. The generated TP describes the scene at the interval during which the tracklets were obtained (Fig. 5.1, right). This scene descriptor, the TP that is, is then exploited with the binning into a tracklet plot histogram

(TPH) for each of the available views. Subsequently, these TPHs can be used on their own for classification (single-view case), or combined (see Sec. 5.2.2) into multi-view descriptors (multi-view TPHs, or MV-TPHs). Since TPs (and therefore TPHs) are obtained from a short interval of time, they are useful for TP-based on-line recognition systems. However, in this work, TPHs are used to describe entire video sequences and subsequently used in a bag-of-words model as covered in Sec. 5.2.3.

Figure 5.1: Overview of the idea behind tracklet plots presented in this chapter. A *tracklet plot* is generated from the tracklets of individuals present in the scene during a given interval.

5.1.1 Tracklet exploitation for event recognition

The field of anomaly detection in automated surveillance has seen many developments in recent years. Algorithms have been developed using very diverse approaches. A review of these by Sodemann et al. [226] brings many of them together, and proposes a classification based on five main aspects of interest: the target(s) of the surveillance, how anomalies are defined, the sensors and feature extraction processes used for analysis, the learning methods employed, and modelling algorithms.

Regarding the definitions of anomaly, most works reviewed in [226] model only 'normal' events, that is, anything deviating from the learnt model will be considered 'abnormal'. This approach has a clear drawback: training examples are needed to cover all possible normal behaviours; when this is not viable, the system is prone to

false positives. The opposite of this approach, that is, to model only abnormalities might seem a better approach, however this has a similar result: anomalous behaviours might differ from those the system was trained with. To clarify, examples of what is considered 'abnormal' include individuals walking in a direction different from that of the majority, and sudden scattering due to a danger, among others. A third approach, that is used when both normal and abnormal events are well defined, and well represented in the dataset, consists in modelling both normal and abnormal behaviours. Finally, if there are more than two classes, and the anomalous events are pre-defined and represent meaningful actions or occurrences, the problem of anomaly detection can be seen as a more general problem of *event classification*, where video sequences are assigned labels, and a meaning can be inferred based on these. Similarly, Ballan et al. [26] draw a parallel between the techniques used for action recognition by a single actor on a single camera, and those employed to recognise events from crowds. They state that there are commonalities between those two fields, since modelling techniques employed (e.g. bag of words), can be employed in a very similar fashion regardless of where the features are extracted from (i.e. a single actor or a crowd of people), the only difference being the features themselves (i.e. information of the joints for a single subject, or other information used for crowds).

Regarding feature extraction methods, in [226] it is explained that two main approaches or categories of works exist: first, works where target tracking or identification is performed (similarly to the concept of *microscopic* analysis in the taxonomy presented above); and those where a general pattern of motion is extracted on a pixel basis, representing the state of whole groups of people or crowds (*meso-* and *macroscopic* levels of analysis). The latter approach is very widely used, and examples abound [17, 77, 95, 127, 290]. Optical flow or variants of it, as well as similar techniques, are among the most widely spread methods in these cases.

Hu et al. [95] are able to construct *supertracks* which represent the dominant, collective motions of the crowd; to do so, motion vectors from a sparse optical flow are used as *tracklets* (their definition differs from the one used in this chapter), which are

in turn linked together using a sink-seeking process. Similarly, Lasdas et al. [127] use tracklets obtained by a Kanade-Lucas-Tomasi (KLT) tracker, and Gárate et al. [77] use features from accelerated segment test (FAST) interest points. Unfortunately, these techniques have two major drawbacks: first, they are only used to detect anomalies as deviations from the inferred dominant motion, but cannot flag other types of events or actions; and second, they can only be used to detect anomalies comprising the whole crowd or a majority of the individuals composing it; and are unsuitable when some of the events that are to be detected involve only a minority or a single individual in the scene.

There exist, however, some *hybrid* techniques, that is, methods that use macroscopic analysis approaches, but somehow limit the extent to regions of interest (ROIs) or use other spatial constraints, that are roughly equivalent to dealing with persons or small groups (as in a micro- or mesoscopic approach). Such are those shown by Dee and Caplier [64] and Zhu et al. [290]. In [64], the KLT tracker is employed, but rather than tracking points over the whole image, a histogram of oriented gradients (HOG) detector is used to determine ROIs where the points are tracked. The tracklets obtained through this method are then used to build histograms of motion detection (HMDs), which can be used to depict the directions of motion of the individuals in the scene. Similarly, in [290], *particle advection* (by clustering) is used to aggregate particles into groups that approximately match the limbs and torsos of people, therefore allowing an analysis at the microscopic level.

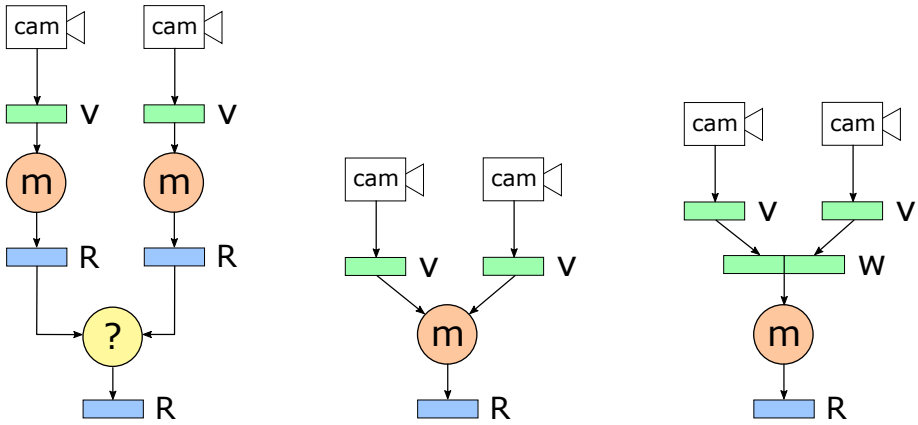
The idea of *tracklet plots* (TPs) presented in this chapter is similar to the HMDs mentioned above; however there are three differences worth mentioning: Firstly, tracklet plots are scene descriptors that can be used directly for analysis as would be images or matrices (depending on their size), or by first obtaining histograms from them; in contrast, HMDs are one-dimensional histograms of motion direction. Secondly, TPs can account for differences in speed among the tracklets, whereas HMDs cannot. Finally, HMDs are used to describe a whole video sequence, therefore they are not suitable for real-time recognition systems, as opposed to TPs, which are scene

descriptors that represent a short lapse of time. This can be advantageous for future TP-based on-line recognition systems.

5.1.2 Multi-view information fusion

When systems use a single view to analyse the scene, challenging situations, such as occlusions need to be tackled. A review of works addressing this is presented in [241]. Some authors, however, consider that single camera systems are inherently unable to overcome the challenge posed by occlusions [111], and therefore, fusion of evidence from multiple cameras is required, although this introduces further challenges, and computational overhead.

In Chaaraoui et al. [44], approaches to fuse evidence from multiple cameras are discussed. Three levels are presented (depicted in Fig. 5.2), where fusion can be performed: *decision level*, *model level*, and *feature level*.



(a) Decision-level (b) Model-level (c) Feature-level

Figure 5.2: Different approaches to fuse evidence from multiple cameras. In the depictions, 'v' stands for feature vector, 'R' stands for response, 'm' stands for model, and each 'cam' represents a camera or view ('w' for combined feature vector).

- In the first case, at the *decision level* (Fig. 5.2(a)), parallel systems are run for each of the views, and it is only at the end (just before the final output of the system) that a *decision* is taken; that is, the fusion is *postponed* until the last moment. In this case, fusion would normally be achieved through voting or

ranking schemes, based on the evidence collected from the different views so far. However, finding appropriate decision rules might not be simple.

- Fusion at the *model level* (Fig. 5.2(b)) entails feeding the features obtained from the different views into the model during the training stage. Features can be fed either labelled [53] (*i.e.* with the view they were extracted from), or unlabelled [265]. The modelling algorithm thus generates a single model for all views, yet changes in the learning scheme might be necessary.

- For *feature-level fusion* (Fig. 5.2(c)), the multiple views need to be synchronised, since the features are extracted for each view separately, but immediately fused into a larger feature (either by concatenation [265] or averaging [151]), that is then fed to the modelling system. Therefore, in this case, no changes are required in the learning scheme, as from the point of view of the model, it is dealing with a single feature that carries more information from the semantic point of view. An additional benefit, is that there is no need for an additional weighting or voting mechanism, as in the case of decision-level fusion. Its major drawback, though, is that the dimensionality of the multi-view feature (in case of using concatenation) will grow linearly with the number of cameras in the system, and this will have an impact on the speed at which a model can be trained.

To overcome the curse of dimensionality when using feature-level fusion by concatenation of features from different views, dimensionality reduction techniques can be exploited. By using these, the dimensionality of the feature can be kept small, while the overall system performance is also maintained. Under these circumstances an advantage exists, even if the addition of new cameras does not improve the recognition rate, but is limited to the best-performing view as the system is faster to train than using several separate models for each view. More interestingly, taking into account that the system does not know which views perform better *a priori*, a multi-view system will benefit from the additional information collected.

To summarise, following the five aspects analysed by Sodemann et al. [226] in

their review (please refer to Sec. 5.1.1 where these were introduced), the method presented in this chapter could be classified as 1) having sparse crowds or large groups of people as its target; 2) modelling both normal and abnormal events, regarding the task as a multi-class event recognition problem; 3) using vision as the only sensors, that is, visible light cameras, and extracting features from each individual in the scene by the use of a visual tracker by identification (high-level features); 4) and 5) using a bag-of-words modelling during the training stage, which internally employs a k-Means clustering to determine the *key words* in an unsupervised fashion; and using a k-Nearest-Neighbour (k-NN) algorithm for classification.

5.2 Methodology

In this section, the two main methods or contributions of this chapter will be explained in more detail. On the one hand, a scene descriptor based on a compact representation of the tracklets during a particular time span is presented (i.e. the *tracklet plots* – or TPs). On the other hand, fusion of evidence from multiple views at the feature level (using TPs) is introduced. The whole work flow will be presented: from people tracking, to feature extraction and fusion, to the recognition of events in new video sequences using k-NN on the trained BoW model.

5.2.1 Tracklet plots for scene description

Tracklet plots (TPs) are envisaged as a scene descriptor, which will subsequently help detect anomalous events occurring within large groups of people, or small crowds. The idea behind this is, to some extent, similar to motion history images (MHI) which were introduced by Bobick and Davis [31]. However, in this case, the superimposed tracklets represent the motion patterns of the people present in the scene, and their arrangement in the tracklet account for differences in speed and direction of motion; different intensity (or density— depending on the histogram technique used, see Sec. 5.2.1.3 below) values in the tracklet plot reveal agreement among individual trajectories: that

is, when people move in "roughly" the same direction, tracklets are "almost" parallel, and therefore, they will be plotted over the same area in the TP; thus, it will have *brighter* areas (or denser ones), representing coherence in the directions (and speeds) of the people (these will be narrow bands when motion patterns are very similar). Some examples, from diverse situations, are depicted in Figure 5.3.

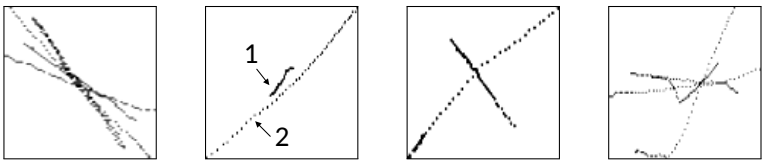


Figure 5.3: Real data examples of different tracklet plots. a) Ordered group of people walking at the same speed and direction; b) a fast biker (b.2) and a slower pedestrian (b.1); c) Two people walking in perpendicular directions; d) A chaotic situation, where people run away. Pictures are shown in inverted intensity and enhanced contrast.

5.2.1.1 Extracting individuals' cues: tracklets

Before introducing tracklet plots, it is necessary to explain what are *tracklets*, or more precisely, what is the definition of tracklets used in this work. To put it shortly, a visual tracker is used for a short number of frames. This is a parameter to the method, and is subsequently referred to as Δ. Please refer to Sec. 5.3.2 and Table 5.2 therein for the value given to this and other parameters introduced here.

Regarding the visual tracker used to obtain the tracklets, 'particle filter'-based trackers (PF) are a very commonly used method, as suggested in [241]. For this reason, the PF variant presented in [188] is used, which is readily available and can easily be run in parallel for multiple targets. PF trackers require *initialisation seeds* to be provided, that is the regions of interest (ROI) where people are found in the first frame need to be provided, so that models are learnt from the given regions, and tracking can then proceed automatically in subsequent frames. Since there will be several targets to be tracked, a multi-target tracker, that runs in parallel for each given individual, is used. Additionally, after the tracklets have been collected, an optional process can be

applied to them, by which the points that make up each tracklet are corrected using a Kalman filter (KF). The idea is to obtain tracklets that represent general patterns of motion; by this procedure *smoother* tracklets are obtained, reducing jitter caused by local decisions of the tracking algorithm.

The reason for limiting the tracking to short intervals is based on the nature of visual trackers: tracking is not perfect, and the longer a visual tracker runs, the higher the probability it will lose the track due to deviations of the current model from the one learnt at initialisation. This is particularly true for algorithms that do not use model updating mechanisms [83, 159, 203] (as seen in Sec. 2.4.1 of Chapter 2). However, in this particular application the emphasis is less on long tracks, but rather to be able to estimate the motion patterns of people, aggregating it into a meaningful descriptor subsequently used for analysis. Furthermore, if this scene description is performed at short intervals, the abstraction layers above can produce responses more frequently, and support on-line event detection.

5.2.1.2 Tracklet plot generation

The tracklets of a given interval of Δ frames obtained from the previous stage are combined to generate a tracklet plot (TP), which is created by superimposing (plotting) several tracklets of people present in the captured scene. They are first normalised using their length (equivalent to speed), using the longest of them. The procedure is elucidated in Algorithm 5.1.

5.2.1.3 TP histogram extraction

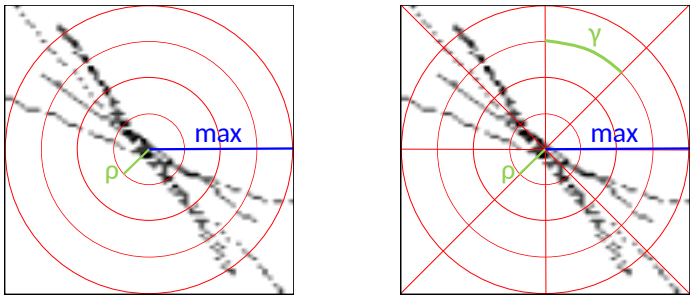
Tracklet plots act as accumulators, as just described, and therefore have high dimensionality. A TP plotted as just described would have $L' \times L'$ dimensions (i.e. *bins* in the accumulator), most of which would be zero (as depicted by white areas in the examples of Fig. 5.3). It is therefore necessary to reduce the dimensionality, to make TPs usable. To do so, different histograms can be extracted from the tracklet plot. In Section 5.4.2, experiments are conducted to determine the validity and value of each

Algorithm 5.1: Tracklet plot generation

Data: Tracklet set T
Result: Tracklet plot TP for T
$d_{\max} = 0$; `/* greatest diagonal */`
$\texttt{box}_{d\max} = \emptyset$; `/* box of the greatest diagonal */`
foreach *tracklet* $t \in T$ **do**
  Find the bounding box b that encloses t;
  Calculate the diagonal d of b;
  if $d > d_{\max}$ **then**
    $d_{\max} = d$;
    $\texttt{box}_{d\max} = b$;
  end
end
Let $L = \max(\texttt{box}_{d\max}\texttt{.height}, \texttt{box}_{d\max}\texttt{.width})$;
Create square image TP of size $L \times L$;
Let $||T||$ be the number of tracklets in T ;
Let $\max_I$ be the maximum intensity value ; `/* 255 for an 8-bit image */`
Let $w = {}^{\max_I}/_{||T||}$;
foreach *tracklet* $t \in T$ **do**
  Cumulatively plot t centred in TP with intensity w;
end
Resize TP to a normalised size of $L' \times L'$; `/* where` L' `is a parameter */`

variety of histograms presented.

Two different types of histograms are introduced *circular* and *polar*. The first type, *circular* histograms, take only speed into account, that is, the histogram has bins dividing the TP into ring-shaped bins, that is, based on the distance to the centre of the TP. Since all tracklets are captured during the same amount of time, longer tracklets correspond to subjects moving faster during that period. The second type, or *polar* histograms, are a variant of circular histograms, that also take direction into account. To do so, additionally to rings, TPs are divided into sectors (that is, angular divisions of the TP). Please refer to Fig. 5.4 for examples of both types of histograms. Fig. 5.4(a) shows a circular histogram, with different regions with their span (ρ), delimited by red lines, and the maximum radius ($\max = \frac{L'}{2}$); and Fig. 5.4(b) depicts a polar histogram, where not only ρ is used, but also γ for the angle span of each sector. Additionally, two modalities are introduced for each type: either using the weights, as defined above (—as w— and therefore binning intensity values as well), or just performing a *count* of the pixels whose intensity in the TP is greater than zero.



(a) (b)

Figure 5.4: Example of the two main histogram extraction modalities presented: a) circular histogram, using only disc-shaped regions r of size ρ); b) polar histogram, using sectors (α, of size γ rad) as well as r regions.

This leads to a total of four different histogram possibilities:

- **Circular histogram.** In this histogram, the TP is divided into concentric disc-shaped regions . R denotes the set of them $R = r_0, r_1, \ldots, r_n, \ldots, r_N$, where each region r_n spans from $n \cdot \rho$ to $(n+1)\rho - 1$, and ρ is a fixed span given by $\rho = \frac{L'}{2N}$. The histogram ($h_{(s,win)}$) for a given short interval (win) of a given sequence (s) is then populated as:

$$h_{(s,\text{win})}(r, y) = \sum_{p_i \in r} I(p_i) \quad \text{if} \;\; I(p_i) = y, \tag{5.1}$$

  where each p_i is a pixel in the region r, $I(\cdot)$ is the intensity value of a given pixel in the TP and y denotes each intensity value. The idea behind this kind of histogram is that it can register the differences in velocity among the people in the scene.

- In case the binning is not performed on the intensity dimension, the bins would be populated as:

$$h_{(s,\text{win})}(r) = \sum_{p_i \in r} 1 \quad \text{if} \;\; I(p_i) > 0, \tag{5.2}$$

- **Polar histogram.** In order to better detect *how orderly* a crowd or group

are, the introduction of *sectors* is considered. These can determine whether all tracklets follow a particular direction, or some deviate from the majority, or the movement is completely chaotic. Therefore, in addition to different speeds, polar histograms can account for differences in the direction of motion among the tracklets. Polar histograms are divided into disc-shaped regions (R) and sectors. The set of all sectors will be denoted as $A = \alpha_0, \alpha_1, \ldots, \alpha_M$. Each α-bin will have a span of $\gamma = \frac{2\pi}{M}$. Therefore in this case, each bin will be populated as:

$$h_{(s,\text{win})}(r, \alpha, y) = \sum_{p_i \in (r,\alpha)} I(p_i) \quad \text{if } I(p_i) = y, \tag{5.3}$$

- Or, in the case no intensity bins are used:

$$h_{(s,\text{win})}(r, \alpha) = \sum_{p_i \in (r,\alpha)} 1 \quad \text{if } I(p_i) > 0, \tag{5.4}$$

5.2.2 Fusion of features from multiple views

Once the tracklet plot histograms (abbreviated as TPH) are extracted for all views, fusion at feature level is applied by concatenating the features from each view (a concatenated feature is named a *multi-view TPH* —or MV-TPH). The simplicity of this method justifies its use, as it will not require changes in the modelling technique used (as would using model-level fusion), while allowing the recognition system to be extended to multiple views. It will not require an additional decision mechanism, either (as opposed to decision-level fusion). However, views will need to be synchronised (this was manually done, please refer to Fig. A.15, in the materials Appendix, p. 185). Figure 5.5 summarises the process of feature extraction and concatenation.

When combining information from multiple views into a single feature vector, the size of the MV-TPHs grows linearly with the number of cameras. This was identified as the main shortcoming of feature-level fusion when different fusion schemes were presented in Sec. 5.2.2. To avoid it, dimensionality reduction (DR) techniques can be used on the MV-TPHs, thus limiting the dimensionality growth of the feature

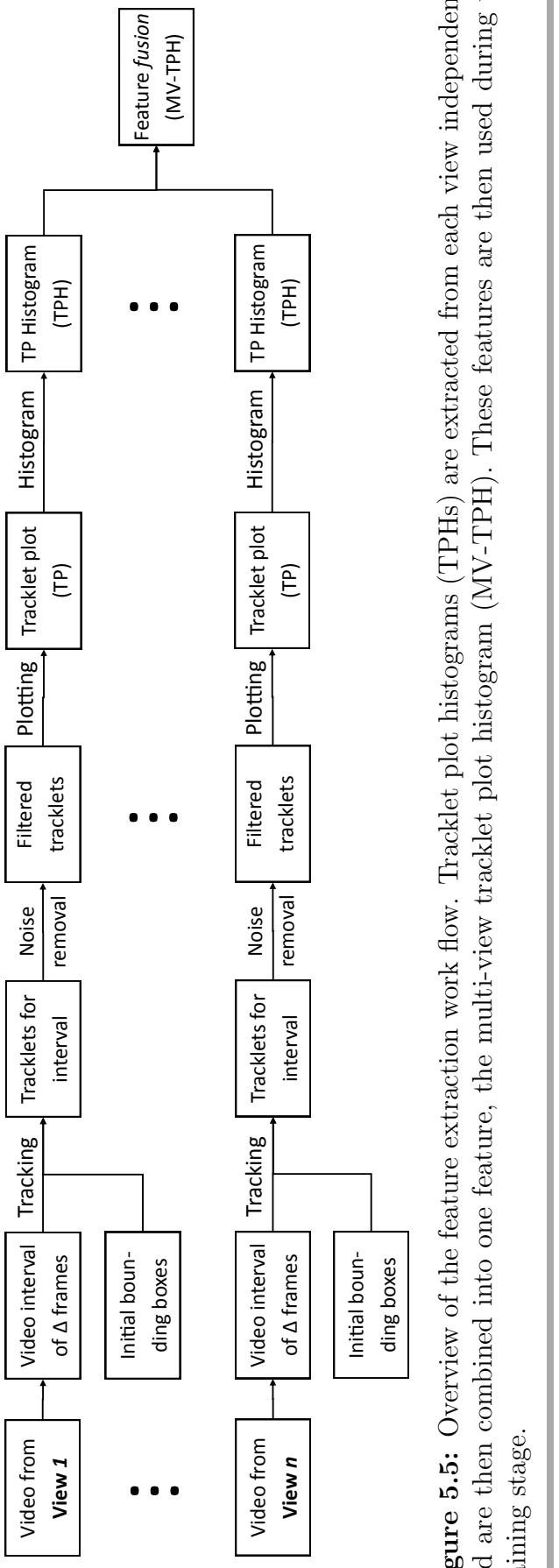


Figure 5.5: Overview of the feature extraction work flow. Tracklet plot histograms (TPHs) are extracted from each view independently, and are then combined into one feature, the multi-view tracklet plot histogram (MV-TPH). These features are then used during the training stage.

vectors. As will be discussed with further detail in the experimentation section (see Sec. 5.3), different tests have been conducted with several DR methods, in order to determine the best-performing method, that can keep feature dimensionality small while maintaining performance. Tests have been conducted with four different dimensionality reduction techniques, namely: principal component analysis (PCA, linear) [106], kernel PCA (with a Gaussian kernel) [166], Isomap [239], and semi-definite embedding (SDE) [259, 260], which is also known as maximum variance unfolding (MVU).

5.2.3 Bag-of-words modelling and recognition

At the end of the process of feature extraction and combination described in the previous section, each multi-view video sequence ($s \in S$) is described by a series of MV-TPHs; that is, each interval in which the sequence is divided is described by one MV-TPH. In order to train the system, a bag-of-words (BoW) modelling is employed; this technique was first applied to the categorisation of text documents in a corpus, and introduced the concept of a *key word frequency* histogram (referred to as η below), to describe each document [26, 224]. As an analogy to the first application of BoW, each video is considered as a document, and each MV-TPH descriptor is a word within a document. Therefore, a video sequence (s) can be replaced by its sequence of descriptors (H_s). The different document categories represent each of the event classes to be recognised. An overview of this process can be seen in Fig. 5.6. To obtain the key word frequency histogram (η), the algorithm proceeds as described in Algorithm 5.2. This algorithm employs a distance function between each descriptor d to the a key word w in the key word set K. This distance is calculated by a symmetric Kullback-Leibler divergence [152, §2.5][42], as:

$$J(d, w) = \frac{KL(d, w) + KL(w, d)}{2}. \tag{5.5}$$

Here, $KL(\cdot, \cdot)$ is the Kullback-Leibler divergence between the key word and the descriptor, or more generally for two discrete probability distributions p, q:

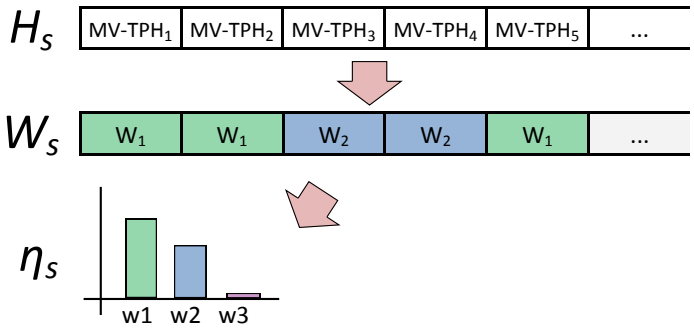


Figure 5.6: Overview of the BoW modelling. After *key words* (w) are obtained via clustering, descriptors in the sequences (H_s) are replaced by their closest *key word* (in W_s), which are then used to generate histograms of key word frequencies (η_s).

$$KL(p,q) = \sum_{i=1}^{||p||} p_i \ln \frac{p_i}{q_i}, \tag{5.6}$$

for all $p_i, q_i \mid p_i \neq 0$ and $q_i \neq 0$.

Once this process is finished, the model is trained, and any future video input can be recognised by means of the k-Nearest Neighbour (k-NN) algorithm.

Algorithm 5.2: key word frequency histogram generation

Data: Multi-view sequence set S.
Result: Sum-1 normalised key word frequency histograms ($\eta_s \quad \forall s \in S$).
/* Step 1. Generate a single descriptor set D will all descriptors (MV-TPHs) regardless of origin (sequence), and feed to kMeans to obtain set of key words K */
$D = [d_0, d_1, \ldots, d_{||H_s||}] \quad \forall s \in S$;
$K = \texttt{kMeans}(D)$;
/* Step 2. Substitute the original sequences (H_s) by sequences of key words W_s. */
foreach $s \in S$ **do**
  $W_s = [\arg\min_{w \in K} J(d, w)] \quad \forall d \in H_s$; /* J described in eq. (5.5) */
end
/* Step 3. Obtain histogram of key-word frequencies $\tilde{\eta}_s$ */
foreach $s \in S$ **do**
  foreach $w \in K$ **do**
    $\tilde{\eta}_s(w) = \sum_{x=1}^{||W_s||} \delta(w, W_s(x))$; /* where $\delta(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$ */
  end
end
/* Step 4. Sum-1 normalisation of each $\tilde{\eta}_s$ into η_s, $\forall s \in S$ */
foreach $s \in S$ **do**
  foreach $w \in K$ **do**
    $\eta_s(w) = \frac{\tilde{\eta}_s(w)}{\sum_{w' \in K} \tilde{\eta}_s(w')}$;
  end
end

5.3 Experimentation

To validate each component of the proposed method, a series of experiments were conducted on a novel dataset (first introduced in [55]). The reason for collecting a new dataset has to do with the fact that existing datasets do not include the type of actions required for the task at hand. Such a dataset would need to be multi-view, collecting footage from several cameras that need to be placed on a high vantage point, and tilted towards the floor, so that the difference in size of the people being closer to the camera, or further away is very small or negligible (e.g. that is not the case for PETS [74]). Also, it would be desirable that the types of actions performed by the actors are similar to those of related datasets, such as the UMN dataset [60, 162], which, unfortunately,

only captures the scene from a single camera. Therefore, the ideal dataset would be a combination of the two just mentioned, that is: containing relevant actions performed by several authors, and recorded from a number of viewpoints with a high vantage point.

5.3.1 The Penrhyn Road Campus Dataset

All experiments have been conducted using a novel dataset which was presented in [55], consisting of 17 video sequences recorded from four different viewpoints (i.e. $17 \times 4 = 68$ videos in total). The cameras were installed on the façade of a building, two of them on the second floor, and two of them on the fourth floor (see Fig. 5.7 for camera locations, and Fig. 5.8 for example captures). In the videos, 20 actors perform several stage group activities:

1. Walking as a single group between two points; or as two crossing groups (starting from opposing points in the courtyard).

2. Walking in one direction, but having some people in the group *abnormally* deviating from the trajectory followed by the rest.

3. Simulating a chaotic event, where everybody runs away from a danger.

These sequences have been labelled into three different categories, namely: `normal`, `abnormal` and `chaotic`, respectively. Examples of video frames from all different video categories can be found in the materials Appendix, Figs. A.16 and A.17, from p. 186. For the purposes of the first experiment described before, only one of the views is used (Bottom-right, #4), for which there is an additional sequence (see Table 5.1).

5.3.2 Experimental set-up and parameters

There are several components that need to be validated through experimentation, and therefore the following experiments have been devised:

Figure 5.7: Camera locations in the façade of the building. Left: as seen by a bystander (© `www.kingston.ac.uk`). Right: as seen from satellite images (Imagery © 2016 Google, Map data © 2016 Google).

Category	Description	Sequences	Length
`normal`	Group(s) walking, crossing	9 [a]	5 min 00 s
`abnormal`	Deviations from the group	5	2 min 56 s
`chaotic`	Panic event (dispersion)	3	1 min 04 s

[a] there is an additional `normal` sequence for the bottom-right view, therefore being 10 `normal` sequences and 18 videos for that view in total (used in Experiment 1).

Table 5.1: Characteristics of the dataset.

- Experiment 1. Analysis of the best parameters for the bag-of-words modelling, and additionally, an analysis of the performance of different TPH extraction techniques, in order to evaluate the strength of the four different proposed TPHs (circular and polar, with or without intensities). A leave-one-out cross-validation is used (LOOCV) [14, 102]: training of the system is done using all sequences but one ($S_{\text{train}} = S - s_{\text{test}}$), and testing on the left-out sequence (s_{test}); doing this for all sequences alternating the sequence that is left out.

- Experiment 2. Baseline approach, that is, performance analysis of each view separately; that is, using TPHs from one view directly for training the model, and without concatenating them into multi-view TPHs (MV-TPHs). This demonstrates the validity of the TP as a scene descriptor. A LOOCV approach is used as before.

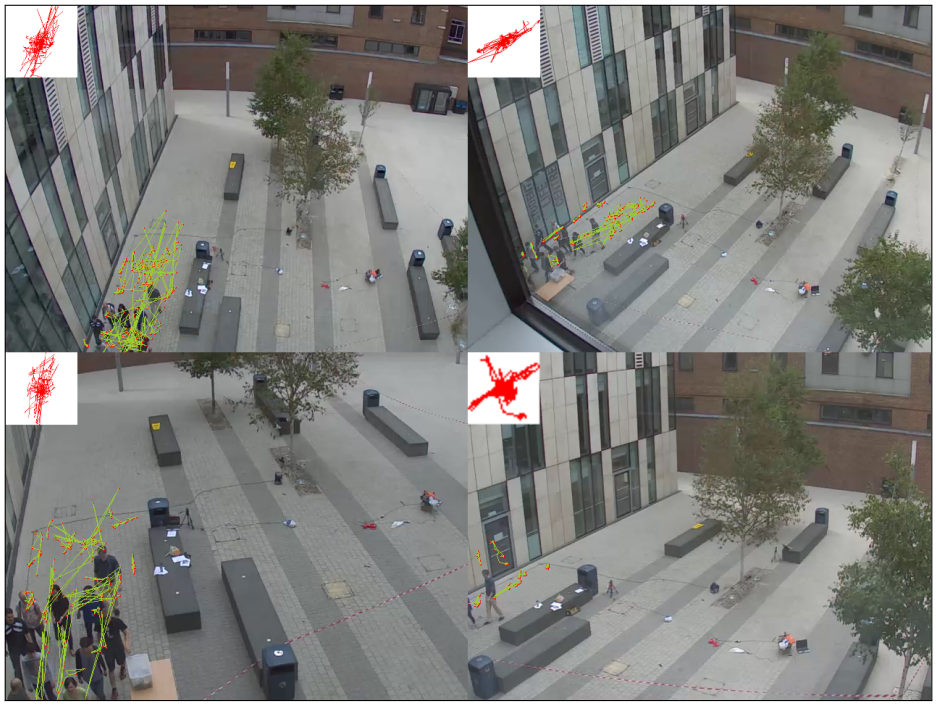


Figure 5.8: A '`normal`' example of the multi-view dataset employed, with superimposed tracklets in green, and *tracklet plots* (top-left of each view, in red). Some tracking error can be observed in the bottom-left, and the bottom-right TP.

- Experiment 3. A performance analysis when each views' TPHs are fused into multi-view descriptors (MV-TPH) and the model is trained on the latter. Again, using a LOOCV framework. Results are compared to those obtained when using different dimensionality reduction methods mentioned in Sec. 5.2.2.

- Experiment 4. An analysis of a K-fold cross-validation, to determine how the system would respond to varying number of folds (K) (that is, with decreasing sizes for the training set).

Moreover, all the steps through the process involve a number of parameter decisions. For instance, there are the parameters regarding the number of frames taken to produce one TP (Δ), a parameter that was introduced in the methodology (Sec. 5.2.1.1), or its normalised size ($L' \times L'$), as seen in Sec. 5.2.1.2. Table 5.2 shows these parameters along with the values used in all experiments to generate the tracklet plots. As for the values employed for the parameters shown in the table, the value for L' was selected heuristically, assuming that a motion of two pixels per frame for an interval (Δ) of 50

frames would need 100 pixels to be represented. Such a motion is quite rapid, as it is observed that in most cases motions of the centroids of people walking normally are less than this. The rationale is that, in the presence of a running person, walking individuals' tracklets will not be changed much by the normalisation, thus preserving their length and clear direction of motion. On the other hand, the reason for picking a Δ value of 50 frames is justified by the fact that it would translate to 2 s on a video with a frame rate of 25 fps, and as stated, a short period is desirable as visual trackers have trouble in the longer run (i.e. the *upper* limit would be loss of track). However, picking a smaller value for delta yields very short tracklets for which it is difficult to assess the direction of motion (i.e. this would be the lower limit). Therefore, the value of Δ was selected by these given constraints.

Parameter	**Value or range**
time interval (Δ)	50 frames
normalised plot size ($L' \times L'$)	100×100 pixels

Table 5.2: Parameters used for the construction of TPs in all experiments

Experiment 1 is conceived as a way to determine the best-performing TPH extraction technique; several TPHs are used, the binning parameters are given in Table 5.3. Furthermore, another goal of that experiment is to determine the best values for the `iter` and `reps` parameters. The parameter `iter` is in reference to the number of *iterations* that the k-Means algorithm is run during the bag-of-words model acquisition; and `reps` is the number of times that the BoW is run per test. Tests are conducted with `iter` = 3 and `reps` = 5; and `iter` = 1 with `reps` = 15.

For all further experiments (Experiments 2–4), parameter values are set based on the results from Experiment 1, these are shown in Table 5.4.

Histogram (TPH) modality	Number of bins
Circular	6 rings × 255 values
Polar	8 sectors × 6 rings × 255 values
Circular (without intensities)	10 rings
Polar (without intensities)	8 sectors × 6 rings

Table 5.3: Number of bins for the histograms compared in Experiment 1.

Parameter	Value or range
Polar histogram[a] bins	6×8 bins
k-Means K	$K = 2, 3, \ldots, 64$ key words
k-Means `iter`	3 iterations
BoW `reps`	5 repetitions

[a] Only these TPH (without intensities) are used, justified by results from Experiment 1 (see Sec. 5.4.1 below).

Table 5.4: Additional parameters used for Experiments 2–4

5.4 Results and Discussion

As explained in the previous section, different experiments are proposed in order to validate the approaches used in different parts of the methodology. Following are the results for each of them.

5.4.1 Experiment 1: Analysis of BoW parameters and TPH extraction techniques

The k-Means clustering algorithm employed to cluster the words, and find their representative key words, has a random initialisation of the cluster centres, and is therefore prone to give different results when run several times. As introduced before, the `iter` parameter defines the number of times the algorithm needs to be run. The clustering error (calculated as the distance from the cluster members to its centre) is calculated, and the result with the lowest error is selected at the end of the process.

As mentioned, this experiment has two main goals, namely 1) to find the best

parameters for the BoW, and 2) to determine the best TPH extraction technique. Therefore in this experiment a single view is used, in this case the bottom-right view. Furthermore, since the goal is to find the best values for some parameters, the optional Kalman filtering step is enabled (described in Sec. 5.2.1.1). In subsequent experiments (Experiments 2 & 3, in Secs. 5.4.2 and 5.4.3, respectively), more will be said about the advantages or disadvantages of using or skipping the optional Kalman filtering of the tracklets, and comparisons given for both cases.

Figures 5.9 to 5.11 present the *classification success rate (CSR)* as the number of correctly classified sequences over the total number of sequences (either normalised over 1 or shown as a percentage value), when using different values of K, that is, different number of *key words* for the k-Means used for the bag-of-words model. However, please read Sec. 5.4.3.1 below for further cues on how to interpret the reported results.

Figure 5.9 shows the results of the comparison between the two configurations for `iter` and `reps`, using either 'circular' or 'polar' histograms, in both cases with intensity values. As can be seen, in general terms, the results of both configurations are very similar, and shows that with polar histograms classification success rate is more homogeneous regardless of the number of key words used in the model. However, in all other experiments, it has been decided that the configuration presented on the left (`iter` $= 3$ and `reps` $= 5$) will be used, since it makes more sense theoretically: that is, using 3 iterations of k-Means is better as a way to overcome the initialisation problem of that algorithm, as opposed to having to *trust* on a single iteration, and then finding the best of 15 models generated (repetitions).

After that initial experiment, results were obtained for all four modalities of TPH extraction. That is, polar and circular histograms (with and without intensity binning) with the selected configuration. Figure 5.10 shows the results when polar histograms are employed, whereas Figure 5.11 shows the results for the circular histograms. The shaded areas in these figures span from the minimum to the maximum results obtained, while the solid lines depict the mean values. As it can be seen, the best results are obtained when using polar histograms with no intensity binning (Fig. 5.10), peaking

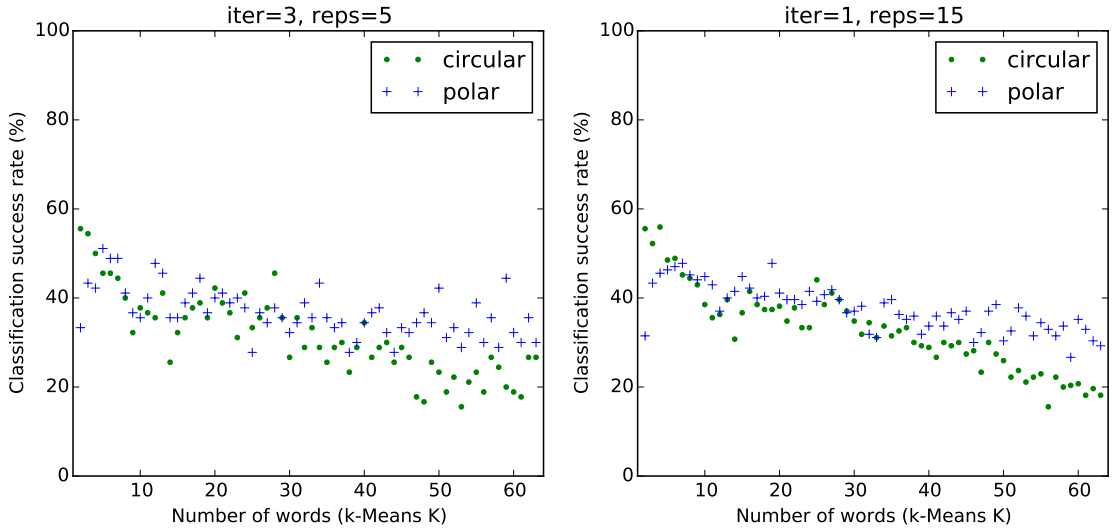


Figure 5.9: Maximum classification success rate comparison for two configurations of `iter` and `reps` with different number of key words, using polar (blue, solid) and circular (green, dotted) histograms (with intensity bins).

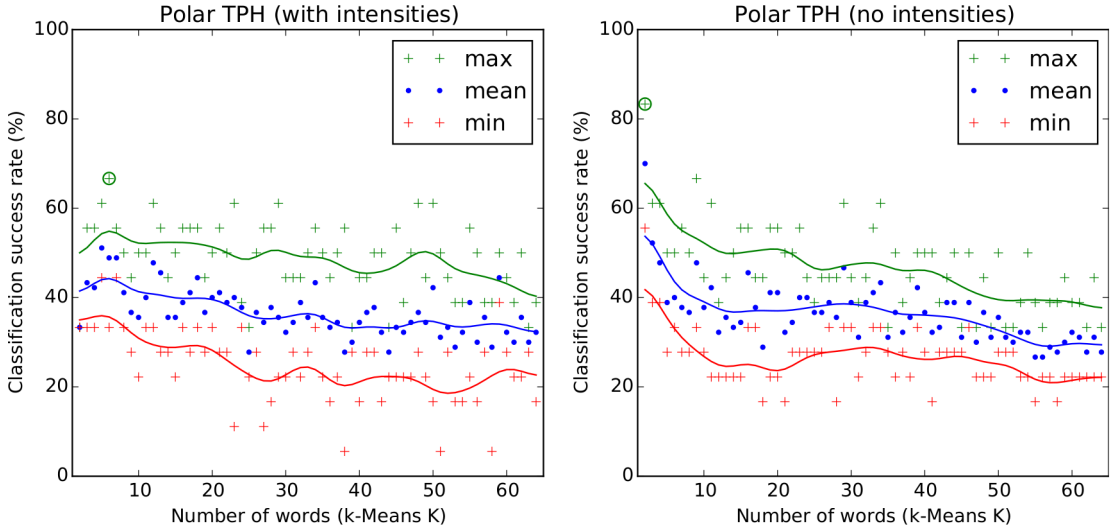


Figure 5.10: Maximum, mean and minimum classification success rates for different number of key words ($K = 2, 3, \ldots, 64$) using **polar** TPHs with and without intensity binning. Best runs appear circled.

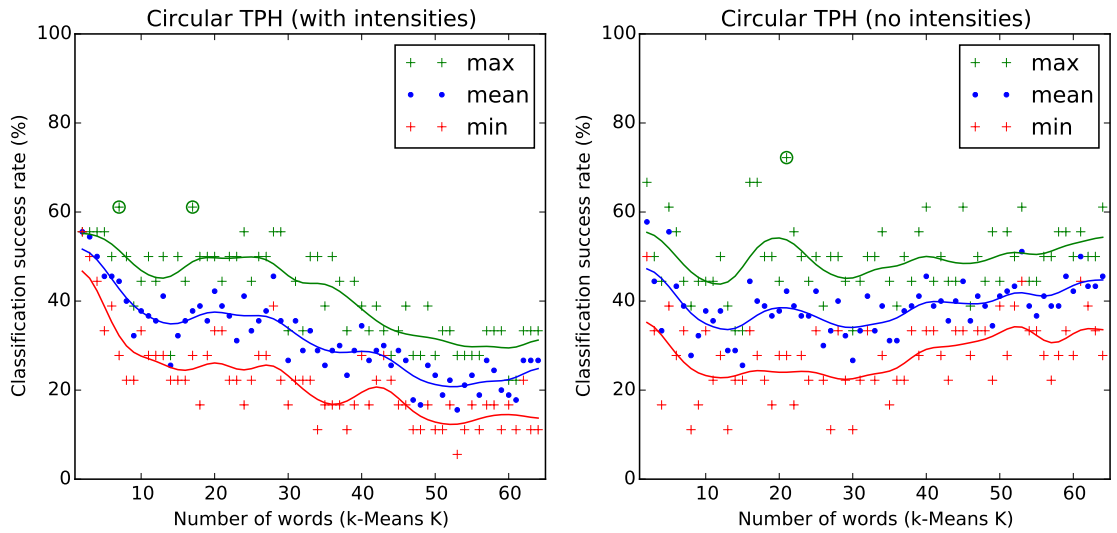


Figure 5.11: Maximum, mean and minimum classification success rates for different number of key words ($K = 2, 3, \ldots, 64$) using **circular** TPHs with and without intensity binning. Best runs appear circled.

at 83.3% for ($K = 2$). If using intensity bins, only 66.7% is obtained ($K = 2$).

Likewise, results are given for circular histograms (Fig. 5.11). As it can be observed, the results are not as promising when using this TPH modality, the maximum classification success rate is achieved at $K = 21$, with a percentage of 72.7% (no intensity binning), and a lower 61.6% ($K = 7$) in the case where intensity bins are used.

Table 5.5 summarises the results in numerical form for clarity (worse minimum result, best maximum results, and best mean). As it can be observed for the polar case, not only the best result is the highest, but also the worse rate is higher than when using intensity bins, which seems indicative of this binning being a drawback, most probably due to its very large sparsity and high dimensionality. The same can be said in the circular histogram case, however, the results are not as good, most probably because this type of TPH binning cannot provide sufficient cues to describe some important parameters of the crowd.

Additionally, results are generally better for lower to moderate[1] values of K (number

[1]Other runs in experiment 3 show their best result with $K = 6$. Furthermore, right graphs in Figs. 5.10 and 5.11 do not show a clear 'elbow' shape. However, experiments conducted later (Exp. 3) show a clearer shape, which makes their results more reliable. See additional Figures (A.18 through A.20) in the materials Appendix (p. 188).

Value	Polar histogram (TPH)				Circular histogram (TPH)			
	With intensities		Without intensities		With intensities		Without intensities	
	CSR (%)	K	CSR (%)	K	CSR (%)	K	CSR (%)	K
Overall min.	5.6%	51	16.7%	18	5.6%	53	11.1%	8
Overall max.	66.7%	6	**83.3%***	2	61.6%	7	**72.7%***	21
Highest mean	51.5%	5	70.0%	2	55.6%	2	57.8%	2

Table 5.5: Maximum, mean and minimum classification success rates (CSR) for all modalities of polar and circular histograms (K stands for the number of words, the k-Means K), using bottom-right view, with 18 sequences (with Kalman-filtered tracklets).

of keywords). This seems reasonable, since the number of different situations to be described is small, it is the combination of these key words that will define what normality, abnormality or a chaotic situation is (for instance, by having TPs labelled as *inconsistent* in different proportions). To summarise, from the results of this experiment, it can be concluded that: 1) `iter` and `reps` should be set to 3 and 5, respectively, for all subsequent experiments; 2) polar histograms should be chosen; 3) binning of intensities is counter-productive and should not be used. These parameters are summarised in Table 5.4, which was introduced earlier (Sec. 5.3.2).

5.4.2 Experiment 2: Baseline results for separate views

In this experiment, the TPHs from the different cameras are not fused into MV-TPHs, but instead are used to feed four separate models, and apply a leave-one-out cross-validation (LOOCV) on each of them. Results are shown in Table 5.6, where it can be seen that the bottom-right view is the best-performing of the four available, and that not all views perform equally well, the reasons could be twofold: on the one hand, top views are further away, and as a result the regions of the targets to track are much smaller (less information to model the targets' appearance); on the other hand, the level of occlusion due to trees or other objects from the different viewpoints is variable. As explained in the methodology, the Kalman filtering of tracklets is an optional step, and therefore, in this experiment, two different configurations were

tested: either using filtered tracks, or using the original non-filtered ones. As it can be observed, the highest results are obtained when no Kalman filtering is used; and in general the performance is the same or slightly better when this optional step is not applied. The reason for this could be explained by the fact that the filtering is not only removing noise, but could potentially also remove some important information (i.e. it oversimplifies the tracks' shapes). Thus its use seems redundant, as it adds to the computational cost that could otherwise be saved, and does not seem to improve the results.

Camera	**Non-filtered tracklets**		**Kalman-filtered tracklets**	
	CSR	K	CSR	K
Top-left (TL)	64.7%	9	70.6%	13, 26
Top-right (TR)	70.6%	12, 22	64.7%	17, 23
Bottom-left (BL)	70.6%	8	**76.5%***	11
Bottom-right (BR)	**82.4%***, [a]	6	70.6%	14, 28

[a] Please note the difference between the value reported here (using 17 sequences) for the bottom-right view, and the one reported in Table 5.5 (18 sequences).

Table 5.6: Results for each viewpoint separately (baseline approach).

5.4.3 Experiment 3: Multi-view fusion and dimensionality reduction

This experiment shows the performance of the entire work flow, including the multi-view fusion described in the methodology (fusion of TPHs into MV-TPHs). The results from the previous experiment will be used here for comparison; that is, to determine how well a multi-view fusion scheme performs as opposed to separate views. Furthermore, dimensionality reduction techniques are applied over the MV-TPHs, and the results compared to the non-reduced fusion scheme. As in previous cases, a LOOCV framework is used for evaluation.

Table 5.7 shows the number of dimensions achieved for the dimensionality reduction techniques that were used (introduced in Sec. 5.2.2). In all cases, where principal

component analysis (PCA), or its variant kernel PCA, were used, the final number of dimensions was selected using values that would keep more than 80% of the variance. For Isomap and semi-definite embedding (SDE, also maximum variance unfolding, or MVU), the number of dimensions is automatically determined by the method itself. As it can be seen, the dimensionality that is achieved is up to two orders of magnitude smaller than the original combined feature, this is justified, apparently, by the fact that the histograms employed are sparse, and therefore much of the information is concentrated in only a few of all the available bins.

Method	Final number of dimensions selected/achieved	
	Kalman-filtered tracklets	Non-filtered tracklets
Original fused feature	192	192
PCA*	25	30
Gaussian Kernel PCA*	25	30
Isomap (radius = 5)	20	20
SDE/MVU	7	7

* $\geq 80\%$ of variance.

Table 5.7: Dimensionality reduction techniques and final dimensions selected.

Table 5.8 presents the results for the all cases: with the original multi-view fusion approach (MV-TPHs with no dimensionality reduction applied); and all other cases (PCA, Gaussian Kernel PCA, Isomap and SDE/MVU). From the results, it can be concluded that, when using fusion of the features of all viewpoints (first row), the accuracy is as high as the best-performing single view available; this shows that the fusion scheme is not causing an overhead, but instead it facilitates for the best available decision to be taken. However, such a model takes more time to train (it has 192 dimensions, and the time to train the model increases linearly with the number of features/viewpoints added). Up to this point, it is not justified to choose this system over one with four separate models (one per viewpoint), that just picks the output from the best-performing camera. However, the camera that gives the best results is only known *a posteriori*; that is, it is known once the system has been trained and subsequently tested.

Method	Non-filtered tracklets		Kalman-filtered tracklets	
	CSR	K	CSR	K
Original fused feature	**82.4%***	20	70.6%	20
PCA	70.6%	2	76.5%	15
Gaussian Kernel PCA	70.6%	11, 29	70.6%	16, 20
Isomap	**82.4%***	17	64.7%	11, 14, ...
SDE/MVU	70.6%	20	70.6%	19

Table 5.8: Results with multi-view fusion (and dimensionality reduction).

Furthermore, with the introduction of dimensionality reduction, and specially seeing the results of Isomap of 82.4% classification success rate, the same accuracy as using the original multi-view fusion scheme with no reduction, it is easy to see the advantage of the multi-view system above having four separate models as just suggested: with only 20 dimensions, the learning of the model is much more rapid, even faster than training one single-view model (48 dimensions), and therefore, the multi-view model is preferred, as it can take advantage of all the available data from all viewpoints to make a better decision, *harvesting* the best result without any *a priori* knowledge about the performance of each camera view. It is also worth noting that, even if the accuracy falls to 70.6% when using SDE/MVU, the dimensionality reduction is drastic in this particular case, reducing the original 192 dimensions to only 7. In this case as well, the Kalman filtering step is not required, as results do not generally tend to improve, as compared to non-filtered tracks.

To finalise this experiment, Figure 5.12 provides confusion matrices showing the classification rates for each class separately using the original combined feature (the reader should note that, the provided matrix corresponds to a total classification rate of 76.5%) and two dimensionality reduction (DR) techniques: Isomap (best result, 82.4%) and PCA (second best, when using Kalman-filtered tracks, 76.5%). In general trends, it can be seen that the best-classified sequences are the `normal` ones, since they are classified correctly in 100% of the cases for the original feature, and on 78% on the results when DR is applied. Sequences labelled `chaotic` follow, with 100% of

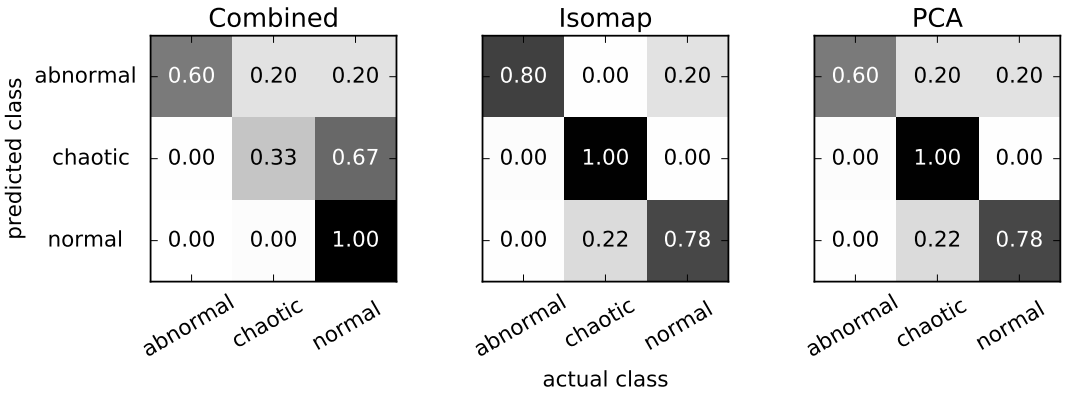


Figure 5.12: Confusion matrices showing the classification success for each class in different configurations: left, using the combined original feature; centre, using Isomap to reduce dimensionality; and right, using PCA.

cases correctly classified with DR, but only 33% classified correctly with the original feature, which is an unexpected result, as the rest are classified as `normal`, which are the most different. This could be due to some poor clustering initialisation in the k-means algorithm for this specific run, and/or the high dimensionality of the original feature, which would explain the excellent results achieved with dimensionally-reduced features. Lastly, `abnormal` sequences seem to be the ones the algorithm gets the most confusion. In part, this is expected as these sequences are somewhere halfway between the two other categories (`normal` and `chaotic`). Nevertheless, the majority of these sequences are still classified correctly (with a minimum of 60% with the original feature and PCA).

5.4.3.1 Impact of an unbalanced dataset

A factor that impacts the interpretation of the presented results is the fact that the proposed dataset is unbalanced, that is, that there are more samples of the `normal` class than there are of other classes. However, the existing imbalance with respect to the other classes is small (1.8:1 and 3:1, for the `abnormal` and `chaotic` classes, respectively), as compared to other datasets and domains (e.g. card fraud) [46].

Having an unbalanced dataset impacts the meaning of the classification results, since a classifier always returning the *majority* class would benefit from the imbalance,

with respect to a *chance* classifier. With a balanced dataset, with three classes as in the dataset in this chapter, both *dummy* classifiers (*majority* and *chance*) should have a classification success rate of one third (33%). However, the imbalance in the number of samples means that a classifier that always assigns `normal` as the output class for all sequences would have a 53% classification success rate.

One way in which this problem can be addressed is via re-sampling techniques that can be applied on the dataset, as is for instance randomly removing samples from the majority class [80], however these have a negative impact as under-sampling the majority class might result in loss of valuable information. Also, with small datasets as is the case, it would be infeasible and counter-productive. Other methods involve adding a penalty for misclassification, so that a dummy classifier that always returns the majority class has the same classification success rate as *chance*. Furthermore those techniques are focused on the effects on training with classifiers tending to simpler models that disregard the samples of a minority class as outliers, rather than how that affects the reported results.

With unbalanced datasets it is important to report not only the classification success for the whole dataset, but also broken down per-class [281]. Classification success rate values normalised over the number samples in each class are shown in the confusion matrices presented in Fig. 5.12. There do not seem to exist consistently misclassified sequence categories. However, the results for the whole dataset presented in Tables 5.6 and 5.8 do not take into account the dataset imbalance. That is, a reported classification success rate of 82% does not mean 49 percentage points over chance (33%), but instead 29 percentage points over a 53% classification success rate of a dummy classifier that has a fixed output (i.e. `normal`).

5.4.4 Experiment 4: Results with descending training set size

The aim of this last experiment is to show how the system responds when the size of the training set is decreased. To do this, a K-fold cross-validation is used with different numbers of folds, so that the less folds, the smaller the size of the training fold will be.

Table 5.9 shows the configurations used (with 10, 5, 4, 3, and 2 folds), along with the sizes of the training and testing splits.

Folds	**Training split (%)**	**Testing split (%)**
10-fold	90%	10%
5-fold	80%	20%
4-fold	75%	25%
3-fold	66.7%	33.3%
2-fold	50%	50%

Table 5.9: K-fold cross validation configurations.

The results for 7 selected samples are presented in Figure 5.13. These series have been selected because any one value surpassed the 0.7 accuracy mark (topping at $K = 5$, $K = 37$ and $K = 52$). From the series in the figure, it can be observed that the general trend, as is logical, indicates that the accuracy goes down as the number of training samples is smaller. However, this trend is not always clear, such as in the series where $K = 3$, or for $K = 37$; but in these cases, the starting and ending accuracies are quite small. More representative examples of that downward trend seem those where $K = 5$, or $K = 46$ (or even that where $K = 52$). Another factor that influences how clear the trend will look is related to the fact that the splits are randomly selected, thus, the selection process could leave all sequences of a given category (e.g. `chaotic`, for which there are only three sequences) outside the training split. That yields a testing split that contains types of sequences that have never been observed during training, which negatively affects the performance.

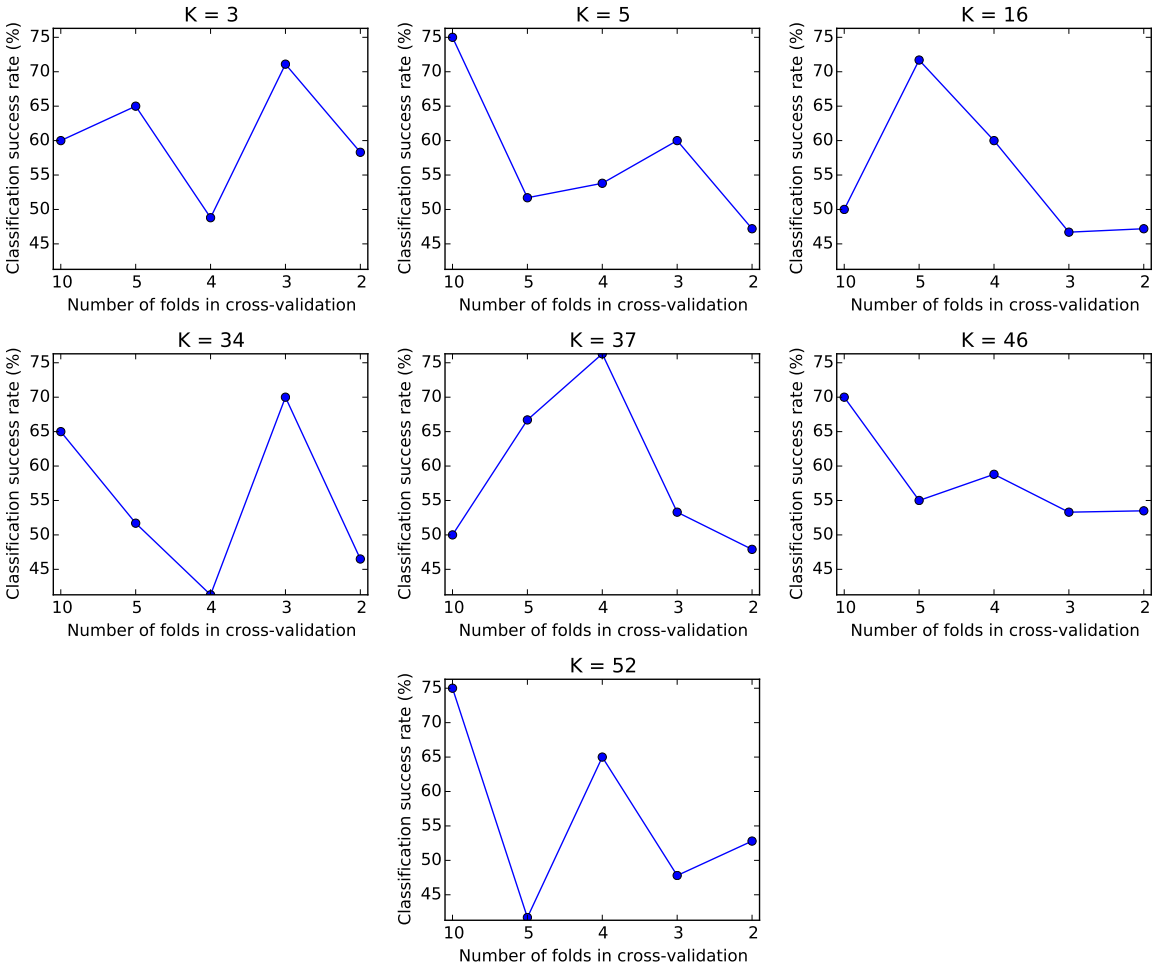


Figure 5.13: Results for the K-fold cross-validation test, on decreasing number of folds (smaller training split sizes). Each series depicts the trend for different values of key words (k-Means K).

5.5 Conclusions

In this chapter, a compact descriptor based on the *tracklets* of the people present in the scene during a time interval of a given video input has been presented. Tracklets are extracted from a time window using a particle filter multi-target tracker. A de-noising algorithm can then be optionally applied, to obtain smooth tracklet trajectories. The tracklets are then plotted in a square image, and different histogram binning techniques can be applied to this compact representation, obtaining tracklet plot histograms (TPHs). After view synchronisation, the TPHs from different views can be merged into a single feature. Furthermore, a bag-of-words modelling has been employed over these

features (MV-TPHs) for crowd event recognition. The proposed method has been validated by several different experiments: first, for the TPH techniques used, and the parameters of the modelling algorithm; then, for the TPHs of each view separately; after that, for all views combined, as well as using dimensionality reduction on the MV-TPHs, and finally using a K-fold cross-validation with descending sizes for the training split, to determine the robustness of the method.

As said in the discussion of Experiment 1 (Sec. 5.4.1), low to moderate values of K (*key words*) seem to perform better (For the first three experiments). The best performing values are always in the first half of the analysed range ($K = 2, 3, \ldots, 64$). Histograms (TPHs) with intensity binning perform worse than their intensity-less counterparts. Polar histograms seem to perform better than circular, which is logical, as was discussed. Regarding TPHs, it could be interesting to develop other types of histogram extraction techniques to capture the motion directions and velocities separately, rather than with a polar histogram, in which each bin captures both the motion direction and speed. Additionally, it could be interesting to rotate the sectors to fit the dominant motion depicted in the TPs, so that it is not split into separate bins, however these variants are left for future work. Furthermore, from Experiments 2 and 3, it can also be seen that Kalman filtering of tracks which was introduced in Experiment 1 as a de-noising step for the tracks is counter-productive.

Regarding the maximum achieved accuracy rate, it is worth mentioning that the presented dataset is a very challenging one, due to the presence of heavy clutter in the form of trees, benches and other objects, as well as inter-occlusions among persons. All this complicates the tracking, which subsequently becomes the bottleneck step. A poor people tracking will result in worse performance in general. Further work needs to be carried out in this regard.

As for Experiment 3, the *complete* method was presented, in which evidence from multiple viewpoints was combined. Novelty resides in the fact that TPs can be combined into one single multi-view feature without a loss in performance, and actually the system can harness the best result from all those available. Regarding

comparison with other datasets, more experiments should be performed using multi-view multi-pedestrian benchmark datasets, such as PETS 2010 [74]. Other benchmark datasets exist, as is the 'UMN dataset' [60, 162], but do not fit the purpose of this work, since they only include data from a single view.

The presented system combines tracklet plot descriptors from several cameras using feature-level fusion. Using this scheme, it was able to perform as well as the best view available. With regards to the mode of fusion used, *model-* and *decision-level* fusion techniques have not been tried, and are also left as future work.

Finally, the dimension of the concatenated feature could be reduced by one order of magnitude without loss in the performance rate; however, it could be interesting to apply more techniques, such as locally linear embedding (LLE) [204]; or to apply the same techniques with alternative configurations; for instance, Isomap, which currently performs as well as the non-reduced data, could be applied with a k-NN rather than a fixed radius.

Chapter 6

Conclusions

Three main contributions have been presented in this thesis, namely: a method for crowd granularity assessment (Chapter 3); telemetry-assisted aerial video surveillance search window correction for tracking, and background modelling (Chapter 4); and finally, tracklet plot scene descriptor fusion from multiple views for event recognition in large groups of people (Chapter 5).

6.1 Contribution highlights

- Crowd classification using a density-entropy signature
  - Most works rely on density as the sole measure for the assessment of the level of danger in crowds.
  - Crowds can also be classified for other purposes, such as selecting the best method for further analysis.
  - A novel density–entropy signature is presented for crowd classification.
  - The entropy expresses the level of *orderliness* of the scene.
  - The obtained signature is *richer* than density-only approaches, since it contains additional cues.
- Telemetry-based airborne video surveillance methods

  - Telemetry-assisted methods can improve purely video-based methods that are affected by poor texture.
  - Search window correction can improve tracking performance by 50% on average, with three-fold improvements in some cases.
  - A telemetry-based background modelling method can outperform corner-based methods with poor texture.
  - In other cases, it can work similarly, with less computational overhead.

- Analysis of crowd behaviour from microscopic analysis
  - A novel scene descriptor for large groups of people called 'tracklet plots' is presented.
  - The proposed descriptor is validated using a single camera workflow.
  - Feature-level fusion from descriptors obtained from multiple views is shown.
  - The combined results are as good as the best-performing view without prior knowledge.
  - Dimensionality reduction increases training speed while maintaining results.

6.2 Discussion

6.2.1 Crowd classification using a density-entropy signature

In this first study, crowd granularity assessment is explored. Most existing works rely on density as the only crowd feature to assess the level of danger in the crowd. However, a very dense crowd can be safe as long as it is orderly. Therefore, in the proposed method, along with density, an orderliness or entropy score is calculated. Few works in the literature seem to use entropy as defined here. Using a density–entropy signature, crowds can be classified. Since methods to further obtain information from the crowd can be different depending on the features captured from the scene, with the method proposed in this chapter, crowds can be labelled accordingly. The results

obtained look promising, illustrating the potential of the method, as indicated by the qualitative, as well as quantitative results provided.

Using a dense optical flow and segmentation with background modelling, density and entropy scores of the crowd were calculated, and used in a density–entropy signature. Using the segmented foreground obtained from the background model, and the optical flow directions, a map for density and a map for entropy are constructed, respectively. These maps are then sum-one normalised to become PDFs, and then compared to uniformly distributed maps of the same size via mutual information (MI). The final scores are then expressed as one minus the normalised MI value obtained. The scores are then used as a point in a $2D$ curve, which can be quantised into several levels (i.e. quadrants), and used for crowded scene classification. The method was validated using human-labelled data on a number of sequences from a well-established dataset.

Findings and Limitations

As observed in the results, each used estimator (density and entropy) performs generally well (80% and 73% on average, respectively). This shows that the proposed analysis methods are a good choice for crowd density and orderliness estimation, respectively. The evaluation results using the combined density–entropy signature average to 60% of instances classified successfully (Table 3.2, p. 78).

This lower result for the combined response can be justified: the evaluation method compares the final scores in the $2D$ point $(\rho, \mathcal{E})$ to the human-labelled quadrants, and both values need to match the human label, as a combination or *product* of the two estimations (logical 'and'). Therefore, it can happen that for a given frame, the density score is within the human-labelled range, whereas the entropy one is not, or vice-versa. If only one score is in the same quantisation level (i.e. *quadrant*), then the whole estimation is considered to be wrong.

There are no other works combining several crowd cues for classification, and most are based on density only. It is easy to see that such methods might have higher

performance, given that only one estimator is used, however, crowd classification in those cases is only based on a single cue, whereas the proposed method takes a combination of several cues into account, and therefore can offer a richer more informative response. Since the entropy results are further away from the human labelling responses, it seems logical to further study how to improve this particular estimator (Fig. 3.8, p. 83).

6.2.2 Telemetry-based airborne video surveillance methods

In the second study of this thesis, two methods using vehicle-provided telemetry data for video surveillance from UAVs were proposed, as an alternative to purely video-based methods for image stabilisation, camera pose estimation, and image matching or registration, which can fail in the absence of texture in the background. On the one hand, a method for the correction of the search window of a tracking algorithm is presented, this is less computationally expensive than performing a full image registration, which is unnecessary for tracking, but is the most common approach used in the literature. On the other, a background modelling technique using a global refinement after crude alignment using telemetry is introduced. The experiments conducted using the OctoXL platform show that telemetry is a reliable source of additional cues for aerial video surveillance methods, and that the techniques using this additional data can perform better compared to well-established methods in the literature, or to baseline counterparts. This is especially true for cases in which purely video-based methods do not work, such as when the texture of the scene background is poor. No previous works present the transformation of the search window of a visual tracker by means of telemetry data. Also, no works were found using telemetry data or corner-based video-only frame registration techniques in conjunction with global registration methods.

In both methods presented in this chapter, data from global positioning and inertial navigation systems (GPS/INS) is first pre-processed, to express the information in units that are useful and easier to manipulate. Once this step is performed, the data

can be used. Since the planarity and orthogonality assumptions are made (Sec. 4.1.1.1, on p. 91), the transformations undergone by the current video frame with respect to the previous frame are limited to translations (in X, Y, expressed from the vehicle's coordinate system, as shown in Fig. 4.2, p. 104), rotations along the Z axis (i.e. yaw, or ϕ), and scaling (translations along the Z axis, i.e. changes in altitude). Therefore, in the first method, using this data, the position of the search window (win) in the current frame can be calculated. In the second method, a similarity matrix S is built, which incorporates refinement translation parameters calculated by a global registration method based on the discrete Fourier transform (DFT). In this way, the pixels of the previous frame are matched in the new frame, and used to update the background model directly.

Findings and Limitations

Experiments are conducted for both methods: the search window correction method is first validated using ground truth data as a *perfect* tracker, and then compared to a baseline approach where tracking is used without correction. The background modelling method is compared to well-established video-only approaches, and a novel approach where these methods are improved with the DFT-based refinement used in the presented approach.

From the experiments conducted on the first method, it can be seen that validation can be conducted successfully using a novel measure (C-measure) accounting for how well contained the target is in the search window after all transformations have been applied to it. The target is found to be inside the expected window in 99.7% of the cases on average, with very low standard deviation of 1.4% (see Table 4.1, p. 109). This demonstrates that the performed search window correction works as expected when assuming perfect tracking, and therefore validates the approach used. A second experiment compares the data from a baseline approach (i.e. using no correction), to using the proposed search window correction method. From the results it can be seen that, on average, the PASCAL scores improve by 50% (1.5 factor), peaking

at improvements greater than three-fold (3.31 factor) for selected sequences where rotations over the yaw axis are very prominent (see Table 4.2, on p. 112). Other results show no improvement, most likely related to the nature of the sequences, in which no prominent fast rotations are present. That is, in those cases, the tracker itself can account for the rotations and translations, and therefore the search window correction algorithm does not make a difference. Finally, in one particular sequence, the results are worse with correction, but that is due to the fact that the tracking algorithm itself performs quite badly (i.e. re-detection fails). It can be demonstrated that it is a tracker issue because the validation results for that same sequence are among the highest ($99.9 \pm 1.3\,\%$, as shown on Table 4.1, p. 109). That is, the target would be within the expected search window, and there should be no problem in re-detection.

The experiments conducted on the second method entailed three different tests. A first test was envisaged to determine the best-performing global registration method. Two methods were compared, one based on the mutual information (MI), and one based on the discrete Fourier transform (DFT). It is shown that, in the presence of brightness change, MI performs better, yet, if using gradient images, rather than colour, DFT can perform at the same level. Since the DFT-based technique is faster, it is selected for all further experiments. After that, purely video-based corner-based techniques (SIFT, Harris), are compared to the proposed method using telemetry and DFT-based refinement. It is shown that, regardless of the background texture, the proposed method outperforms the compared methods. Furthermore, if adding the proposed refinement step to the compared methods (and this would be a novelty), it can be seen that these can perform better than the proposed method. However, this is only true in the instances where the background texture is prominent, since matching would fail otherwise. Furthermore, the proposed method has real-time capabilities, whereas most interest point detection algorithms can be very slow. It can be concluded that the proposed method is preferred in poorly textured scenarios, but could be used in conjunction with purely video-based techniques in the presence of richly textured scenarios, to harness the best results of both methods.

Regarding limitations of the presented approaches, there is the dependence on GPS/INS units, which could not work when working in scenarios where a GPS signal is not available (e.g. among tall buildings or indoors). In the search window correction method, re-identification of people that leave the field of view has not been addressed. Regarding the background modelling method, the disadvantage lies in the fact that due to the high learning rate necessary, stationary foreground objects are quickly absorbed into the background model. This can be mitigated if using a visual tracker that does not rely on foreground information in conjunction to the background modelling method, that can take over and continue to track the foreground object, even after it has disappeared from the estimated foreground mask.

6.2.3 Analysis of crowd behaviour from microscopic analysis

In the third, and last, study of this thesis, *tracklet plots* are presented, which are a compact representation of the 'short tracks' or *tracklets* present in a time window of a given video input, which allows describing the motion patterns of a small- to medium-sized group of people in a given short time span. These can be then be used as *words* in a bag-of-words model. Novel video sequences, can then be analysed to detect whether an abnormal or chaotic situation is present. First, a workflow with a single camera is tested, then evidence from multiple viewpoints is combined in a multi-view workflow. By obtaining tracklet plots for each of the views, and synchronising the available video streams, a *feature-level* fusion method by concatenation can be applied. The presented system is able to recognise specific events in large groups of people from multiple cameras, and to perform equally well as compared to the best single view available. Furthermore, the dimension of the concatenated feature can be reduced by one order of magnitude without loss of performance.

Using a visual tracker in parallel on each present person in the scene, and gathering all the short tracks during a given interval of time, a *tracklet plot* describing the directions of motion and speed of the individuals can be produced. Once this is done, tracklet plots histograms (TPHs) can be obtained, using one of several methods

(circular, polar; with intensity binning or without it, four possible combinations in total). Training sequences can therefore be represented as a series of TPHs; or *words*, as is said in the bag-of-words literature; that can be clustered via k-Means to obtain cluster representatives, or *key words*. The frequency of key words in a given sequence is used for classification of events or actions that unfold in the scene. Furthermore, since views are synchronised, the TPHs from each view can be concatenated in a multi-view TPH (MV-TPH), and fed into the bag-of-words model as is done in the single-view case. Advantage over the single-view case is obtained thanks to dimensionality reduction techniques.

Findings and Limitations

As described in the chapter, four experiments were conducted: a first experiment to determine the best values for the parameters used; a second to test the performance of the single-view workflow on each separate view; a third one showing the performance of the multi-view approach as well as the justification based on dimensionality reduction, and a fourth one shows how the algorithm performs as the size of the training set is reduced (K-fold cross-validation).

From the first experiment, it can be concluded that polar histograms have better performance than circular, most likely because the former have higher expressive power, since directions of motion and not only speeds are accounted for (i.e. the length of the tracklets, represent the speed of the individual, therefore 'ring-shaped' regions are able to distinguish different tracklet speeds). Sector-shaped regions are used for different directions of motion in polar histograms. However, binning different intensity values separately (which would be akin to accounting for density or number of people following a given direction), is counter-productive, as observed in the results. This might be due to the sparsity in that modality of histograms, as well as its high dimensionality. Finally, from this experiment, it can also be observed that, in general, lower values of K (number of key words, or cluster representatives) tend to perform better. This seems logical, as in principle, a combination of a few different key words should suffice

to express different situations that are produced in the training sequences, which when combined into a sequence of key words express the evolution of the given sequence.

In the second experiment, baseline results (single-view workflow) are obtained for each view. It can be observed that a particular view (bottom-right, see Table 5.6, on page 146), is the best performing one, with a success rate of 82.4% (with $K = 6$). It can also be seen, that in general, when using Kalman-filtered tracklets, results are lower (maximum result is 76.5% for the bottom-left view). This could be because the filtering removes important information regarding the shape of the tracks apart from noisy tracklet points. Furthermore, regarding the maximum success rate achieved, it could be explained due to the performance of the underlying tracker used, which becomes the bottleneck of the whole process. The used dataset is very challenging as there exist many objects and clutter (trees, benches) as well as inter-target occlusions that make it difficult to obtain good tracks in all cases. This is partially overcome by using short intervals of time, but this does not completely solve the issue for all cases.

The results of the third experiment show that when combining information from multiple views the system can harness the result from the best-performing view. That is, the combination scheme is not causing an overhead, but facilitating for the best available decision to be used. Yet, since the combined feature (each MV-TPH) has 192 dimensions, it takes much longer to train the system than it does for each separate view (48-dimensional), and since it does not give an advantage over the best performing view, it makes no sense to adopt the combining approach. This, however, is not true, given that dimensionality reduction techniques are employed, that can achieve a significant reduction by one order of magnitude (to 20 dimensions) of the original combined feature, without loss of performance. It is only after dimensionality reduction that the proposed system is justified: the training time is faster than it would be for a single view, with the advantage that the system can harness the best available result without prior knowledge of which view is providing the best response.

6.3 Possibilities of integration

This section explores some ways in which the presented works could be integrated. As already stated on Chapter 3, the assessment of crowd density and entropy can be used to determine which types of methods could ensue: if the density and entropy are high, it might not be possible to obtain more detailed information about the individuals forming the crowd, whereas in a sparser scenario, it would be possible to analyse each individual's behaviour as part of the group. Figure 6.1 reflects the interactions that would required among the contributions of this thesis for this purpose:

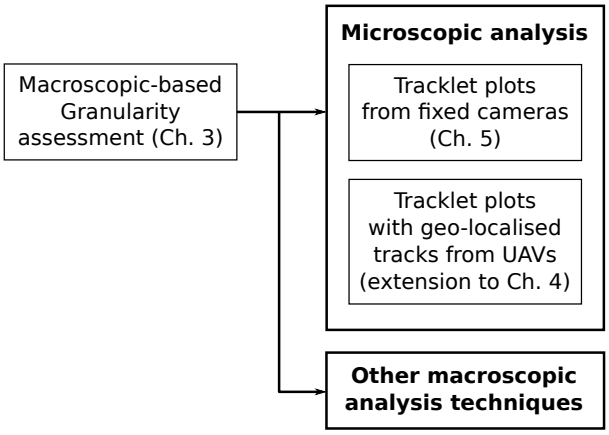


Figure 6.1: Proposed integrated system, as discussed in Chapter 3.

Another interesting system that could integrate every contribution in this thesis would be a system for detecting and classifying crowds from an aerial platform. That is, using all available cues from the aerial platform, one could: first, assess the level of density and entropy in the crowd, and based on the results, if the crowd is sparse, then obtain the geo-localised tracklets of the individuals present in the scene to classify the observed actions into different categories.

Figure 6.2 shows the interactions that would be necessary among the different contributions of this thesis to construct the described system. As depicted, the background subtraction results obtained from an aerial platform can be used for crowd assessment, along with the directions of motion obtained from multiple tracking of ground subjects. This can be used to assess the level of density and entropy of the

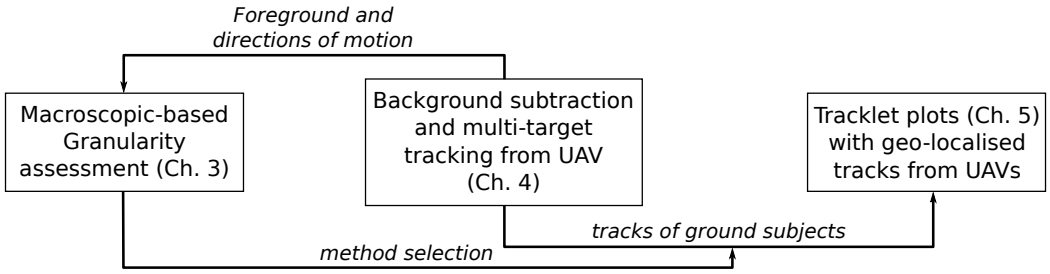


Figure 6.2: Proposed integrated system for crowd assessment and activity classification from an aerial platform.

crowd, as presented in the first contribution of this thesis, and depending on the result, further analysis at the microscopic level using tracklet plots generated from geo-localised tracks can be used to classify crowd activities.

6.4 Future Work

Additional testing with larger datasets needs to be carried out for the proposed crowd granularity assessment method. Nevertheless, this is a very time-consuming task, since comparison is performed against human-labelled data obtained from several subjects. Also, further exploration of estimators is needed, that is different ways to estimate how orderly the crowd is, to determine the estimators that come closer to the human labelling the most. Using the curve shown in Fig. 3.8 (p. 83), several methods for density and entropy can be compared to determine which one yields a higher number of correctly classified instances with the lowest error allowance.

Concerning the second contribution on telemetry-assisted methods for aerial video surveillance, testing with other visual trackers is left for future work, as well as addressing the re-identification of targets that left and re-enter the field of view of the camera. This could be done by keeping a database of identities linked to their appearance models (as last seen), as is done in the field of people re-identification. As stated in the chapter, the region covariance descriptor has been used to that end [24, 202]. Another possible future work could be to use the foreground mask as a detection algorithm to initialise the tracking of humans present in the scene, and to

employ the presented telemetry-corrected tracking for continued tracking.

On the third contribution using tracklet plots for event recognition in large groups, several things could be improved in the future: firstly, exploring other types of tracklet plot histograms, apart from the proposed polar, and also, rotating the plots so that the most prominent direction of motion is always on the *first bin*, so that scenes in which only the direction of motion changes are considered to be very similar. Secondly, using an alternative tracker that can deal better with occlusions. Furthermore, finding comparable datasets that are publicly available. As discussed, PETS or UMN cannot be used, because the vantage point is too low, and the actions performed do not conform to the classes established, respectively. Lastly, trying other dimensionality reduction techniques, such as Local Linear Embedding (LLE) or Isomap with a different configuration (using k-NN instead rather than a fixed neighbourhood radius).

Finally, as stated, one could have a fully integrated system, that employs outputs of a certain methodology as input for others. Two possible integrated systems have been shown, along with the changes that would be required in the presented contributions to carry out the integration.

6.5 Epilogue - Final Statement

To summarise, it has been demonstrated that: crowd granularity assessment via a density–entropy signature contributes with additional information to the decision-making process via an *orderliness* measure. That is, it provides information on the level of potential target inter-occlusions.

Additionally, telemetry-assisted aerial video surveillance methods can: first, improve tracking via search window correction by up to 50% on average; and second, outperform purely video-based techniques based on corner detection on poorly textured video sequences, and perform in similar terms otherwise, but with a much lower computational overhead.

Finally, tracklet plots, combined from multiple views, have been shown to be a useful scene descriptor for event detection in small crowds or large groups of people.

The combined system can harness the result from the best-performing camera with faster training rates.

Appendix A

Additional material

A.1 Introduction

This appendix presents additional materials for each contribution chapter in this thesis. Specifically it introduces figures that were excluded from the chapters initially, as well as video strips showing the range content of the video datasets used in contribution chapters (i.e. Chapters 3–5).

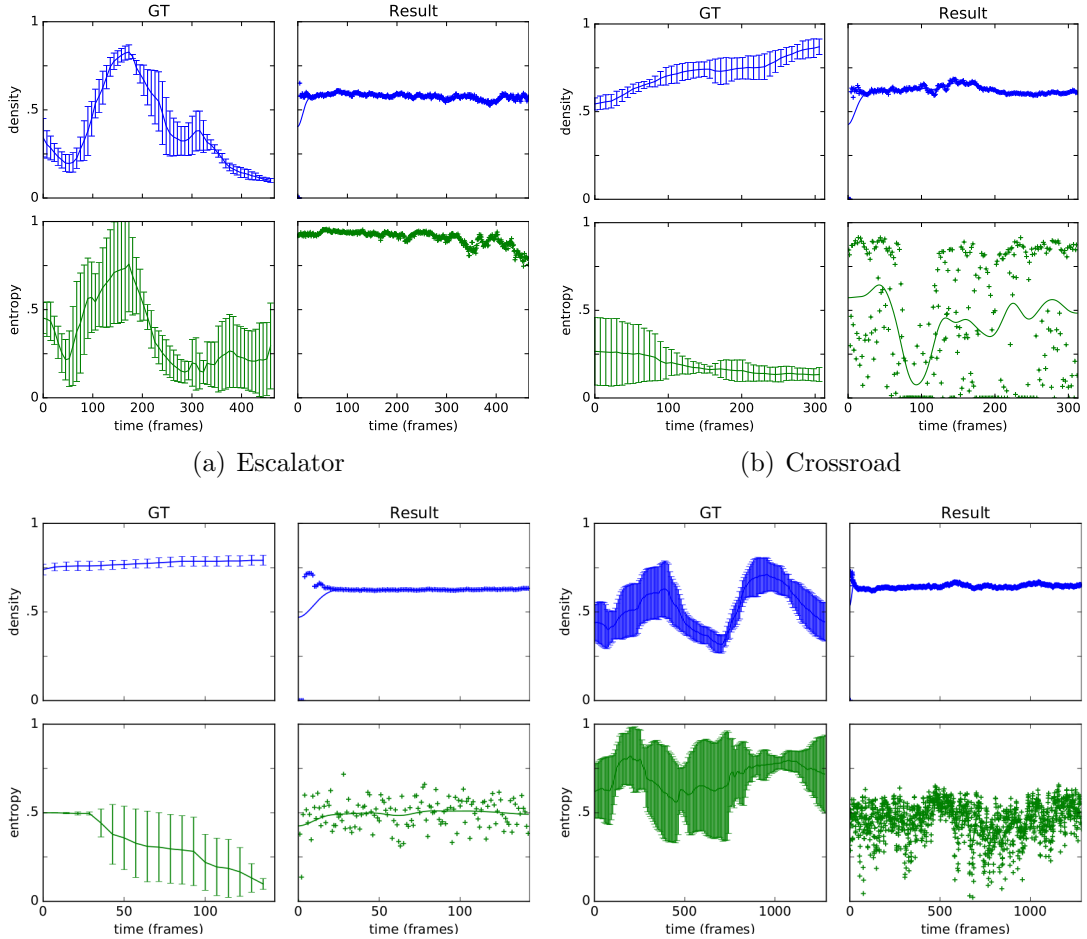
A.2 Additional materials to Chapter 3

A.2.1 Parameter selection (δ, L)

δ values	$L = 10$	$L = 20$	$L = 40$
5	**0.68**	**0.65**	**0.65**
10	0.12	0.13	0.24
15	0.14	0.14	0.14
20	0.14	0.12	0.12

Table A.1: Parameter selection based on average correct classification of a subset of sequences using various values on the proposed dataset

A.2.2 Other results



(a) Escalator (b) Crossroad

(c) Running (d) Street

Figure A.1: Analysis results for the remaining sequences each sub-figure shows the human-labelled ground truth average and standard deviations (left column) and estimations of the presented algorithm (right column) for density (top row, in blue) and entropy (bottom row, in green).

A.2.3 Example frames and ground truth

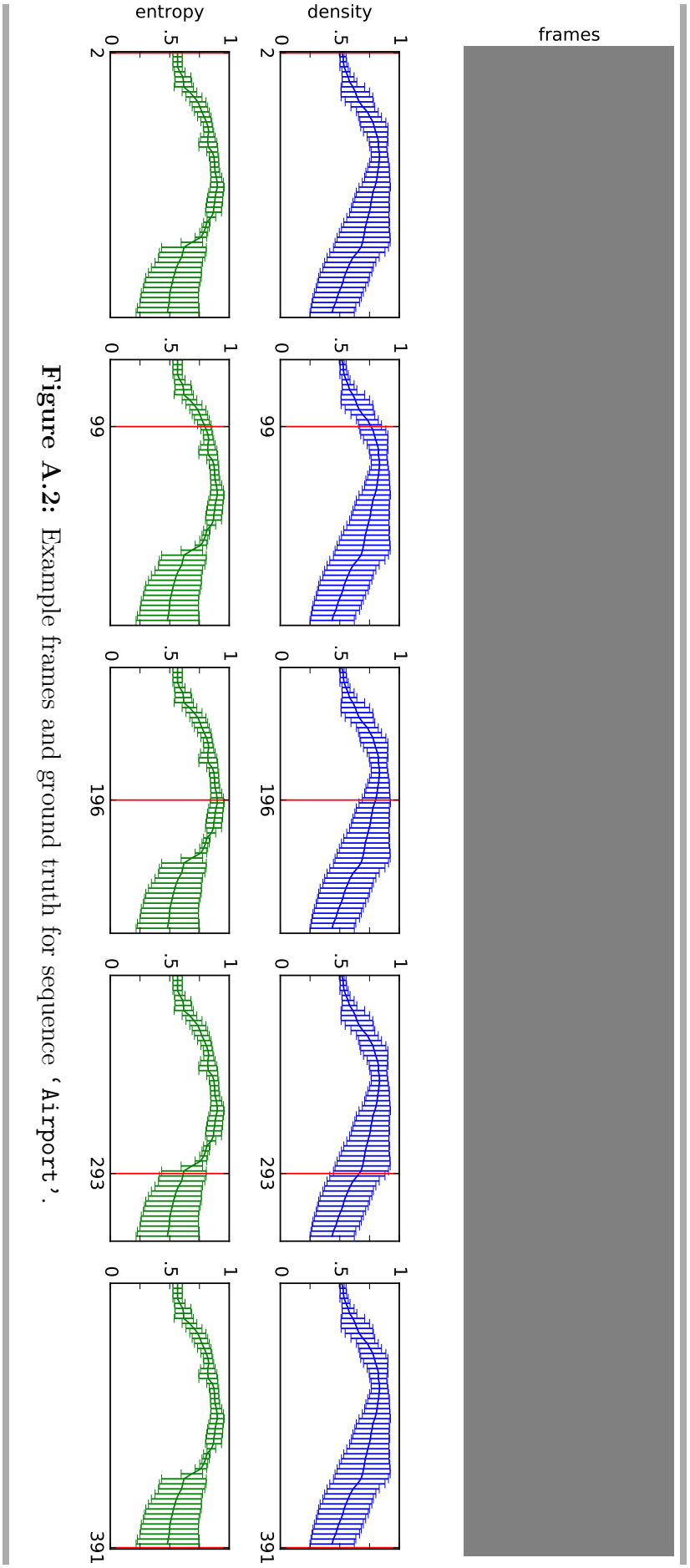


Figure A.2: Example frames and ground truth for sequence '`Airport`'.

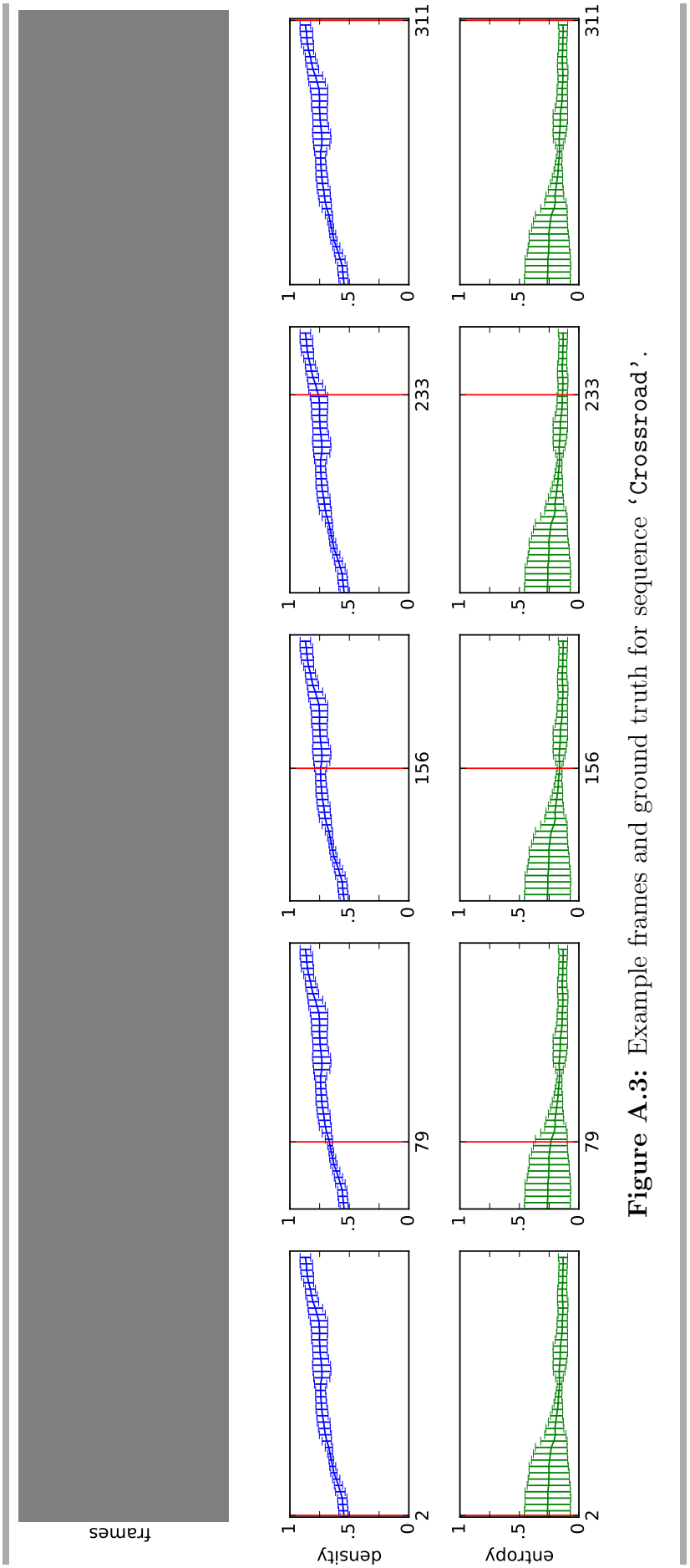


Figure A.3: Example frames and ground truth for sequence ‘`Crossroad`’.

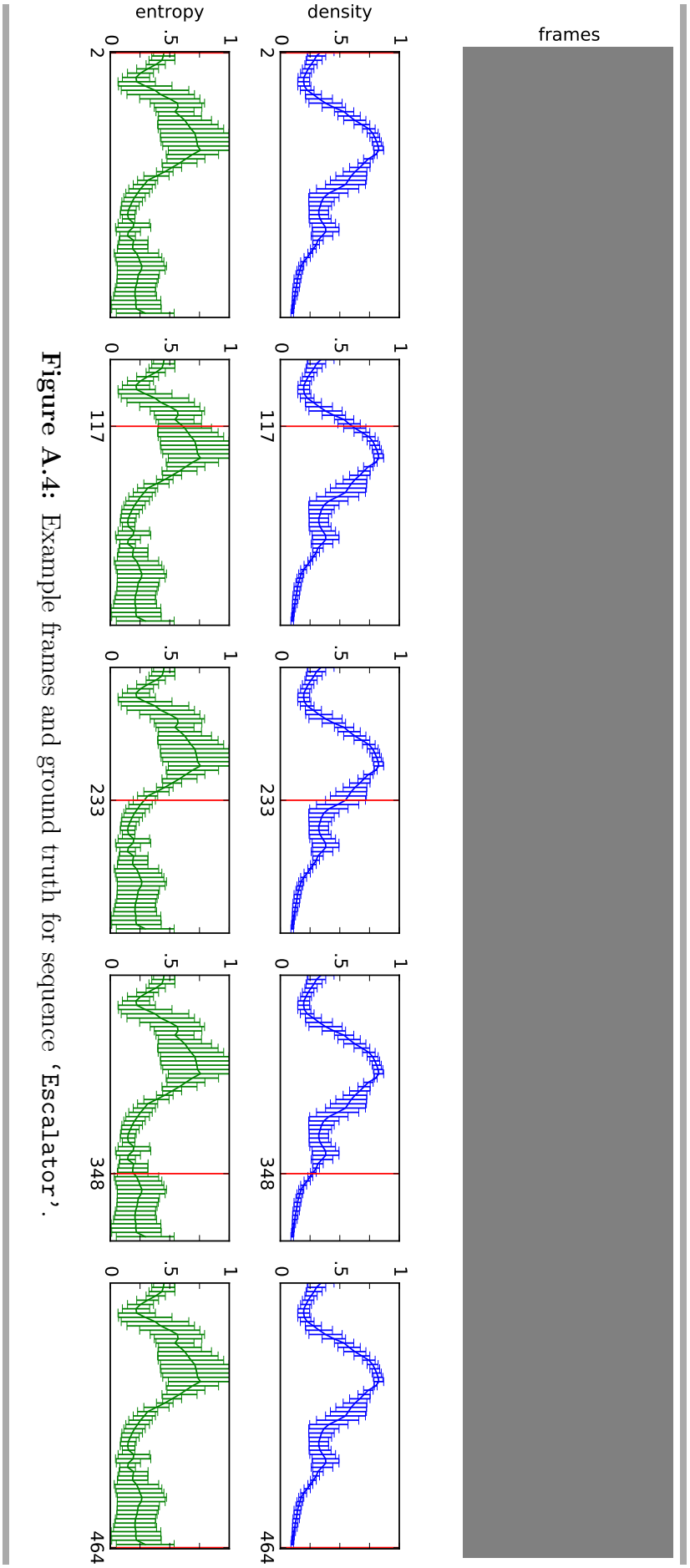


Figure A.4: Example frames and ground truth for sequence 'Escalator'.

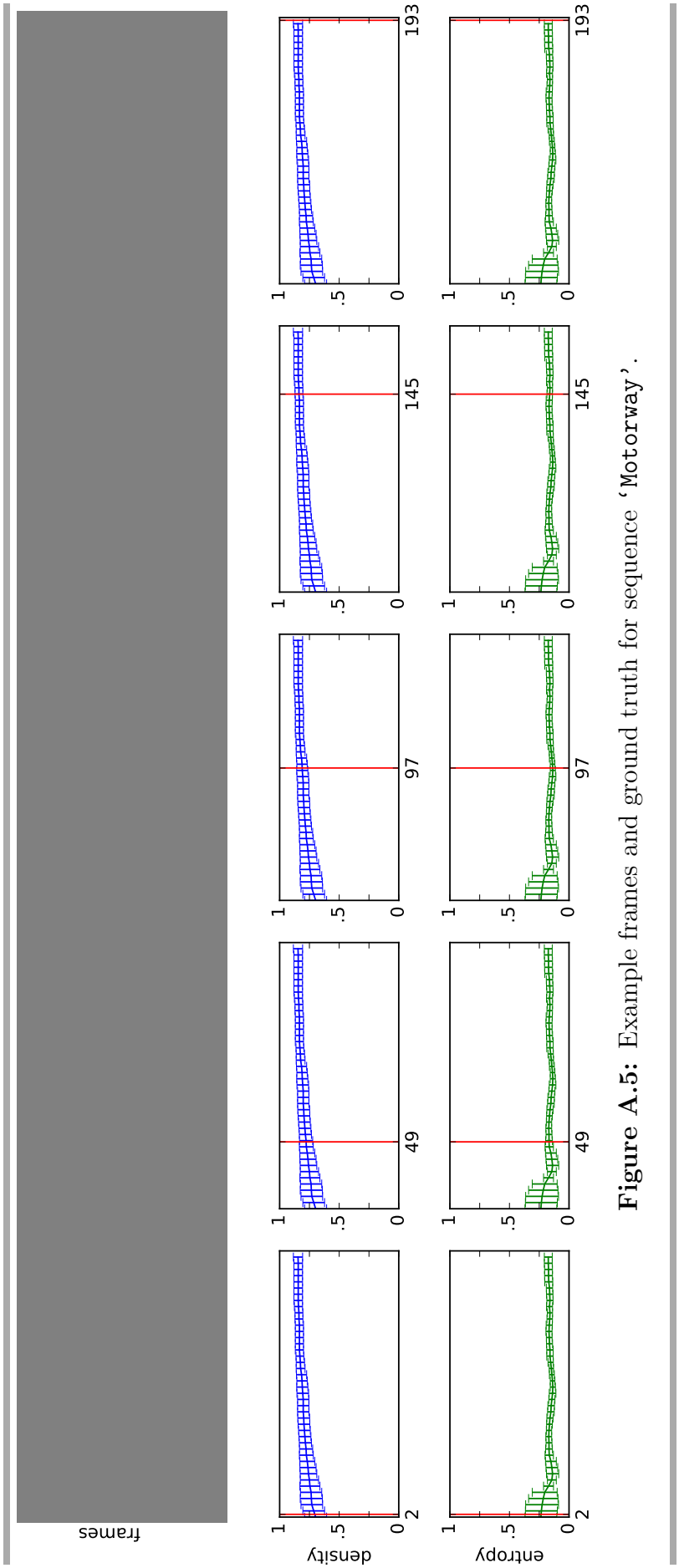


Figure A.5: Example frames and ground truth for sequence ‘`Motorway`’.

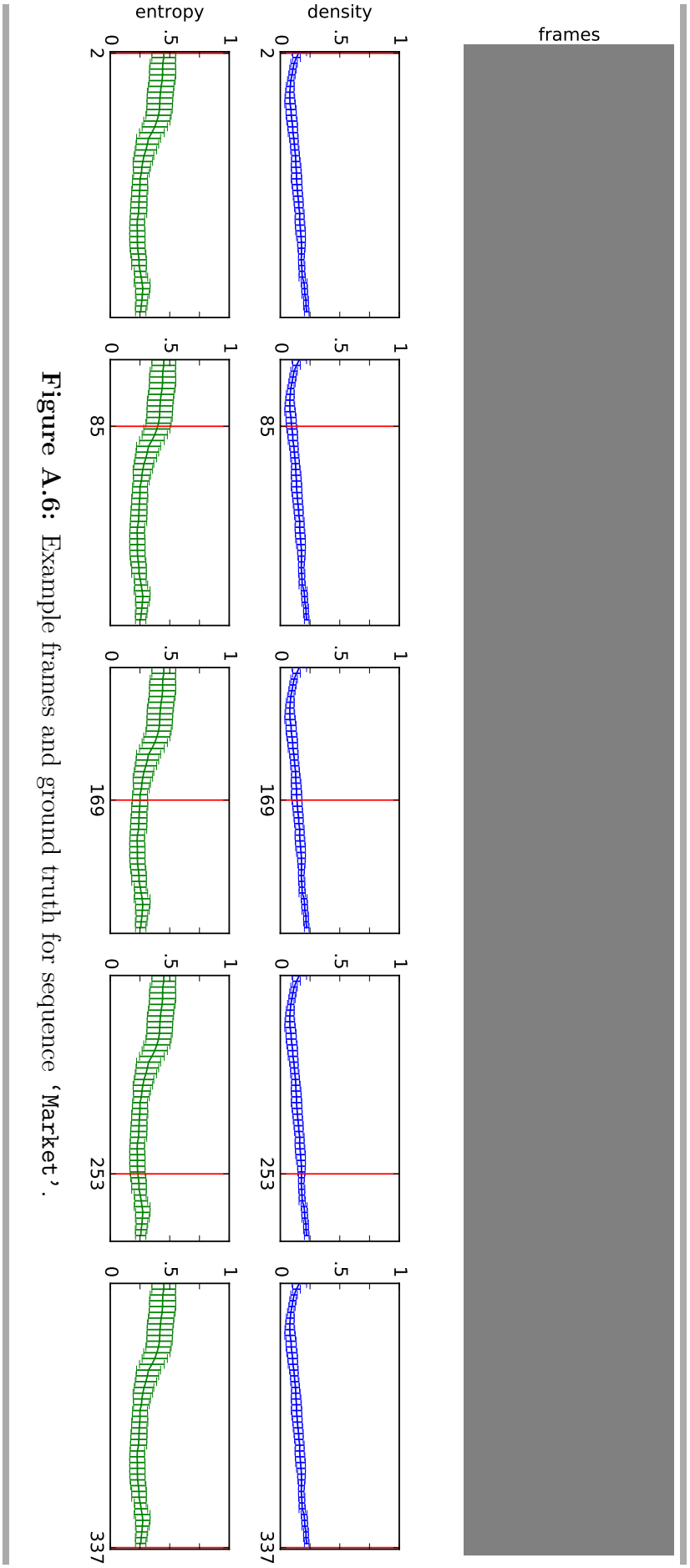


Figure A.6: Example frames and ground truth for sequence ‘`Market`’.

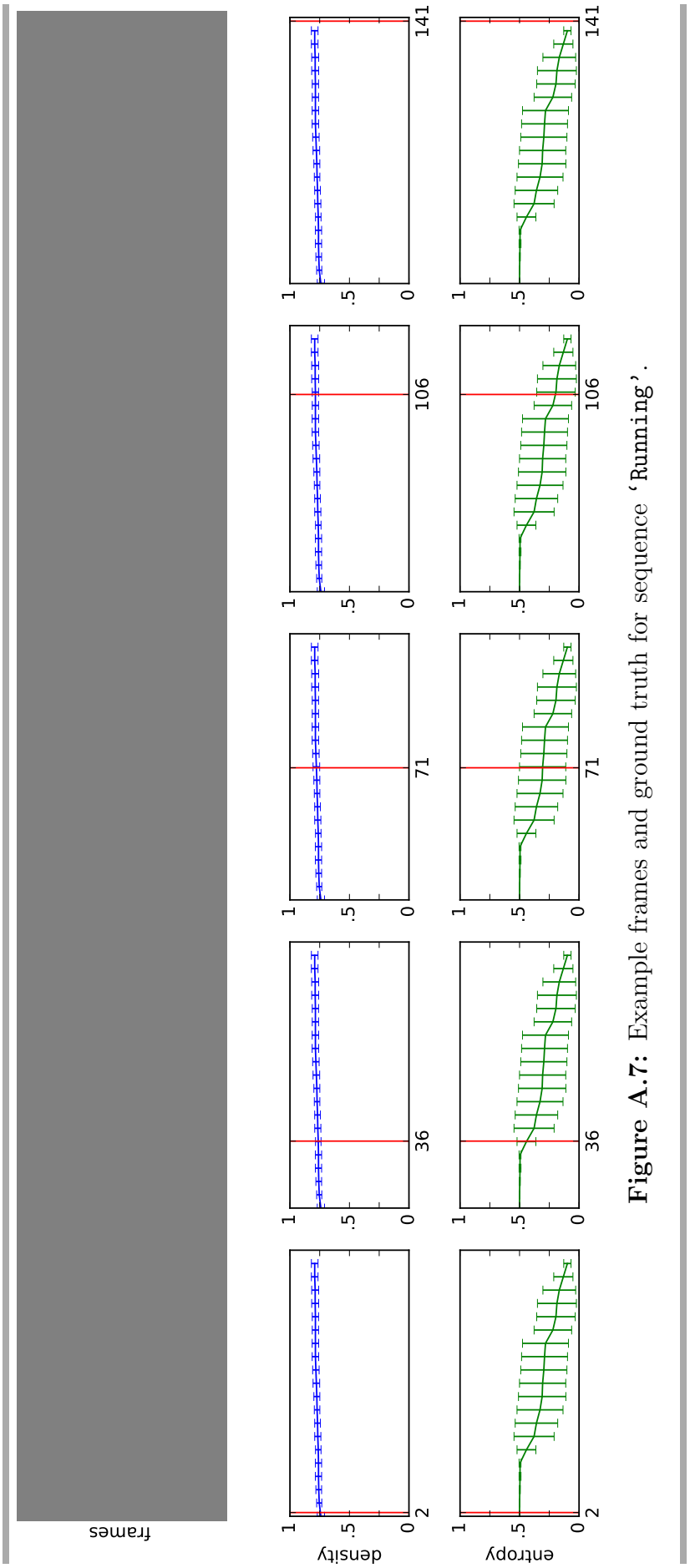


Figure A.7: Example frames and ground truth for sequence `'Running'`.

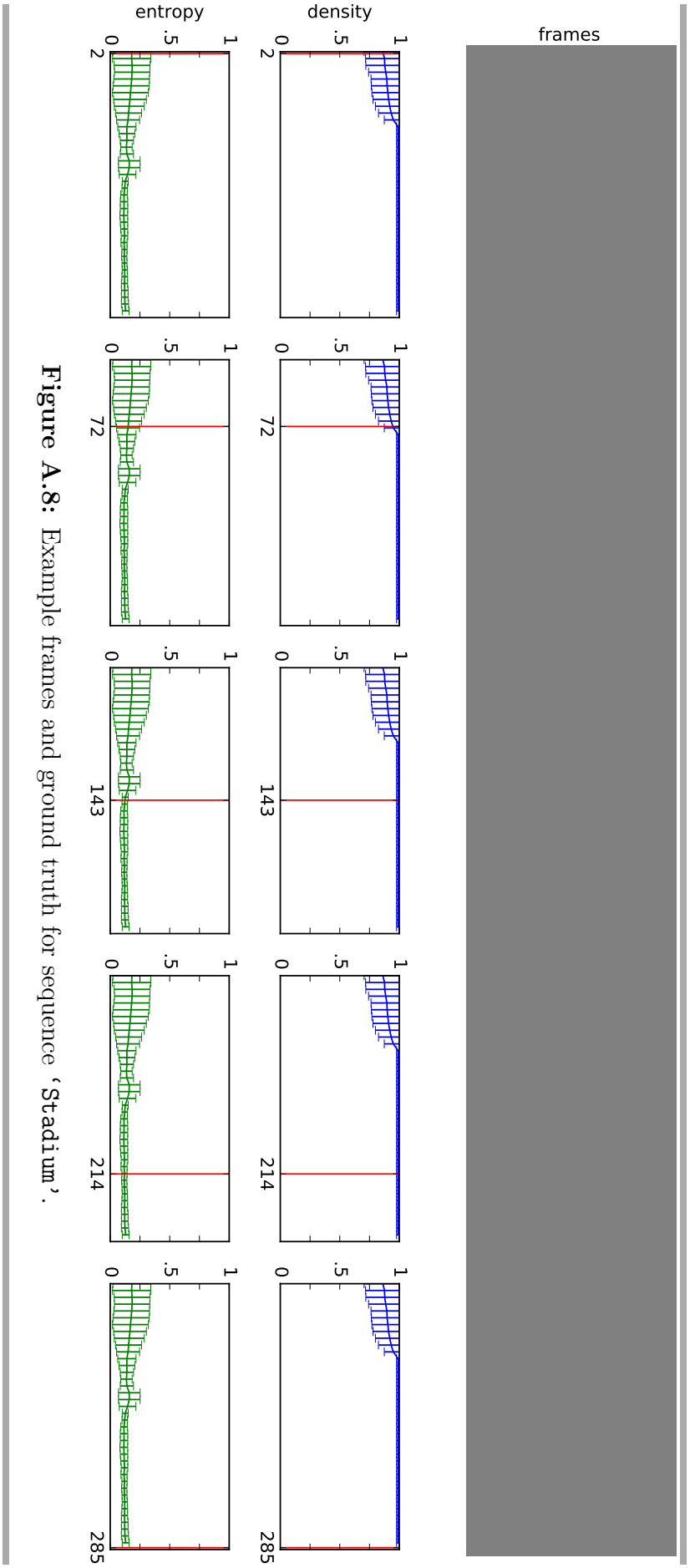


Figure A.8: Example frames and ground truth for sequence '`Stadium`'.

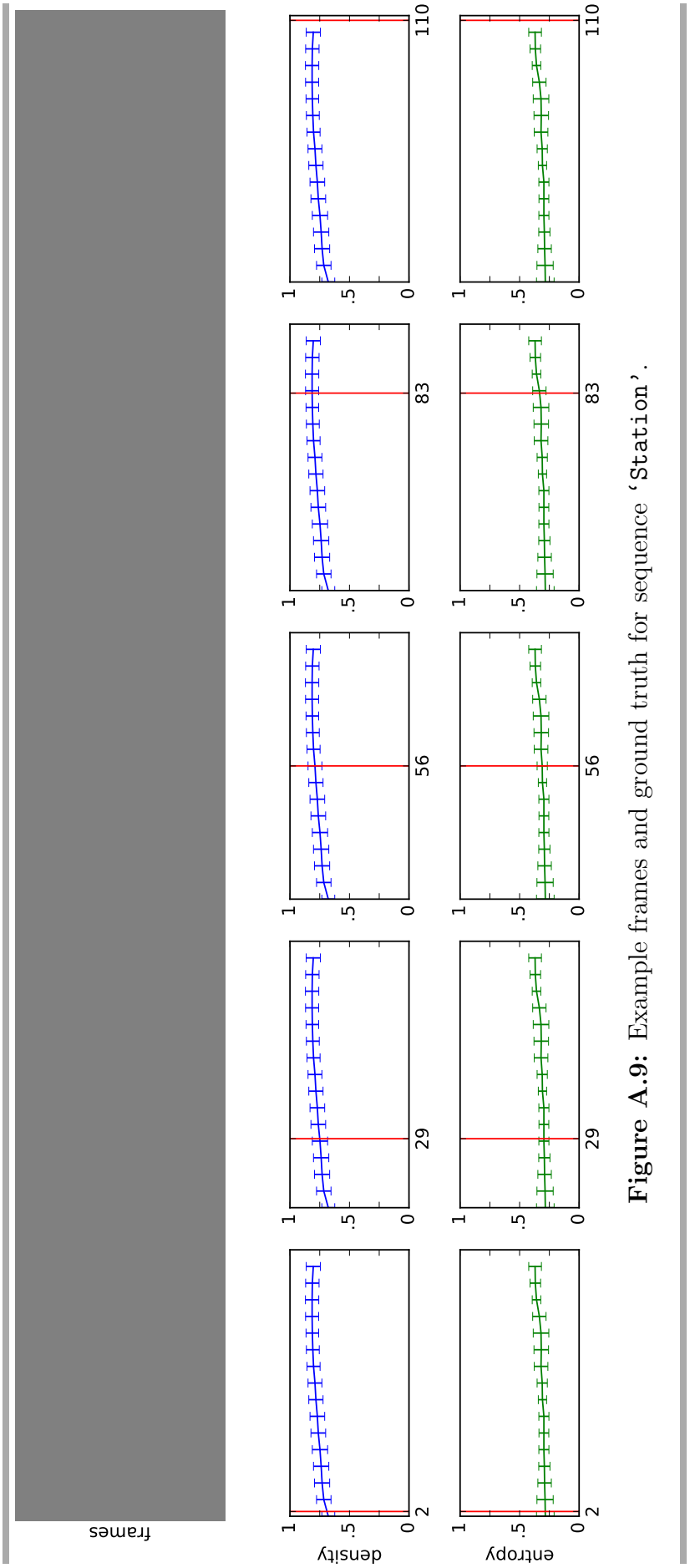


Figure A.9: Example frames and ground truth for sequence `'Station'`.

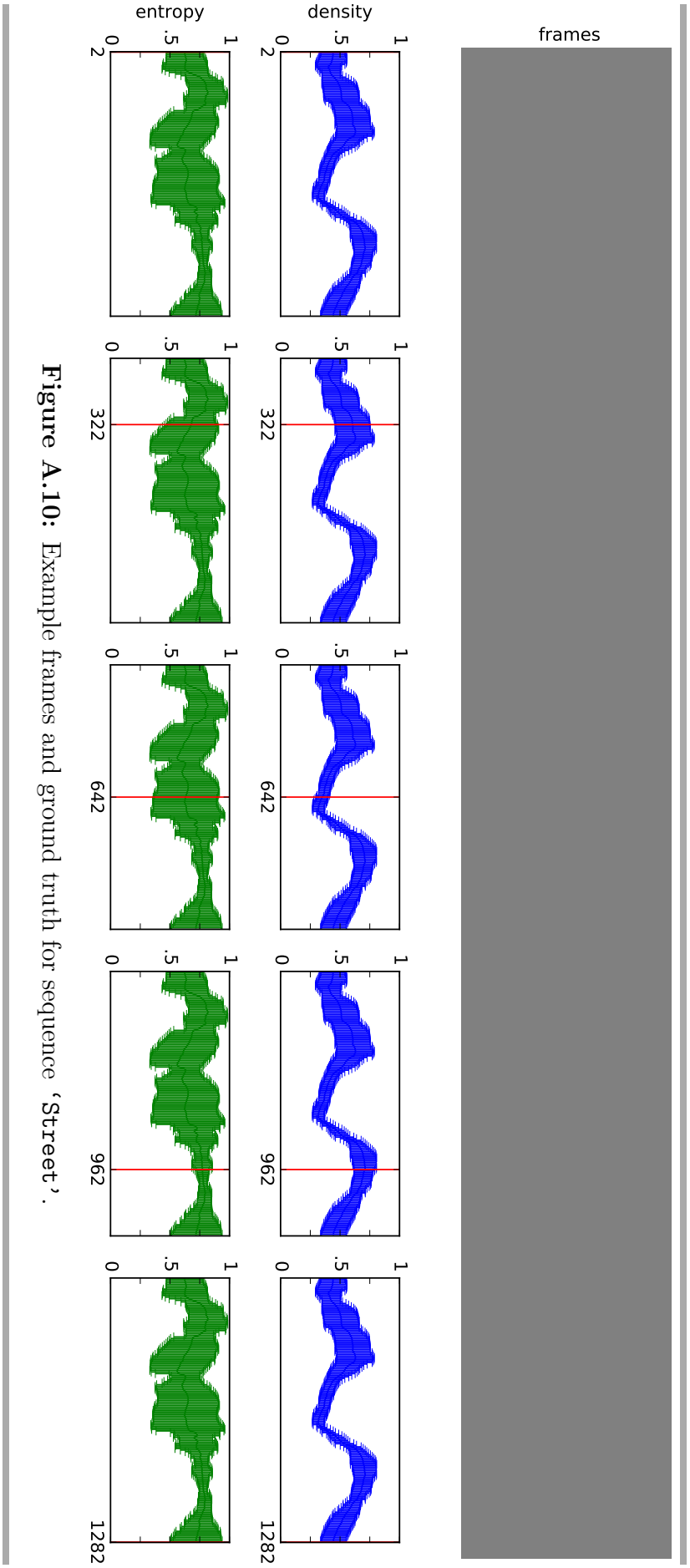


Figure A.10: Example frames and ground truth for sequence ‘`Street`’.

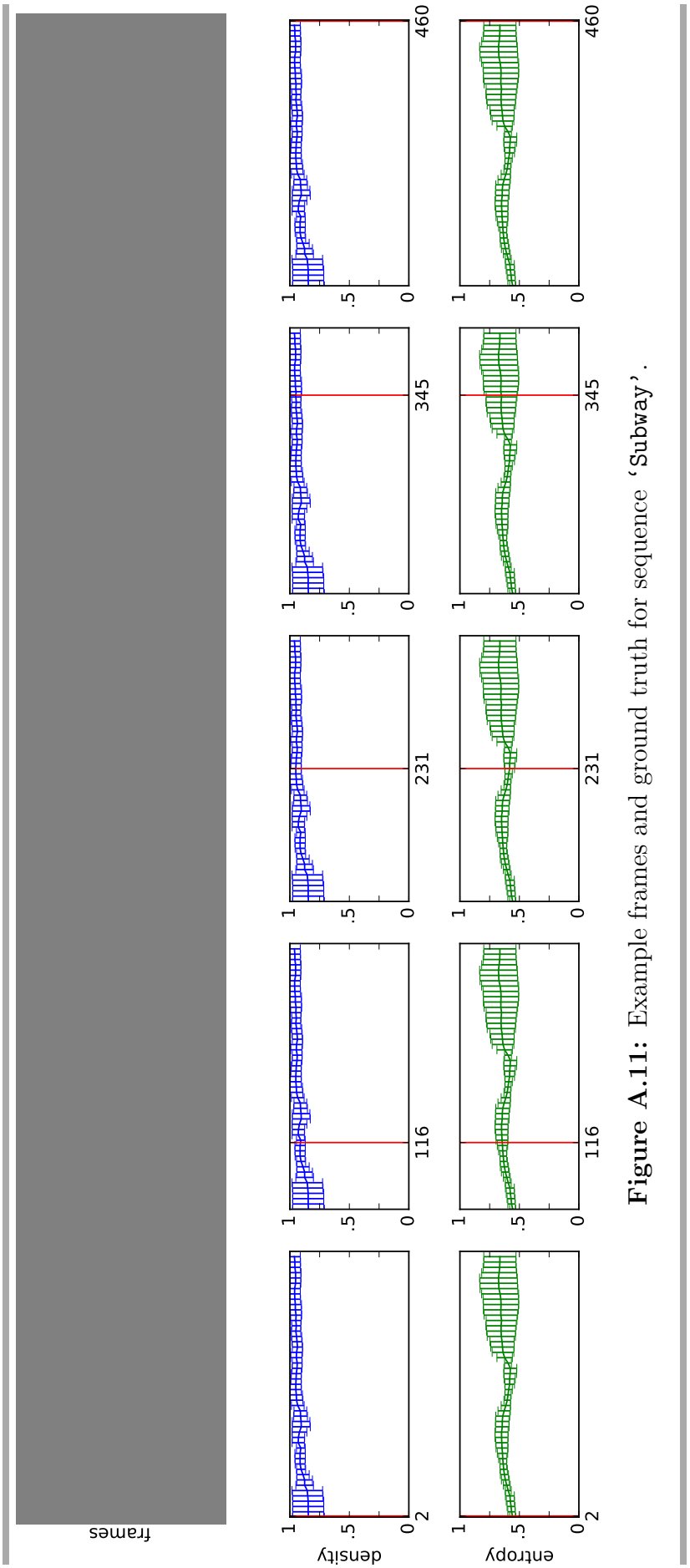


Figure A.11: Example frames and ground truth for sequence ‘Subway’.

A.3 Additional materials to Chapter 4

A.3.1 First dataset (for tracking window correction)

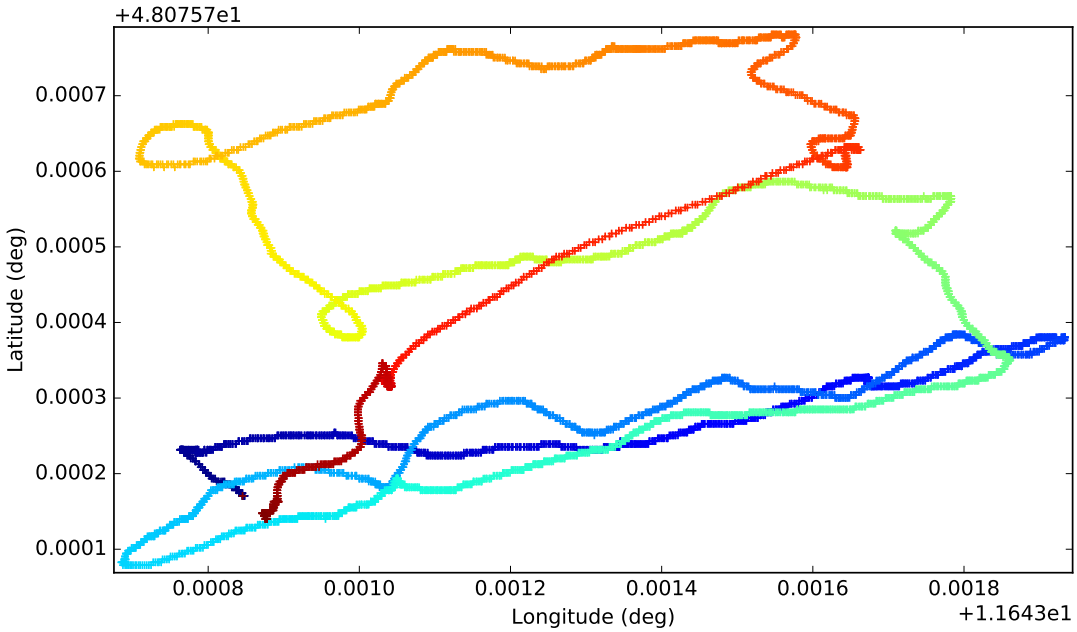


Figure A.12: Latitude and longitude of the aerial vehicle (first dataset). Warmer colours represent more recent vehicle positions.

A.3.2 Second dataset (for background subtraction)

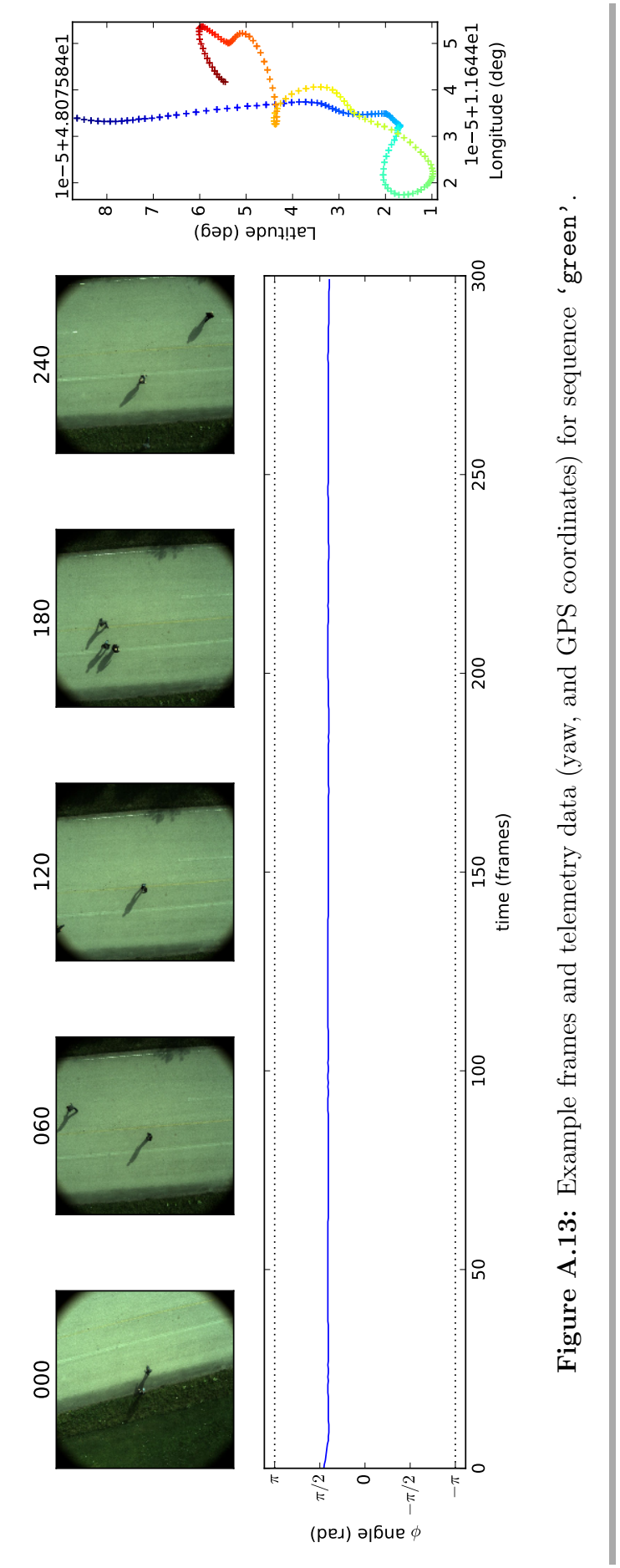


Figure A.13: Example frames and telemetry data (yaw, and GPS coordinates) for sequence `'green'`.

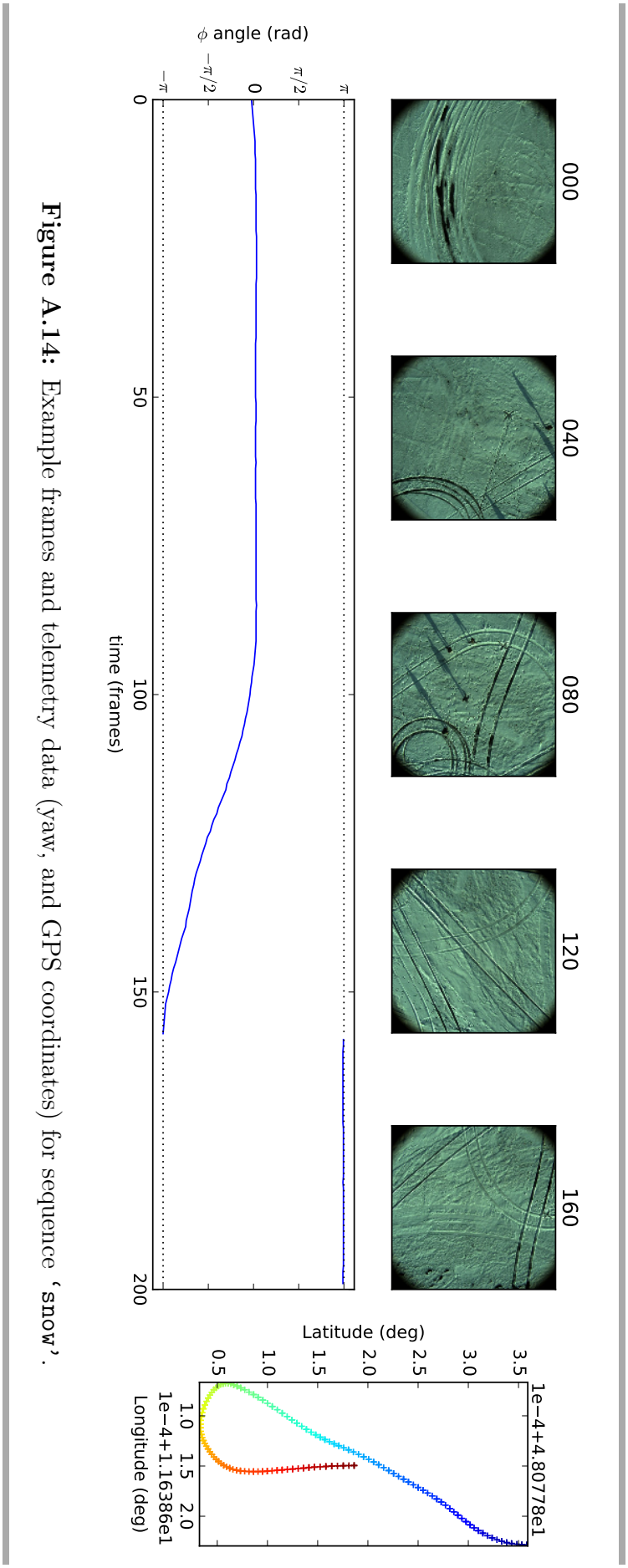


Figure A.14: Example frames and telemetry data (yaw, and GPS coordinates) for sequence '`snow`'.

A.4 Additional materials to Chapter 5

A.4.1 Synchronisation

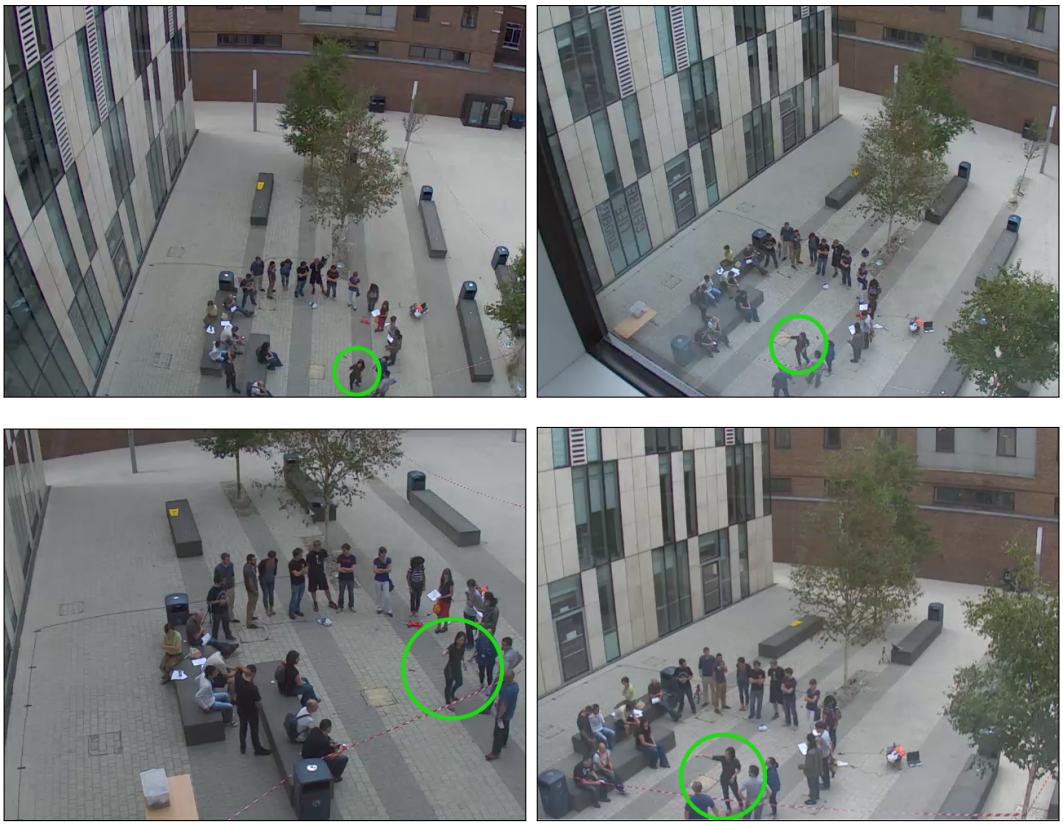


Figure A.15: Example frames from the Penrhyn Road campus dataset, showing the moment that was used to synchronise the video streams. The lady inside the green circle is lowering her arm. The cameras are synchronised at the instant in which her arm is in a straight angle to her torso in all views.

A.4.2 Example sequences

A.4.3 Experiment 3: Evaluation of the number of clusters

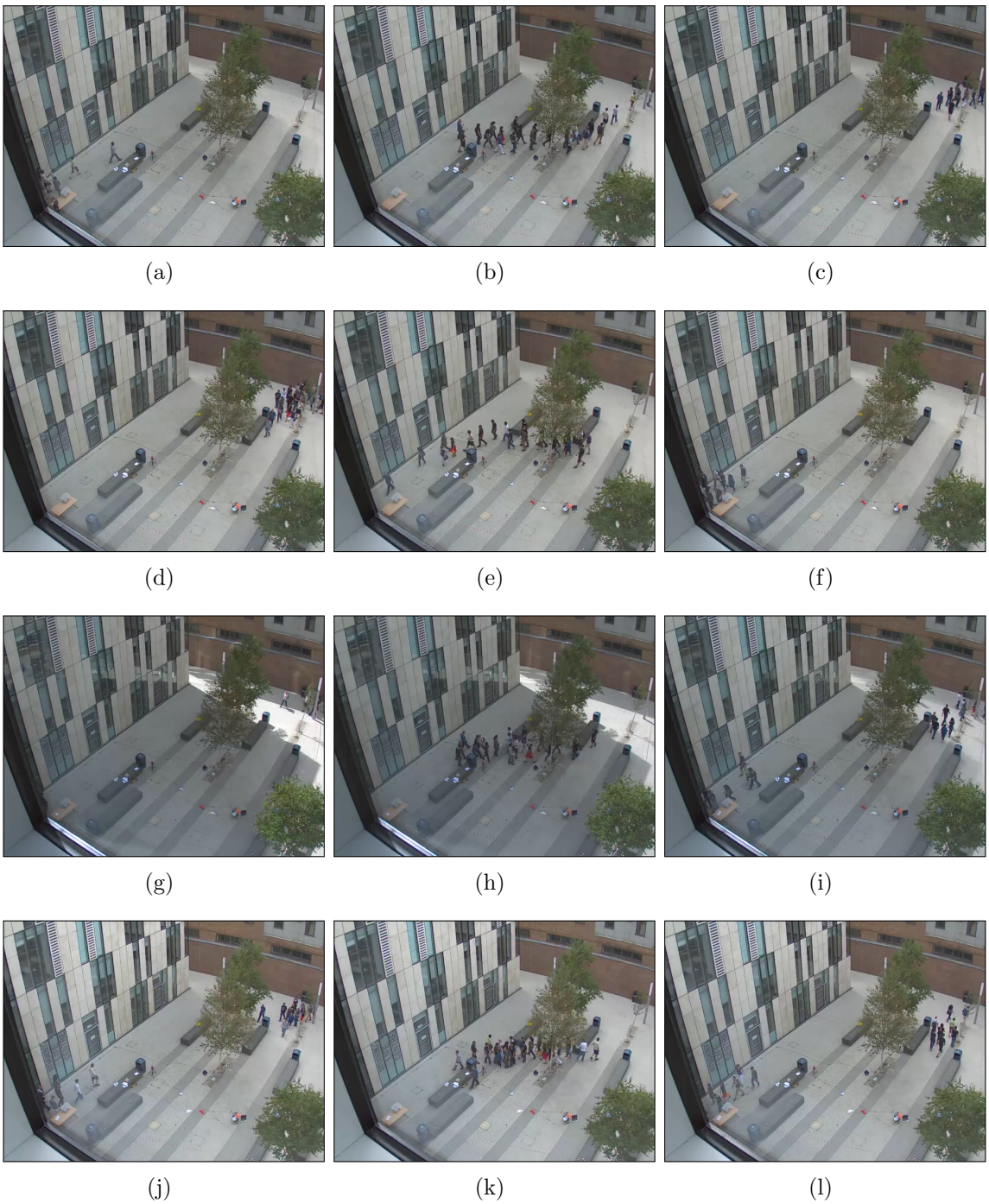


Figure A.16: Examples of four `normal` sequences (one per row, as seen from view 2, i.e. top-right). Top two rows "walk" sequences: (a–c) and (d–f). Bottom two rows "cross" sequences: (g–i) and (j–l).

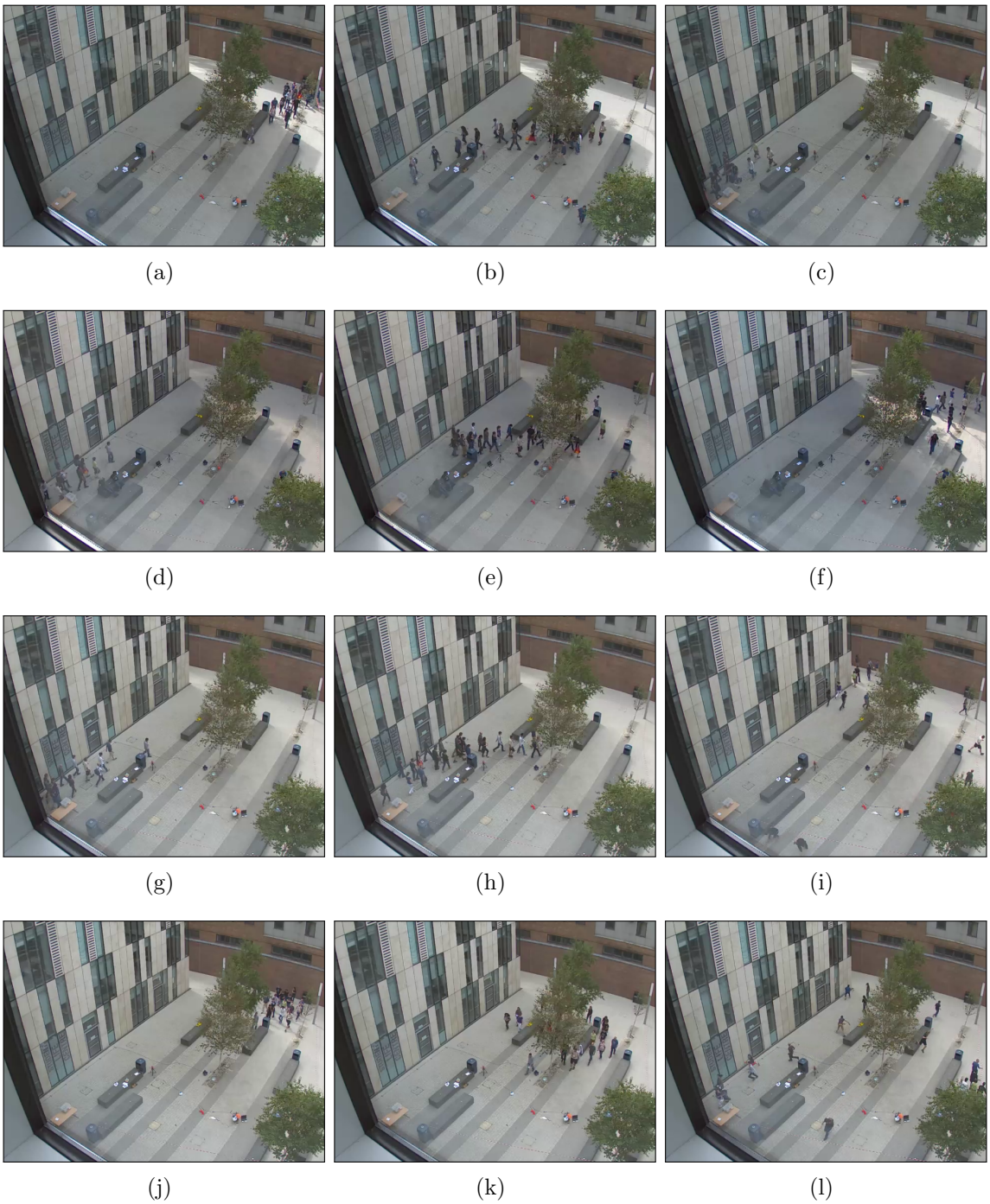


Figure A.17: Examples of two `abnormal` and two `chaotic` sequences (one per row, as seen from view 2, i.e. top-right). Top two rows "abnormal" sequences: (a–c) and (d–f). Please note in (b) and (f) some subjects are not following the rest of the group. Bottom two rows "chaotic" sequences: (g–i) and (j–l). Please observe in (i) and (l) people running away in all directions due to an acoustic signal of danger.

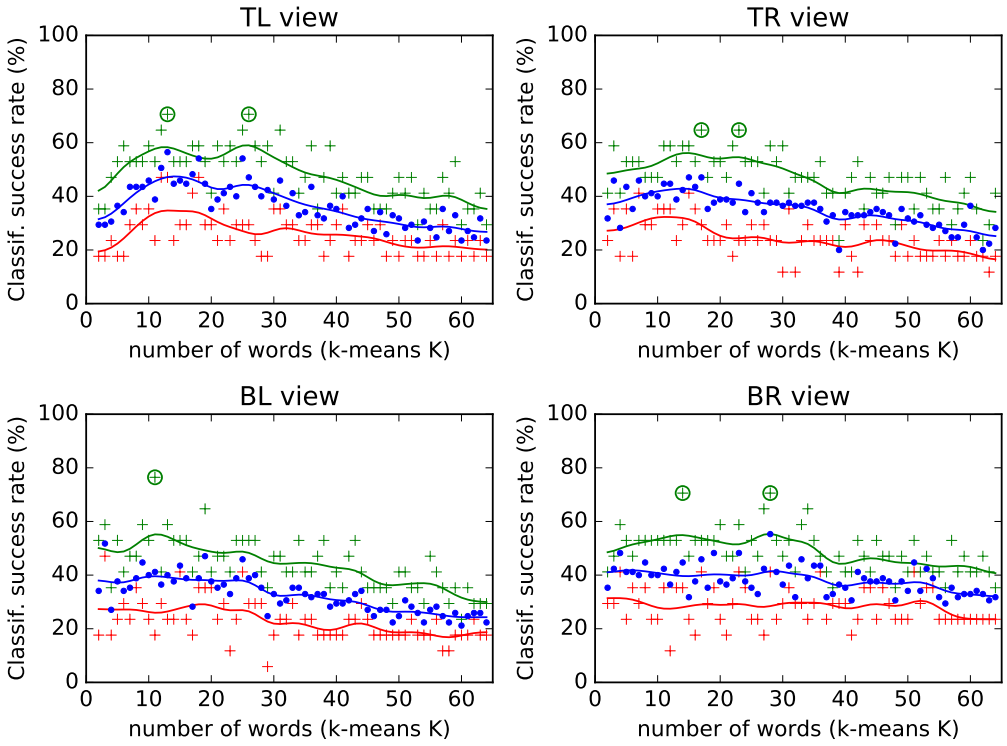


Figure A.18: Maximum, mean and minimum classification success rates for different number of key words ($K = 2, 3, \ldots, 64$) using **polar** TPHs without intensity binning, for each separate view (using Kalman-filtered tracks). Best runs appear circled.

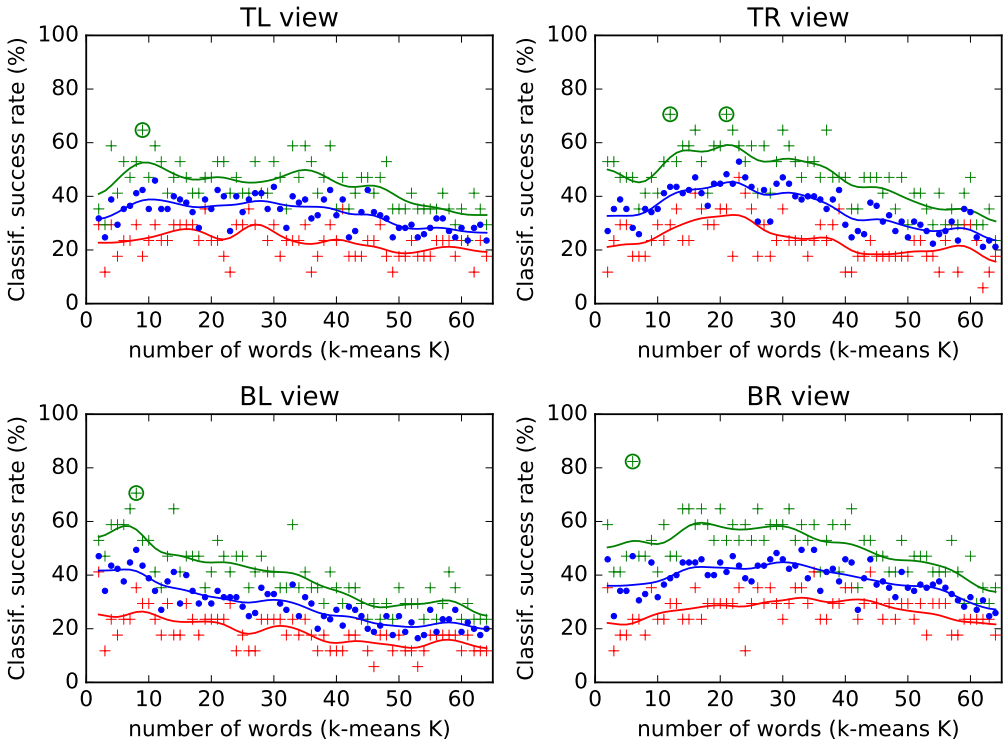
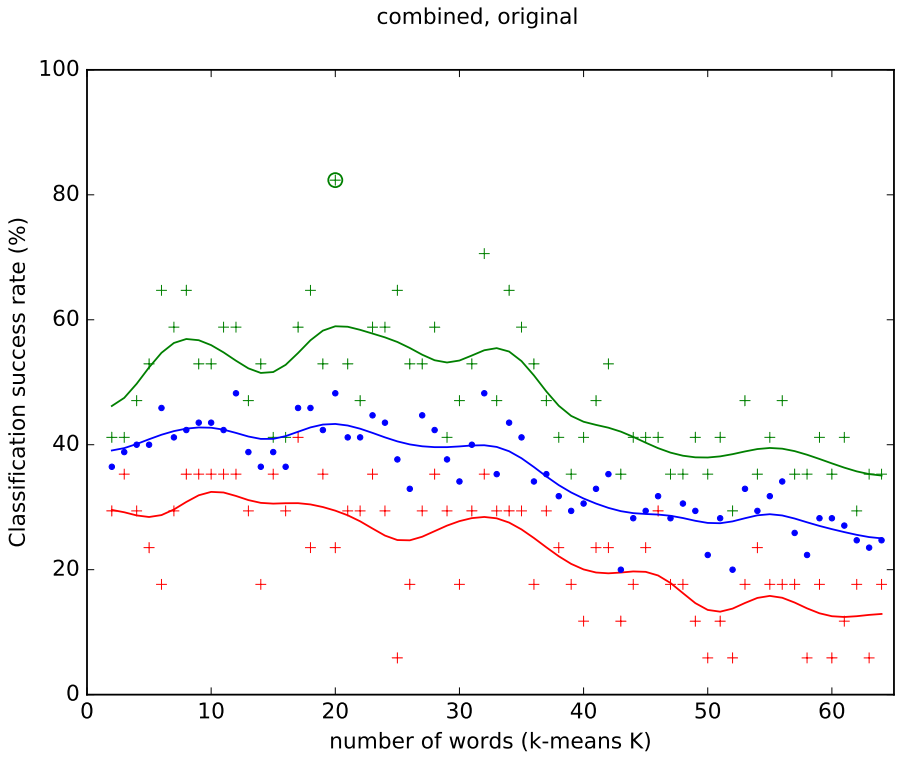
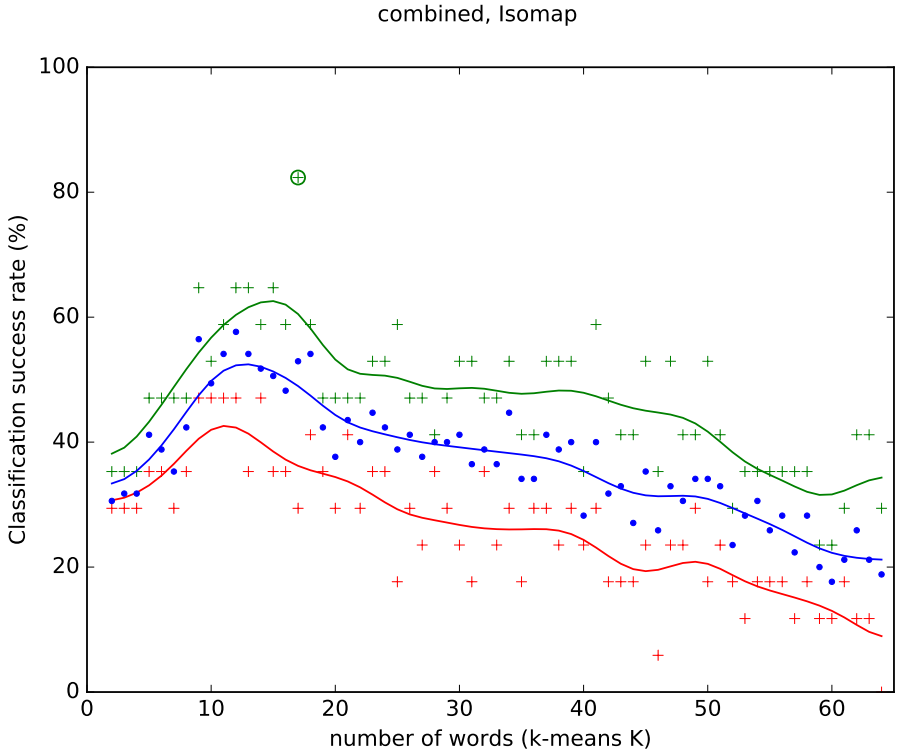


Figure A.19: Maximum, mean and minimum classification success rates for different number of key words ($K = 2, 3, \ldots, 64$) using **polar** TPHs without intensity binning, for each separate view (using **non-**Kalman-filtered tracks). Best runs appear circled.



(a)



(b)

Figure A.20: Maximum, mean and minimum classification success rates for different number of key words ($K = 2, 3, \ldots, 64$) using **polar** TPHs without intensity binning, for the combined case: (a) original, (b) reduced by Isomap. Best runs appear circled.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805, June 2006. 30, 31

[2] C. Aeschliman, J. Park, and A. Kak. Tracking Vehicles Through Shadows and Occlusions in Wide-Area Aerial Video. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(1):429–444, January 2014. 92, 93

[3] I. Ali and M. N. Dailey. Multiple Human Tracking in High-density Crowds. In *Advanced Concepts in Intelligent Vision Systems*, pages 540–549, 2009. 53, 54

[4] S. Ali and M. Shah. COCOA: tracking in aerial imagery. *Proc. SPIE*, 6209: 62090D–62090D–6, 2006. 92, 93

[5] S. Ali and M. Shah. A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6, Minneapolis, Florida, June 19-21 2007. IEEE. 16, 74

[6] S. Ali and M. Shah. Floor Fields for Tracking in High Density Crowd Scenes. In *Proceedings of European Conference on Computer Vision*, pages 1–14, Marseille, France, 2008. Springer. 16

[7] J. G. Allen, R. Y. D. Xu, and J. S. Jin. Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces. In *Proceedings of the Pan-*

Sydney Area Workshop on Visual Information Processing, VIP '05, pages 3–7, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc. 23

[8] V. Alvarez-Santos, X. Pardo, R. Iglesias, A. Canedo-Rodriguez, and C. Regueiro. Feature analysis for human recognition and discrimination: Application to a person-following behaviour in a mobile robot . *Robotics and Autonomous Systems*, 60(8):1021 – 1036, 2012. 12

[9] E. Andrade and R. Fisher. Simulation of Crowd Problems for Computer Vision. In *Proceedings of 19th International Conference on Pattern Recognition*, volume 3, pages 71–80, November 2005. 16

[10] E. Andrade, R. Fisher, and S. Blunsden. Modelling Crowd Scenes for Event Detection. In *Proceedings of 19th International Conference on Pattern Recognition*, volume 1, pages 175–178, September 2006. 16

[11] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 21

[12] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, S. Roth, and B. Schiele. Vision based victim detection from unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1740–1747, Oct 2010. 89, 91

[13] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani. Methods of analyzing traffic imagery collected from aerial platforms. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):99–107, June 2003. 93

[14] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. 138

[15] N. Artner. A comparison of mean shift tracking methods. In *12th Central European Seminar on Computer Graphics*, 2008. 23

[16] M. S. Arulampalam, S. M. N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. 24

[17] I. Atmosukarto, B. Ghanem, and N. Ahuja. Trajectory-based Fisher kernel representation for action recognition in videos. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3333–3336, Nov 2012. 123

[18] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010. 12, 56

[19] B. Auffarth, M. López, and J. Cerquides. Comparison of Redundancy and Relevance Measures for Feature Selection in Tissue Classification of CT Images. In *ICDM*, pages 248–262. Springer, 2010. 72

[20] S. Avidan. Ensemble Tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, Feb 2007. 27, 32, 46, 47, 49, 50

[21] M. Azahar, M. Sunar, D. Daman, and A. Bade. Survey on Real-Time Crowds Simulation. In Z. Pan, X. Zhang, A. El Rhalibi, W. Woo, and Y. Li, editors, *Technologies for E-Learning and Digital Entertainment*, volume 5093 of *Lecture Notes in Computer Science*, pages 573–580. Springer Berlin Heidelberg, 2008. 12

[22] B. Babenko, M.-H. Yang, and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, Aug 2011. 22, 30, 36, 41, 46, 47, 48, 49, 50

[23] K. Bai. Particle filter tracking with Mean Shift and joint probability data association. In *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pages 607–612. IEEE, 2010. 24

[24] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Person Re-identification Using Spatial Covariance Regions of Human Body Parts. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 435–440, Aug 2010. 98, 165

[25] S. Bakas. Computer-Aided Localisation, Segmentation and Quantification of Focal Liver Lesions in Contrast-Enhanced Ultrasound, 2014. xi

[26] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra. Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications*, 51(1):279–302, 2011. 123, 134

[27] D. Barber, J. Redding, T. McLain, R. Beard, and C. Taylor. Vision-based Target Geo-location using a Fixed-wing Miniature Air Vehicle. *Journal of Intelligent and Robotic Systems*, 47(4):361–382, 2006. 60, 91

[28] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss. The Autonomous City Explorer: Towards Natural Human-Robot Interaction in Urban Environments. *International Journal of Social Robotics*, 1(2):127–140, 2009. 13

[29] L. Bazzani, M. Cristani, and V. Murino. Decentralized particle filter for joint individual-group tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1886–1893, June 2012. 24

[30] S. Bhattacharya, H. Idrees, I. Saleemi, S. Ali, and M. Shah. Moving Object Detection and Tracking in Forward Looking Infra-Red Aerial Imagery. In *Machine Vision Beyond Visible Spectrum*, volume 1 of *Augmented Vision and Reality*, pages 221–252. Springer Berlin Heidelberg, 2011. 92, 93, 113

[31] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001. 127

[32] F. Boehm and A. Schulte. Air to ground sensor data distribution using IEEE802.11N Wi-Fi network. In *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, pages 4B2–1–4B2–10, Oct 2013. 105

[33] B. Boghossian and S. Velastin. Motion-based machine vision techniques for the management of large crowds. In *Electronics, Circuits and Systems, 1999.*

Proceedings of ICECS '99. The 6th IEEE International Conference on, volume 2, pages 961–964 vol.2, Sep 1999. 12

[34] M. Boltes and A. Seyfried. Collecting pedestrian trajectories. *Neurocomputing*, 100(0):127–133, 2013. 56

[35] H. Bouma, S. Borsboom, R. J. M. den Hollander, S. H. Landsmeer, and M. Worring. Re-identification of persons in multi-camera surveillance under varying viewpoints and illumination. In *Proc. SPIE*, volume 8359, pages 83590Q–83590Q–10, 2012. 55

[36] O. Brdiczka, J. Maisonnasse, P. Reignier, and J. Crowley. Detecting small group activities from multimodal observations. *Applied Intelligence*, 30(1):47–57, 2009. 15

[37] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2860–2865, May 2006. 91

[38] F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Unmanned Aerial Vehicle Localization Based on Monocular Vision and Online Mosaicking. *Journal of Intelligent and Robotic Systems*, 55(4-5):323–343, January 2009. 59, 91

[39] Y. Cai, N. de Freitas, and J. J. Little. Robust Visual Tracking for Multiple Targets. In *Proceedings of Eighth European Conference on Computer Vision*, volume 3954, pages 107–118. IEEE, 2006. 53

[40] J. Candamo, M. Shreve, D. Goldgof, D. Sapper, and R. Kasturi. Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *Intelligent Transportation Systems, IEEE Transactions on*, 11(1):206–224, 2010. 12, 121

[41] L. Cehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 540–547, March 2014. 44

[42] S.-H. Cha. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007. 25, 134

[43] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta. A review on vision techniques applied to Human Behaviour Analysis for Ambient-Assisted Living. *Expert Systems with Applications*, 39(12):10873 – 10888, 2012. 56

[44] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta. An efficient approach for multi-view human action recognition based on bag-of-key-poses. In *Human Behavior Understanding*, pages 29–40. Springer Berlin Heidelberg, 2012. 125

[45] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011. 68

[46] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations Newsletter*, 6(1):1–6, June 2004. 149

[47] A.-h. Chen, B.-q. Yang, and Z.-g. Chen. A Timely Occlusion Detection Based on Mean Shift Algorithm. In W. Deng, editor, *Future Control and Automation*, volume 173 of *Lecture Notes in Electrical Engineering*, pages 51–56. Springer Berlin Heidelberg, 2012. 52

[48] S. Chen. Kalman Filter for Robot Vision: A Survey. *Industrial Electronics, IEEE Transactions on*, 59(11):4409–4420, Nov 2012. 23

[49] S. Cheraghi and U. Sheikh. Moving object detection using image registration for a moving camera platform. In *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, pages 355–359, Nov 2012. 93

[50] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Efficient similarity search for covariance matrices via the Jensen-Bregman LogDet Divergence.

In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2399–2406, Nov 2011. 25

[51] W. Choi and S. Savarese. A Unified Framework for Multi-target Tracking and Collective Activity Recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, volume 7575 of *Lecture Notes in Computer Science*, pages 215–230. Springer Berlin Heidelberg, 2012. 54

[52] H. Chu, S. Ye, Q. Guo, and X. Liu. Object Tracking Algorithm Based on Camshift Algorithm Combinating with Difference in Frame. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 51–55, Aug 2007. 23

[53] R. Cilla, M. A. Patricio, A. Berlanga, and J. M. Molina. A probabilistic, discriminative and distributed system for the recognition of human actions from multiple views . *Neurocomputing*, 75(1):78–87, 2012. 126

[54] P. Climent-Pérez, G. Lazaridis, G. Hummel, M. Russ, D. N. Monekosso, and P. Remagnino. Telemetry-based search window correction for airborne tracking. In *International Symposium on Visual Computing*, pages 457–466, 2014. 89

[55] P. Climent-Pérez, A. Mauduit, D. N. Monekosso, and P. Remagnino. Detecting events in crowded scenes using tracklet plots. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, volume 2, pages 174–181, 2014. 120, 136, 137

[56] P. Climent-Pérez, D. N. Monekosso, and P. Remagnino. Multi-view event detection in crowded scenes using tracklet plots. In *22nd International Conference on Pattern Recognition*, pages 4370–4375, 2014. 120

[57] I. Cohen and G. Medioni. Detecting and Tracking Moving Objects in Video from an Airborne Observer. In *Image Understanding Workshop*, pages 217–222, 1998. 92

[58] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27 (10):1631–1643, Oct 2005. 22, 27, 38, 46, 47, 49, 50, 51

[59] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24 (5):603–619, May 2002. 23

[60] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456, June 2011. 136, 154

[61] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005. 66, 87

[62] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*, pages 1–11. BMVA Press, 2014. 37

[63] A. Davies, J. Yin, and S. Velastin. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, 7(1):34–47, 1995. 12, 65, 121

[64] H. M. Dee and A. Caplier. Crowd behaviour analysis using histograms of motion direction. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1545–1548. IEEE, 2010. 124

[65] A. Dehghan, H. Idrees, A. Zamir, and M. Shah. Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities. In U. Weidmann, U. Kirsch, and M. Schreckenberg, editors, *Pedestrian and Evacuation Dynamics 2012*, pages 3–19. Springer, 2014. 14, 64, 68

[66] V. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo. Vision-based tracking

and motion estimation for moving targets using small UAVs. In *American Control Conference, 2006*, pages 1428–1433, June 2006. 60, 91

[67] P. Drews, J. Quintas, J. Dias, M. Andersson, J. Nygårds, and J. Rydell. Crowd behavior analysis under cameras network fusion using probabilistic methods. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, July 2010. 12, 14, 55

[68] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 14

[69] R. Eshel and Y. Moses. Tracking in a Dense Crowd Using Multiple Cameras. *International Journal of Computer Vision*, 88(1):129–143, 2010. 12, 55

[70] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada. Normalized mutual information feature selection. *Neural Networks, IEEE Transactions on*, 20(2): 189–201, 2009. 72

[71] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 42, 108, 112

[72] G. Farnebäck. Two-Frame Motion Estimation Based on Polynomial Expansion. In J. Bigun and T. Gustavsson, editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg, 2003. 73, 83

[73] R. Farrell and L. Davis. Decentralized discovery of camera network topology. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–10, Sept 2008. 12, 55

[74] J. Ferryman and A. Ellis. PETS2010: Dataset and Challenge. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 143–150, Aug 2010. 12, 136, 154

[75] H. Fradi and J.-L. Dugelay. Towards crowd density-aware video surveillance applications . *Information Fusion*, 24:3 – 15, 2015. 64, 65, 66, 68, 70

[76] H. Fu and H. Ma. Real-time Crowd Detection Based on Gradient Magnitude Entropy Model. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 885–888, New York, NY, USA, 2014. ACM. 66, 67, 68

[77] C. Gárate, P. Bilinsky, and F. Bremond. Crowd event recognition using HOG tracker. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–6, 2009. 123, 124

[78] W. Ge, R. Collins, and B. Ruback. Automatically detecting the small group structure of a crowd. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–8, Dec 2009. 14, 15

[79] W. Ge, R. Collins, and R. Ruback. Vision-Based Analysis of Small Groups in Pedestrian Crowds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):1003–1016, May 2012. 15

[80] D. Ghazi, D. Inkpen, and S. Szpakowicz. Prior and contextual emotion of words in sentential context. *Computer Speech & Language*, 28(1):76 – 92, 2014. 150

[81] A. Gilbert and R. Bowden. Multi Person Tracking within Crowded Scenes. In *Proceedings of Workshop on Human Motion*, pages 166–179, 2007. 53

[82] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised On-Line Boosting for Robust Tracking. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, pages 234–247. Springer Berlin Heidelberg, 2008. 22, 29, 30, 32, 34, 46, 47, 49, 50

[83] S. Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11(1):23–63, 1987. 22, 26, 129

[84] X. Gu, J. Cui, and Q. Zhu. Abnormal crowd behavior detection by using the particle entropy . *Optik - International Journal for Light and Electron Optics*, 125(14):3428 – 3433, 2014. 67, 68, 70

[85] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. Efficient subpixel image registration algorithms. *Opt. Lett.*, 33(2):156–158, Jan 2008. 102, 115

[86] O. Hamdoun, F. Moutarde, B. Stanciulescu, and B. Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–6, Sept 2008. 55

[87] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *2011 International Conference on Computer Vision*, pages 263–270, Nov 2011. 30, 37

[88] A. Haselhoff, L. Hoehmann, C. Nunn, M. Meuter, and A. Kummert. On Occlusion-Handling for People Detection Fusion in Multi-camera Networks. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, volume 149 of *Communications in Computer and Information Science*, pages 113–119. Springer Berlin Heidelberg, 2011. 55

[89] T. Hassner, Y. Itcher, and O. Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–6, June 2012. 67, 68

[90] D. Helbing and P. Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282–4286, May 1995. 16

[91] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, March 2015. 37, 40

[92] G. Heredia, F. Caballero, I. Maza, L. Merino, A. Viguria, and A. Ollero. Multi-Unmanned Aerial Vehicle (UAV) Cooperative Fault Detection Employing Differential Global Positioning (DGPS), Inertial and Vision Sensors. *Sensors*, 9(9): 7566–7579, 2009. 60, 91

[93] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing*, 5(3):236 – 252, 2009. Pervasive Health and Wellness Management. 56

[94] H.-H. Hsu, Z. Cheng, T. Huang, and Q. Han. Behavior Analysis with Combined RFID and Video Information. In J. Ma, H. Jin, L. Yang, and J.-P. Tsai, editors, *Ubiquitous Intelligence and Computing*, volume 4159 of *Lecture Notes in Computer Science*, pages 176–181. Springer Berlin Heidelberg, 2006. 12, 56

[95] M. Hu, S. Ali, and M. Shah. Detecting Global Motion Patterns in Complex Videos. In *Proceedings of International Conference on Pattern Recognition*, pages 1–5, Tempa, Florida, 2008. IEEE. 16, 123

[96] M. Hu, S. Ali, and M. Shah. Learning Motion Patterns in Crowded Scenes Using Motion Flow Field. In *Proceedings of International Conference on Pattern Recognition*, pages 1–5, Tempa, Florida, 2008. IEEE. 16

[97] N. Hu, H. Bouma, and M. Worring. Tracking individuals in surveillance video of a high-density crowd. In *Proc. SPIE*, volume 8399, pages 839909–839909–8, 2012. 24

[98] C. Huang, Y. Li, and R. Nevatia. Multiple Target Tracking by Learning-Based Hierarchical Association of Detection Responses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):898–910, April 2013. 54

[99] C.-H. Huang, Y.-T. Wu, J.-H. Kao, M.-Y. Shih, and C.-C. Chou. A Hybrid Moving Object Detection Method for Aerial Images. In *Advances in Multimedia*

Information Processing - PCM 2010, volume 6297 of *Lecture Notes in Computer Science*, pages 357–368. Springer Berlin Heidelberg, 2010. 93

[100] C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):791–812, Jul 2002. 24

[101] G. Hummel, L. Kovács, P. Stütz, and T. Szirányi. Data Simulation and Testing of Visual Algorithms in Synthetic Environments for Security Sensor Networks. In N. Aschenbruck, P. Martini, M. Meier, and J. Tölle, editors, *Future Security*, volume 318 of *Communications in Computer and Information Science*, pages 212–215. Springer Berlin Heidelberg, 2012. 90

[102] R. J. Hyndman. Why every statistician should know about cross-validation. `http://robjhyndman.com/hyndsight/crossvalidation/`, 2010. Accessed: 2015-11-26. 138

[103] A. Ibrahim, P. W. Ching, G. Seet, W. Lau, and W. Czajewski. Moving Objects Detection and Tracking Framework for UAV-based Surveillance. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 456–461, Nov 2010. 59

[104] M. Isard and A. Blake. CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 24

[105] J. C. S. Jacques Junior, S. R. Musse, and C. R. Jung. Crowd Analysis Using Computer Vision Techniques: A survey. *Signal Processing Magazine, IEEE*, 27 (5):66–77, 2010. 12, 64, 66, 68, 121

[106] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005. 134

[107] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, July 2012. 33, 37, 46, 47, 49, 50

[108] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita. An Affective Guide Robot in a Shopping Mall. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, HRI '09, pages 173–180, New York, NY, USA, 2009. ACM. 12, 56

[109] P. Kelly, N. E. O'Connor, and A. F. Smeaton. Robust pedestrian detection and tracking in crowded scenes. *Image and Vision Computing*, 27(10):1445 – 1458, 2009. Special Section: Computer Vision Methods for Ambient Intelligence. 14

[110] S. M. Khan and M. Shah. A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 133–146. Springer Berlin Heidelberg, 2006. 14, 55

[111] S. M. Khan and M. Shah. Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):505–519, March 2009. 15, 53, 55, 121, 125

[112] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, Nov 2005. 15, 53

[113] P. Kilambi, O. Masoud, and N. Papanikolopoulos. Crowd Analysis at Mass Transit Sites. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 753–758, Sept 2006. 12

[114] J. Kim and K. Grauman. Observe Locally, Infer Globally: A space-time MRF for detecting abnormal activities with incremental updates. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2921–2928, 2009. 16

[115] D. Klein, D. Schulz, S. Frintrop, and A. Cremers. Adaptive real-time video-tracking for arbitrary objects. In *Intelligent Robots and Systems (IROS), 2010*

IEEE/RSJ International Conference on, pages 772–777, Oct 2010. 14, 31, 32, 46, 47, 49, 50

[116] D. Kong, D. Gray, and H. Tao. A Viewpoint Invariant Approach for Crowd Counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187–1190, 2006. 14

[117] L. Kratz and K. Nishino. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 693–700, June 2010. 14

[118] L. Kratz and K. Nishino. Spatio-Temporal Motion Pattern Modelling of Extremely Crowded Scenes. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis*, Marseille, France, October 12-18 2008. 16

[119] L. Kratz and K. Nishino. Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, Maimi Beach, Florida, June 20-25 2009. 16

[120] M. Kristan and L. Cehovin. Visual Object Tracking Challenge (VOT2013) Evaluation Kit. *Visual Object Tracking Challenge*, 2013. 19, 44

[121] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, and T. Vojir. The vot2013 challenge: overview and additional results. In *Computer Vision Winter Workshop*, 2014.

[122] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernández, A. Lukežič, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangeršič, G. Häger, F. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Choi, J.-W. Choi, J. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. Yi, L. Qin, L. Wen, M. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. Torr, Q. Huang,

R. Bowden, S. Hare, S. Lim, S. Hong, S. Liao, S. Hadfield, S. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. Niu. The Visual Object Tracking VOT2014 Challenge Results. In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014 Workshops*, volume 8926 of *Lecture Notes in Computer Science*, pages 191–217. Springer International Publishing, 2015. 19, 44

[123] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10):1518–1539, Oct 2001. 90

[124] S. Kwak, W. Nam, B. Han, and J. H. Han. Learning occlusion with likelihoods for visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1551–1558, nov. 2011. 51

[125] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276, June 2010. 33, 46, 47, 49, 50, 60

[126] J. Kwon and K. M. Lee. Wang-Landau Monte Carlo-Based Tracking Methods for Abrupt Motions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):1011–1024, 2013. 22, 40, 46, 47, 49, 50, 58, 60

[127] V. Lasdas, R. Timofte, and L. Van Gool. Non-parametric motion-priors for flow understanding. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 417–424. IEEE, 2012. 123, 124

[128] L.-K. Lee, S.-Y. An, and S. young Oh. Robust visual object tracking with extended CAMShift in complex environments. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 4536–4542, Nov 2011. 23

[129] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1683–1698, 2008. 24

[130] C. Leistner, H. Grabner, and H. Bischof. Semi-supervised boosting using visual similarity learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 29

[131] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2290–2297, Dec 2009. 34

[132] A. Li, M. Lin, Y. Wu, M. H. Yang, and S. Yan. NUS-PRO: A New Visual Tracking Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):335–349, Feb 2016. 45

[133] H. Li, Y. Li, and F. Porikli. Robust Online Visual Tracking with a Single Convolutional Neural Network. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *12th Asian Conference on Computer Vision*, pages 194–209. Springer, 2014. 39, 40

[134] J. Li, X. Lu, L. Ding, and H. Lu. Moving Target Tracking via Particle Filter Based on Color and Contour Features. In *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*, pages 1–4, Dec 2010. 24

[135] M. Li, Z. Zhang, K. Huang, and T. Tan. Rapid and robust human detection and tracking based on omega-shape features. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2545–2548, Nov 2009. 56

[136] X. Li, F. Sun, and Y. Y. Liu. Fusion Tracking Algorithm of Mean-shift and Particle Filter Based on EMD. In *Computer Science Service System (CSSS), 2012 International Conference on*, pages 1896–1899, Aug 2012. 24

[137] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12): 5630–5644, Dec 2015. 38

[138] G. Lidoris, F. Rohrmüller, D. Wollherr, and M. Buss. The Autonomous City Explorer (ACE) project — mobile robot navigation in highly populated urban environments. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1416–1422, May 2009. 13

[139] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in neural information processing systems*, pages 793–800, Cambridge, 2005. MIT Press. 26

[140] M. K. Lim, C. S. Chan, D. Monekosso, and P. Remagnino. Refined particle swarm intelligence method for abrupt motion tracking. *Information Sciences*, 283:267 – 287, 2014. New Trend of Computational Intelligence in Human-Robot Interaction. 22, 40, 46, 47, 49, 50, 60

[141] Y. Lin, Q. Yu, and G. Medioni. Efficient detection and tracking of moving objects in geo-coordinates. *Machine Vision and Applications*, 22(3):505–520, 2011. 92, 93, 102, 113

[142] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and Fast Collaborative Tracking with Two Stage Sparse Optimization. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 624–637. Springer Berlin Heidelberg, 2010. 36, 46, 47, 49, 50

[143] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and K-selection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1313–1320, June 2011. 36

[144] B. Liu, J. Huang, C. Kulikowski, and L. Yang. Robust Visual Tracking Using Local Sparse Appearance Model and K-Selection. *Pattern Analysis and Machine*

Intelligence, IEEE Transactions on, 35(12):2968–2981, Dec 2013. 36, 46, 47, 49, 50

[145] X. Liu, H. Chu, and P. Li. Research of the Improved Camshift Tracking Algorithm. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 968–972, Aug 2007. 23

[146] B. P. L. Lo, J. Sun, and S. A. Velastin. Fusing Visual and Audio Information in a Distributed Intelligent Surveillance System for Public Transport Systems. *Acta Automatica Sinica*, 29(3):393–407, 2003. 12

[147] B. Lo and S. Velastin. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 158–161, 2001. 12

[148] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 83, 91

[149] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical Convolutional Features for Visual Tracking. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082, December 2015. 39

[150] Y. Ma and P. Cisar. Activity Representation in Crowd. In *Proceedings of the* 2008 *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 107–116, Florida, USA, December 4-6 2008. Springer-Verlag. 16

[151] T. Määttä, A. Härmä, and H. Aghajan. On Efficient Use of Multi-view Data for Activity Recognition. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 158–165. ACM, 2010. 126

[152] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. 134

[153] E. Maggio and A. Cavallaro. Hybrid Particle Filter and Mean Shift tracker with adaptive transition model. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages 221–224, March 2005. 24

[154] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly Detection in Crowded Scenes. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, San Francisco, June 2010. 16

[155] J. Mairal, F. Bach, and J. Ponce. Sparse Modeling for Image and Vision Processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3): 85–283, 2014. 35, 36

[156] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM. 35

[157] S. Mann and R. Picard. Video orbits of the projective group a simple approach to featureless estimation of parameters. *Image Processing, IEEE Transactions on*, 6(9):1281–1295, Sep 1997. 93

[158] A. Marana, S. Velastin, L. Costa, and R. Lotufo. Automatic estimation of crowd density using texture . *Safety Science*, 28(3):165 – 175, 1998. 14

[159] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):810–815, June 2004. 22, 24, 29, 129

[160] R. Mazzon and A. Cavallaro. Multi-camera tracking using a Multi-Goal Social Force Model . *Neurocomputing*, 100:41 – 50, 2013. Special issue: Behaviours in video. 12, 55

[161] R. Mazzon, S. F. Tahir, and A. Cavallaro. Person re-identification in crowd. *Pattern Recognition Letters*, 33(14):1828 – 1837, 2012. Novel Pattern Recognition-Based Methods for Re-identification in Biometric Context. 12, 15, 55, 56

[162] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942, June 2009. 14, 16, 136, 154

[163] X. Mei, H. Ling, and D. Jacobs. Sparse representation of cast shadows via L1-regularized least squares. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 583–590, Sept 2009. 36, 46, 47, 49, 50

[164] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient ℓ-1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264, June 2011. 36, 46, 47, 49, 50, 51

[165] B. Michel, A. Gianluca, and W. Mats. Behavioural Dynamics for Pedestrians. *Lecture Notes in Computer Science*, pages 1–18, August 2003. 16

[166] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and De-Noising in Feature Spaces. In *NIPS*, volume 11, pages 536–542, 1998. 134

[167] K. Mikolajczyk and C. Schmid. An Affine Invariant Interest Point Detector. In *Computer Vision — ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 128–142. Springer Berlin Heidelberg, 2002. 113

[168] S. Mittal, T. Prasad, S. Saurabh, X. Fan, and H. Shin. Pedestrian detection and tracking using deformable part models and Kalman filtering. In *SoC Design Conference (ISOCC), 2012 International*, pages 324–327, Nov 2012. 23

[169] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis . *Computer Vision and Image Understand-*

ing, 104(2–3):90 – 126, 2006. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour. 12

[170] I. F. Mondragón, P. Campoy, C. Martínez, and M. A. Olivares-Méndez. 3D pose estimation based on planar object tracking for UAVs control. In *Robotics and Automation ICRA 2010 IEEE International Conference on*, pages 35–41, 2010. 91

[171] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1114–1127, Aug 2008. 19, 121

[172] I. Mtir, K. Kaaniche, M. Chtourou, and P. Vasseur. Aerial sequence registration for vehicle detection. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6, March 2012. 92, 93

[173] D. Mušicki, S. Suvorova, and S. Challa. Multi target tracking of ground targets in clutter with LMIPDA-IMM. In *Proceedings of the Seventh International Conference on Information Fusion, FUSION 2004*, volume 2, pages 1104–1110, 2004. 54

[174] T. Nawaz and A. Cavallaro. PFT: A protocol for evaluating video trackers. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2325–2328, Sept 2011. 41, 42, 44

[175] T. Nawaz and A. Cavallaro. A Protocol for Evaluating Video Trackers Under Real-World Conditions. *Image Processing, IEEE Transactions on*, 22(4):1354–1361, April 2013. 41, 43, 44

[176] A. Nemra and N. Aouf. Robust feature extraction and correspondence for UAV map building. In *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, pages 922–927, June 2009. 59, 91

[177] Z. Ni, S. Sunderrajan, A. Rahimi, and B. Manjunath. Distributed particle filter tracking with online multiple instance learning in a camera sensor network. In

Image Processing (ICIP), 2010 17th IEEE International Conference on, pages 37–40. IEEE, 2010. 24, 55

[178] K. Nordberg, P. Doherty, G. Farnebäck, P.-E. Forssén, G. Granlund, A. Moe, and J. Wiklund. Vision for a UAV helicopter. In *International Conference on Intelligent Robots and Systems (IROS)*, 2002. 57, 58, 59

[179] O.-D. Nouar, G. Ali, and C. Raphael. Improved Object Tracking With Camshift Algorithm. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II, May 2006. 23

[180] K. Nummiaro, E. Koller-Meier, and L. V. Gool. An adaptive color-based particle filter . *Image and Vision Computing*, 21(1):99 – 110, 2003. 24

[181] D. Oberhagemann. Static and dynamic crowd densities at major public events. Technical report, Technical Report vfdb TB 13-01, German Fire Protection Association, 2012. 121

[182] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 28–39. Springer Berlin Heidelberg, 2004. 53

[183] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally Orderless Tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1940–1947, June 2012. 34, 35, 46, 47, 49, 50

[184] N. Oxtoby, J. Ralph, C. Durniak, and D. Samsonov. Myriad target tracking in a dusty plasma. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–6, July 2011. 54

[185] P. Panangaden. Knowledge and information in probabilistic systems. In *Proceedings of the 19th International Conference on Concurrency Theory*, CONCUR '08, pages 4–4, Berlin, Heidelberg, 2008. Springer-Verlag. 65

[186] Y. Pang and H. Ling. Finding the Best from the Second Bests - Inhibiting Subjective Bias in Evaluation of Visual Tracking Algorithms. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 45

[187] R. Pelapur, S. Candemir, F. Bunyak, and M. Poostschi. Persistent target tracking using likelihood fusion in wide-area and full motion video sequences. In *Information Fusion (FUSION), 2012 Proceedings of the 15th International Conference on*, pages 2420–2427, 2012. 91

[188] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *European Conference on Computer Vision 2002*, pages 661–675, 2002. 24, 31, 128

[189] H. Plinval, P. Morin, P. Mouyon, and T. Hamel. Visual servoing for underactuated VTOL UAVs: a linear, homography-based framework. *International Journal of Robust and Nonlinear Control*, pages 1–24, 2013. 59, 91

[190] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 728–735. IEEE, 2006. 22, 24, 25, 41, 46, 47, 49, 50, 98, 107, 109

[191] H. Possegger, T. Mauthner, and H. Bischof. In Defense of Color-based Model-free Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2113–2120, 2015. 22, 38, 46, 47, 49, 50

[192] M. Pouzet, P. Bonnin, J. Laneurit, and C. Tessier. A robust real-time image algorithm for moving target detection from unmanned aerial vehicles (UAV). In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 01, pages 266–273, Sept 2014. 93

[193] H. Rahmalan, M. S. Nixon, and J. N. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545, June 2006. 66, 67

[194] V. Reddy, C. Sanderson, and B. Lovell. Improved Anomaly Detection in Crowded Scenes via Cell-based Analysis of Foreground Speed, Size and Texture. In *MLvMA Workshop, IEEE Conference on Computer Vision and Pattern Recognition*, pages 57–63, Colorado Springs, USA, June 2011. IEEE. 16

[195] V. Reilly, H. Idrees, and M. Shah. Detection and Tracking of Large Number of Targets in Wide Area Surveillance. In *Computer Vision – ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 186–199. Springer Berlin Heidelberg, 2010. 93, 103

[196] P. Reinartz, M. Lachaise, E. Schmeer, T. Krauss, and H. Runge. Traffic monitoring with serial images from airborne cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(3-4):149–158, December 2006. 90

[197] P. Reisman, O. Mano, S. Avidan, and A. Shashua. Crowd detection in video sequences. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 66–71, June 2004. 12

[198] P. Remagnino, P. Brand, and R. Mohr. Correlation techniques in adaptive template matching with uncalibrated cameras. In *Proc. SPIE*, volume 2356, pages 252–263, 1995. 59, 91

[199] X. Ren and J. Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17 vol.1, Oct 2003. 35

[200] J. Rivera-Bautista, A. Marin-Hernandez, and L. Marin-Urias. Using color histograms and range data to track trajectories of moving people from a mobile robot platform. In *Electrical Communications and Computers (CONIELECOMP), 2012 22nd International Conference on*, pages 288–293, Feb 2012. 12

[201] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430, Nov 2011. 14

[202] A. Romero, M. Gouiffés, and L. Lacassagne. Covariance Descriptor Multiple Object Tracking and Re-identification with Colorspace Evaluation. In J.-I. Park and J. Kim, editors, *Computer Vision - ACCV 2012 Workshops: ACCV 2012 International Workshops, Daejeon, Korea, November 5-6, 2012, Revised Selected Papers, Part II*, pages 400–411, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 98, 165

[203] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 26, 36, 46, 47, 49, 50, 129

[204] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 154

[205] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2): 99–121, 2000. 25, 34

[206] M. Russ, M. Schmitt, C. Hellert, and P. Stütz. Airborne sensor and perception management: Experiments and Results for surveillance UAS. *AIAA Infotech@Aerospace (I@A) Conference*, pages 1–16, 2013. 94

[207] I. Saleemi, L. Hartung, and M. Shah. Scene Understanding by Statistical Modeling of Motion Patterns. In *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2069–2076, San Francisco, June 2010. 16

[208] S. A. M. Saleh, S. A. Suandi, and H. Ibrahim. Recent survey on crowd density estimation and counting for visual surveillance . *Engineering Applications of Artificial Intelligence*, 41:103 – 114, 2015. 64, 65, 66, 68

[209] A. Salhi and A. Y. Jammaoussi. Object tracking system using Camshift, Meanshift and Kalman filter. *World Academy of Science, Engineering and Technology*, 6(4):598 – 603, 2012. 24

[210] A. Sanna, B. Pralio, F. Lamberti, and G. Paravati. A Novel Ego-Motion Compensation Strategy for Automatic Target Tracking in FLIR Video Sequences taken from UAVs. *Aerospace and Electronic Systems, IEEE Transactions on*, 45 (2):723–734, April 2009. 56, 60

[211] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 723–730, June 2010. 26, 32, 41, 42, 44, 46, 47, 49, 50, 84

[212] S. Saxena, F. Brémond, M. Thonnat, and R. Ma. Crowd Behavior Recognition for Video Surveillance. In J. Blanc-Talon, S. Bourennane, W. Philips, D. Popescu, and P. Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 5259 of *Lecture Notes in Computer Science*, pages 970–981. Springer Berlin Heidelberg, 2008. 14

[213] F. Schubert and K. Mikolajczyk. Robust Registration and Filtering for Moving Object Detection in Aerial Videos. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2808–2813, Aug 2014. 93

[214] S. Schwertfeger, A. Birk, and H. Bulow. Using iFMI spectral registration for video stabilization and motion detection by an Unmanned Aerial Vehicle (UAV). In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 61–67, Nov 2011. 59

[215] C. Sharp, O. Shakernia, and S. Sastry. A vision system for landing an unmanned aerial vehicle. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1720–1727, 2001. 90

[216] A. Shastry and R. Schowengerdt. Airborne video registration and traffic-flow parameter estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 6(4):391–405, Dec 2005. 89, 90

[217] J. Sherrah, D. Kamenetsky, R. Whatmough, and N. Redding. Online Tracking of People through a Camera Network. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 579–584, Dec 2011. 14

[218] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994. 83

[219] M. Shiomi, T. Kanda, D. Glas, S. Satake, H. Ishiguro, and N. Hagita. Field trial of networked social robots in a shopping mall. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2846–2853, Oct 2009. 12

[220] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. Interactive Humanoid Robots for a Science Museum. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, HRI '06, pages 305–312, New York, NY, USA, 2006. ACM. 12, 56

[221] M. Siam and M. ElHelw. Robust autonomous visual detection and tracking of moving targets in UAV imagery. In *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, volume 2, pages 1060–1066, Oct 2012. 60

[222] M. Siam and M. ElHelw. Enhanced Target Tracking in UAV Imagery with P-N Learning and Structural Constraints. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 586–593, Dec 2013. 92

[223] D. Simonnet, S. Velastin, J. Orwell, and E. Turkbeyler. Selecting and evaluating data for training a pedestrian detector for crowded conditions. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 174–179, Nov 2011. 17

[224] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to ob-

ject matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477 vol.2, 2003. 134

[225] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: An Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014. 45

[226] A. A. Sodemann, M. P. Ross, and B. J. Borghetti. A Review of Anomaly Detection in Automated Surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):1257–1272, 2012. 121, 122, 123, 126

[227] B. Song, T.-Y. Jeng, E. Staudt, and A. Roy-Chowdhury. A Stochastic Graph Evolution Framework for Robust Multi-target Tracking. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 605–619. Springer Berlin Heidelberg, 2010. 54, 55

[228] B. Song, R. Sethi, and A. Roy-Chowdhury. Wide Area Tracking in Single and Multiple Views. In T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, editors, *Visual Analysis of Humans*, pages 91–107. Springer London, 2011. 54, 55

[229] A. Sorokin, D. Berenson, S. Srinivasa, and M. Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2117–2122, Oct 2010. 13

[230] S. Stalder, H. Grabner, and L. Van Gool. *Cascaded Confidence Filtering for Improved Tracking-by-Detection*, pages 369–382. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 24

[231] J. A. Stankovic. Real-time computing. *BYTE*, 17(8):155–162, 1992. 8

[232] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time

tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 246–252, 1999. 102

[233] G. K. Still. *Introduction to Crowd Science*. CRC Press, 2014. 121

[234] R. Stolkin, I. Florescu, and G. Kamberov. An adaptive background model for camshift tracking with a moving camera. In *Proceedings of the 6th International Conference on Advances in Pattern Recognition*, pages 147–151. World Scientific, 2007. 23

[235] P. Stütz, G. Hummel, M. Kaiser, A. Schulte, N. Theissing, P. Climent-Pérez, P. Remagnino, D. Lund, A. Magzoub, Y. Tsado, J. Gozdecki, and K. Loziak. UAV integration aspects within the PROACTIVE network. In *Proceedings of the International Conference on World of UAV (Unmanned Aerial Vehicles)*. [in press], 2015. 94, 95

[236] W. Sun, L. Chen, L. Ren, B. Guo, and X. Wu. Objects detecting and tracking with a new particle filter. In *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pages 3340–3343, April 2012. 24

[237] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-Tracking Using Semi-Supervised Support Vector Machines. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. 28, 29, 32, 46, 47, 49, 50, 60

[238] S. L. Tang, Z. Kadim, K. M. Liang, and M. K. Lim. Hybrid blob and particle filter tracking approach for robust object tracking. *Procedia Computer Science*, 1(1):2549 – 2557, 2010. {ICCS} 2010. 17, 52

[239] J. B. Tenenbaum. Mapping a manifold of perceptual observations. *Advances in neural information processing systems*, pages 682–688, 1998. 134

[240] M. Teutsch and W. Kruger. Detection, Segmentation, and Tracking of Moving Objects in UAV Videos. *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 313–318, September 2012. 23, 91

[241] M. Thida, Y. L. Yong, P. Climent-Pérez, H.-l. Eng, and P. Remagnino. A Literature Review on Video Analytics of Crowded Scenes. In A. Cavallaro and P. K. Atrey, editors, *Intelligent Multimedia Surveillance: Current Trends and Research*, pages 17–36. Springer Berlin Heidelberg, 2013. 11, 14, 17, 18, 51, 55, 121, 125, 128

[242] A. Treptow, G. Cielniak, and T. Duckett. Real-time people tracking for mobile robots using thermal vision . *Robotics and Autonomous Systems*, 54(9):729 – 739, 2006. Selected papers from the 2nd European Conference on Mobile Robots (ECMR'05)2nd European Conference on Mobile Robots. 12

[243] D. Turner, A. Lucieer, and C. Watson. An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SfM) point clouds. *Remote Sensing*, 4 (5):1392–1410, 2012. 89

[244] O. Tuzel, F. Porikli, and P. Meer. Region Covariance: A Fast Descriptor for Detection and Classification. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3952 of *Lecture Notes in Computer Science*, pages 589–600. Springer Berlin Heidelberg, 2006. 22, 24, 25, 98, 107, 109

[245] G. Tzanidou, P. Climent-Perez, G. Hummel, M. Schmitt, P. Stutz, D. Monekosso, and P. Remagnino. Telemetry assisted frame registration and background subtraction in low-altitude UAV videos. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6, Aug 2015. 89

[246] M. Versichele, T. Neutens, M. Delafontaine, and N. V. de Weghe. The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities . *Applied Geography*, 32(2):208 – 220, 2012. 12, 57

[247] R. Vezzani, C. Grana, and R. Cucchiara. Probabilistic people tracking with appearance models and occlusion classification: The AD-HOC system. *Pattern Recognition Letters*, 32(6):867–877, 2011. 52

[248] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734–741 vol.2, Oct 2003. 17

[249] P. Viola and M. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 17

[250] A. Walha, A. Wali, and A. Alimi. Video stabilization with moving object detecting and tracking for aerial video surveillance. *Multimedia Tools and Applications*, pages 1–23, 2014. 92, 113

[251] J. Wang, Y. Ma, C. Li, H. Wang, and J. Liu. An Efficient Multi-object Tracking Method Using Multiple Particle Filters. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 6, pages 568–572, March 2009. 24

[252] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual Tracking With Fully Convolutional Networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3119–3127, December 2015. 39

[253] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pages 809–817, 2013. 39, 40

[254] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015. 39

[255] P. Wang, W. Li, W. Zhu, and H. Qiao. Object tracking with serious occlusion based on occluder modeling. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 1960–1965, aug. 2012. 52, 53

[256] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1323–1330, Nov 2011. 35, 46, 47, 49, 50

[257] Z. Wang, X. Yang, Y. Xu, and S. Yu. CamShift guided particle filter for visual tracking. *Pattern Recognition Letters*, 30(4):407 – 413, 2009. 24

[258] Z. Wang and A. Bovik. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *Signal Processing Magazine, IEEE*, 26(1):98–117, Jan 2009. 113

[259] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1): 77–90, 2006. 134

[260] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, pages 106–113. ACM, 2004. 134

[261] Z. Wen, Z. Peng, X. Deng, and S. Li. Particle Filter Object Tracking Based on Multiple Cues Fusion . *Procedia Engineering*, 15:1461 – 1465, 2011. {CEIS} 2011. 24

[262] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 Optical Flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 108.1–108.11, London, UK, September 2009. 84

[263] J. Winter. Using the Student's t-test with extremely small sample sizes. *Practical Assessment, Research & Evaluation*, 18(10):1–12, 2013. 76

[264] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan. Sparse Representation for Computer Vision and Pattern Recognition. *Proceedings of the IEEE*, 98(6):1031–1044, June 2010. 35, 36, 51

[265] C. Wu, A. H. Khalili, and H. Aghajan. Multiview Activity Recognition in Smart Homes with Spatio-temporal Features. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 142–149. ACM, 2010. 126

[266] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A Scalable Approach to Activity Recognition based on Object Use. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. 56, 57

[267] P. Wu, L. Kong, F. Zhao, and X. Li. Particle filter tracking based on color and SIFT features. In *2008 International Conference on Audio, Language and Image Processing*, pages 932–937. IEEE, July 2008. 24

[268] S. Wu, B. E. Moore, and M. Shah. Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2054–2060, San Francisco, CA, USA, June 13-18 2010. IEEE. 16

[269] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai. Real-Time Probabilistic Covariance Tracking With Efficient Model Update. *Image Processing, IEEE Transactions on*, 21(5):2824–2837, May 2012. 25

[270] Y. Wu, J. Lim, and M.-H. Yang. Online Object Tracking: A Benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418, June 2013. 44

[271] L. Xu and A. Stentz. An efficient algorithm for environmental coverage with multiple robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4950–4955, May 2011. 12

[272] Y. Xu and D. Song. Systems and algorithms for autonomous and scalable crowd surveillance using robotic PTZ cameras assisted by a wide-angle camera. *Autonomous Robots*, 29(1):53–66, 2010. 12, 55

[273] F. Yang, H. Lu, and Y. wei Chen. Bag of Features Tracking. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 153–156, Aug 2010. 31, 46, 47, 49, 50

[274] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review . *Neurocomputing*, 74(18):3823 – 3831, 2011. 18, 19, 20, 21, 51

[275] Y. Yang, J. Liu, and M. Shah. Video Scene Understanding using Multi-Scale Analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1669–1676, Kyoto Japan, September 26 - October 4 2009. IEEE. 16

[276] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), December 2006. 18, 19, 20, 21, 51, 52, 53

[277] M. Yin, J. Zhang, H. Sun, and W. Gu. Multi-cue-based CamShift guided particle filter tracking. *Expert Systems with Applications*, 38(5):6313 – 6318, 2011. 24

[278] Q. Yu, T. Dinh, and G. Medioni. Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 678–691. Springer Berlin Heidelberg, 2008. 29, 46, 47, 48, 49, 50

[279] X. Yu, X. Chen, and H. Zhang. Accurate Motion Detection in Dynamic Scenes Based on Ego-Motion Estimation and Optical Flow Segmentation Combined Method. In *Photonics and Optoelectronics (SOPO), 2011 Symposium on*, pages 1–4, May 2011. 59, 91

[280] C. Yuan, F. Recktenwald, and H. Mallot. Visual steering of UAV in unknown environments. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3906–3911, Oct 2009. 59, 91

[281] M. R. Zare, A. Mueen, M. Awedh, and W. C. Seng. Automatic classification of

medical X-ray images: hybrid generative-discriminative approach. *IET Image Processing*, 7(5):523–532, July 2013. 150

[282] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, April 2008. 12, 14, 66, 121

[283] C. Zhang, J. Xu, A. Beaugendre, and S. Goto. A KLT-based approach for occlusion handling in human tracking. In *Picture Coding Symposium (PCS), 2012*, pages 337–340, may 2012. 52

[284] J. Zhang, S. Ma, and S. Sclaroff. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *13th European Conference on Computer Vision*, pages 188–203. Springer, 2014. 34

[285] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *Computer Vision–ECCV 2014*, pages 127–141. Springer, 2014. 37, 38

[286] S. Zhang. Object Tracking in Unmanned Aerial Vehicle (UAV) Videos Using a Combined Approach. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages 681–684, March 2005. 91

[287] T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang, and X. He. A new method for violence detection in surveillance scenes. *Multimedia Tools and Applications*, pages 1–23, 2015. 67, 68

[288] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust Visual Tracking via Structured Multi-Task Sparse Learning. *International Journal of Computer Vision*, 101(2):367–383, 2013. 24

[289] Q. Zhong, Z. Qingqing, and G. Tengfei. Moving Object Tracking Based on Codebook and Particle Filter. *Procedia Engineering*, 29:174–178, 2012. 52

[290] X. Zhu, J. Liu, J. Wang, W. Fu, and H. Lu. Weighted Interaction Force Estimation for Abnormality Detection in Crowd Scenes. In K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, editors, *Computer Vision – ACCV 2012*, volume 7726 of *Lecture Notes in Computer Science*, pages 507–518. Springer Berlin Heidelberg, 2013. 123, 124

[291] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31, Aug 2004. 72