Tracking interacting targets in multi-modal sensors

Taj, Murtaza

Tracking Interacting Targets in Multi-modal Sensors

A dissertation presented

by

Murtaza Taj

to

The School of Electronic Engineering and Computer Science

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Electronic Engineering

Queen Mary University of London

Mile End Road

E1 4NS, London, UK

September 2009

"In the name of God, most Gracious, most Compassionate"

I confirm that the work presented in this thesis is my own and the work of other persons is appropriately acknowledged.

Sincerely yours,

Murtaza Taj

Thesis supervisor
Andrea Cavallaro

Author
Murtaza Taj

Tracking Interacting Targets in Multi-modal Sensors

Abstract

Object tracking is one of the fundamental tasks in various applications such as surveillance, sports, video conferencing and activity recognition. Factors such as occlusions, illumination changes and limited field of observance of the sensor make tracking a challenging task. To overcome these challenges the focus of this thesis is on using multiple modalities such as audio and video for multi-target, multi-modal tracking. Particularly, this thesis presents contributions to four related research topics, namely, pre-processing of input signals to reduce noise, multi-modal tracking, simultaneous detection and tracking, and interaction recognition.

To improve the performance of detection algorithms, especially in the presence of noise, this thesis investigate filtering of the input data through spatio-temporal feature analysis as well as through frequency band analysis. The pre-processed data from multiple modalities is then fused within Particle filtering (PF). To further minimise the discrepancy between the real and the estimated positions, we propose a strategy that associates the hypotheses and the measurements with a real target, using a Weighted Probabilistic Data Association (WPDA). Since the filtering involved in the detection process reduces the available information and is inapplicable on low signal-to-noise ratio data, we investigate simultaneous detection and tracking approaches and propose a multi-target track-before-detect Particle filtering (MT-TBD-PF). The proposed MT-TBD-PF algorithm bypasses the detection step and performs tracking in the raw signal. Finally, we apply the proposed multi-modal tracking to recognise interactions between targets in regions within, as well as outside the cameras' fields of view.

The efficiency of the proposed approaches are demonstrated on large uni-modal, multi-modal and multi-sensor scenarios from real world detections, tracking and event recognition datasets and through participation in evaluation campaigns.

Table of Contents

Previously Published Work

Book Chapter

[Ch1] M. Taj and A. Cavallaro. Recognizing Interactions in Video. *Intelligent Multimedia Analysis for Security Applications*, Springer Verlag GmbH, 2009.

Journals

[J1] F. Daniyal, M. Taj and A. Cavallaro. Content and task-based view selection from multiple video streams. *Data semantics from multi-media systems, Special issue: semantic multimedia*, September 2009.

[J2] H. Zhou, M. Taj and A. Cavallaro. Target detection and tracking with heterogeneous sensors. *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 505–513, August 2008.

[J3] E. Maggio, M. Taj and A. Cavallaro. Efficient multi-target visual tracking using random finite sets. *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1016–1027, August 2008.

[J4] S. Karlsson, M. Taj and A. Cavallaro. Detection and tracking of humans and faces *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp 1–9, 2008.

Conferences

[C1] M. Taj and A. Cavallaro, "Multi-camera track-before-detect", in *ACM/IEEE Int. Conf. on Dist. Smart Cameras*, Como, IT, 30 August–02 September 2009.

[C2] M. Taj and A. Cavallaro, "Audio-assisted trajectory estimation in non-overlapping multi-camera networks," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Taipei, TW, 19–24 April 2009.

[C3] F. Daniyal, M. Taj and A. Cavallaro, "Content-aware ranking of video segments," in *ACM/IEEE Int. Conf. on Dist. Smart Cameras*, Stanford, CA, USA, 07–11 September 2008.

[C4] M. Taj and A. Cavallaro, "Object and scene-centric activity detection using state occupancy duration modeling," in *IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Santa Fe, NM, USA, 1–3 September 2008.

[C5] G. Kayumbi, P. Mazzeo, P. Spagnolo, M. Taj and A. Cavallaro. Distributed visual sensing for virtual top-view trajectory generation in football videos in *Proc. of ACM Int. Conf. on Image and Video Retrieval*, Niagara Falls, CA, 7–9 July 2008.

[C6] H. Zhou, M. Taj, and A. Cavallaro, "Audiovisual tracking using STAC sensors," in *ACM/IEEE Int. Conf. on Dist. Smart Cameras*, Vienna, AT, 25–28 September 2007.

[C7] M. Taj and A. Cavallaro, "Multi-camera scene analysis using an object-centric continuous distribution hidden Markov model," in *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio, TX, USA, 16–19 September 2007.

[C8] M. Bregonzio, M. Taj, and A. Cavallaro, "Multi-modal particle filtering tracking using appearance, motion and audio likelihoods," in *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio, TX, USA, 16–19 September 2007.

[C9] M. Taj, E. Maggio, and A. Cavallaro. Objective evaluation of pedestrian and vehicle tracking on the CLEAR surveillance dataset. in *Classification of Events, Activities and Relationships Workshop*, Springer LNCS 4625, Baltimore, MD, USA, 8–11 May 2007.

[C10] N. Anjum, M. Taj, and A. Cavallaro, "Relative position estimation of non-overlapping cameras," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, USA, 15–20 April 2007.

[C11] A. Cavallaro, R. Chandrasekera, and M. Taj, "Hands-on experience in image processing: the automated lecture cameraman," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, USA, 15–20 April 2007.

[C12] M. Taj, E. Maggio, and A. Cavallaro. Multi-feature graph-based object tracking. in *Classification of Events, Activities and Relationships Workshop*, Springer LNCS 4122, pp. 190–199, Southampton, UK, 6–7 April 2006.

Miscellaneous

[M1] M. Taj, F. Daniyal, and A. Cavallaro, "Event analysis on TRECVid 2008 London Gatwick dataset," in *TREC Video Retrieval Workshop*, Gaithersburg, MD, USA, 17–18 November 2008.

[M2] M. Taj, N. Anjum, and A. Cavallaro, "Multi-camera calibration and event recognition with outlier behaviour detection," in *Proc. of Int. Crime Science Conf*, London, UK, 16–17 July 2007.

Electronic preprints are available on the Internet at the following URL:
http://www.elec.qmul.ac.uk/staffinfo/andrea/publications.html

Acknowledgments

First, I would like to express my gratitude to my supervisor Dr. Andrea Cavallaro for his continuous support throughout my time at Queen Mary. His guidance and dedication have greatly contributed to my work. Then, I would like to thank my second supervisor Prof. Mark Plumbley and independent assessor Prof. Alan Pearman for their constructive feedback during various stages of the degree.

Many appreciations to all my colleagues and students working under my supervision, the collaboration with whom has a significant impact on my research. In particular, I would like to acknowledge the contribution of Nadeem Anjum, Matteo Bregonzio, Fahad Daniyal, Stefan Karlson, Emilio Maggio and Huiyu Zhou. I would also like to thank all the people in the MMV group, particularly all the guys in room 160, CS438 and CS441 for their assistance in data recordings and particularly for being supportive, friendly and sharing with me the experience of pursuing a PhD degree.

Above all, I am grateful to God and to my family for their continuous love and support. The continuous encouragement of my parents, siblings, friends and colleagues has been instrumental in achieving my goals. Many thanks to all my colleagues, family members and friends.

To my family

List of abbreviations

AA-KF	Arrival angle estimation after Kalman filter
AB-KF	Arrival angle estimation before Kalman filter
AD	Automatic Detection
APIDIS	Autonomous production of images based on distributed and intelligent sensing
AV	Audiovisual
AVDW	Audiovisual Fusion using Dynamic Weighting
AVKF	AVDW with Trajectory Smoothing using Kalman filter
BC	Broadway/Church
BR	Brannigans
C-T	Centre of mass tracking
CAVIAR	Context Aware Vision using Image-based Active Recognition
CB-T	Centre of mass and bounding box tracking
CBD-T	Centre of mass, bounding box and direction tracking
CBDH-T	Centre of mass, bounding box, direction and histogram tracking
CHMM	Coupled Hidden Markov Models
CLEAR	Classification of Events, Activities and Relationships
DBN	Dynamic Bayesian Network
DKF	Decentralised Kalman filter
EM	Expectation Maximisation
FN	False Negatives
FP	False Positives

GCC	Generalised Cross-correlation
GCC-PHAT	Generalised cross-correlation Phase Transform
GM	Graph Matching
GMM	Gaussian Mixture Model
GMMPHD	Gaussian Mixture Model Probability Hypothesis Density
GP-PF	GCC-PHAT arrival angle estimation and Particle filter based AVDW tracker
GPM	Graphical Models
GT	Ground Truth
HCI	Human Computer Interaction
HMM	Hidden Markov Models
HSMM	Hidden Semi-Markov Models
i-Lids	Imagery Library for Intelligent Detection Systems
IER	Interaction event recognition
IISA	Institute of Intelligent Systems for Automation
IR	Infrared
KF	Kalman filter
KF-PF-P	KF audio detection and the Particle filter-based audiovisual tracker with PDA
KF-PF-WP	KF and Particle filter based audiovisual tracker with WPDA
LDA	Linear Discriminant Analysis
MAP	Maximum a Posterior
MB	Multi-band Frequency Analysis
MFA	Multi-feature Analysis
ML	Maximum Likelihood
MODA	Multi-object Detection Accuracy
MODP	Multi-object Detection Precision
MOMC-HMM	Multi-Observation-Mixture+Counter Hidden Markov Models
MOTA	Multi-object Tracking Accuracy

MOTP	Multi-object Tracking Precision
MS	Mean-shift
MT-TBD-PF	Multi-target Track-before-detect Particle filter
PDA	Probabilistic Data Association
PETS	Performance Evaluation of Tracking and Surveillance
PF	Particle filter
PHAT	Phase Transform
PHD	Probability Hypothesis Density
QW	Queensway
RF	Reverberation filtering
SCOT	Smoothed Coherence Transform
SNR	Signal-to-Noise Ratio
STAC	Stereo Audio and Cycloptic Vision
STF	Spatio-temporal filtering
TBD	Track-before-detect
TDNN	Time Delay Neural Networks
TDOA	Time Difference of Arrival
TP	True Positives
VLHMM	Variable-length Hidden Markov Models
WPDA	Weighted Probabilistic Data Association

Glossary

AD	automatic detection.
GT	ground truth.
N_k^{AD}	number of an entity defined by superscript (automatic detections in this case) at time k.
S	set of discrete events.
X_k^r	set of target states (track) up to time k.
Z_k	set of measurements up to time k.
$\hat{\mathbf{y}}_{ik}$	audio signal received at i^{th} receiver at time k.
$\mathbf{A}$	bold upper case letters indicate matrices.
$\mathbf{x}_k$	single-target state at time-step k.
$\mathbf{z}_k$	measurement at time-step k.
$\mathcal{H}$	contribution of the target intensity at pixel position (i, j) at time k.
ω	weight.
$h_k(i, j)(\mathbf{x}_k)$	contribution of the target intensity at pixel position (i, j) at time k.
k	time-step (frame) index.
$z_k(i, j)$	measurement intensity at position (i, j).

Chapter 1

Introduction

1.1 Motivation

In the last decades video cameras have been increasingly used in applications such as surveillance, health care monitoring and entertainment. However, until recently, most video processing was done manually, for example by an operator observing a CCTV camera or by a technician observing a medical video. The ever increasing amount of captured video has created the need for automatic processing to enable an effective utilisation of the data.

Although commercial applications exist that address the issue of automatic video processing (e.g., obstacle detection [1], flaw detection [2] and arbitrary view point generation [3]), most applications work in constrained or well defined scenarios. In fact, these applications rely mainly on detection and tracking modules that are affected by environmental conditions such as reflections, illumination changes and occlusions.

The limitations of current automatic video processing are due to the complexity of real scenes as well as to the limited capability of cameras themselves, such as for example their fields of view. Although multiple cameras can be used to increase the observed area, in many cases even multiple cameras cannot cover the entire scene. A possible solution is to use multi-modal sensors, combining cameras with other sensors with a wider field of observation. An example of such sensors is microphones.

The use of multiple sensors introduces new challenges related to the synchronisation and to the fusion of the data streams. In this thesis, we investigate the use of multi-modal sensors composed of a camera coupled with a microphone pair, here referred to as Stereo Audio and Cycloptic Vision (STAC) sensor. Compared to large microphone arrays used in conventional settings [4], a STAC sensor is viable in realistic scenarios. However, the use of fewer sensors increases the uncertainty in the acquired data thus

creating the need for an effective signal filtering and manipulation. This thesis will try and answer the following questions: (i) How can we extract *multiple targets* in realistic scenarios? (ii) How can we *track* these targets with single and multiple (heterogeneous) sensors? (iii) How can the resulting trajectories be used to detect *activities* performed by the targets either individually or while interacting with each other?

1.2 Main contributions

The main contributions of this thesis are as follows:

- We propose the first multi-target multi-sensor track-before-detect Particle filter (MT-TBD-PF) algorithm for visual applications [C1]. Unlike conventional tracking algorithms that incorporate measurement at the likelihood level [5] to track multiple objects, track-before-detect uses signal intensity in the state representation. The proposed algorithm, unlike conventional techniques, does not hard threshold the input signal. To enable multiple target tracking we have introduced a cluster step and performed distributed updating and resampling. The proposed approach does not require manual initialisation of the targets nor prior knowledge of the number of clusters, as we use mean-shift on the particles [C1].

- We propose a multi-modal tracking algorithm by fusion of audio and video to track targets consistently under visual occlusions. We reduce the uncertainty in the estimation of the angle of arrival through reverberation filtering based on precedence effects and multi-band analysis [C8]. This estimation is then refined by applying weighted probabilistic data association (WPDA) (which is a joint contribution with Dr. Zhou) to increase robustness against reverberation and noise [C6, J2]. The modalities are finally fused at the likelihood level within particle filtering.

- We propose an interaction event recognition framework which is the first to perform recognition of interactions in regions uncovered by the cameras by using audio [C2]. The activities are modelled as either interactions between dynamic and static objects (*dynamic-static interactions*) or as interactions between multiple dynamic objects (*dynamic-dynamic interactions*) [Ch1]. To this extent, we propose scene-centric and object-centric models for dynamic-static interaction recognition incorporating duration distribution in the state estimation using HMM Viterbi decoding [C7]. The use of Viterbi decoding eliminates the limitation of recognising only known state sequence

templates. For dynamic-dynamic interaction we modelled full coupling between the interacting processes based on the Coupled Hidden Markov Model.

- We have performed an extensive evaluation of the proposed image-based localisation, tracking and event recognition algorithm by participating in the CLEAR and the ETISEO evaluations. The performance of various blocks of the algorithm using different parameter values is demonstrated on the evaluation dataset consisting of over 1 hour 20 minutes of annotated sequences.

1.3 Outline of the thesis

The focus of this thesis is on multi-target, multi-modal tracking and interaction recognition particularly using audio and visual modalities. Part I of the thesis is about tracking in multi-modal sensors, whereas interaction recognition is discussed in Part II. The first three contributions are mentioned in Chapter 3, Chapter 4 and Chapter 6 respectively followed by conclusions and future work. The results from CLEAR and ETISEO evaluation are in experimental section of Chapter 3 and Chapter 6. The state of the art, of uni-modal and multi-modal tracking approaches is discussed in Chapter 2 whereas that of interaction recognition is reviewed in Chapter 5.

Chapter 2 discusses the related work on tracking using multiple sensors of different types (*heterogeneous* sensors). Information fusion is an important step towards robust multi-sensor tracking, and we presented the problem as either based on the *track-before-fuse* or *fuse-before-detect* strategies. The track-before-fuse strategy performs fusion of estimated trajectories that can be from uni-modal sensors for which image based target localisation and tracking techniques are presented for both single and multiple sensors. The fuse-before-track strategy on the other hand is more applicable in the case of multiple modalities. To this extent we introduce localisation using an audio signal as a complementary modality and discuss tracking using heterogeneous sensors. In the case of low signal-to-noise ratio signals the detection step is not favourable. To cater for these types of scenarios, we discuss simultaneous detection and tracking approaches and introduce techniques based on the *track-before-detect* strategy. We also discuss how these approaches can bypass the localisation step and are more applicable in *fuse-before-track* strategy.

Chapter 3 answers the first question presented in Sec. 1.1. In this chapter we firstly propose post-processing techniques to improve the image-based localisation and then extend an established graph-based multi-target tracking strategy by employing colour histograms for better modelling of target appearance. We then extend the tracking from a

single camera to multiple cameras and present two algorithms for tracking on the top-view containing fused information from all cameras. To this extent we propose a multi-camera track-before-detect algorithm based on novel multi-target particle filtering (MT-TBD-PF).

Chapter 4 is based on multi-modal tracking and answers the second question presented in Sec. 1.1. It also contributes towards answering the first question. To this extent we present pre-processing for increasing the accuracy of the estimation of the angle of arrival. Next we apply Weighted Probabilistic Data Association to further increase the robustness of multi-modal tracking. Then we introduce our particle filtering based multi-modal fusion strategy where fusion of multiple cues is performed at the likelihood level. In the second half of the chapter we extend this to multiple multi-modal sensors for extended tracking in regions uncovered by cameras.

Chapter 5 reviews the literature on interaction recognition using trajectories. In this context we introduce interactions as either between dynamic and static objects (*dynamic-static interactions*) or between multiple dynamic objects (*dynamic-dynamic interactions*). We introduce algorithms based on dynamic graphical models for recognising both these types of interactions.

Chapter 6 answers the third question presented in Sec. 1.1 and presents the proposed interaction recognition framework which takes as input the trajectories generated using the work presented in previous two chapters. To this extent we model interaction between dynamic objects with other static objects and interaction between multiple dynamic objects.

Chapter 7 summarises the achievements of this thesis and its possible future directions.

Part I

Tracking

Chapter 2

State of the Art

2.1 Introduction

Object tracking is a fundamental task in various applications such as surveillance, sports, video conferencing and activity recognition. Object tracking can either be (i) interactive or (ii) automatic. In interactive tracking, track initialisation is formed manually when the target appears in the scene [6, 7] (tag-and-track). The frame-to-frame linking is then performed using appearance based features such as colour histograms [6, 8], or shape-based features such as contours [9]. More autonomous approaches employ a detection mechanism for track initialisation [10–12]. These detection algorithms can be based on estimation of direction of arrival of the signal [13], image-based segmentation [14] or model-based classifiers [15–17]. In this chapter we will focus our discussion on the latter approach that does not require manual track initialisation.

Factors such as occlusions, bad lighting and limited field of observance of the sensor make tracking a challenging task. To overcome these challenges, multiple sensors of multiple modalities, such as a network of cameras and microphones, can be used. With the advancement of technology and the use of these sensors at mass level, it is now affordable to employ such sensors. Tracking in such heterogeneous sensor networks reduces the uncertainty of tracks due to redundancy of information, as well as allowing extended tracking over wider areas by utilising complementary data [18]. We will limit our discussion to two complementary modalities, namely video and audio. However, many techniques may be readily applicable to other modalities such as radar and radio frequency signals.

Data fusion is desired for effective utilisation of the information in sensor networks. The information fusion can be performed using two strategies (i) *track-before-fuse*, and (ii) *fuse-before-track*. In track-before-fuse the tracks generated on each camera-view

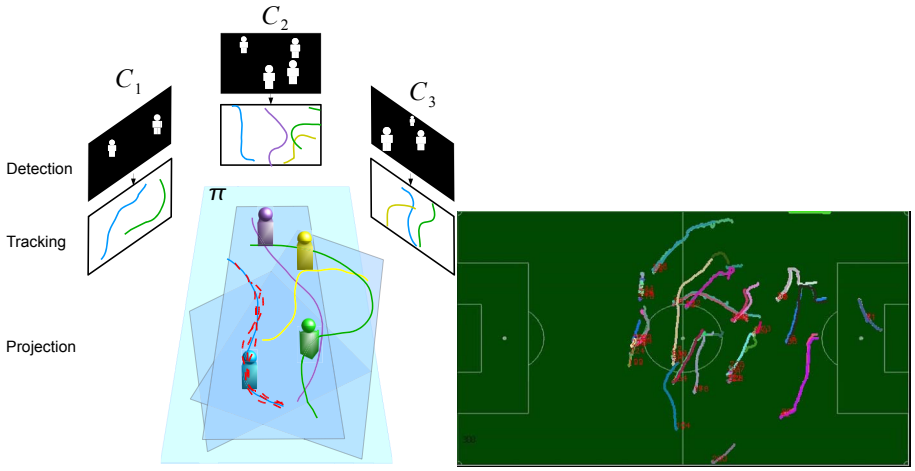


Figure 2.1: Sample *track-before-fuse* and intermediate results on the top-view. (a) Illustration of the track-before-fuse approach. (b) Projection of tracks from multiple-views for fusion of corresponding tracks. (C_1: camera 1; C_2: camera 2 and C_3: camera 3).

are fused together, whereas in the fuse-before-track, tracking is performed once on the fused data only. Literature review on track-before-fuse algorithms is discussed in Sec. 2.2 whereas on fuse-before-detect strategies is given in Sec. 2.3.

Both these strategies require a detection step either before or after fusion of information. A third category is *track-before-detect* where tracking can be performed without going through any additional detection phases [19, 20]. The related work on *track-before-detect* algorithms is discussed in Sec. 2.4.

2.2 Track-before-fuse

In *track-before-fuse* first objects are localised on each camera-view followed by single camera tracking [21–31]. These short tracks from individual sensors are then fused to improve tracking accuracy and to obtain extended tracks. Figure 2.1 shows the typical intermediate result of *track-before-fuse* algorithms where tracks from multiple-views are projected onto the top-view. The problem to be solved here is how to fuse the multiple tracks belonging to the corresponding targets.

In this section we will limit our discussion to uni-modal *track-before-fuse* techniques using the video only modality. We will first introduce the state of the art of image-based localisation techniques, followed by that of single camera tracking. Finally, we will review the literature on multi-camera *track-before-fuse*. Figure 2.2(a-b) shows a

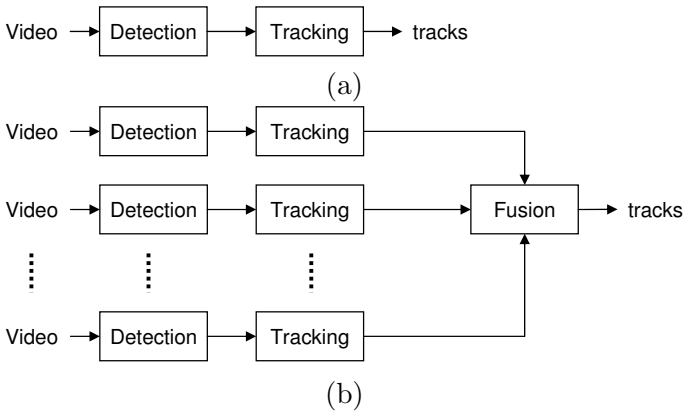


Figure 2.2: Generic block diagram of detection and tracking algorithms using uni-modal sensor(s). (a) Single sensor detection and tracking algorithms. (b) Multi-sensor *fuse-after-track* detection and tracking algorithms.

generic block diagram of *track-before-fuse* detection and tracking systems based on single and multiple video sensors only.

2.2.1 Image-based localisation

Object detectors can be based on an *object model* or a *background model*. The first class of detectors initially learns a model for objects of interest and then use a classifier that is generally applied to each frame of the sequence [32]. Object model based techniques [16, 32] learn local representative features of the object appearance and perform detection by searching for similar features in each frame. Edgelets [33] or Haar wavelets [17] are used in Adaboost algorithms as weak object classifiers that combined in a cascade form a strong classifier [34]. Approaches based on learned classifiers are also used after background subtraction to categorise the detections (i.e. to differentiate pedestrians from vehicles) [35]. Similarly, Support Vector Machines using simple object features, such as object size and width-height ratio, can be used [36]. Although these approaches are also appropriate in applications with non-static cameras, they can only detect object classes belonging to the training dataset.

In the second class the detection is performed by learning a model of the background and then by classifying pixels as either foreground or background [37–39]. These approaches all have a mathematical formulation equivalent to the following. Let I_k be the frame at time k; a simple foreground segmentation technique is to perform image difference

(a) (b) (c) (d)

Figure 2.3: Comparison of background subtraction results with and without update of the background model. (a) Reference frame, (b) frame 585, (c) sample result without background update and (d) sample result with background update.

between current and reference frame $I_k^{'}$ followed by binarisation using thresholding

$$I_k^f = \begin{cases} 1 & \left| I_k^{'} - I_k \right| > \hbar \\ 0 & \text{otherwise} \end{cases}, \tag{2.1}$$

where $\hbar$ is the threshold and I_k^f is the extracted foreground image at time k. There are two constraints associated with this method. First, the reference background $I_k^{'}$ needs to be computed and second the presence of additive noise makes it difficult to adjust the threshold $\hbar$. The reference background can be obtained by taking the average or median of previous frames as $I_k^{'} = \frac{1}{T}\sum_{t=k-T}^{k-1} I_t$, where T is the length of the time window. An alternative is to take an exponentially decaying weighted average of the previous frames as

$$I_k{}' = \omega I_{k-1} + (1-\omega) I_{k-1}^{'}, \tag{2.2}$$

where ω is the mixing weight and serves as a learning rate. The choice of ω depends on a trade-off between the update capabilities and the resilience to assimilating stopped or slow foreground objects in the background model. Figure 2.3 shows the comparison between the static and the adaptive background, indicating significant reduction in false positives with adaptive modelling. The drawback of these approaches is that the objects that become stationary may become part of the background resulting in missed detections. Equation 2.2 can be modified to address this problem as

$$I_k^{'}(i,j) = \omega I_{k-1}(i,j) + (1-\omega) I_{k-1}^{'}(i,j),\ \forall\ (i,j) \notin I_{k-1}^f, \tag{2.3}$$

where I_{k-1}^f is the set of pixels belonging to the foreground only at time $k-1$. This solves the problem with foreground objects that become stationary; however the errors due to

false foreground detections (such as due to sensor noise and rapid illumination changes) will not be rectified. Assuming that the colour value at each pixel follows a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with mean μ and variance σ^2, the statistics of the visible pixels can be recursively updated, in each frame, using an adaptive filter [40]

$$\mu_k(i,j) = \omega I_{k-1}(i,j) + (1-\omega)\mu_{k-1}(i,j) \tag{2.4a}$$

$$\sigma_k^2(i,j) = \omega(I_{k-1}(i,j) - \mu_{k-1}(i,j))^2 + (1-\omega)\sigma_{k-1}^2(i,j). \tag{2.4b}$$

In the above, recursive updating has to be applied for each detected object as well as for the background. Computation of the support map is also required for each object, indicating its occupancy [40]. Alternatively, assuming that the additive noise affecting each image of the sequence respects a Gaussian distribution with mean μ_k and variance σ_k^2 [38], the value of σ_k^2 is computed by analysing image differences $d(i,j) = \left|I_k^{'}(i,j) - I_k(i,j)\right|$. Here $d(i,j) \neq 0$ only because of camera noise and not because of other factors like scene changes due to moving object or illumination change. Based on this hypothesis, which we call H_0, the conditional probability density function $p(d(i,j)|H_0)$ is defined as

$$p(d(i,j)|H_0) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{d^2(i,j)}{2\sigma_k^2}\right). \tag{2.5}$$

This approach can only be applied to uni-modal backgrounds. In the case of the background distribution being multi-modal, a mixture of multiple Gaussians is used [37]. In this approach, each Gaussian is weighted according to the proportion of data it models and the probability of observing a current pixel value is computed as

$$p(I_k(i,j)) = \sum_{l=1}^{N^g} \omega_{lk}\mathcal{N}(I(i,j), \mu_{lk}, \sigma_{lk}). \tag{2.6}$$

where the distribution $\mathcal{N}$ is a Gaussian distribution defined as

$$\mathcal{N}(I(i,j), \mu_{lk}, \sigma_{lk}) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(I(i,j)-\mu_{lk})^T(I(i,j)-\mu_{lk})}{2\sigma_{lk}^2}\right) \tag{2.7}$$

and N^g is the number of Gaussians. The mean μ and variance σ are updated at each time k, for each Gaussian, using running average and weights ω are recomputed. These parameters are updated to cope with slow changes in natural light conditions. However, when an object becomes static it is gradually assimilated into the background model.

The update speed for the parametric model is usually a trade-off between a fast update required to cope with sudden illumination changes and a slow update necessary to allow the detection of slow or stopping objects. A possible solution is to modify the learning rate in the region around a moving object depending on its speed [41]. Edge information can also help in detecting objects when they become static [42]. Once the edge structure of the background is learned, a pixel is classified as foreground by comparing its gradient vector with the gradient distribution of the background model. A generalisation of the Gaussian Mixture model based background learning is to use Kernel density estimator $\mathcal{K}$ [43]

$$p(I_k(i,j)) = \frac{1}{T \times h} \sum_{t=k-T}^{k-1} \mathcal{K}\left(\frac{I_k(i,j) - I_t(i,j)}{h}\right), \tag{2.8}$$

where $\mathcal{K}$ is some kernel and h is a smoothing parameter called the bandwidth. The function $\mathcal{K}$ can be assumed to be Gaussian with zero mean and unit variance and thus the variance is controlled indirectly through the bandwidth h. Although being general and non-parametric, it's impractical due to its memory requirements. The recursive density approximation can be done efficiently by employing Mean-Shift (MS) [44] to detect density modes (at which the probability density function attains its maximum value) and propagating them over time. This makes the approach memory efficient as well as maintaining the flexibility of being non-parametric. However, heuristic methods are used for mode merging. In another approach [45], neurons are used instead of Gaussians to calculate the probability of a pixel belonging to background or foreground. In this technique a General Regression Neural Network (GRNN) is employed which can model the underlying foreground/background density. The weights of the connections between the neurons in the network are updated for each pixel of each frame $I_k(i,j)$ recursively as

$$\omega_k^{if} = (1-\alpha)\omega_{k-1}^{if} \tag{2.9a}$$

$$\omega_k^{ib} = (1-\alpha)\omega_{k-1}^{ib} + \alpha \tag{2.9b}$$

or

$$\omega_k^{if} = (1-\alpha)\omega_{k-1}^{if} + \alpha \tag{2.10a}$$

$$\omega_k^{ib} = (1-\alpha)\omega_{k-1}^{ib} \tag{2.10b}$$

depending upon which type of neuron has the maximum response, where ω_k^{ib} is the weight of the connection between i^{th} pattern neuron and the background summation neuron and ω_k^{if} is the weight of the connection between i^{th} pattern neuron and the foreground sum-

mation neuron and α is the mixing weight. This approach is also limited in its capability to consistently detect objects when they become static.

The aforementioned foreground segmentation methods cannot deal with sudden changes such as on/off of the light. A way to solve such a situation is to consider the pixel intensity variations as discrete states corresponding to certain events. This involves training Hidden Markov Models (HMM) [46] and then selecting the model corresponding to a certain event [47]. The advantage of this is that different states of the system, which the unsupervised background modelling approaches cannot handle, can be learned during training. Another method that requires prior training is based on eigenvalue decomposition [48]. Principal component analysis (PCA) is applied to the sequence of images to compute the eigenbackgrounds. At each time k the image I_k is projected onto the eigenvector sub-space and then reconstructed. The reconstructed image serves as the background image at time k. The foreground subtraction is then performed as in Eq. 2.1. A similar approach can be to sample background values at each pixel into a codebook [39] which represents a compressed form of the background model for long sequences. This approach can encode variations in the background model without having huge memory constraints. However, its performance is highly dependent on the codebook generation during the training phase. In [49, 50] incremental PCA decomposition over a sequence of non-overlapping spatio-temporal blocks is performed to detect motion. In this method the image is segmented into patches of dimension $8 \times 8 \times w$ where w is the temporal window size. These patches are disjoint in space but overlap in time and are concatenated and reshaped to form a 2D background map. The Haar wavelet transform is then applied at each location (i, j) on the background map $I_k^{'}(i, j)$ over windows of size 3×3 to obtain L corresponding wavelet coefficients, $c_o(l), l = 1, \cdots, L$. The dissimilarity between wavelet coefficients of each patch $p(i, j, m)$ and $p(i, j, n)$ is computed as

$$\zeta(p(i,j,m), p(i,j,n)) = \sqrt{(\sum_{l=1}^{L}(c_m(l) - c_n(l))^2)}. \tag{2.11}$$

Finally, a threshold is applied to this dissimilarity matrix to classify changed and unchanged blocks [51].

These solutions are mainly used to detect moving objects in the scene. A major problem with background-based detection algorithms is the difficulty of dealing with object interactions, such as object proximity and occlusions. In such cases, multiple objects that are close are likely to generate a merged foreground region that produces a single detection

(a) (b)

Figure 2.4: Sample background-based and model-based detection results. (a) Outdoor test sequence and (b) corresponding background-based motion segmentation result.

only, instead of multiple detections. Furthermore, rapid illumination changes result in a large number of false detections as well as affecting the segmentation of actual objects, which can result in track loss. Projection histograms can be used to split the merged objects [41], thus allowing a single blob to represent two objects [36]. However, they cannot resolve merging when an object is completely overlapping with another object. In Sec 3.2 we will propose some techniques to overcome some of these problems, particularly rapid illumination changes and object merging. A survey on image-based segmentation algorithms can be found in [52]. Figure 2.4 shows typical output from a background-based detection algorithm [10].

2.2.2 Single camera tracking

Once objects are detected, the second step aims at linking different instances of the same object over time (i.e., data association). A typical problem for data association is to disambiguate objects with similar appearance and motion. For this reason data association for object tracking can be considered as a motion correspondence problem. Several statistical and graph-based algorithms for tracking have been proposed in the literature. A significant amount of work has been reported on detecting and tracking single or multiple moving objects using Kalman filters (KF) [53,54], particle filters (PF) [5,55,56] and variants of Probabilistic Data Association (PDA) [57, 58]. Smoothing or target state estimation can be performed by initialising a Kalman filter for each target [53] and by assuming that the posterior density at every time step follows a Gaussian distribution. This limiting assumption can be alleviated by using particle filters. Tracking can be based on adaptive multi-feature tracking [5] using colour and orientation information under a particle filtering framework. Similarly, colour and edge features are used in [59] to track single targets, such as faces and hands, using a trust-region method. Multi-target tracking algorithms include Mixture particle filters (MPF) [55], where individual interacting PFs

perform distributed resampling to avoid track loss due to sample depletion. Similarly, Boosted particle filtering (BPF) [11] uses proposal distributions with a mixture model that contains contributions from a detector and the target dynamic model. In [19], the target state is augmented with an existence variable to model the number of targets in the Bayesian estimation. This leads to a hybrid estimation problem solved using a jump Markov model [60], as one component of the state vector is discrete valued, while the rest are continuous valued. Two methods based on statistics are the Joint Probabilistic Data-Association Filter [61] and Multiple Hypotheses Tracking [62]. Smoothing and clutter filtering can also be performed prior to data association using a Probability Hypothesis Density (PHD) filter [63], a Bayesian recursive method with linear complexity (with respect to the number of targets). The PHD filter approximates the multi-target statistics by propagating only the first order moments of the posterior probability. The major drawbacks of these methods are the large number of parameters that need to be tuned and the assumptions that are needed to model the state space [64].

An alternative to probabilistic methods is mean-shift (MS), a non-parametric kernel-based method used for target localisation [36]. Another alternative is to model the problem with a graph where the nodes are associated with the detections and the edges represent the likelihood that two detections in consecutive frames are generated by the same object. An example of graph-based method is Greedy Optimal Assignment [65], which requires a batch processing to deal with occlusions and detection errors, and assumes that the number of objects is constant over time. A variable number of objects is allowed when dummy nodes are introduced in the graph in order to obtain a constant number of nodes per frame [66]. A more elegant solution [64] has also been proposed where the best motion tracks are evaluated across multiple frames based on a simple motion model. Next, node linking is performed after pruning unlikely motions [64]. Data association can also be performed by matching blob contours using the Kullback-Leibler distance [67]. However, this method needs large targets to compute blob contours accurately, and the correspondence is limited to two consecutive frames. Finally two-frame bipartite graph matching can be used to track objects in aerial videos based on grey level templates and centroid positions [68]. A more comprehensive survey on tracking algorithms can be found in [69].

2.2.3 Multi camera tracking

In multi-camera *track-before-fuse* algorithms, firstly tracking is performed on each camera-view. The short tracks from each camera-view are then projected onto a common

space for fusion. The algorithms for multi-camera tracking can be categorised based on type of calibration and tracking employed. We divide the algorithms into (i) single target tracker with manual calibration, (ii) multi-target tracker with manual calibration, (iii) automatic calibration and tracking (iv) and without calibration. In this section we will separately discuss the algorithms belonging to each of these categories.

Several approaches performed multi-target tracking using manually calibrated cameras with a single target tracker in each camera. In [29] tracking is performed in each camera-view using a Bayes classifier to locate the most likely match of the target in the next frame. The likelihood is computed by finding the minimum sum of the corresponding Mahalanobis distances of the features given the estimated feature vector. The features used are 2D location, height and intensity. The target's features are projected from the current view, from where the target visibility decreases, to the view which gives maximum visibility, by applying camera-to-camera homography. The features are fused between multiple views by matching epipoles. However, selection of next best view is not a trivial task and requires analysis of content in each view [70]. Moreover, applying camera-to-camera homography requires the computation of all possible combinations of homographies and assumes overlap between views. Several approaches eliminate this limitation by computing the projection to and from a reference-view [27] or top-view [22,28]. In [22], the targets are first tracked using a particle filter in each view and then the particles are projected onto the top-view generated using manually calibrated homography. To compute the precise location of the target in the top-view, the principle axis of the target is defined in each view as the vertical line from the bottom (feet in case of person) to the top (head in case of person) of the target. These principle axes are then projected on the common-view and their intersection is used as the target feet location. The closeness of the particle to the principle axis is used as the likelihood criterion in the particle filter applied on the top-view. To improve the results on individual camera-views using top-view tracking, the particles in each view are sampled both from camera-view particles and top-view particles using homography. Similarly, in [28], multiple independent regular particle filters (MIPFs) are used to track each target in camera-view. The posterior in the other camera is computed by using measurements from all the cameras. The correspondence between two views is done using epipolar geometry. It also uses a repulsion model to resolve merging of interacting targets.

The above approaches employ single target trackers on each camera-view which makes them computationally expensive. In [21] as compared to other approaches the tracks in each view are generated using a multi-target tracker based on graph theory. In

Table 2.1: Multi-camera track-before-fuse tracking algorithms. (Key: GMPHD = Gaussian Mixture Probability Hypothesis Density; MTT = Multi-target tracker)

Refs.	**Features**	**Tracking algo.**	**Calib.**	**MTT**
[29]	2D position, height and intensity	Bayes tracker	Manual	No
[27]	2D position, size, velocity	Kalman filter	Manual	No
[28]	5D state space using ellipses	Particle filter	Manual	No
[22]	2D position, size	Particle filter	Manual	No
[21]	2D position, size, velocity	Graph matching	Manual	Yes
[26]	position, size and colour histogram	GMPHD filter	Manual	Yes
[23]	position, velocity, size and colour features	not mentioned/any	Auto	NA
[30]	field of view lines	not mentioned/any	Auto	NA
[25]	pixels, manifold learning	Caratheodory-Fejer interpolation	No	Yes

this approach, the fusion is applied on the top-view. The fusion of the multiple tracks, belonging to the corresponding objects, is first performed on the top-view using feature clustering. In the case when more than one candidate track has been selected to be fused with the selected track, all the candidate tracks are projected on the image-view for validation. Similarly, in [26], a multi-target tracker is used. This approach uses a Probability Hypothesis Density (PHD) filter for target tracking in each view as well as in the top-view. The complexity of the PHD filter increases linearly with the number of targets in comparison to other approaches. Furthermore, this approach used a Gaussian mixture based PHD (GMPHD) implementation which is faster than particle implementation as it does not requires particles for state estimation. The features used are position, size and a colour histogram. The 2D estimates of target state from each camera-view are projected onto the top-view and are used as observations for the GMPHD filter for 3D tracking in the common-view. The track labelling is performed by assigning a label to each Gaussian component. However, it is assumed that the projection from camera-view to the top-view (3D view) is available. This method also assumes that cameras are calibrated and overlapping.

The limitation of the above approaches is that they require manual computation of correspondence between multiple views. This limitation is addressed in [23] by using a trajectory correspondence model (TCM). This approach assumes that reliable tracks are pre-computed in each camera-view using one or more trackers. This tracking information from each view is used to establish correspondence between tracks, belonging to the same object in different camera-views, using position, velocity, size and colour features. The points from corresponding tracks are then used to automatically compute the homography

matrix. In [27], a least mean square search is applied to identify corresponding trajectory points in multiple views to compute the homographic mapping. This method uses 2D and 3D Kalman filters to perform tracking in the camera-view and reference-view respectively. This method also performs 3D trajectory prediction to track targets in unobserved regions between adjacent views and during occlusion. In [30] field of view (FOV) lines are estimated to disambiguate between multiple possibilities for correspondence and also to recover homography. These FOV lines are recovered automatically by observing motion in the environments. The single camera tracking is performed by two different trackers to show independence from tracking algorithm. This approach assumes that each camera should overlap with at least one other camera. The correspondence between objects in multiple views is done as they enter or exit the scene because at that moment they appear on FOV lines in other overlapping views. However, such approaches fail when the object appears from the middle of the scene, such as a person getting out of a car. Such situations are handled using the homography computed through FOV lines.

In case camera calibration information is not available, or cannot be computed efficiently, or the assumption that the world is planar is not applicable, most of the previous approaches are difficult to apply. In [25] manifold learning using Locally Linear Embedding of the data is applied to solve the multi-camera tracking problem under such conditions. The view correspondence is done by computing the embedding of the views. The tracking is performed by employing Caratheodory-Fejer (CF) interpolation theory to identify the dynamic evolution of the data on the manifolds. The limitation of this approach is that it requires prior training and assumes that multiple views are highly overlapping. A summary of the state of the art of multi-camera tracking approaches is shown in Table 2.1.

The *track-before-fuse* approaches are computationally expensive as they require frame-to-frame correspondence at each camera-view and on the common-view (reference-view, top-view). In most of these approaches not all the available information is used effectively. The alternative approach is to use *fuse-before-track* in which there is no need to apply tracking in each sensor and is discussed next.

2.3 Fuse-before-track

Tracking in individual sensors may result in error-prone tracks because uncertainty and fusion of this noisy data may not yield optimal results. Fuse-before-track is an alternative in which the tracking step is deferred until all the information has been accumulated on a reference-view [71–74]. In this case, instead of tracks, the object localisation

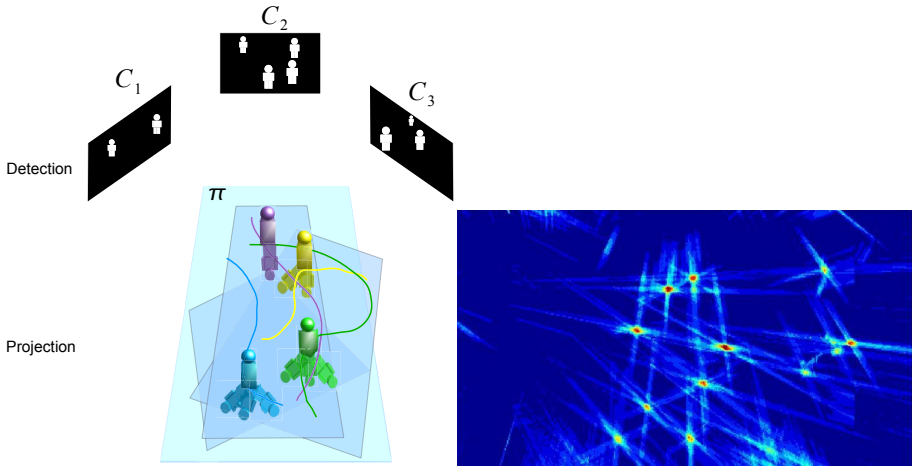


Figure 2.5: Sample *fuse-before-track* intermediate results on the top-view. (a) Illustration of the fuse-before-track approach. (b) Detection volume obtained by projection and fusion from multiple camera-views. (C_1: camera 1; C_2: camera 2 and C_3: camera 3).

information is fused and tracking is applied afterwards. This approach is particularly useful in case of multi-modal sensor networks where availability of complementary information can significantly improve the results. Figure 2.5 shows the typical intermediate result of *fuse-before-track*, where detections from multiple cameras are projected and fused on the top-view to perform tracking.

In this section, we will first introduce the multi-camera *fuse-before-track* algorithm. Since fusion preceded by tracking is more effective in the case of multiple modalities, we will also introduce the audio localisation in this section. Finally, we discuss the state of the art on audiovisual multi-modal tracking approaches.

2.3.1 Multi camera tracking

In recent years a new paradigm for multi-camera tracking has been proposed in which, firstly, information from all cameras is fused and then tracking in performed [71–74] (Fig. 2.7). In [74], similar to [22], the vertical axes of the target across views are mapped on the top-view plane and their intersection point on the ground is computed to obtain the feet location of the target (Fig 2.6). The vertical axis in each view is obtained by least mean squares fitting, minimising the perpendicular distances between the pixels and the axis. The projection from each view to top-view is performed using pre-computed planar homography matrices. These top-view feet locations are then tracked using a particle filter. Projecting only feet locations make this approach very sensitive to detection errors

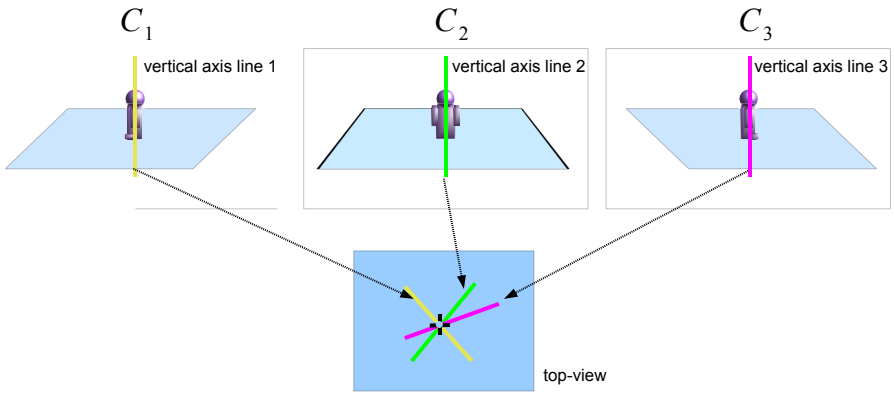


Figure 2.6: Illustration showing intersection of vertical axis of target on top-view.

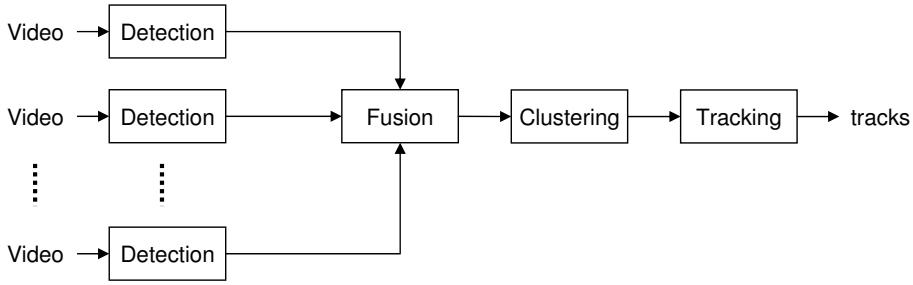


Figure 2.7: Generic block diagram of video only *fuse-before-track* detection and tracking algorithms.

in camera-views and inapplicable in crowded scenarios where feet locations may not be visible. To avoid this problem, most approaches project and fuse information from entire segmented foreground regions. In [72] the foreground mask from each camera-view is projected onto the top-view to obtain an occupancy map. This occupancy map uses colour and motion information in a generative model which explicitly handles complex occlusions and interactions between individuals. The tracking of each object is performed using the Viterbi algorithm. The Greedy approach, that makes the locally optimal choice at each stage with the hope of finding the global optimum, is used to avoid the combinatorial explosion due to joint posteriors. Contrary to most of the other tracking methods that perform state estimation using frame-to-frame correspondence only, this method computes global optima of scores summed over many frames. This makes it more robust against persistent and prolonged occlusion. However, this approach can only process a batch of N frames at a time and hence the results are delayed which make it unsuitable for real-time applications.

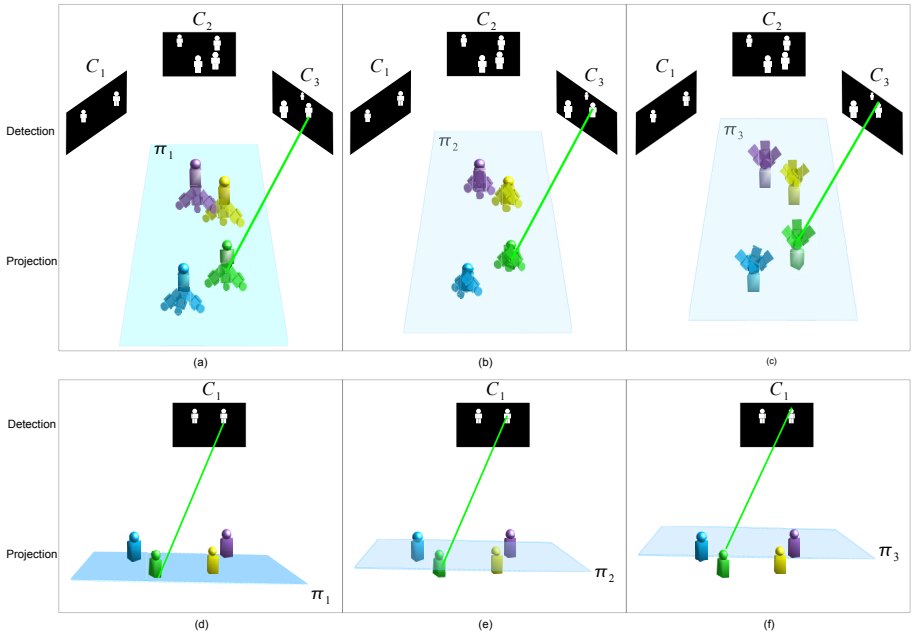


Figure 2.8: Illustration showing projection of detections from camera-views to 3 parallel planes on top-view. (a,d) Projection to a ground plane. (b,e) Projection to a mid-level plane. (c,f) Projection to a head-level plane.

To further improve the effectiveness of tracking in the fused domain, multi-level homography is proposed in [75]. In [73] three homography planes are used, one at feet level, one at head level while the third plane is in between these two planes (Fig 2.8). The homography for these planes are computed using manually selected control points. This method also assumes that cameras are highly overlapping. The fusion between multiple-views is performed by projecting the intensities in the foreground regions of each view (obtained through background subtraction) and computing the variance of these intensities at each point in the top-view. The low-variance indicates higher probability of the presence of target heads. Firstly, head detection is performed by thresholding the variance map and by employing floor level homographic projections. Finally the candidate head-top positions are estimated by clustering using double threshold hysteresis. K-means clustering is than applied for splitting of merged blobs. Tracking is performed by applying prediction on the candidate head locations. This approach requires the person to be fully inside the camera-view, calibration information to be available, and the cameras used to be mounted at a significant height from where full heads are easily visible. The detection and tracking procedure also depends upon various thresholds. Although this method was shown to work well on the sequences on which it was demonstrated, its application in various other camera configurations may not be possible. The multi-level homography approach proposed in [75]

Table 2.2: Multi-camera fuse-before-track tracking algorithms

Refs.	**Features**	**Tracking algo.**
[74]	person vertical axis, ground position	Particle filter
[73]	head position	Bayes tracker
[72]	colour and motion	Viterbi algorithm
[71]	multiple planes occupancy map	Minimum graph cut

is further extended in [71]. The correspondence between multiple views is performed automatically using SIFT features followed by RANSAC to reject outliers. This procedure gives a planar homography which is then used to compute multi-level homographies by moving along the vertical vanishing points to have projection planes parallel to the planar top-view. The foreground likelihood probabilities from each plane of each view at each time are projected onto the corresponding plane of the common-view to obtain a 4D spatio-temporal occupancy map. Graph-cut trajectory segmentation is then applied on this 4D spatio temporal data to estimate tracks for each individual target using the minimum cut algorithm. Although this approach shows promising results, it is computationally expensive and cannot work in real-time as it requires a 4D occupancy map to be created before applying the minimum cut procedure. The summary of the state of the art multi-camera tracking approaches is shown in Table 2.2.

The drawback of most of the *fuse-before-track* approaches is that projection of complete segmentation information from each view to common-view requires more computation as compared to projection of particles or tracks only. Furthermore, they perform detection at two steps: first in each view and then in the common-view. This makes them computationally expensive. The computation complexity due to projections can be reduced by only projecting foreground pixels [76] rather then projecting entire binary mask images. To reduce the complexity due to the additional detection step, a *track-before-detect* approach that does not require a detection step can be used. The state of the art on *track-before-detect* approaches is discussed in Sec. 2.4 whereas the details of the proposed Multi-target track-before-detect particle filtering (MT-TBD-PF) approach is discussed in Sec. 3.5.

The fusion mechanisms employed may differ from modality to modality as they generate data of different dimensions. The aforementioned *fuse-before-track* category of multi-sensor tracking can thus be extended by simply adding a fusion step per modality (Fig 2.11). The next section (Sec. 2.3.2) introduces the audio modality and discusses audio source localisation techniques. Literature on multi-modal *fuse-before-track* algorithms is reviewed in Sec. 2.3.3.

2.3.2 Audio source localisation

Localisation of a source emitting waves has been an active area of research for more than half a century [77]. It has its application in vast areas including under-water surveillance, wireless communication and border security. Recently it has been used in applications such as speaker localisation for teleconferencing, video coding and surveillance. These waves can either be electromagnetic or sound waves; however both have similar characteristics and follow the same propagation model which is the key factor in their localisation.

There are three propagation models [13] that are considered: (i) single-path model, (ii) multi-path model and (iii) reverberation model. These are explained as follows: let each target generate a sound which is received at an array of microphones. The *ideal propagation model* [13] assumes that the original signal undergoes attenuation and delay before reaching each microphone. Let $\mathbf{y}_k$ be a sound wave generated by the source at time k. The signals $\hat{\mathbf{y}}_{ik}$ received at the i^{th} microphone can be expressed as

$$\hat{\mathbf{y}}_{ik} = \Gamma_i \mathbf{y}_{k-n-f_i(\tau)} + \mathcal{N}_{ik}, \tag{2.12}$$

where, for each i^{th} microphone, Γ_i is the attenuation factor, n is the propagation time for the signal to reach the first microphone, τ is the relative delay between two consecutive microphones; $f_i(\tau)$ is the delay between the first and the i^{th} microphone in the array of N^r microphones and $\mathcal{N}_i$ is the process noise at microphone i which is assumed to be uncorrelated between the sensors.

In case of *multi-path propagation models* [78–81], the direct path signal as well as reflected versions of the signal are considered. This is based on the fact that real world environments are composed of various obstacles such as walls and furniture. In this case the recorded signal can be expressed as

$$\hat{\mathbf{y}}_{ik} = \sum_{j=1}^{N} \Gamma_{ij} \mathbf{y}_{k-n-\tau_{ij}} + \mathcal{N}_{ik}, \tag{2.13}$$

where N is the total number of paths.

The problem with the multi-path model is that it is impractical if N is large and is used in ocean surveillance where there are only three possible paths (the direct path and two reflections from surface and bottom respectively). In case of indoor scenarios each microphone receives a large number of echoes and hence N is large. A *reverberation model* is used in such scenarios. The reverberation model also considers the fact that there will

be reflections due to surroundings and can be expressed as

$$\hat{\mathbf{y}}_{ik} = h_i * \mathbf{y}_{ik} + \mathcal{N}_{ik}, \tag{2.14}$$

where h_i is the channel impulse response between the source and the i^{th} sensor and $*$ indicates convolution. It should be noted that in Eq. 2.14 there is no time delay τ, hence there is no plain solution to the problem with the reverberation model. The reverberation model requires prior knowledge about the source signal and then the impulse response for each microphone needs to be computed for which there is a closed-form solution only in the case of an indoor room environment [82]. In case of an outdoor environment, it is difficult to compute the impulse response [13]. In cases where the original source signal is not available, this method is used by approximating h_i using domain knowledge. However, it is a very challenging problem [13]. In this model, the time delay is computed by identifying the two direct paths of the sound signal to the microphone pair. However, identifying the two direct paths is a blind channel identification problem which is a hard problem particularly in indoor environments [13].

Due to the computational difficulty of multi-path propagation models and the challenging nature of reverberation models, the aforementioned ideal propagation model is widely used for source localisation. The basic method of source localisation via time difference of arrival estimation (TDOA) is cross-correlation (CC). Considering a single-path model and only 2 microphones (Fig 2.9), receiving signals $\hat{\mathbf{y}}_{1k}$ and $\hat{\mathbf{y}}_{2k}$ respectively, which satisfy

$$\hat{\mathbf{y}}_{1k} = \mathbf{y}_{k-n} + \mathcal{N}_{1k}, \tag{2.15a}$$

$$\hat{\mathbf{y}}_{2k} = \Gamma \mathbf{y}_{k-n-\tau} + \mathcal{N}_{2k}, \tag{2.15b}$$

the cross-correlation function between these two signals can be written as

$$R_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = E_k[\hat{\mathbf{y}}_{1k}\hat{\mathbf{y}}_{2k-\tau}], \tag{2.16}$$

where E_k denotes expectation across a suitable range of k values. The delay is found by maximising the cross-correlation function. However, the cross-correlation function is not enough to obtain a valid delay estimation using real audio data. In order to improve the accuracy of the delay estimation τ, it is desirable to pre-filter $\hat{\mathbf{y}}_{1k}$ and $\hat{\mathbf{y}}_{2k}$ prior to integration in Eq. 2.16. The CC method suffers from the noise present in the signal and pre-filtering is needed. The improvement over CC, referred to as the generalised cross-

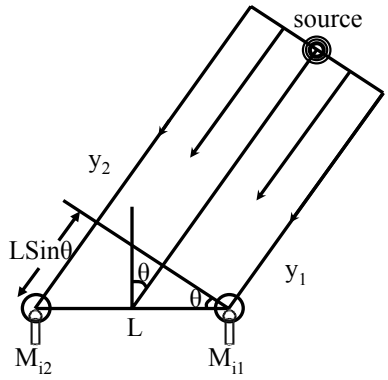


Figure 2.9: Source-receiver geometry for a STAC sensor in the far field. The distance between the microphones M_{i1} and M_{i2} is denoted by L and the arrival angle by θ. The sound wave has to travel an additional distance of $L \sin\theta$ to reach microphone M_{i2}.

correlation function (GCC), which includes pre-filtering, is defined as

$$\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \int_{-\infty}^{\infty} \Phi(f) G_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f) \exp(j2\pi f\tau) df, \tag{2.17}$$

where $G_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2} = E[Y_1(f)Y_2^*(f)]$ is the cross-power spectrum, $*$ indicates the complex conjugate operator, $Y_i(f)$ is the Fourier transform of $\hat{\mathbf{y}}_i$ and $\Phi(f)$ is a pre-filter that serves as a weighting function. In practice only an estimate $\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)$ of $G_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)$ can be obtained from finite observations. To improve the delay estimation, a general frequency weighting/pre-filtering transform $\Phi(f)$ has to be defined. Commonly used weighting functions $\Phi(f)$ include the constant weighting (in this case, the GCC becomes the frequency domain implementation of the cross-correlation defined in Eq. 2.16), the smoothed coherence transform (SCOT) [83], the Roth processor [83], the Echart filter [83], the Phase transform (PHAT) [83], and the Maximum-likelihood processor (ML) [83]. In the Roth processor the weighting function $\Phi_R(f)$ is defined as

$$\Phi_R(f) = \frac{1}{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)} \tag{2.18}$$

and

$$\hat{R}^R_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)}{|\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)|} \exp(j2\pi f\tau) df. \tag{2.19}$$

If $\mathcal{N}_i \neq 0$, as is generally the case for Eq. 2.15, then

$$\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f) = \hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f) + \hat{G}_{\mathcal{N}_1\mathcal{N}_2}(f) \tag{2.20}$$

and

$$\hat{R}^{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \delta(k_s - \tau) \int_{-\infty}^{\infty} \frac{\Gamma \hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f)}{\hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f) + \hat{G}_{\mathcal{N}_1\mathcal{N}_2}(f)} \exp(j2\pi f\tau)df, \tag{2.21}$$

where k_s is the range of time shifts until a peak is obtained. The Roth processor has the desirable effect of suppressing those frequency regions where $\hat{G}_{\mathcal{N}_1\mathcal{N}_2}(f)$ is large and $\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)$ is likely to be in error. In SCOT, the weighting function $\Phi_S(f)$ is defined as

$$\Phi_S(f) = \frac{1}{\sqrt{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_1}(f)\hat{G}_{\hat{\mathbf{y}}_2\hat{\mathbf{y}}_2}(f)}}. \tag{2.22}$$

SCOT can be considered as a pre-whitening filter and is equivalent to Roth if $\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_1}(f) = \hat{G}_{\hat{\mathbf{y}}_2\hat{\mathbf{y}}_2}(f)$. In Eckart the weighting function $\Phi_E(f)$ is defined as

$$\Phi_E(f) = \frac{\Gamma \hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f)}{\hat{G}_{\mathcal{N}_1\mathcal{N}_1}(f)\hat{G}_{\mathcal{N}_2\mathcal{N}_2}(f)}. \tag{2.23}$$

Eckart, similar to SCOT, suppresses frequency bands with high amount of noise. The PHAT transform has the same weighting function $\Phi_P(f)$ as in the case of Roth; however, when noise is uncorrelated, $\Phi_P(f)$ becomes

$$\Phi_P(f) = \frac{1}{|\Gamma \hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f)|}. \tag{2.24}$$

Ideally, when $\hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f) = G_{\mathbf{y}_1\mathbf{y}_2}(f)$,

$$\frac{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)}{|\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)|} = \exp(j\phi(f)) = \exp(j2\pi f\tau) \tag{2.25}$$

has unit magnitude and

$$\hat{R}^{P}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \delta(k - \tau), \tag{2.26}$$

where $\phi(f)$ is the signal phase. Hence only phase is preserved which is described in $\exp(j2\pi f\tau)$. The phase transform (PHAT) is an ad-hoc technique to pre-whiten the signals before computing the cross-correlations in order to get a sharp peak. The time delay information is present in the phases of the various frequencies and these are not modified by the weighting function $\Phi(f)$. The weighting function tends to enhance the true delay and suppresses all spurious delays. In a real situation, this property demonstrates a low sensitivity with respect to drawbacks due to reverberation and multi-path distortion.

To further improve the time delay τ estimation, generalisation of the cross-correlation function is introduced in [84]. This approach models the reverberation com-

ponent as additive noise and tries to filter it. The reverberation as additive noise can be expressed by extending Eq. 2.15 as

$$\hat{\mathbf{y}}_{1k} = \mathbf{y}_{k+n} + h_1^r * \mathbf{y}_{k+n} + \mathcal{N}_{1k}, \tag{2.27a}$$

$$\hat{\mathbf{y}}_{2k} = \Gamma \mathbf{y}_{k+n+\tau} + h_2^r * \mathbf{y}_{k+n} + \mathcal{N}_{2k}, \tag{2.27b}$$

where h_i^r is representative of the environment and describes the additive effect of reverberation. For example, h_i^r can be considered as a sum of shifted Dirac deltas describing the time delays between the reverberation components and the original signal. Assuming the two reverberation transfer functions in Eq. 2.27 have the same power spectrum $|H^r(f)|^2$, the overall noise power spectrum component $|\mathcal{N}^{'}(f)|^2$ can be represented as

$$|\mathcal{N}^{'}(f)|^2 = |H^r(f)|^2 |Y(f)|^2 + |\mathcal{N}(f)|^2. \tag{2.28}$$

Taking into account this additive noise component, the optimum cross-correlation estimator from Eq. 2.19 can be written as

$$\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)}{|H^r(f)|^2 |Y(f)|^2 + |\mathcal{N}(f)|^2} \exp(j2\pi f\tau) df. \tag{2.29}$$

Assuming that the reverberant energy is proportional to the direct signal energy; the following approximations can be obtained

$$|H^r(f)|^2 |Y(f)|^2 \propto (\hat{G}_{\mathbf{y}_1\mathbf{y}_2}(f) - |\mathcal{N}(f)|^2), \tag{2.30}$$

where the parameter γ satisfies $0 < \gamma < 1$. Then, Eq. 2.28 can be re-written as

$$|\mathcal{N}^{'}(f)|^2 = \gamma \hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f) + (1-\gamma)|\mathcal{N}(f)|^2 \tag{2.31}$$

and the optimum cross-correlation estimator can be expressed in terms of γ as

$$\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)}{\gamma \hat{G}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f) + (1-\gamma)|\mathcal{N}(f)|^2} \exp(j2\pi f\tau) df. \tag{2.32}$$

And the estimated value of time delay τ is given as

$$\tau = \arg\max_{k_s} (\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(k_s)), \tag{2.33}$$

which indicates that the time shift k_s that maximises the cross-correlation is the estimated

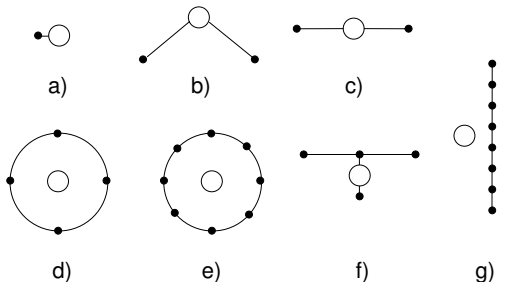


Figure 2.10: Examples of sensor configurations for audiovisual object detection and tracking (filled circles indicate microphones; empty circles indicate cameras): (a) single microphone-camera pair; (b-c) Stereo Audio and Cycloptic Vision (STAC) sensors; (d-e) circular microphone array with single camera; (f) triangular microphone array with single camera; (g) linear microphone array with single camera.

delay. Then the arrival angle, θ_i is estimated as $L_{M_i} Sin\theta = v_c\tau$ i.e., $\theta_i = \arcsin(v_c\tau / L_{M_i})$, where v_c is the speed of sound in air and L_{M_i} is the distance between the two microphones.

Despite the robustness of generalised cross-correlation phase transform (GCC-PHAT) against reverberation, a significant amount of noise is still present in the signal which can deteriorate the localisation estimate from the true target location. In Sec 4.2 we will discuss our solution to reduce the effect of noise and reverberation. The next subsection discusses the multi-modal tracking algorithms using audio and video modalities.

2.3.3 Multi-modal tracking

Localisation and object tracking using audiovisual measurements is an important module in applications such as surveillance and human-computer interaction. The effectiveness of fusing video and audio features for tracking was demonstrated in [85–87]. The success of the fusion strategy is mainly because each modality may compensate for the weaknesses of the other or can provide additional information [88], [89]. For example, a speaker identified via audio detection may trigger the camera zooming in a teleconference. The main challenges for audiovisual localisation are reverberations and background noise. Therefore, the audiovisual sensor networks (with camera and microphone arrays) have been used to address these problems using a variety of sensor configurations. Figure 2.10 shows a summary of these configurations, which range from a single microphone-camera pair to single or stereo cameras with stereo, circular arrays or linear arrays of microphones. Camera-microphone pairs are used for speaker detection in environments with limited reverberation under the assumption that the speakers face the microphone [88]; single or stereo cameras with multiple microphones are used in meeting rooms and teleconferenc-

Table 2.3: Multi-modal tracking algorithms. (Key: PF=Particle filter, KF=Kalman filter, DKF=Decentralised KF, LDA=Linear discriminant analysis, TDNN=Time delay neural networks, GPM= Graphical models, BT = Bayes tracker, MFA=Multi-feature analysis, HCI=Human computer interaction)

Refs.	**Sensor types**	**Algo.**	**Applications**
[91]	Stereo camera and circular microphone array	PF	Multi-modal user interface
[93]	2 cameras and 4 microphone arrays	PF	Indoor multiple person tracking
[87]	Camera and 10 microphone circular array	PF	Outdoor surveillance
[90]	Panoramic camera and 4 omni-microphones	MFA	Face detection
[89]	Wide-angle camera and a microphone array	I-PF	Meeting rooms
[94]	PTZ camera and 2 microphones	PF	Teleconferencing
[88]	Camera and microphone	TDNN	Lip reading, HCI
[95]	Camera and 2 microphones	GPM	Indoor environment
[96]		BT	Indoor environment
[97]		TDNN	Surveillance
[85, 98]		PF	Surveillance and teleconferencing
[99, 100] [101]	Multiple cameras and microphone arrays	KF, DKF	Smart rooms
[102]		LDA	Smart rooms
[92, 103] [104, 105]		PF	Meeting rooms

ing [90,91]. Gatica-Perez *et al.* use cameras and eight microphones to capture interactions in meeting scenarios [92]. A significant amount of work has been reported on detecting and tracking single or multiple moving objects using Kalman filters (KF) [99, 100], particle filters (PF) [87, 106] and variants of probabilistic data association (PDA) [57, 58]. Multi-modal multi-sensor configurations are used for object tracking [18, 69, 106, 107] to compensate for failure of each modality. Tracking can be performed using the video modality only [6, 64, 108–110], the audio modality only [111–114] or using audio and video simultaneously [85, 87, 90, 92, 93, 104–106]. Many approaches address audiovisual tracking for smart multi-modal meeting rooms [91, 92, 94, 103–105] where the speakers are multiple interacting meeting participants. Tracking of multiple non-simultaneous speakers is described in [93] whereas in [94, 104] the authors track a single speaker using variants

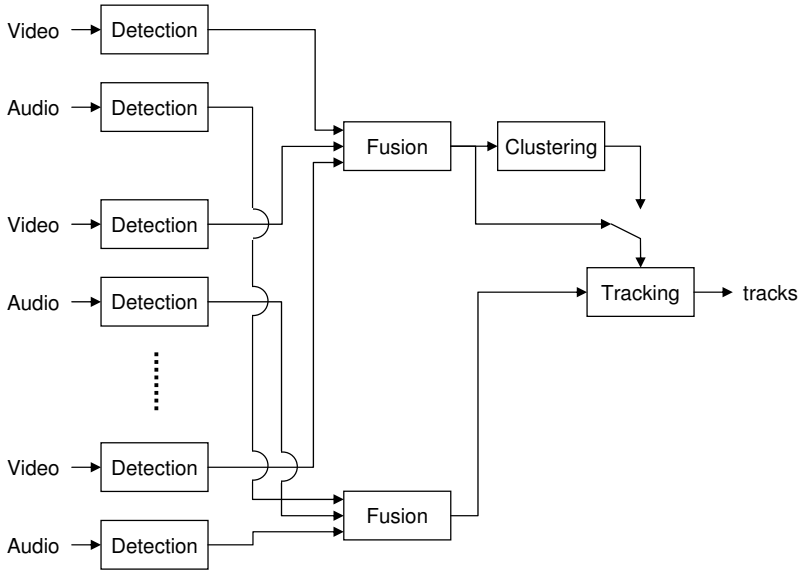


Figure 2.11: Generic block diagram of *fuse-before-track* multi-modal detection and tracking.

of the classical particle filter in smart rooms. In meeting scenarios, interaction of multiple speakers is modelled using mixed-state dynamical graph models [92, 103]. Similarly, non-simultaneous speakers can be recognised by semantic analysis of the scene using trajectories generated via an audiovisual particle filter [105]. Moving speakers can be tracked using Bayesian hidden variable sequence estimation [91]. This approach is equivalent to extending the Bayesian network to a dynamic Bayesian network in order to account for the dynamics of the state of the sound sources [91]. Face and upper body parts can be detected using contour extraction by performing edge and motion analysis and then combining with audio detection in a particle filter framework [85, 98, 115]. Gehrig *et al.* [99] apply audio detection to generate face positions that could also be observed by multiple cameras.

Unlike meeting rooms, more challenging scenarios are uncontrolled environments (e.g., indoor and outdoor surveillance) where it is not practical to use complex microphone configurations requiring sophisticated hardware for installation and synchronisation. Recently, simple configurations (e.g., one camera and two microphones) were adapted using Time-Delay Neural Networks (TDNN) and Bayesian Networks (BN) [97]. Audio features are detected by computing the spectrogram coefficients of foot-step sounds via the Short-Time Fourier Transform (STFT). TDNN is then used to fuse the audio and visual features, where the latter is generated using a modified background subtraction scheme. However, it

is unclear how object detection is achieved when visual features are unavailable. Moreover, this algorithm relies on a pre-training stage that leads to intensive processing. Furthermore, like other approaches [91, 92, 94, 103–105] this work also focuses on optimal fusion of modalities. Multi-modal localisation and tracking can be improved by accounting for both integration as well as segregation of modalities through Bayesian modelling [96]. Probabilistic reasoning for multi-modal data association can then be used to segment, associate and track multiple targets in audiovisual sequences obtained through similar sensor configurations consisting of a camera mounted between two microphones. Sensors similar to Stereo Audio and Cycloptic Vision (STAC) sensors, with a pan, tilt and zoom (PTZ) camera are used to detect speakers in the near field with unscented particle filter for data fusion [94]. When the target dynamics and measurements are linear and Gaussian, a closed-form solution can be uniquely determined. Such target dynamics can be modelled using the Kalman filter to fuse the audio and video modalities [100]. The Kalman filter cannot effectively handle non-linear and non-Gaussian models [58, 100, 106], although an extended Kalman filter can linearise models with weak non-linearities around the state estimate [58, 116]. The particle filter is a popular choice to model non-linear and non-Gaussian systems [85, 94, 98, 115]. Cevher *et al.* [87] use a particle filter to combine acoustic and video information in a single state space. They adapt the Kullback-Leibler divergence measure to decrease the probability of divergence of the individual modalities. Vermaak *et al.* [85] combine particle filter based head tracking with the acoustic time difference of arrival (TDOA) measurements to track speakers in a room. Bregonzio *et al.* [117] use colour-based change detection and TDOA for generic object tracking. In most approaches, the detection mechanism uses TDOA or beamforming [87, 111, 118–120] for audio detection. Speakers can also be detected using a recognition mechanism. In this case, Mel-Frequency Cepstral Coefficients are used for speech recognition, and video recognition can be done using linear subspace projection methods [102]. A summary of multi-modal tracking algorithms is presented in Table 2.3.

Large arrays of microphones are difficult to use in many real world scenarios such as wide-area surveillance. In this area, this thesis presents a method for performing detection and tracking of targets using a STAC sensor (Chapter 4). A STAC sensor, composed of a single camera mounted between two microphones (Fig. 2.10(b-c)), makes the designed system simpler, cheaper and portable. STAC sensors are used to perform audiovisual tracking with a probabilistic graph model and fusion by linear mapping [95] or with particle filtering [98]. The cost of using such a simple sensor against an array of microphones is its sensitivity to noise and reverberations. Since STAC sensors are sensitive

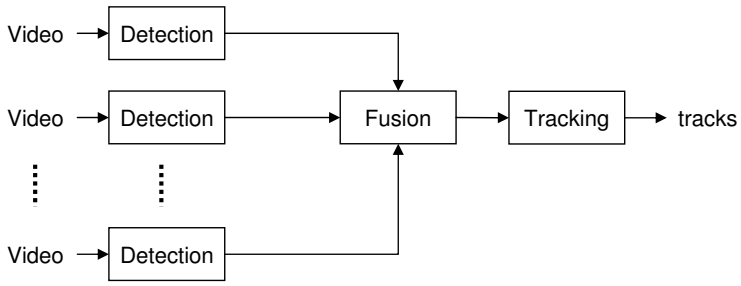


Figure 2.12: Generic block diagram of multi-sensor *fuse-before-track* with *track-before-detect* (Note: there is no detection/clustering step after fusion).

to reverberations, the proposed approach applies multi-band analysis and precedence effect to filter the signal (Sec. 4.2). This thesis also proposes a fusion strategy based on the Weighted Probabilistic Data Association filter (WPDA), which associates the hypotheses and the measurements with a real target (Sec. 4.3 and Sec. 4.4).

2.4 Track-before-detect

The drawback of the aforementioned detection and tracking algorithms (Sec. 2.2.1 and Sec. 2.2.2) is the thresholding of data, this makes them inapplicable for tracking targets with low observability. In many single and multi-sensor applications the Signal-to-Noise Ratio (SNR) of the input or pre-processed signal is relatively low. Examples of such signals are the far-field of infrared (IR) images, bearing frequency distributions (sonar) and range-Doppler maps (radar). Examples of sensors whose signals are fused include cameras [75], microphones [121] and radars [122]. Fusion involves triangulation of noisy information that can result in much larger number of solutions than desired. To address this type of data, simultaneous detection and tracking can be performed via the *track-before-detect* (TBD) approach. In TBD the entire input signal is considered as a measurement. This measurement is a highly non-linear function of the target state and can be solved either by discretisation of the state [123] or by employing non-linear state estimation techniques such as particle filtering [56], which are computationally less expensive.

A recursive Bayesian single target TBD is proposed in [124] using particle filtering. This method assumes a point target and extends the target state with the signal intensity based on the assumption that the return intensity from the target is unknown. Similar to multi-target PF, multi-target TBD-PF approaches are also based on extending the target state with an existence variable and solved with a jump Markov model [125]. An

approach based on dynamic programming is used in [126] to track aircraft through TBD. In this context, conventional change-based detection cannot be applied because targets are very small and the presence of clouds makes them dimmer. In [121] multiple microphones were used to track multiple speakers using TBD on steered beamforming results. In this approach a conditional probability density is used that characterises uncertainty in both target state and target number, given the measurements. The polar Hough transform is used in the fusion between multiple radar signals [122]. As the co-ordinate measurement errors (range, azimuth) degrade the accumulation of a signal in each cell of the Hough space (i.e., reduce the output SNR and the output signal peak, while increasing the output side lobes peak), TBD is applied for target tracking.

Most TBD algorithms have been demonstrated on simulated data [122, 124, 125, 127]. Two exceptions are [121, 126]: [121] is a multi-target multi-sensor tracking algorithm applied on audio sensors and [126] is applied to IR sequences from a single sensor only. To the best of our knowledge, work proposed in Sec. 3.5 of this thesis is the first adaptation of the TBD concepts to multi-camera tracking.

2.5 Summary

This chapter has reviewed the literature on autonomous multi-target, multi-modal detection and tracking. The algorithms can be categorised as simply detection and tracking techniques in the case of a single sensor; detection, tracking and fusion in the case of multiple sensors (*track-before-fuse*); and detection, fusion and tracking also in the case of multiple sensors (*fuse-before-track* and *track-before-detect*) where there is the same number of fusion blocks as modalities.

The chapter has provided a literature review of all these categories of algorithms for two modalities, namely video (Sec. 2.2.1) and audio (Sec. 2.3.2). The detection in the case of the video modality can be classifier-based or based on the background model used. The classifier-based approaches are appropriate when the only prior knowledge is the class of objects that need to be detected (such as faces or pedestrians). On the other hand, background model based techniques can segment all kinds of objects. However, they are highly sensitive to deviation in the background model, such as due to rapid illumination changes, moving vegetation and various environmental effects. These can result in a significant amount of noise and clutter in the detections. They are also sensitive to shadows and object merging. Despite these limitations, they are preferred as they can be applied without any prior training to detect any class of object, as long as the object

is not part of the background model. Section 3.2 will discuss the proposed methodology to overcome some of these limitations.

Similarly, information obtained from audio modality is also highly affected by the presence of noise such as due to reverberation and background noise. The noise in the case of the audio modality is highly related to signal propagation characteristics (such as absorption, reflection, superposition and attenuation). Three models, namely the ideal-propagation modal, the multi-path propagation model and the reverberation model, have been discussed in this chapter, and the multi-path model was rejected due to its impractical nature in finding the solution. A literature review on approaches related to time-difference of arrival using cross-correlation techniques has been presented. To reduce the effect of noise and reverberation we will extend these techniques in Sec. 4.2.

Next, tracking algorithms based on these detections were discussed. These algorithms were discussed separately for a single camera (Sec. 2.2.2), multiple cameras (Sec. 2.2.3 and Sec. 2.3.1) and multiple modalities (Sec. 2.3.3), as these techniques differ significantly based on how fusion is applied. In the case of single camera tracking, discussion on approaches based on Markov models and those based on graph theory have been presented in this chapter. The multi-camera tracking state of the art on the other hand has been categorised based on the fusion step within the algorithms. Tracking strategies can be based on detections only or the approach of simultaneous detection and tracking (*track-before-detect*). The drawback of the detection-based algorithms is the thresholding of information at the initial stage, which results in data loss and computational load due to the detection phase. *Track-before-detect* approaches are inherently immune to these issues as they perform simultaneous detection and tracking by totally bypassing the detection phase. The track-before-detect approaches do not rely on detections; instead they tend to track only the targets that follow a certain motion model. This makes them useful for tracking targets in the presence of noise. To track targets with different motion models a bank of track-before-detect filters can be used. Section 3.5 discusses a proposed extension for multi-target, multi-sensor track-before-detect algorithms. Finally, the state of the art on audiovisual detection and tracking algorithms has been discussed in Sec. 2.3.3. Section 4.3 and Sec. 4.4 will present a proposed approach for this problem.

Chapter 3

Uni-modal tracking

3.1 Introduction

As discussed in Chapter 2, tracking strategies can be based on detections only or on simultaneous detection and tracking approaches. In this chapter we discuss both these categories of tracking algorithms and apply them in various scenarios, for multi-target tracking. We will limit our discussion to the single modality only, namely video.

The organisation of the chapter is as follows: in Sec. 3.2 we discuss the improvements to overcome some of the limitations of image-based localisation of targets. Multi-target tracking on these detections is discussed in Sec. 3.3. Multi-target multi-sensor fusion is discussed in Sec. 3.4. Multi-target multi-camera track-before-detect is then explained in Sec. 3.5 followed by the results and evaluation in Sec. 3.6. Finally, the chapter is summarised in Sec. 3.7.

3.2 Image-based localisation

Let $X_k = \{\mathbf{x}_k^1, \mathbf{x}_k^2, \cdots, \mathbf{x}_k^{N^{\mathbf{x}}}\}$ be the set of $N^{\mathbf{x}}$ detected targets at time k. Ideally X_k should only contain the position of each target in the scene. In practice, however depending upon the type of detection mechanism employed, it may contain anomalies such as: (i) false detections; (ii) $\mathbf{x}_k^i$ may represent the position of more than one object (due to object merging); and (iii) some targets may not be present in X_k at all (missed detections). These anomalies are due to the type of sensor, the nature of the detection algorithm and several environmental and physical conditions. In this section we discuss the localisation strategy employed in this work for the vision sensor and the proposed improvements in pre-processing and post-processing steps to improve the results. Particularly, we improve

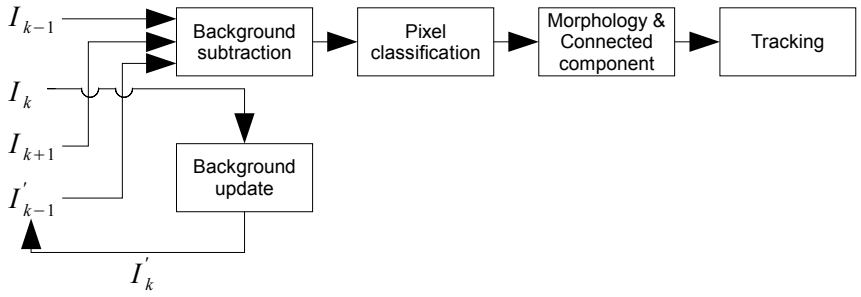


Figure 3.1: Block diagram of image-based localisation algorithm.

background estimation (Sec. 3.2.1) to segment temporarily stopped objects. We also improve motion-based segmentation (Sec. 3.2.2) by filtering false detections due to rapid illumination changes and splitting merged objects. Figure 3.1 shows the high-level block diagram of the proposed localisation algorithm.

3.2.1 Background estimation

We employ a background modelling based on a foreground extraction approach, where each frame I_k is subtracted from a reference background model I'_{k-1} to obtain foreground pixels $I^f_k = |I'_{k-1} - I_k|$. For *background modelling*, we perform the adaptive update of the reference frame using Eq. 2.2. One of the problems associated with the change-based object extraction modules is that it segments overlapping objects as one single object (Fig. 3.2(a-c)). We use a multiple layer background subtraction approach [128] to segment moving objects that are overlapping with stopped objects. Unlike [128], we detect stopped objects (using tracking information) as a whole rather than detecting stopped pixels. The motivation behind object-level (high-level) processing as opposed to pixel-level (low-level) processing is to avoid the introduction of noise and the segmentation of partial objects. The pixels belonging to the stopped objects are copied onto the background frame at the corresponding locations. This process allows us to create an additional reference frame that contains stopped objects as part of the background (Fig. 3.2(d)). The foreground extraction is then performed with both background frames at each time k until all the stopped objects start moving again. Figure 3.2 shows the sample result, with and without layered background.

The adaptive background update helps in reducing false detection due to slow illumination changes. However, rapid illumination changes can still generate a large number of false detections. In the next section we will address this problem together with other pre-processing and post-processing steps to improve the results.

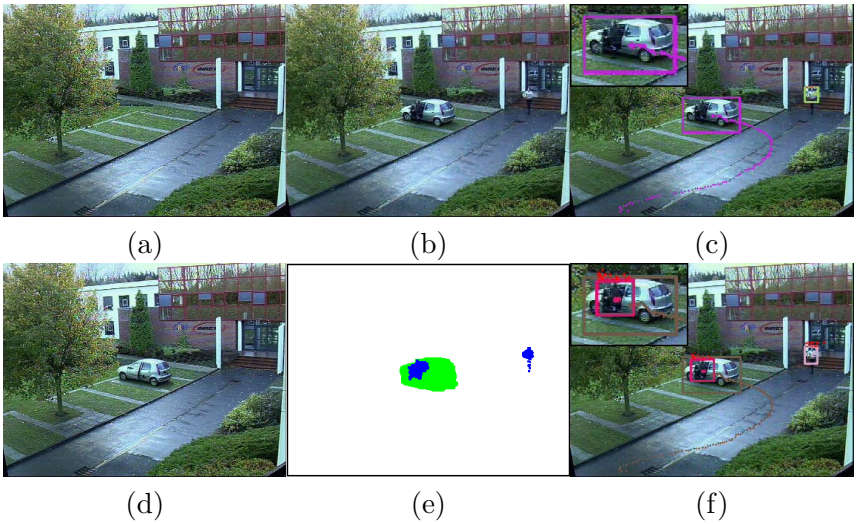


Figure 3.2: Comparison of object extraction results with and without layered background subtraction. (a) Reference background; (b) current frame; (c) segmentation without layered background (box on image); (d) layered background; (e) mask (colour-coded in green) using background frame and mask (colour-coded in blue) using layered background frame; (f) segmentation with layered background.

3.2.2 Motion-based segmentation

The frame-differencing-based foreground segmentation poses several challenges for a correct segmentation of objects. These include global and local illumination changes and object merging and need to be addressed separately. Rapidly changing illumination conditions can lead to a situation where most of the pixels are classified as foreground pixels. This results in large number of false positive detections, especially in regions in the shade of buildings or trees. Results from the frame-differencing shows that these illumination changes generate positive values. Although these differences, belonging to false detections, are of low magnitude as compared to actual foreground objects (Fig. 3.3(c)), they may not be filtered during the pixel classification process depending upon the classification parameters.

We propose the adjustment of pixel differences such that the lower values are further minimised whereas high difference values are further increased. We assume that there is no illumination change between the initial reference background $I_0^{'}$ and first frame I_0 of the sequence.

The difference image not suffering from rapid illumination changes will have a higher variance whereas those suffering from rapid illumination changes will have gradual changes in pixel values resulting in lower variance. The variance of such difference images is thus increased by adjusting the brightness and contrast values (*contrast enhancement*).

(a) (b) (c) (d)

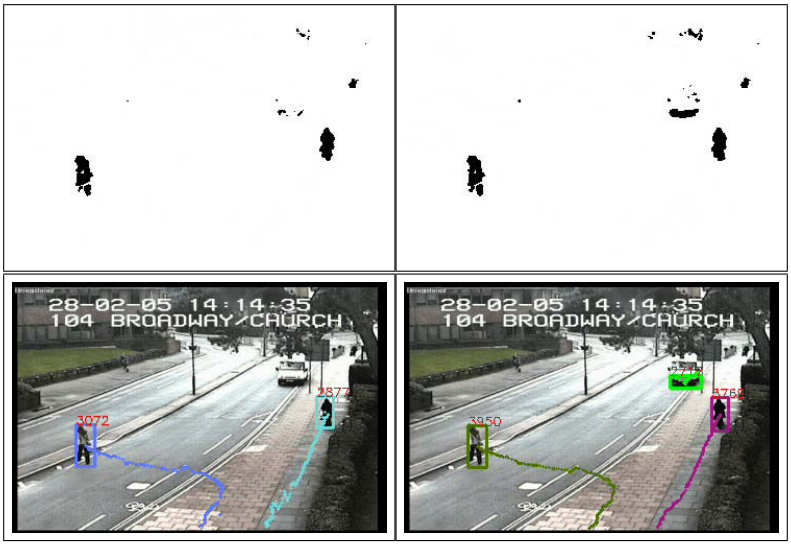
Figure 3.3: Contrast enhancement for improving object detection. (a) Reference frame; (b) current frame; (c) image difference before contrast enhancement; (d) image difference after contrast enhancement.

Let $\beta = 100$ and $\zeta_0 = 100$ be the empirically defined brightness and the initial contrast, respectively; and let $\sigma_k^2 = \sigma^2(\mid I_k^{'} - I_k \mid)$. The contrast of the current difference image is modified at each iteration i using $\zeta_i = \zeta_{i-1} \pm n$ until the condition $\sigma_{ik}^2 > \sigma_0^2$ is satisfied, where n is a fixed step size and σ_0^2 is variance of $I_0^f = \mid I_0^{'} - I_0 \mid$, i.e., the difference between the reference and the first frame. The pixel values $I_k^{fi}(x,y)$ in each channel of the difference image are modified, for an 8-bit image, according to

$$I_k^{fi}(x,y) = \begin{cases} 0 & \text{if } a_i \times I_k^{fi}(x,y) + b_i < 0 \\ 255 & \text{if } a_i \times I_k^{fi}(x,y) + b_i > 255 \\ a_i \times I_k^{fi}(x,y) + b_i & \text{otherwise} \end{cases}, \tag{3.1}$$

where $I_k^{fi}(x,y) \in [1, 255]$ is the pixel value, $a_i = \frac{1}{1 - w \times \Delta_i}$, $b_i = a_i \times (\beta - \Delta_i)$, $w = 2/255$ and $\Delta_i = \frac{\zeta_i}{w \times \zeta_0}$. Figure 3.3(d) shows a sample frame with increased contrast.

The disadvantage of using the contrast adjustment is that it further reduces the chances of segmenting objects with appearances similar to the background. Furthermore, slow moving objects can be erroneously classified into the background whereas sudden local illumination changes in specific regions of the scene can still generate false detections. To address these problems, we perform *Edge Analysis* (EA), which enhances the difference image obtained after background subtraction using an edge detector. An edge-based post-processing is performed using selective morphology that filters out misclassified foreground regions by dilating strong foreground edges and eroding weak foreground edges. The dilation of strong foreground edges enables detection of stopped edges whereas erosion of weak foreground edges helps in eliminating pixels that are segmented due to local illuminations such as due to vehicle headlights. In our implementation we compute the edges by taking the difference between consecutive frames. To further reduce the effect on the object detector of short-term illumination variations we use a *spatio-temporal filtering*

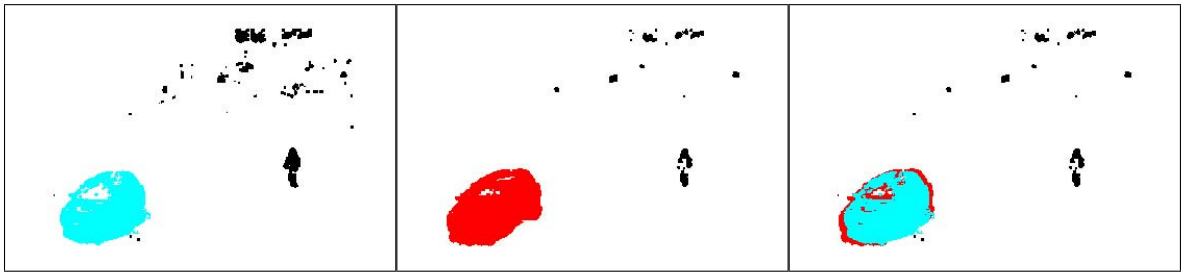


(a) (b)

Figure 3.4: Sample results with and without change detection enhanced by edge analysis (EA). (Row 1) Change detection mask (Row 2) Detection and tracking result. (a) Without edge analysis; (b) with edge analysis.

(STF) on the result of the frame differences between consecutive frames. In STF, for each pixel value an n-frame temporal window is used, centred at the current frame. The median value within each of these temporal windows is selected to smooth the output using past and future information. These edges are then added into the difference image by taking the weighted average using equal weights. Figure 3.4 shows an example of a correct detection of a vehicle despite it having stopped moving, by applying EA and STF. The price to pay for these correct detections is an artificial enlargement of the blobs produced by fast moving objects (Fig. 3.5).

The foreground-background *pixel classification* is performed by classifying pixels that are unchanged or that have changed only due to sensor noise as background and classifying the remaining pixels as foreground. This method checks the hypothesis H_0 that $I_k^f(x,y) \neq 0$ because of the camera noise as opposed to other factors like moving object or illumination changes. Based on this hypothesis, H_0, the conditional probability density function $f(I_k^f(x,y)|H_0)$ is obtained using Eq. 2.5. The noise amplitude is experimentally estimated for each colour channel which is then applied to the entire sequence. To account for camera perspective and to preserve small blobs associated with objects in regions far from the camera (top of the frame), we empirically adapt σ according to the spatial location. Figure 3.6 shows sample detection results: the high values of σ do not allow the detection of the small pedestrians on similar background, whereas with $\sigma = 0.8$

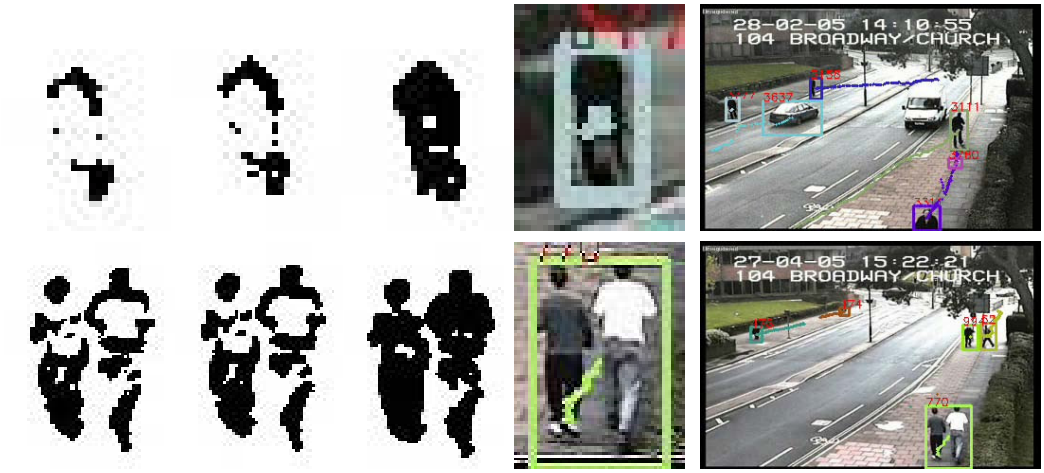


(a) (b) (c)

Figure 3.5: Comparison of background subtraction results obtained with and without edge analysis. (a) Without edge analysis the results contain a large number of spurious blobs. (b) With edge analysis the spurious blobs are partially removed; however this generates holes in the pedestrian and an enlarged mask for the vehicle. (c) Superimposed result showing the extra pixels (halo) around the vehicle.

the classification of most of the pixels belonging to the object is correct. Next, *morphological* operations, namely erosion and dilation, are performed to further eliminate any isolated noise [129]. First the entire binary image is dilated twice using a 3×3 rectangular structuring element. This allows merging of multiple blobs belonging to a single target. Erosion is then applied twice, again using 3×3 rectangular structuring element. The erosion eliminates or reduces the size of blobs that are generated due to noise, some of which may be filtered later by applying a threshold on the minimum allowed blob size. The erosion does not nullify the effect of dilation as it is applied only on the boundary of the blobs, hence multiple blobs belonging to the same target that are merged into a single blob remain merged. Finally, 8-neighbour *connected components analysis* is performed to extract the foreground objects.

Multiple objects in proximity to each other may be grouped into one blob by background subtraction based detection algorithms. In order to maintain a separate identity for these objects, a possible solution is to analyse the histograms of the pixels of a blob projected onto one of the two Cartesian co-ordinates [130]. We analysed the histogram along the horizontal axis that is computed, at each time k, as $\mathcal{H}_x = \sum_{y=1}^{H} I^f(x,y), x = \{1, \cdots, W\}$, where $\mathcal{H}_x$ is the x^{th} bin of the W-bin projection histogram $\mathcal{H}$. This solution assumes that the peaks of the histogram correspond to the different pedestrians, and these can be split by separating the modes. The modes were identified using mean-shift [131] with a kernel bandwidth of 7 pixels. An example of tracks obtained with and without splitting is shown in Fig. 3.7. The merged blobs associated to the two pedestrians on the right are constantly split by analysing the projection histograms. Figure 3.8 shows the block diagram of the proposed localisation algorithm.



(a) (b) (c) (d)

Figure 3.6: Comparison of background masks of pedestrians by changing the model of the sensor noise. (a) $\sigma = 1.2$, (b) $\sigma = 1.0$, (c) $\sigma = 0.8$. (d) Sample results. (All objects are correctly detected except the white van which is part of the background frame).

Figure 3.7: Sample results obtained with and without blob spitting using projection histograms. (Row 1) without blob splitting. (Row 2) with blob splitting.

3.3 Single camera tracking

Data association is a challenging problem due to track management issues such as appearance and disappearance of objects, occlusion, false detection due to clutter and noisy measurement. Furthermore, data association has to be verified throughout several frames to validate the correctness of the tracks.

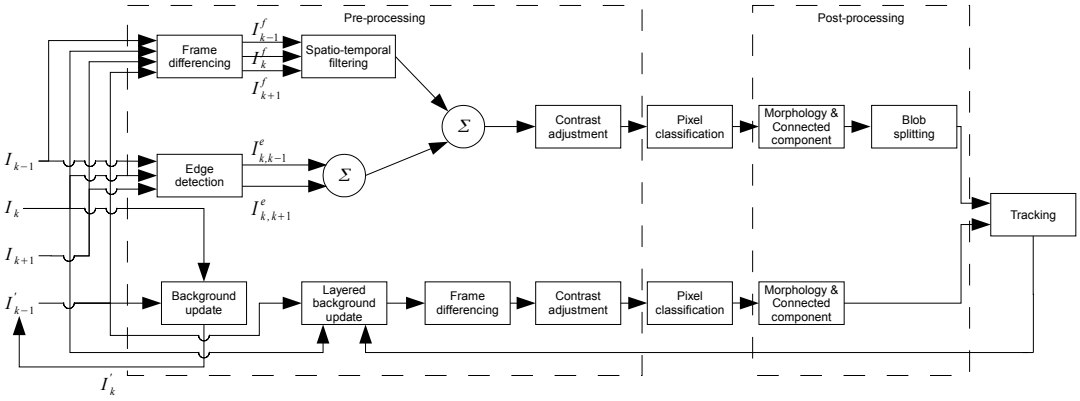


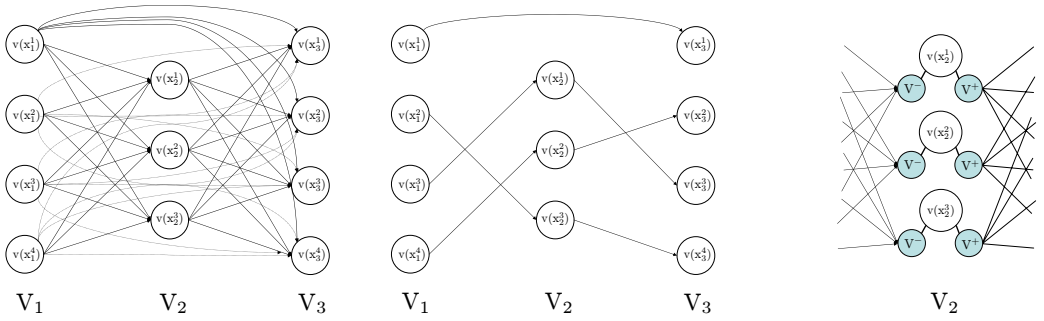
Figure 3.8: Detailed block diagram of image-based localisation algorithm showing pre- and post-processing steps.

Let us define the state of each object as

$$\mathbf{x} = (x, \dot{x}, y, \dot{y}, h, w, \mathcal{H}), \tag{3.2}$$

where (x, y) is the centre of mass of the object, $(\dot{x}, \dot{y})$ are the vertical and horizontal velocity components, (h, w) are the height and width of the bounding box, and $\mathcal{H}$ is the colour histogram. Let $\{X_1, \cdots, X_K\}$ be K sets of target detections, and $v(\mathbf{x}_i^a) \in V_i$ the set of vertices representing the detected targets at time i. Each $v(\mathbf{x}_i^a)$ belongs to D, a bi-partitioned digraph (i.e., a directional graph), such as the one shown in Fig. 3.9 (a). The candidate correspondences at different observation times are described by the gain g associated with each edge $e(v(\mathbf{x}_i^a), v(\mathbf{x}_j^b)) \in E$ that links vertices $v(\mathbf{x}_i^a)$ and $v(\mathbf{x}_j^b)$ where E is the set of edges. To obtain a bi-partitioned graph, a split of the graph $D = (V, E)$ is performed and two sets, V^+ and V^-, are created as copies of V. After splitting, each vertex becomes either a source (V^+) or a sink (V^-). Each detection $\mathbf{x}_i^a \in X_i$ is therefore represented by twin nodes $v^+(\mathbf{x}_i^a) \in V^+$ and $v^-(\mathbf{x}_i^a) \in V^-$ (Fig. 3.9 (c)). The graph is formed by iteratively creating new edges from the vertices $v^+(\mathbf{x}_i^a) \in V^+$ to the sink nodes $v^-(\mathbf{x}_K^b)$ associated with the new object observations X_K of the last frame.

Edges represent all possible track hypotheses, including missed detections and occlusions (i.e., edges between two vertices $v(\mathbf{x}_i^a)$ and $v(\mathbf{x}_j^b)$, with $j - i > 1$). The best set of tracks is computed by finding the maximum weight path cover of D, as illustrated in Fig. 3.9 (b). This step can be performed using the algorithm by Hopcroft and Karp [132] with complexity $O((N^v)^{2.5})$, where N^v is the number of vertices in D. After the maximisation procedure, a vertex without backward correspondence models a new target, and a



(a) (b) (c)

Figure 3.9: Example of digraph D for 3-frame motion correspondence. (a) The full graph. (b) A possible maximum path cover. (c) Bi-partition of some nodes of the graph.

vertex without forward correspondence models a disappeared target. The depth of the graph determines the maximum number of consecutive misdetected or occluded frames during which an object track can still be recovered. Note that despite larger values of depth allow dealing with longer term occlusions, the larger the value of depth, the higher is the probability of wrongly associating different targets.

The gain g between two vertices is computed using the information in X_i, where the elements of the set X_i are the vectors $\mathbf{x}_i^a$ defining $\mathbf{x}$, the state of the object. The velocity is computed based on the backward correspondences of the nodes. If a node has no backward correspondence (i.e., object appearance), then $\dot{x}$ and $\dot{y}$ are set to 0. The gain for each pair of nodes $\mathbf{x}_i^a, \mathbf{x}_j^b$ is computed based on the position, direction, appearance and size of a candidate target. The *position gain* g_1 based on the predicted and observed position of the point, is computed as

$$g_1(\mathbf{x}_i^a, \mathbf{x}_j^b) = 1 - \sqrt{\frac{[x_j^b - (x_i^a + \dot{x}_i^a(j-i))]^2 + [y_j^b - (y_i^a + \dot{y}_i^a(j-i))]^2}{W^2 + H^2}}, \tag{3.3}$$

where W and H are the height and width of the image, respectively. Since the gain function is dependent on the backward correspondences (i.e. the speed at the previous step), the greedy suboptimal version of the graph matching algorithm is used [64]. The *direction gain* g_2 aimed at penalising large deviations in the direction of motion (Fig. 3.10), is

$$g_2(\mathbf{x}_i^a, \mathbf{x}_j^b) = 1 - \frac{(x_j^b - x_i^a)\dot{x}_i^a(j-i) + (y_j^b - y_i^a)\dot{y}_i^a(j-i)}{W^2 + H^2}. \tag{3.4}$$

The *appearance gain* g_3 is the distance between colour histograms of objects using the

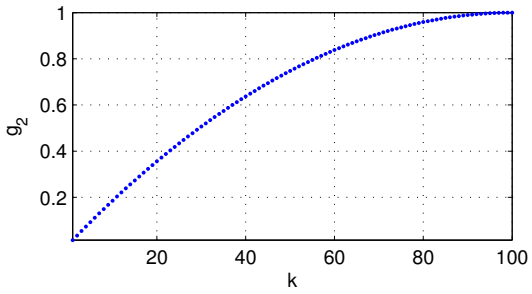


Figure 3.10: Gain g_2 computed at time $k = 100$ with target represented by node $\mathbf{x}_i^a$ has candidate positions between $(1, 1)$ to (W, H) between time 1 to 100, i.e., along the diagonal of the image, and the target represented by node $\mathbf{x}_i^b$ is at position (W, H) at time $k = 100$.

correlation method [133]:

$$g_3(\mathbf{x}_i^a, \mathbf{x}_j^b) = \frac{\sum_{n=1}^{N^{\mathcal{H}}}(\mathcal{H}'_{ia}(n) \cdot \mathcal{H}'_{jb}(n))}{\sqrt{\sum_{n=1}^{N^{\mathcal{H}}}(\mathcal{H}'_{ia}(n)^2)} \cdot \sqrt{\sum_{n=1}^{N^{\mathcal{H}}}(\mathcal{H}'_{jb}(n)^2)}}, \tag{3.5}$$

where $\mathcal{H}^{'}(n) = \mathcal{H}(n) - \frac{1}{N^{\mathcal{H}} \cdot \sum_{z=1}^{N^{\mathcal{H}}} \mathcal{H}(z)}$, and $N^{\mathcal{H}}$ is number of histogram bins.

Finally, the *size gain* g_4 is the gain computed as the absolute difference between the width and height of the objects represented by the nodes, as follows:

$$g_4(\mathbf{x}_i^a, \mathbf{x}_j^b) = 1 - \frac{1}{2}\left(\frac{\left|w_j^b - w_i^a\right|}{\max(w_j^b, w_i^a)} + \frac{\left|h_j^b - h_i^a\right|}{\max(h_j^b, h_i^a)}\right). \tag{3.6}$$

The *overall gain* g is a weighted linear combination of the position, direction, size and appearance gain, defined as

$$g(\mathbf{x}_i^a, \mathbf{x}_j^b) = \omega_1 \cdot g_1(\mathbf{x}_i^a, \mathbf{x}_j^b) + \omega_2 \cdot g_2(\mathbf{x}_i^a, \mathbf{x}_j^b) + \omega_3 \cdot g_3(\mathbf{x}_i^a, \mathbf{x}_j^b) + \omega_4 \cdot g_4(\mathbf{x}_i^a, \mathbf{x}_j^b) - (j - i - 1) \cdot \varpi \tag{3.7}$$

where $j > i$, $\sum_{i=1}^{4} \omega_i = 1$ and ϖ is a constant that penalises the choice of shorter tracks. Since graph matching links nodes based on the highest weights, two trajectory points far from each other can be connected. To overcome this problem, gating is used and an edge is created only if $g > 0$.

In the next section we will apply this tracking algorithm to the detections from the multi-camera fusion mask (*detection volume*) to perform extended tracking.

3.4 Multiple camera fusion

Let a wide-area be monitored by a set $C = \{C_1, \cdots, C_N\}$ of N cameras. To perform multi-camera tracking, the foreground from each c^{th} camera is projected onto the hypothetical top-view π (Fig. 3.11(a)). Such a projection can be performed through a projection matrix computed using corresponding points [134]. Let $\mathbf{H}_{c\pi}$ be the homographic matrix that performs projection from c^{th} camera-view to the top-view π as

$$I^f_{c\pi} = \mathbf{H}_{c\pi} I^f_c, \tag{3.8}$$

where I^f_c represents the foreground pixels in the c^{th} camera and $I^f_{c\pi}$ is the plane obtained by projections from the c^{th} camera to the top-view π. In the case of partially overlapping cameras, pixels from more than one camera can be projected onto the same pixel position on the top-view, thus on the top-view the occupancy map of foreground objects is computed as

$$I^f_\pi = \sum_{c=1}^{N} I^f_{c\pi}. \tag{3.9}$$

Each point in I^f_π is then normalised with the number of overlapping cameras in that region. The homographic matrix $\mathbf{H}_{c\pi}$ projects pixels from the ground plane in each view to the top-view. Homography from multiple planes parallel to the ground planes can be computed to perform projection on planes parallel to the top-view (Fig. 3.11(b)). Let $\mathbf{H}_{c\pi_i}$ be the homographic matrices that project pixels from the i^{th} plane in c^{th} camera-view (I^f_{ci}) to the i^{th} top-view plane (π_i) defined as

$$I^f_{\pi_i} = \sum_{c=1}^{N} \mathbf{H}_{c\pi_i} I^f_{ci}. \tag{3.10}$$

These projections onto multiple planes can be treated separately to obtain information about object shape [71] or can be collapsed to obtain a detection volume (consisting of an accumulated amount of change created by objects from each camera-view) with less noise. Figure 3.13 shows the configuration of 5 overlapping cameras, a sample view for each camera and their detection volumes on the top-view using single and multiple plane homographies.

The detection volumes shown in Fig. 3.13(g) show the objects' occupancy on the top-view. However, a segmentation mechanism needs to be employed to extract the position information for each object. The object detection can be formed by thresholding

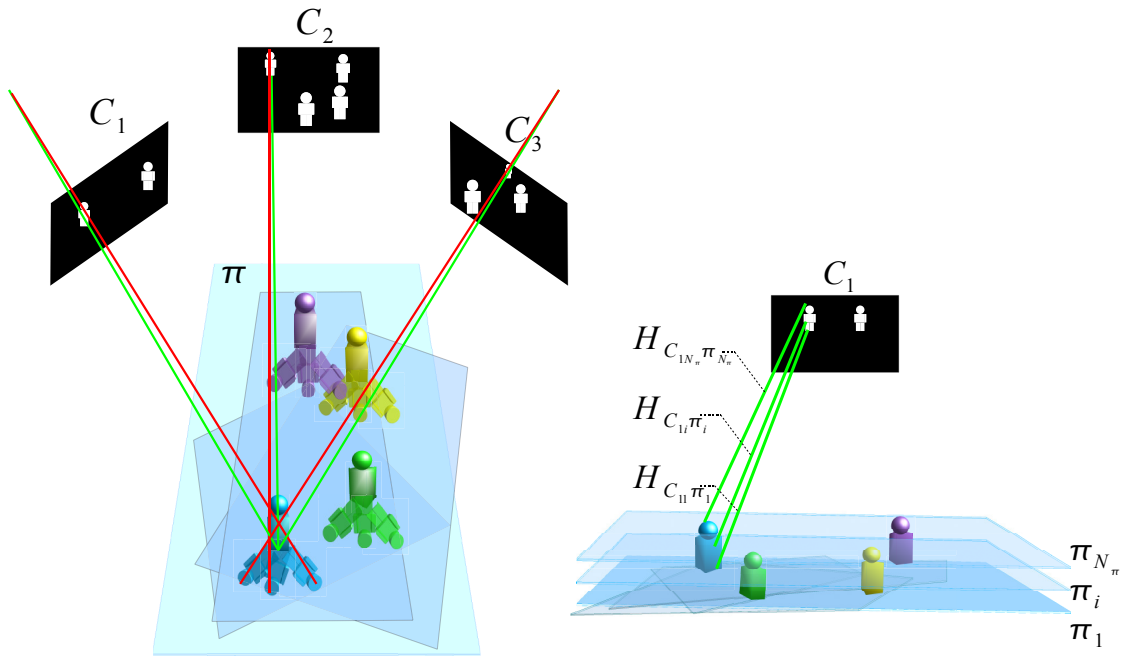


Figure 3.11: Schematic diagram showing projection of detections from multiple camera-views to top-view. (a) Projection from multiple views to a single plane. (b) Projection to multiple parallel planes.

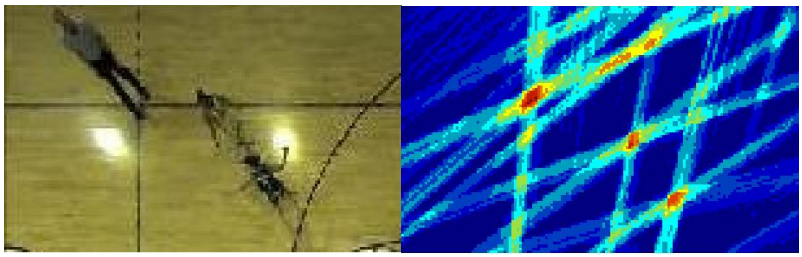


Figure 3.12: Example showing parallax error. (a) Top-view showing 3 targets. (b) Detection volume showing 3 high intensity regions due to targets and several others due to parallax error.

the detection volume to obtain a binary image followed by connected-component analysis (as in the case of a single camera) to obtain position and size for each object. This may result in multiple objects being detected as a single merged object due to their physical closeness in the scene as well as due to camera parallax (Fig. 3.12). Contrary to the binary mask on image plane, the occupancy of objects in the detection volume is not the same at each pixel position. Instead, there are more overlapping pixels at object centres and this overlap decreases as parallax increases. Hence each object may have a distinguished peak. Clustering techniques such as mean-shift [131] and fuzzy clustering [135] can be applied

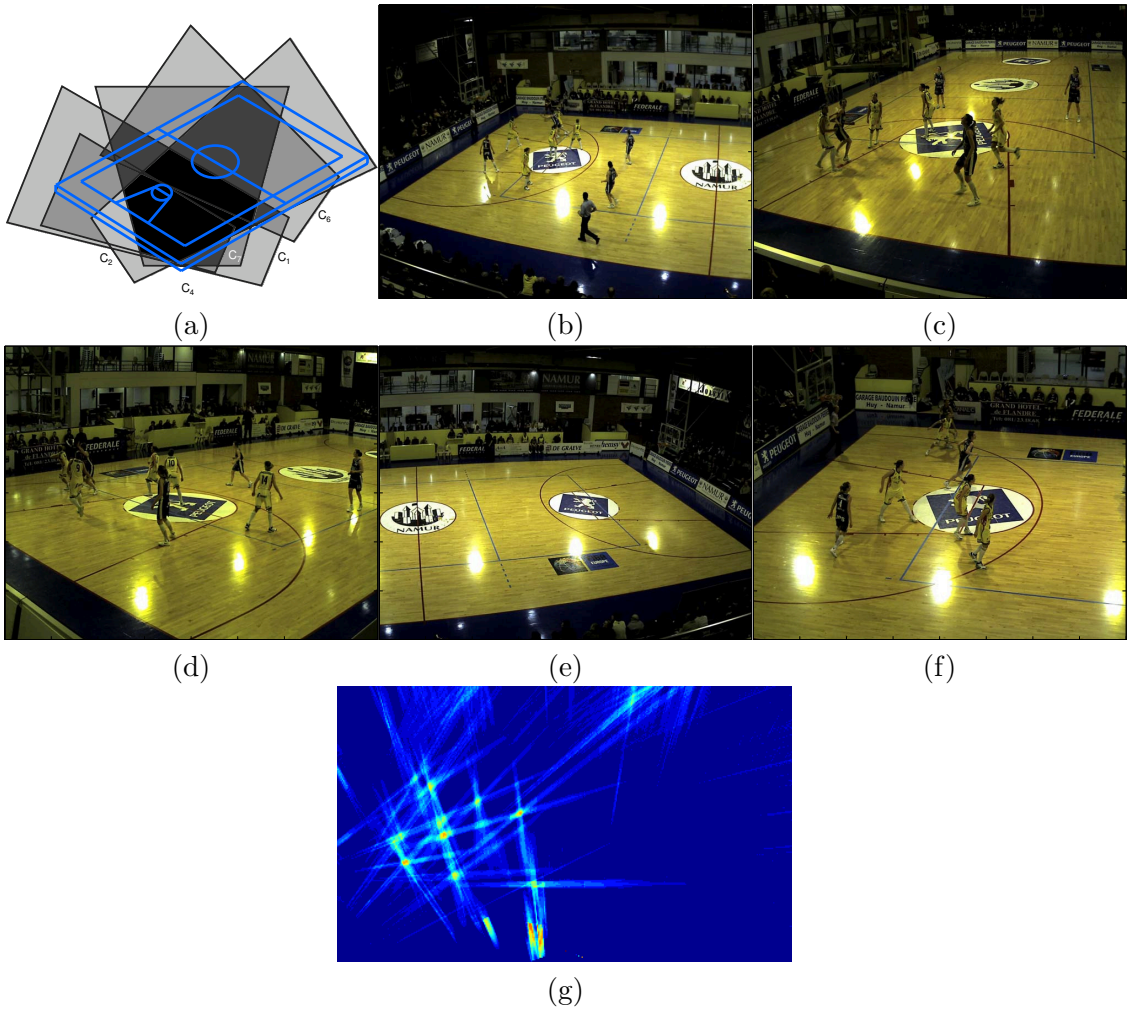


Figure 3.13: Camera views and projection of motion segmentation mask. (a) Configuration of cameras in basket ball court (excluding two top-mounted cameras with fish eye lenses). (b) Camera 1. (c) Camera 2. (d) Camera 4. (e) Camera 6. (f) Camera 7. (g) Sample projection of detection mask from multiple cameras to a top-view using multi-level homography.

to segment targets in such data.

Once the detections are obtained, the graph matching tracker, discussed in Sec. 3.3, can be applied for multiple target tracking. In this case only target positions represented by cluster centres are tracked. In the case of unknown numbers of targets, clustering may miss objects that have low observability. To address this issue, we propose to apply the track-before-detect approach which also eliminates the computational load due to detection step and is discussed next.

3.5 Multiple camera track-before-detect

In this section, we consider the multi-sensor detection volume as a *meta-sensor*. In this context we propose an algorithm that given the meta-sensor as input can perform simultaneous detection and tracking of multiple targets. In this section we first introduce the single target track-before-detect formulation based on particle filtering [124]. Next, we discuss the proposed multi-target track-before-detect particle filtering (MT-TBD-PF). Finally, we describe the mean-shift clustering and identity propagation approach within MT-TBD-PF.

3.5.1 Single target track-before-detect

Let $\mathbf{x}_k$ be the target state vector at time k, using a discrete time model with a fixed sampling period τ. The state can be defined as

$$\mathbf{x}_k = (x_k, \dot{x}_k, y_k, \dot{y}_k, I_k)^T, \tag{3.11}$$

where (x_k, y_k) are the position components, $(\dot{x_k}, \dot{y_k})$ are the velocity components and I_k is the value of the target signal strength (intensity) at time k at position (x_k, y_k). The state evolution can be modelled as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathcal{N}_k^p), \tag{3.12}$$

where $f(.)$ is the state transition function and $\mathcal{N}_k^p$ is the process noise. For a linear stochastic process, the state evolution can be expressed as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathcal{N}_k^p, \tag{3.13}$$

where $\mathbf{F}$ is the state transition matrix, defined as

$$\mathbf{F} = \begin{bmatrix} B & 0_{2\times2} & 0_{2\times1} \\ 0_{2\times2} & B & 0_{2\times1} \\ 0_{1\times2} & 0_{1\times2} & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \tag{3.14}$$

where $0_{m\times n}$ denotes an $m \times n$ matrix of zeros and τ is the sampling interval. The process noise $\mathcal{N}_k^p$ models the disturbances affecting the target state and is generally modelled as

a zero mean Gaussian random variable [60] with covariance $\mathbf{Q}$, defined as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D} & 0_{2\times 2} & 0_{2\times 1} \\ 0_{2\times 2} & \mathbf{D} & 0_{2\times 1} \\ 0_{1\times 2} & 0_{1\times 2} & q_2\tau \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \frac{q_1}{3}\tau^3 & \frac{q_1}{3}\tau^2 \\ \frac{q_1}{2}\tau^2 & q_1\tau \end{bmatrix}, \tag{3.15}$$

where q_1 and q_2 are the process noise in target motion and intensity.

Let $\mathbf{z}_k = \{z_k(i,j) : i = 1, \cdots W, j = 1, \cdots, H\}$ be the measurement, at each time k, encoded in a $W \times H$ resolution image. At each pixel position, the measurement intensity $z_k(i,j)$ is either due to the presence of the target or due to measurement noise $\mathcal{N}_k^m$; that is

$$z_k(i,j) = \begin{cases} h_k(i,j)(\mathbf{x}_k) + \mathcal{N}_k^m(i,j) & \text{if target is present} \\ \mathcal{N}_k^m(i,j) & \text{if target is not present} \end{cases}, \tag{3.16}$$

where the measurement noise $\mathcal{N}_k^m$ models the disturbances affecting the measurement. $\mathcal{N}_k^m$ is modelled as a zero mean Gaussian sequence which is assumed to be mutually independent from the process noise. $h_k(i,j)(.)$ is the contribution of the target intensity at pixel position (i,j). In the case of a point target, the distribution of the target intensity over the surrounding pixels will be only due to the sensor point spread function and can be approximated as [60]

$$h_k(i,j)(\mathbf{x}_k) \approx \frac{\Delta_x \Delta_y I_k}{2\pi A^2} \exp\left(-\frac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2A^2}\right), \tag{3.17}$$

where A models the amount of blurring introduced by the sensor and $\Delta_x \times \Delta_y$ is the size in pixels of the segment centred at $(i\Delta_x, j\Delta_y)$. This indicates that each target occupies multiple pixels in the measurement $\mathbf{z}_k$, instead of being a point target (Fig. 3.19(a-b)).

Given the set of measurements $Z_k = \{\mathbf{z}_m | m = 1, \cdots, k\}$ up to time k, the objective is to recursively quantify some degree of belief in the state $\mathbf{x}_k$ taking different values, i.e., to estimate the posterior *pdf* $p(\mathbf{x}_k|Z_k)$. Using the Bayesian recursion, the posterior *pdf* $p(\mathbf{x}_k|Z_k)$ can be computed in two steps: *prediction* and *update*. In the *prediction* step, the prior density of the state at time k is obtained using the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|Z_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|Z_{k-1}) d\mathbf{x}_{k-1}, \tag{3.18}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition density defined by the target model (Eq. 3.12) and $p(\mathbf{x}_{k-1}|Z_{k-1})$ is the posterior at time $k-1$. The *update* step is carried out using the measurement at time k by applying Bayes' rule:

$$p(\mathbf{x}_k|Z_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|Z_{k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|Z_{k-1})d\mathbf{x}_k}, \tag{3.19}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood function.

The above algorithm is implemented using a Sampling Importance Resampling (SIR) particle filter [56] where a posterior density is represented by a set of particles each with associated weight $\{\omega_k^n, \mathbf{x}_k^n\}$.

In the *prediction* step, we draw two sets of particles to estimate the predicted density, namely *new-born* particles and *surviving* particles. The new-born particles are the set of J_k particles for which the target state is drawn as a sample from a proposal distribution $p(\mathbf{x}_k|Z_k)$. The proposal distribution $p(\mathbf{x}_k|Z_k)$ could be any appropriate distribution such as an uniform distribution where at each position (x_k, y_k) in the measurement $\mathbf{z}_k$, equal number of particles are drawn. Such a distribution is appropriate when signal-to-noise ratio (SNR) is very low. In case of moderate or high SNR the proposal distribution $p(\mathbf{x}_k|Z_k)$ can be the measurement $\mathbf{z}_k$ itself, normalised between zero and 1 such that at each position (x_k, y_k) in the measurement $\mathbf{z}_k$ the number of particles drawn is proportional to the signal intensity $I_k(x_k, y_k)$ (Fig. 3.19(c)). The surviving particles are the set of L_{k-1} particles that continue to stay alive. These particles are generated from the proposal density $q_k(\mathbf{x}_k|\mathbf{x}_{k-1}, Z_k)$ based on the target dynamic model such that the current state of each of the surviving particles is estimated by applying Eq. 3.13 (Fig. 3.19(d)).

Particle filtering approximates the densities $p(\mathbf{x}_k|Z_k)$ with a sum of $L_{k-1} + J_k$ Dirac functions centred in $\{\mathbf{x}_k^n\}_{n=1,\ldots L_{k-1}+J_k}$ as

$$p(\mathbf{x}_k|Z_k) \approx \sum_{n=1}^{L_{k-1}+J_k} \omega_k^n \delta\left(\mathbf{x}_k - \mathbf{x}_k^n\right), \tag{3.20}$$

where ω_k^n are the weights associated with the particles. The weights are calculated in [56] as

$$\omega_k^n \propto \omega_{k-1}^n \frac{p(\mathbf{z}_k|\mathbf{x}_k^n)p(\mathbf{x}_k^n|\mathbf{x}_{k-1}^n)}{q(\mathbf{x}_k^n|\mathbf{x}_{k-1}^n, \mathbf{z}_k)}. \tag{3.21}$$

$q(.)$ is the importance density function. When $q(.) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^n)$ (i.e., the transitional prior), then

$$\omega_k^n \propto \omega_{k-1}^n p(\mathbf{z}_k|\mathbf{x}_{k-1}^n). \tag{3.22}$$

In the *update* step, for each pixel (i,j), the likelihood $p(z_k(i,j)|\mathbf{x}_k^n)$, for the combined set of $L_{k-1}+J_k$ particles is computed. Given the sensor model defined in Eq. 3.16, the likelihood function can be expressed as [124]

$$p(\mathbf{z}_k|\mathbf{x}_k) = \begin{cases} \prod_{i=1}\prod_{j=1} p_{S+\mathcal{N}}(z_k(i,j)|\mathbf{x}_k^n) & \text{if target is present} \\ p_{\mathcal{N}}(z_k(i,j)) & \text{if target is not present} \end{cases}, \tag{3.23}$$

where $p_{\mathcal{N}}(z_k(i,j))$ is the *pdf* of the background noise in pixel (i,j) and $p_{S+\mathcal{N}}(z_k(i,j)|\mathbf{x}_k^n)$ is the likelihood of the target signal affected by noise in pixel (i,j). The product between the *pdf* values computed for each pixel (i,j) is based on the assumption that the measurement noise $\mathcal{N}_k^m(i,j)$ is independent from pixel to pixel.

The final likelihood is obtained by taking the likelihood ratio in pixel (i,j) for a target in state $\mathbf{x}_k^n$ as

$$\begin{aligned} p(z_k(i,j)|\mathbf{x}_k^n) &= \frac{p_{S+\mathcal{N}}(z_k(i,j)|\mathbf{x}_k^n)}{p_{\mathcal{N}}(z_k(i,j))} \\ &= \exp\left(-\frac{h_k(i,j)(\mathbf{x}_k)(h_k(i,j)(\mathbf{x}_k)-2z_k(i,j))}{2A^2}\right). \end{aligned} \tag{3.24}$$

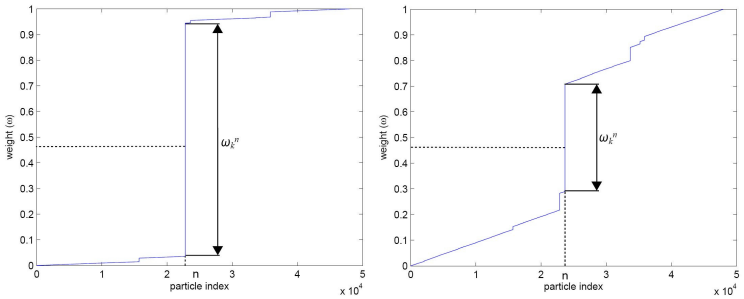
Since the pixels are assumed to be conditionally independent, the likelihood of the whole image is computed by taking the product over the pixels; thus, the updated particle weights are computed as

$$\tilde{\omega}_{k|k-1}^n = \prod_{i=w_i(\mathbf{x}_{k|k-1}^n)} \prod_{j=w_j(\mathbf{x}_{k|k-1}^n)} p(z_k(i,j)|\mathbf{x}_k^n), \tag{3.25}$$

where $w_i(.)$ and $w_j(.)$ indicates that only the pixels affected by the target are used in the likelihood computation which are selected by using a fixed size window. The weights are finally normalised with the sum of all weights $\Omega_k = \sum_{n=1}^{L_{k-1}+J_k} \omega_{k|k-1}^n$ as

$$\omega_{k|k-1}^n = \frac{\tilde{\omega}_{k|k-1}^n}{\Omega_k}. \tag{3.26}$$

The variance of these importance weights $\omega_{k|k-1}$ can only increase over time [136]. This means that after certain number of particle filtering steps, all but one particle will have negligible normalised weights (Fig. 3.14(a)) and this phenomenon is called the de-



(a) (b)

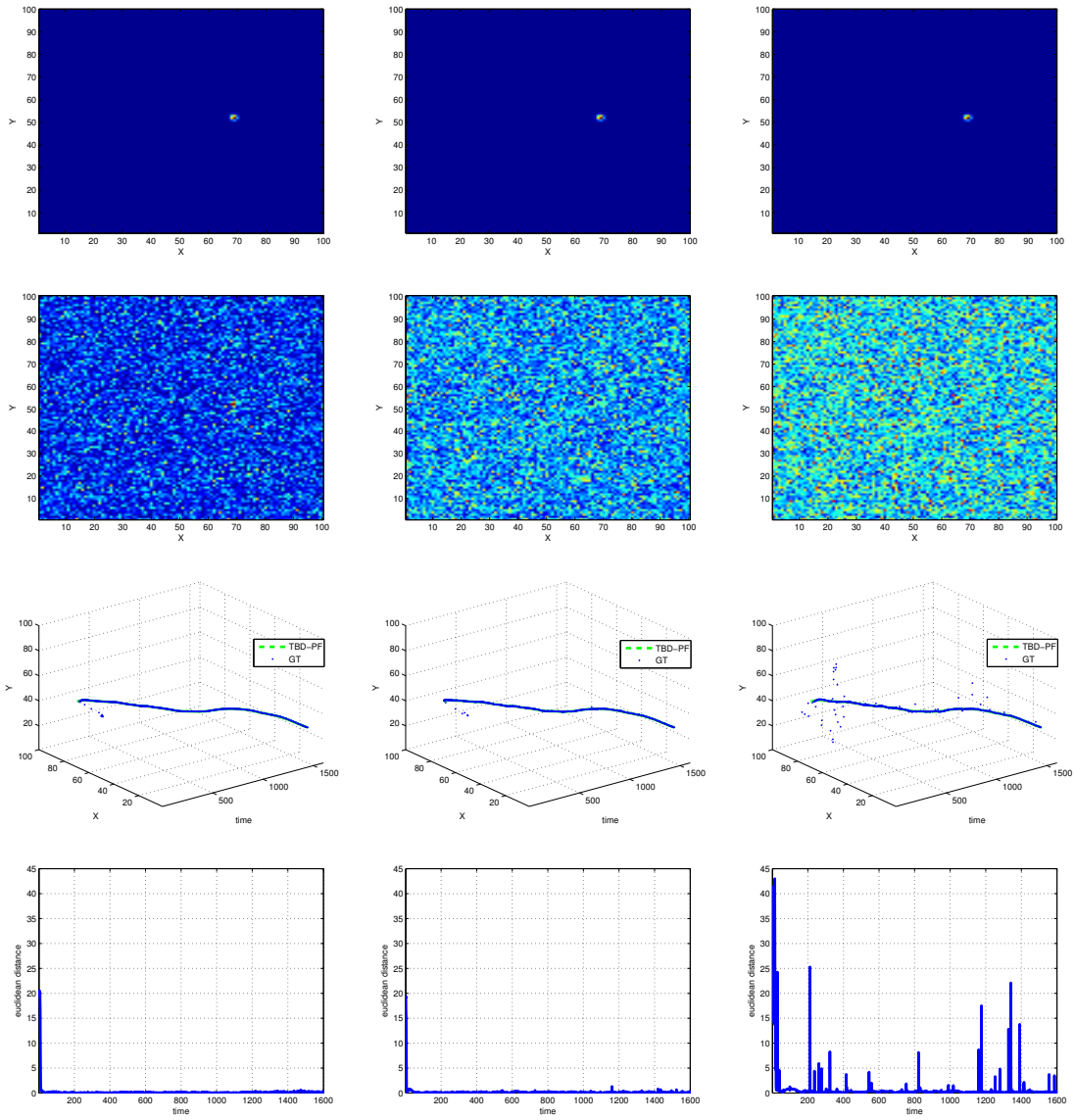
Figure 3.14: Sample cumulative of particle weights. (a) Degeneracy problem showing all but one particle having negligible normalised weights. (b) The n^{th} particle $\mathbf{x}_k^n$ has higher chances of being selected due to having high weight ω_k^n.

generacy problem [60]. To avoid the degeneracy of particles, resampling in applied which eliminates samples with low importance weights and multiplies samples with high importance weights by using the cumulative sum of particle weights (Fig. 3.14(b)). The combined set of $L_{k-1} + J_k$ particles are resampled to reduce the number to L_k only by selecting particles for which $\omega_k^n > \lambda_\omega$, where λ_ω is the minimum allowed particle weight. If $\omega_k^n > \lambda_\omega, \forall n$ then $L_k = L_{k-1} + J_k - J_{k+1}$ where $J_{k+1} = N^{min}$ and N^{min} is the minimum number of new-born particles at each time k. This process involves generating L_k random variable from the uniform distribution on the interval [0 1]. For each of the L_k values, a particle whose weight correspond to that value is propagated. The resampled particles weights are set to $\omega_{k-1}^n = 1/(L_{k-1} + J_k)\ \forall\, n$. This means there is no need to pass on the importance weights from one time step to the next and Eq. 3.22 can be simplified to

$$\omega_k^n \propto p(\mathbf{z}_k|\mathbf{x}_k^{n-1}). \tag{3.27}$$

That is, the weights are proportional to the likelihood function.

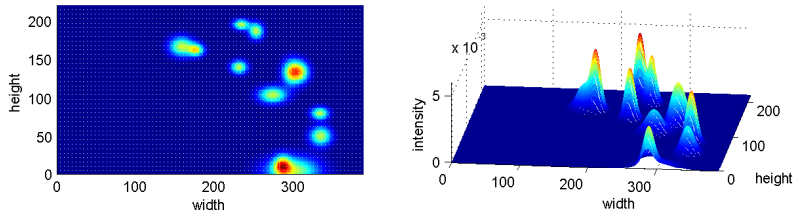
Figure 3.15 shows an example of our single target track-before-detect particle filter using three different SNR values of synthetic data. The synthetic data is generated by computing a target track using a motion model. This track is then converted into an input image of resolution WxH where the position of the target is represented by a Gaussian with standard deviation of 2 pixels (Fig 3.15 (Row 1)). White Gaussian noise is then added on this image multiple times to achieve a signal with different SNR values (Fig 3.15 (Row 2)). Although with SNR= 8.6969dB and SNR= 6.2613dB the target cannot be observed visually due the noise (Fig 3.15 (Row 2)(b,c)), it was correctly tracked (Fig. 3.15(Row 3)(b,c)). When SNR= 6.2613dB, the algorithm had some difficulties in



(a) (b) (c)

Figure 3.15: Sample single target track-before-detect results with varying SNR values: (Row 1) sample frames from input data without noise indicating a target with hotter values. (Row 2) sample frames from input data with noise illustrating that target is difficult to detect by visual inspection at low SNR. (Row 3) Tracking results (mean particle position for every k). (Row 4) Euclidean distance between ground truth and estimated target position; (a) SNR = 18.3422dB, (b) SNR = 8.6969dB and (c) SNR = 6.2613dB. (Blue dots: estimated positions; green dashes: ground truth).

identifying the target location; however, once enough particles were drawn around the target, it was tracked consistently. Note that here we use only one particle per pixel as compared to other approaches [137], where 4 times more particles were used.



(a) (b)

Figure 3.16: Example showing that the measurement at each pixel can have a contribution from other targets. (a-b) 2D and 3D visualization of all the targets showing each target intensity has contribution from other targets.

3.5.2 Multi-sensor, multi-target track-before-detect

In the case of multiple targets, the measurement at each pixel (i, j) can have a contribution from all the targets (Fig. 3.16) and Eq. 3.16 can be modified to

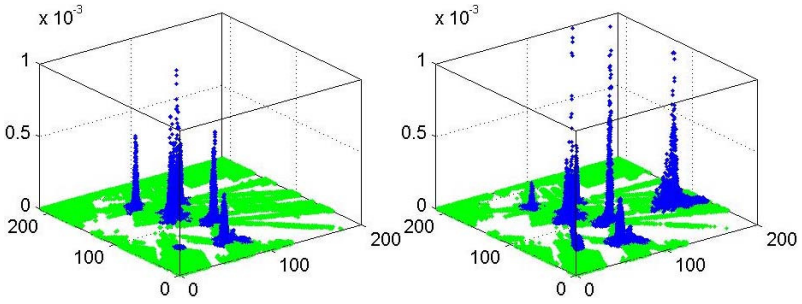
$$z_k(i,j) = \begin{cases} \sum_{t=1}^{N_k^O} h_k(i,j)(\mathbf{x}_k^t) + \mathcal{N}_k(i,j) & \text{if } N_k^O \text{ targets present at time } k \\ \mathcal{N}_k(i,j) & \text{if no target present} \end{cases}, \tag{3.28}$$

The approximation shown in Eq. 3.17 is based on a point target assumption and is a truncated 2D Gaussian density with circular symmetry. Similar approximation can be used in the case of multiple targets in the projected domain by tuning the values for Δ_x, Δ_y and A, respectively. This enables the filtering out of noise that is due to parallax error.

The particle filter may perform poorly when the posterior is multi-modal as the result of multiple-targets [55]. To solve this problem, instead of using the existence variable and the jump Markov model [19,125], we employ clustering of the particles. The prediction step remains the same as in the case of single targets. If all targets follow the same motion model, this prediction step is correct as each particle contains the velocity components $(\dot{x}_k, \dot{y}_k)$ of the target it represents. Tracking targets with a different dynamic model can be performed by incorporating Interacting Multiple Models (IMM) [138].

As different targets may have different intensity levels and in TBD the weight update is a function of the target intensity, this results in lower weight assignment to weaker targets. To address this issue we consider each target individually in the update step and Eq. 3.25 can be re-written for multiple targets TBD as

$$\tilde{\omega}_{k|k-1}^{nt} = \prod_{i \in w_i(\mathbf{x}_{k|k-1}^{nt})} \prod_{j \in w_j(\mathbf{x}_{k|k-1}^{n,t})} p(z_k(i,j)|\mathbf{x}_k^n), \tag{3.29}$$



(a) (b)

Figure 3.17: Example of particle weights and positions. (a) Without the proposed update strategy (one target has very small weights and another one is missing); (b) with the proposed update strategy. As the weights for weak targets are very low without the proposed update, this results in track losses.

where $\mathbf{x}_{k|k-1}^{nt}$ is the n^{th} particle at time k belonging to the t^{th} target. The weights are normalised with the sum of all weights associated to t^{th} target $\Omega_k^t = \sum_{n \in t} \omega_{k|k-1}^{nt}$ as

$$\omega_{k|k-1}^{nt} = \frac{\tilde{\omega}_{k|k-1}^{nt}}{\Omega_k \Omega_k^t}. \tag{3.30}$$

Here the component Ω_k is used to further normalise the weights so that they lie between 0 and 1. This is used instead of the number of targets as there are some particles generated using another proposal density $p(\mathbf{x}_k|Z_k)$. Figure 3.17 shows a comparison between the evolutions of particle weights with and without the proposed update strategy. It can be seen that without the proposed update strategy (Fig. 3.17(a)), one of the targets is completely missed while the other one has very low weight and is lost in the next frame. Following the update step, the particles are clustered using mean-shift for the association of an identity with each particle. Mean-shift clustering climbs the gradient of a probability distribution to find the nearest dominant mode or peak [131]. Mean-shift is preferred here as it is a non-parametric clustering technique that does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.

Given $L_{k-1} + J_k$ particles $\{\mathbf{x}_k^n, n = 1, \cdots, L_{k-1} + J_k\}$ on a 2-dimensional space $\mathbb{R}^2$ using (x_k, y_k) only, the multivariate kernel density estimate obtained with kernel $\mathcal{K}(\mathbf{x})$ and bandwidth h is

$$f(\mathbf{x}_k) = \frac{1}{(L_{k-1} + J_k)h^2} \sum_{n=1}^{L_{k-1}+J_k} \mathcal{K}\left(\frac{X_k - \mathbf{x}_k^n}{h}\right). \tag{3.31}$$

The bandwidth h is set as $h = 2q_1$ based on the target covariance Q (see Eq. 3.15). The

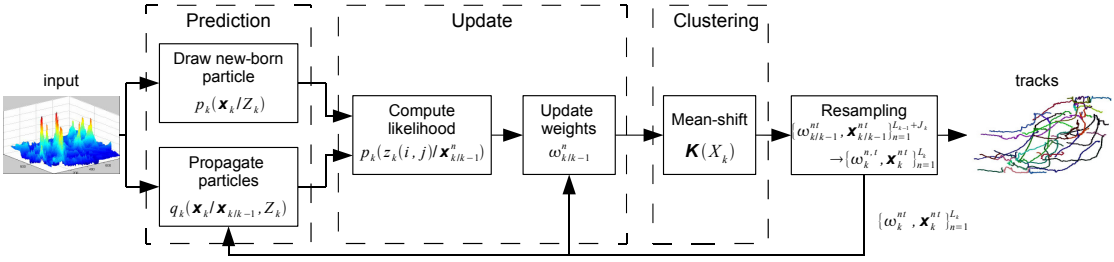


Figure 3.18: Block diagram of the proposed approach.

mean-shift algorithm tends to maximise the density whose modes are located at the zeros of the gradient $\bigtriangledown f(\mathbf{x}_k)$.

After clustering, a cluster merging process is performed to fuse similar clusters. The fusion is based on proximity $\gamma_{op} < \lambda_\mu$ and $\beta_{op} < \lambda_A$, where λ_μ and λ_A are the mean and covariance thresholds, γ_{op} is the Euclidean distance between centre of clusters o and p, and β_{op} is the covariance of the merged clusters. Finally, an identity is assigned to each particle based on its cluster membership. If all the particles in a cluster are new-born, then a new identity is issued; otherwise all cluster members are assigned the identity with the highest population within the cluster. Figure 3.19(e) shows the particles before clustering, whereas the clustered particles are shown in Fig. 3.19(f-h). In Fig. 3.19(f-h) each colour indicates a unique cluster and particles coloured in dark blue in Fig. 3.19(f) are the pruned particles.

To avoid the degeneracy problem [56], we resample the particles. Resampling is performed according to the particle weights. Here again the single target resampling strategy based on the cumulative distribution function *cpdf* of particle weights will not work as it is insensitive to the particle location. Particles with lower weights (such as those associated to new-born targets) will not be able to have enough representation in the mixture distribution. As shown in Fig. 3.20, ω_k^{n1}, ω_k^{n2} and ω_k^{n3} are the weights for a particle representing the state of target 1, 2 and 3 respectively. The particle for target 1 having higher weight ω_k^{n1} will be multiplied more times that of particle for target 2 which will be multiplied more times that of target 3. This will result in an unfair resampling where more particles will be used to represent the state of a particular target than another depending upon their weights. This will create a hindrance in initialising new tracks in the presence of existing targets. To this extent, the resampling is performed individually for each cluster. For each cluster the weights of only those particles that are associated with the cluster are used to create a cumulative distribution function *cpdf*. The resampling

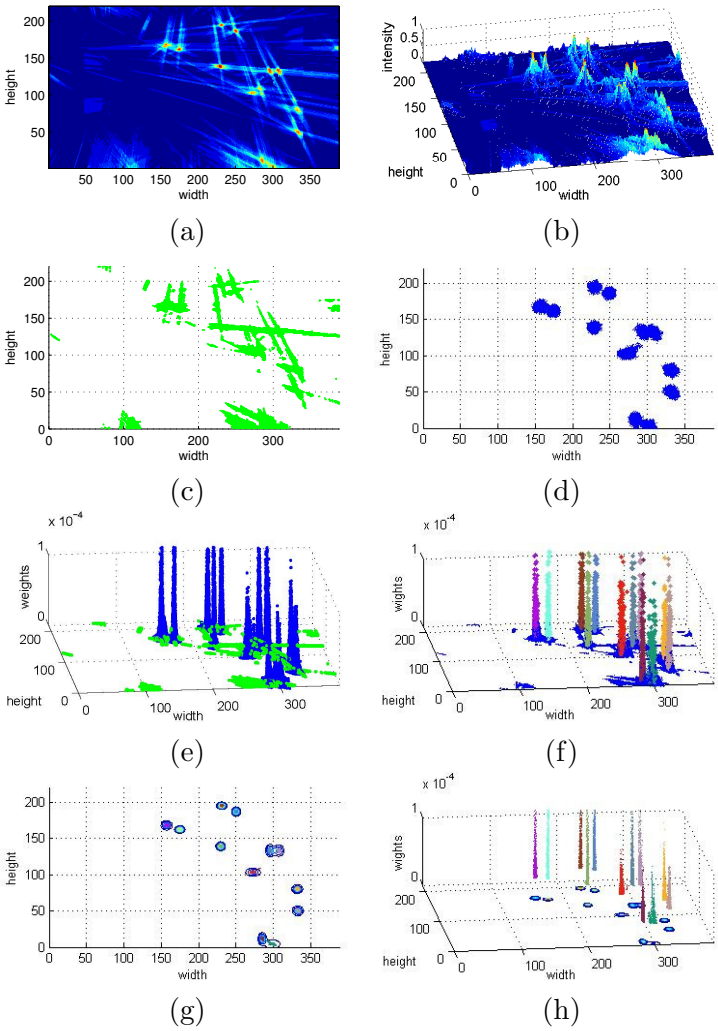


Figure 3.19: Sample input and output at intermediate steps of the algorithm. (a-b) 2D and 3D view of multi-camera detection volume (meta-sensor). (c-d) Output at prediction step showing new-born and propagated particles. (e) Weight assignment at update step. (f) Uniquely colour-coded clusters of particles corresponding to targets and pruned particles (dark blue colour). (g-h) Uniquely colour-coded clusters of particles.

is then performed for each cluster individually in the same way as done for single target (Fig 3.14(b)).

The block diagram of the proposed approach is shown in Fig. 3.18 and sample outputs at intermediate steps of the algorithm are shown in Fig. 3.19.

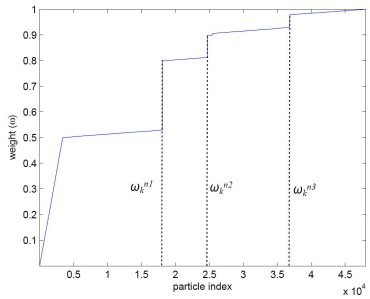


Figure 3.20: Sample cumulative of particle weights showing weights associated with particles belonging to different targets (ω_k^{n1}, ω_k^{n2} and ω_k^{n3} represents weights for a particle representing state of target 1, 2 and 3 respectively).

3.6 Results

3.6.1 Evaluation metrics

The single camera detection and tracking performance is measured by participating in evaluation challenges, namely CLEAR [139] and ETISEO [140]. These two evaluations define two different set of scores. The evaluation measures defined in CLEAR evaluation are Multi-Object Detection Precision (MODP), Detection Accuracy (MODA), Tracking Precision (MOTP) and Tracking Accuracy (MOTA). These scores give a weighted summary of the detection and tracking performance in terms of False Positives (FP), False Negatives (FN) and object identity switches. They are defined as follows. First the detection scores with

$$MODP_k = \frac{\sum_{i=1}^{N^{tm}} \frac{|GT_k^i \cap AD_k^i|}{|GT_k^i \cup AD_k^i|}}{N^{tm}}, \tag{3.32}$$

where the term in numerator is a measurement of the overlap between the i^{th} ground truth object GT_k^i (bounding box) and corresponding automatic detection AD_k^i in frame k, N^{tm} is the number of targets mapped and

$$MODA_k = 1 - \frac{c_{FN} N_k^{FN} + c_{FP} N_k^{FP}}{N_k^{GT}}, \tag{3.33}$$

where c_{FN} and c_{FP} are the costs associated with the false negatives and false positives and N_k^{FN} and N_k^{FP} are the number of false negatives and false positives in frame k and N_k^{GT} is the number of objects in the ground truth. The values used for c_{FN} and c_{FP} for the evaluation of the proposed algorithm is 1. Then the tracking scores are computed

similarly as

$$MOTP = \frac{\sum_{i=1}^{N^{tm}} \sum_{k=1}^{N^{fr}} \left(\frac{| \, GT_k^i \cap AD_k^i \, |}{| \, GT_k^i \cup AD_k^i \, |} \right)}{\sum_{k=1}^{N^{fr}} N_k^{tm}} \tag{3.34}$$

and

$$MOTA = 1 - \frac{\sum_{k=1}^{N^{fr}} (c_{FN} N_k^{FN} + c_{FP} N_k^{FP}) + log_e(N^{id})}{\sum_{k=1}^{N^{fr}} N_k^{GT}}, \tag{3.35}$$

where N^{id} is the number of false identity switches for all objects during the sequence and N^{fr} is the number of frames. Note that the range of precision is [0 1] whereas that of accuracy is $(-\infty \; 1]$. The accuracy becomes negative when the score deductions due to false and missed detections exceed the score obtained through true positives. The log of N^{id} is used in the calculation of MOTA, hence in case of fewer identity switches both MODA and MOTA will have similar values. Similarly, given each detected object has been tracked consistently without multiple large number of identity switches, the MODP and MOTP will also have similar values. The CLEAR data annotations are done separately for person and vehicles. In order to use the CLEAR evaluation tool and the available ground truth, a simple pedestrian/vehicle classifier is added to the system, whose decision is based on the ratio between the width and height of the bounding box, followed by a temporal voting mechanism. These four matrix were used for the evaluation of results on the CLEAR dataset only (Fig. 3.21(a-c)).

The ETISEO evaluation defines 2 scores for detection, 5 for localisation and 5 for tracking evaluation [140]. These scores measure the precision and sensitivity of the algorithm. Let FP be the number of false positive detections, TP the number of true positive detections, and FN the number of false negative detections. The *precision* is defined as

$$P = \frac{TP}{TP + FP} \tag{3.36}$$

and the *sensitivity* is defined as

$$S = \frac{TP}{TP + FN}. \tag{3.37}$$

The precision and sensitivity is computed per pixel as well as per object and details can be found in [140]. The ETISEO evaluation also defines scores to measure splitting and merging of blobs. The object area fragmentation or splitting score measures the number of detected objects associated with each object in the ground truth at each time k and is

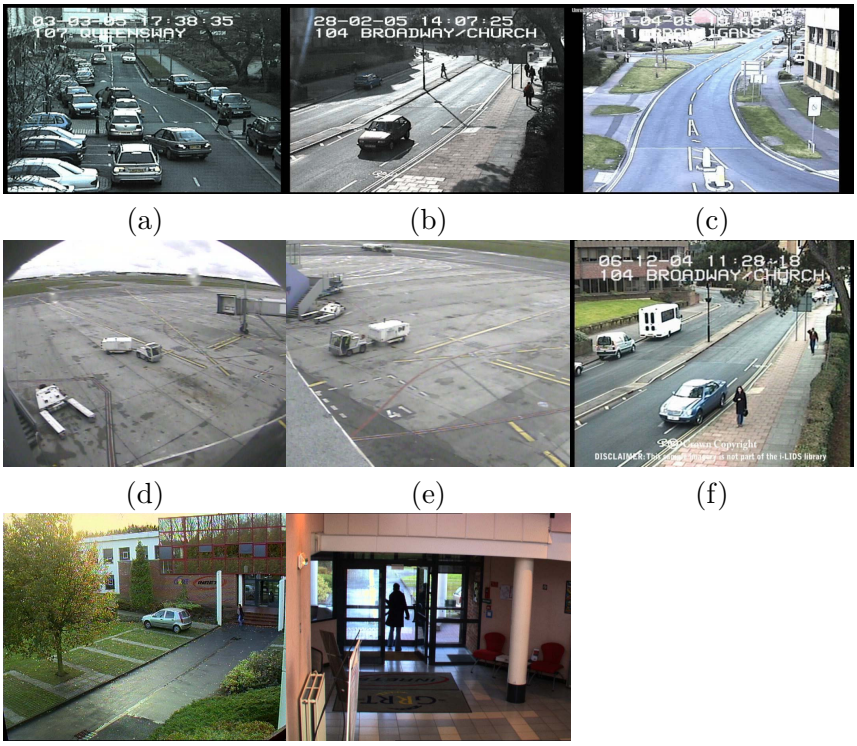


Figure 3.21: The evaluation surveillance datasets. (a-c) CLEAR Surveillance scenario (BC, QW and BR respectively) (d-e) ETISEO airport scenario (AP-11 C4 and C7) (f) ETISEO road scenario (RD-6 C7) (g) ETISEO building entrance scenario (BE-19 C1 and C4).

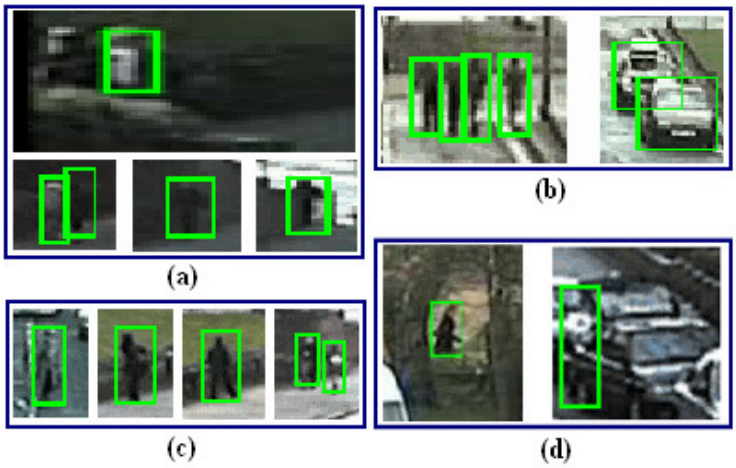


Figure 3.22: Examples of challenging situations for the pedestrian and vehicle detection and tracking task in the CLEAR dataset (the ground-truth detection are shown in green). (a) Objects in low visibility regions. (b) Objects in close proximity. (c) Objects with low contrast compared to the background. (d) Occluded objects.

Table 3.1: Summary of the datasets used in the experiments

Dataset	**Seq.** name	**Cameras**	**Resol.**	**No. of frames**	**Frame rate (Hz)**
CLEAR	BC	C1	720×480	72750	25
	QW	C1	720×480	51309	25
	BR	C1	720×480	15542	25
ETISEO	AP-11	C4, C7	720×576	805 , 805	12.5
	BE-19	C1, C4	768×576	1025 , 950	25
	RD-6	C7	720×576	1201	25
IISA	IISA	C1, C2	1920×1088	5996	25
		C3-C6	1920×1080	11992	25
APIDIS	AP	C1-C7	1600×1200	8931	25
Total number of frames				171306	-

defined as

$$split_k = \frac{1}{N_k^{GT}} \sum_{i=1}^{N_k^{GT}} \frac{1}{N_k^{AD|GT^i}}, \tag{3.38}$$

where $N_k^{AD|GT^i}$ is the number of automatically detected objects AD that overlaps with the i^{th} object in the ground truth GT^i at time k . Similarly the score measuring the merging of blobs is defined as

$$merge_k = \frac{1}{N_k^{AD}} \sum_{i=1}^{N_k^{AD}} \frac{1}{N_k^{GT|AD^i}}, \tag{3.39}$$

where N^{AD} is the number of detected objects at time k and $N^{GT|AD^i}$ is the number of ground truth objects GT that overlap with the i^{th} detected object AD. The value of 1.0 for split and merge indicates that each object in the ground truth corresponds with only one of the detected objects and and vice versa. Further details about the ETISEO metrics can be found in [140].

The evaluation of multi-camera tracking algorithms is also performed by computing precision and sensitivity.

3.6.2 Experimental set-up

The proposed single camera detection and tracking has been evaluated on the CLEAR [139], ETISEO [140] and IISA [141] datasets (Table 3.1). The CLEAR evaluation dataset consists of 50 sequences with ground truth annotation, for interval of 139601 frames (i.e., approximately 1 hour 22 minutes of recorded video). To reduce the computational time, the sequences were processed at a half the original resolution (i.e., 360×240 pixels).

The dataset consists of outdoor surveillance sequences of urban areas (Fig. 3.21(a-c)). The ETISEO dataset consists of 5 sequences from 3 different scenarios, namely *Airport*, *Building entrance* and *Road* with ground truth annotation, for a total of 4786 frames (i.e., approximately 4 minutes 44 seconds of recorded video). The ETISEO dataset consists of both indoor and outdoor surveillance scenarios sequences (Fig. 3.21(d-h)). The annotations from both these datasets provide the bounding boxes of pedestrians and vehicles in the scene and have been used for quantitative evaluation of the proposed algorithm.

The evaluation of detection and tracking algorithms requires a large amount of (annotated) data from real word scenarios because performance varies depending on different environmental conditions. The CLEAR dataset provides a large test bed, making it easier to evaluate how different features impact on the final detection and tracking results. The complexity of the CLEAR dataset is particularly related to the challenges discussed earlier in this chapter. Examples illustrating the following difficult situations in the CLEAR dataset are shown in Fig. 3.22: objects with low visibility located in the shade generated by a building (Fig. 3.22 (a)); merged detections (Fig. 3.22 (b)) due either to physical closeness or to the camera perspective view; objects with little contrast compared to the background (Fig. 3.22 (c)); partial and total occlusions (Fig. 3.22 (d)).

The IISA dataset depicts a soccer match observed by 6 cameras with partial overlap (Fig. 3.23(a-f)). The dataset consists of 6 sequences (one per camera) without ground truth annotation and is used for qualitative analysis of both single and multi-camera tracking algorithms. The algorithm has also been qualitatively evaluated on CAVIAR [142], PETS 2006 [143] and MediaPro [144] datasets. The evaluation using these datasets will be discussed together with the interaction recognition in Chapter 6.

We have evaluated the multi-camera detection and tracking algorithm on synthetic and real datasets. The synthetic data consisted of 12 simulated targets moving with moderate speed with some manoeuvring. The real data is from APIDIS [145] and IISA datasets. The APIDIS dataset consists of a basketball match scenario captured using five partially overlapping cameras (Fig. 3.23(j-k)) and two top-mounted with fish eye lenses (Fig. 3.23(l-m)). There are in total 12 targets in the video (10 players and 2 referees). The players have similar appearances and are difficult to distinguish from the background colour. The algorithm has been quantitatively evaluated on the APIDIS dataset and qualitatively evaluated on both the APIDIS and the IISA datasets. The IISA dataset consists of 22 targets in the video (20 players and 2 referees).

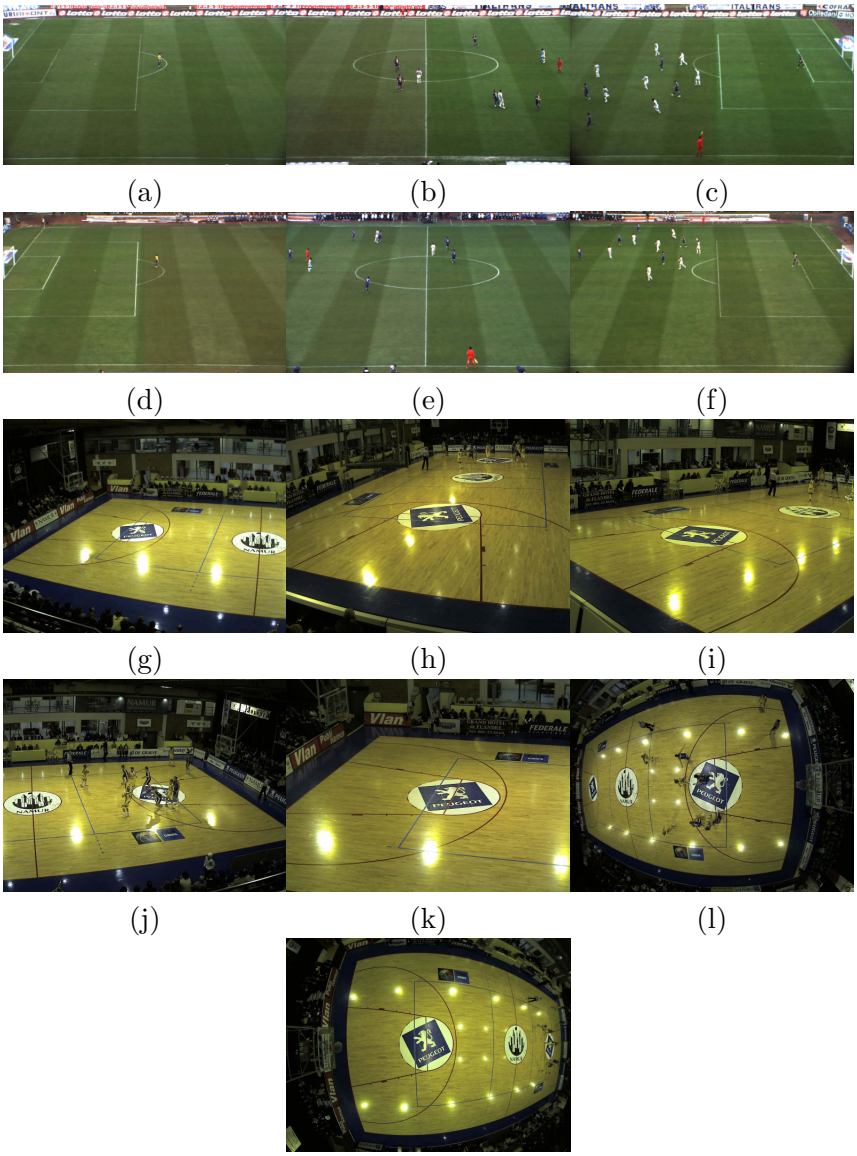
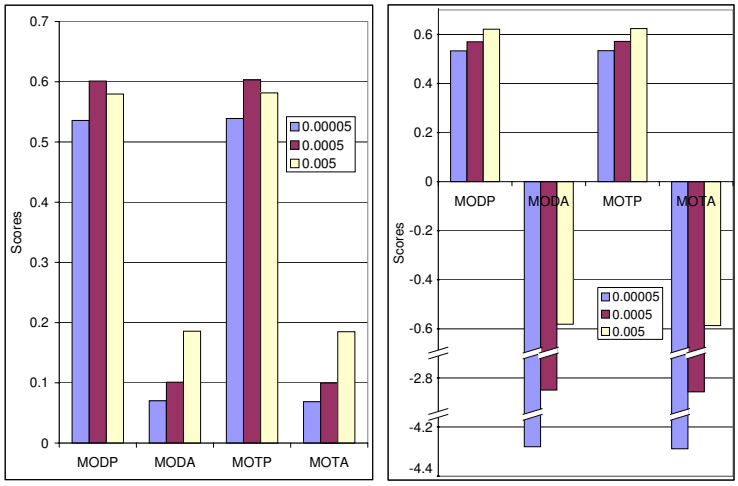


Figure 3.23: The evaluation sports datasets. (a-f) IISA football scenario (C1-C6) (g-m) APIDIS basket ball scenario (C1-C7).

3.6.3 Single camera detection and tracking

In this section we will first analyse the results from CLEAR evaluation and discuss the effect on performance with and without various modules of the detection algorithm as well as by changing parameter values. The discussion on CLEAR evaluation will be followed by analysis of scores obtained from ETISEO evaluation. Finally we show some qualitative results on CLEAR, ETISEO and IISA datasets followed by discussion on failure modes.



(a) (b)

Figure 3.24: Comparison of tracking results with different update factors (0.005, 0.0005, 0.00005) for the background model. (a) Pedestrian tracking. (b) Vehicle tracking.

Evaluation using CLEAR dataset and metrics

Using the CLEAR dataset, the performance of different modules of the proposed algorithm has been evaluated and the performance comparison with and without them has been performed. Particularly, in order to quantify the performance of the proposed detection and localisation algorithm, this subsection discusses the effect of: (i) changing background update factor (Sec. 3.2.1), (ii) including and excluding spatio-temporal filtering (Sec. 3.2.2), (iii) changing σ of the noise in the pixel classification (Sec. 3.2.2) and (iv) including and excluding blob splitting (Sec. 3.2.2). In each of these experiments, the evaluation has been performed on the entire CLEAR dataset. The tracker performance is evaluated by using different combinations of the aforementioned features.

Figure 3.24 shows a comparison of the effect on performance of varying the *update factor*, over the range $[0.00005, 0.005]$. Increasing update factor from 0.00005 to 0.0005 improves precision and accuracy (false positives are reduced without a significant increase of false negatives). However, when increasing update factor to 0.005, the accuracy improves significantly for vehicle tracking with slight decrease for pedestrian tracking. A value of $\alpha = 0.005$ is therefore a good compromise between accuracy and precision. Figure 3.25 shows how the model update manages to reduce false positives; however, the car that stopped on the road becomes part of the background model thus producing a false negative.

Once the background is estimated using an adaptive update factor, this background is used in the next frame to perform frame differencing. The frame differencing

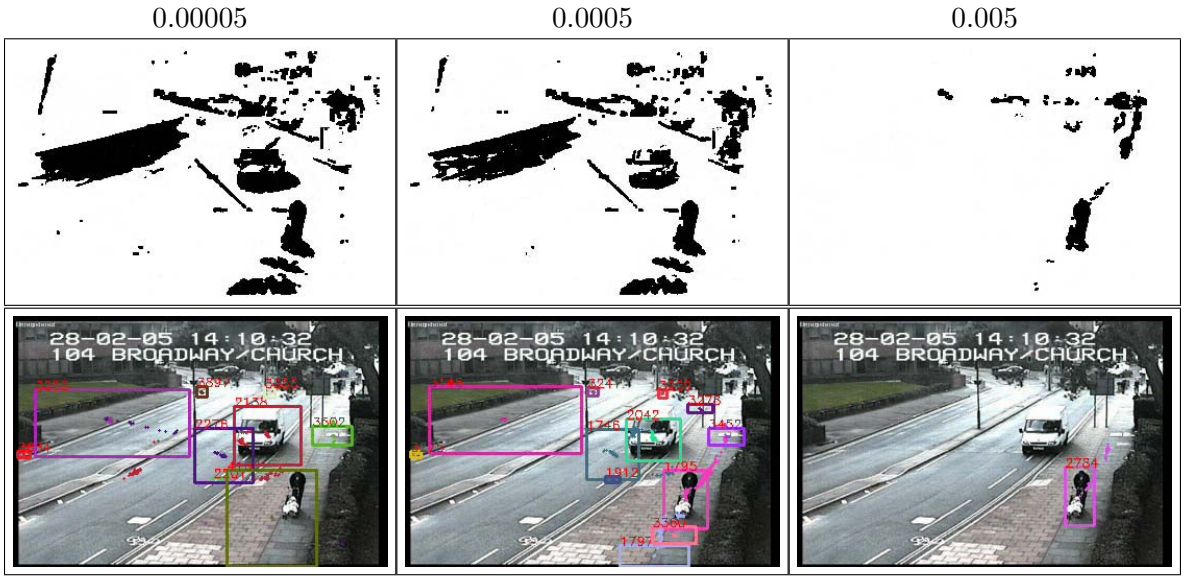
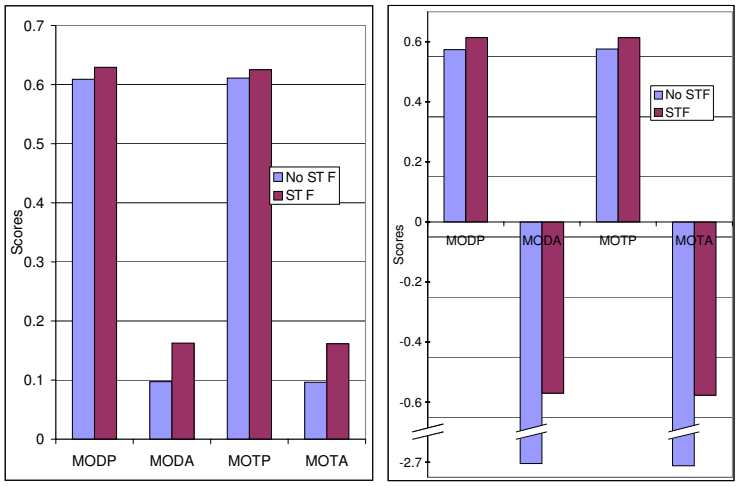


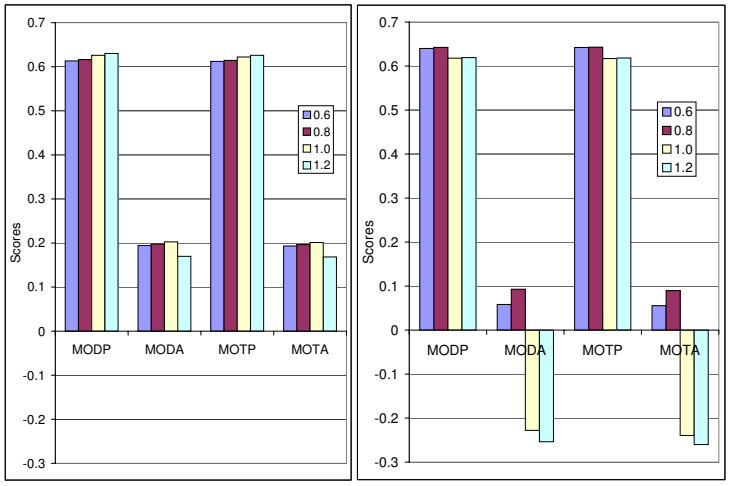
Figure 3.25: Sample tracking results with different update factors for the background model. (Row 1) Change detection mask. (Row 2) Detection and tracking results. A reduction of false positives is observed by increasing update factor (from left to right). When update factor is set to 0.005 no false positives are returned at the cost of one false negative which is a parked vehicle.



(a) (b)

Figure 3.26: Comparison of tracking results with and without spatio-temporal filtering (STF). (a) Pedestrian tracking, (b) vehicle tracking. The scores show a significant improvement especially in terms of accuracy.

results are enhanced using spatio-temporal filtering (STF). Figure 3.26 shows the comparative results using pixel wise temporal filtering. The accuracy and precision improved due to STF by 2.13 and 0.04, respectively, for vehicle detection and by 0.07 and 0.04,



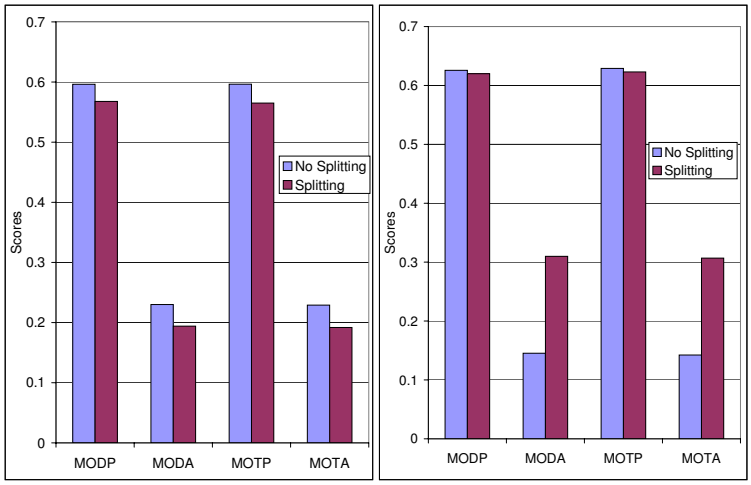
(a) (b)

Figure 3.27: Comparison of tracking results by changing the model parameter of the sensor noise $(0.6, 0.8, 1.0, 1.2)$. (a) Pedestrian tracking. (b) Vehicle tracking.

respectively, for pedestrian tracking. This improvement in accuracy indicates that STF has helped reducing the flickering objects generated due to noise as well as reinforcing the change produced due to weak and small objects.

The frame differencing result, after being improved through pre-processing (Fig. 3.8) including STF, is then passed through a classification process to obtain a binary image. The classification depends upon the σ of the sensor noise. The higher the value of σ the more the pixels are classified as noise. Figure 3.27 shows the impact of σ on vehicle and pedestrian tracking. The value $\sigma = 1.0$ produces better results for pedestrians but also an important performance decrease in terms of accuracy for vehicle tracking. Note here that we empirically divide the image into three horizontal regions and apply three different multipliers to σ, namely 0.75, 1.0 and 1.25. The actual value of σ is used in the vertical centre of the scene whereas the multiples 0.75 and 1.25 are used on the top and bottom regions of the scene respectively. The top region tends to contains targets in far field hence lower σ values and vice versa.

The binary mask obtained may contain merged objects for which the proposed approach applies blob splitting using projection histograms. Figure 3.28 shows the tracking performance comparison with and without the use of the projection histogram based blob splitting. The impact of this procedure on the scores is biased by the vehicle-pedestrian classification. As the classification depends on the width-height ratio of the bounding boxes, the splitting allows assignment of the correct label to groups of pedestrians and



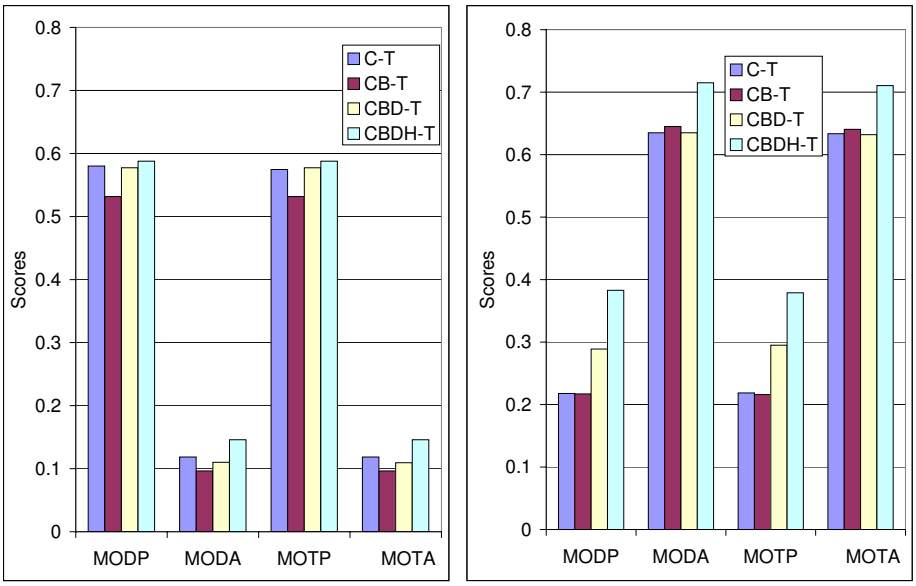
(a) (b)

Figure 3.28: Comparison of tracking results obtained by splitting the blobs associated to more than one target using projection histograms. (a) Pedestrian tracking: small decrease in the scores. (b) Vehicle tracking: large accuracy improvement.

(a) (b)

Figure 3.29: Example of over splitting of pedestrian. (a) Without splitting. (b) With splitting.

therefore the accuracy of vehicle tracking (MOTA) increases by 0.16 (Fig 3.7). However, some of the pedestrians that had elongated shadows were also split into multiple detections resulting in an increase in the number of false positives (Fig 3.29). This resulted in a reduction in detection and tracking accuracies for pedestrians. To evaluate the benefits introduced by different features in graph based tracking and the improvements introduced particularly by colour histograms, four configurations have been compared: C-T, the baseline system with centre of mass only; CB-T the system with centre of mass and bounding box; CBD-T, the system with centre of mass, bounding box and direction; and CBDH-T, the proposed system with all the previous features and the appearance model based on colour histograms (Sec. 3.3).

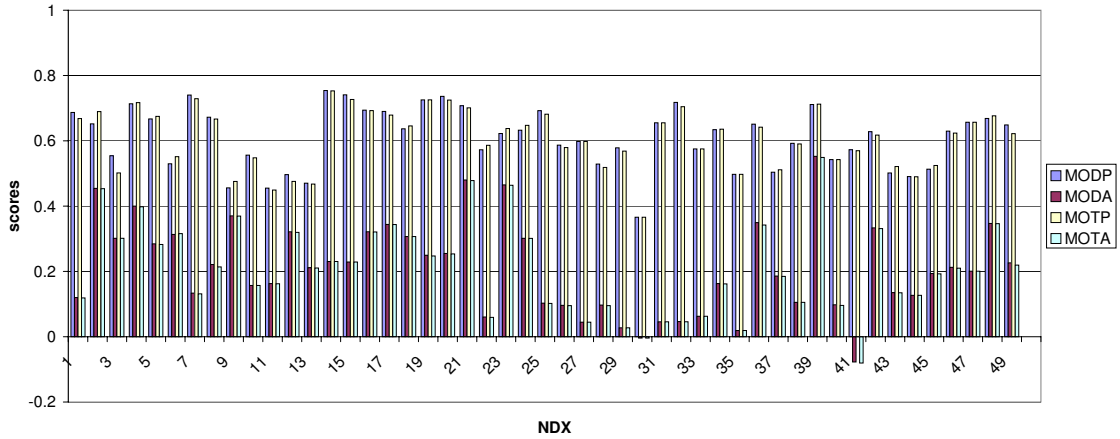


(a) (b)

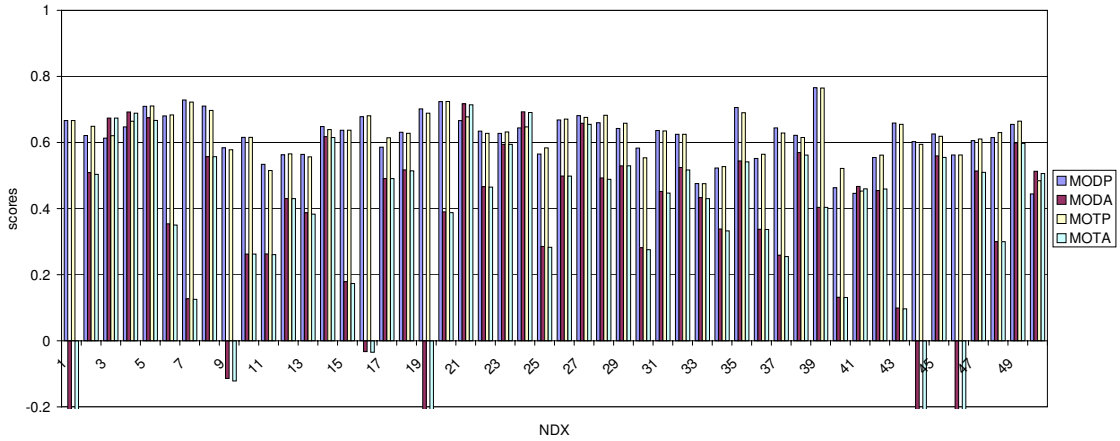
Figure 3.30: Comparison of objective results using different sets of features for detection and tracking on the Broadway/Church scenario, from the CLEAR dataset (C-T: centre of mass only; CB-T: centre of mass and bounding box; CBD-T: centre of mass, bounding box and direction; CBDH-T, the proposed system with all the previous features and the appearance model based on colour histograms). (a) Score for person detection and tracking. (b) Score for vehicle detection and tracking.

The parameters used in the simulations were the same for all scenarios. For colour features, a 32-bin histogram is used for each colour channel. The weights used for graph matching (set empirically) were: $\omega_1 = 0.40$ (position), $\omega_2 = 0.30$ (direction), $\omega_3 = 0.15$ (histogram), $\omega_4 = 0.15$ (size), and $\varpi = 0.043$.

Scores obtained with the different combinations of features are shown in Fig. 3.30. The results on the 4 scores show that the proposed algorithm (CBDH-T) produces a consistent improvement, especially in the case of vehicle tracking. This performance is not surprising as vehicles tend to have more distinctive colours than pedestrians. The use of direction as a feature improves detection and tracking precision more than detection and tracking accuracy (Fig. 3.30 CBD-T vs. CB-T). The higher values of detection and tracking accuracy (MODA and MOTA) in case of Fig. 3.30 (b), as compared to precision values is having only 2-3 annotated vehicles in the interval of the scene used for evaluation. These vehicles were detected in most of the frames resulting in very low number of missed detections. Furthermore, the illumination conditions also remained unchanged during the selected interval, hence resulting in very few false detections. Having both low FP and



(a)



(b)

Figure 3.31: CLEAR evaluation scores for detection and tracking, showing the overall performance on the entire dataset. (a) Person detection and tracking. (b) Vehicle detection and tracking. (NDX: index number corresponding to the sequence name and frame span).

FN resulted in higher accuracy. The lower precision on the other hand could be due to slightly enlarged detections from the actual objects as a result of pre- and post-processing (Fig. 3.8).

The final evaluation scores on the CLEAR dataset for vehicle and person detection and tracking are shown in Fig 3.31(a-b). The accuracy of all except 2 out of 50 sequences for person detection and tracking have positive scores whereas 6 out of 50 sequences for vehicle detection and tracking have positive scores and indicates the successful detection and tracking over a major portion of the evaluation dataset. These failures in person detection and tracking are due to the large number of missed detections (Fig. 3.32), whereas those in vehicle detection and tracking are due to the large numbers of false de-

Figure 3.32: Sample person detection and tracking failure modes on the CLEAR Broadway/Church (BC) and Queensway (QW) scenarios showing failures due to missed detections. (a,c) Ground truth and Detection and tracking result of sequence id 30. (b,d) Ground truth and Detection and tracking result of sequence id 41.

Table 3.2: Average evaluation scores for QW and BC sequences

			MODP	**MODA**	**MOTP**	**MOTA**
Without Post-Processing	**QW**	**Person**	0.2929	-0.8612	0.2946	-0.8673
		Vehicle	0.5187	-0.1102	0.5265	-0.1243
	BC	**Person**	0.4979	0.1918	0.5006	0.1875
		Vehicle	0.6092	-0.2967	0.6108	-0.3033
With Post-Processing	**QW**	**Person**	0.5885	0.1063	0.5852	0.1051
		Vehicle	0.5946	0.4289	0.6005	0.4252
	BC	**Person**	0.6210	0.2659	0.6208	0.2649
		Vehicle	0.6342	0.2505	0.6356	0.2480

tections that are generated due to missclassification of groups of pedestrians as vehicles (Fig. 3.33). Figure 3.33 shows sample failure modes for each of these sequences with negative accuracy.

Table 3.2 shows the comparison between average scores for the Queensway (QW) and Broadway/Church (BC) scenarios obtained with and without post-processing (Fig. 3.2.2). Improvement in both precision and accuracy can be seen particularly in accuracy, which has gone from negative values to much higher positive scores. In general there is an improvement of approximately 0.20 to 0.50 precision and accuracy scores. The comparison of the evaluation scores with [32] which is among the highest-scoring algorithms in the CLEAR evaluation is shown in Table 3.3. In [32] a classifier for person detection is used and hence it is less prone to errors due to classification. The MODA and MOTA of the proposed approach has suffered due to classification error, which has resulted in several

Figure 3.33: Sample vehicle detection and tracking failure modes on the CLEAR Broadway/Church (BC) and Queensway (QW) scenarios showing failures due to missclassification of pedestrians as vehicles. (a,d) Ground truth and Detection and tracking result of sequence id 1. (b,e) Ground truth and Detection and tracking result of sequence id 9. (c,f) Ground truth and Detection and tracking result of sequence id 16. (g,j) Ground truth and Detection and tracking result of sequence id 19. (h,k) Ground truth and Detection and tracking result of sequence id 44. (i,l) Ground truth and Detection and tracking result of sequence id 46.

consistent tracks being misclassified resulting in decreasing the scores (Fig 3.33(h,k)). Note each misclassification generates false positive for one class and missed detections in another and hence has double effects on the scores. Despite this, the result shows that the approach proposed in this work has better detection and tracking precision in the case of vehicles, whereas the precision for person detection and tracking is comparable with [32].

Table 3.3: Comparison of proposed approach with another highest ranking submission in the CLEAR evaluation

		MODP	**MODA**	**MOTP**	**MOTA**
Proposed	**Person**	0.6103	0.2076	0.6085	0.2064
	Vehicle	0.6200	0.3147	0.6230	0.3118
[32]	**Person**	0.6194	0.5148	0.6230	0.4988
	Vehicle	0.6020	0.6790	0.6160	0.6400

Evaluation using ETISEO dataset and metrics

The ETISEO evaluation consisted of 5 tasks, namely detection, localisation, tracking, classification and event recognition. The evaluation on the first three tasks is discussed here whereas, the classification task is out of the scope of this work. The evaluation of event recognition is discussed in Chapter 6 where we extend our detection and tracking framework for interaction recognition.

Table 3.4 compares the scores of the proposed approach with the mean, variance, minimum and maximum scores of the ETISEO evaluation. These scores are computed by the evaluation organisers without disclosing the information about the other algorithms, hence only a high-level analysis is performed here. The proposed approach has obtained maximum scores in case of the airport (AP-11) sequence, whereas for the road (RD-6) sequence the scores are above the mean scores. The reason for not having highest scores is that several objects that never move in the scene such as parked vehicles and contextual objects like door and door control are also part of the annotated ground truth. This is because in this work we do not address the problem of segmenting contextual objects hence the scores are affected. This has significantly affected the performance in the case of the BE-19 scenarios. Figure 3.34 shows the frame by frame evaluation for BE-19 scenario using all 4 metrics. The decrease in precision during frames 188 to 291 in Fig. 3.34(a) is due to false detection caused by the reflections of the car on the glass door. The lower sensitivity after frame 291 is due to missed detection of building and car doors (Fig. 3.34(g)). However, the total area detected is similar to that in the ground truth annotations as indicated by the higher precision and sensitivity scores in Fig. 3.34(c). This can be seen in Fig. 3.34(e) which shows that several objects in the ground truth are detected as merged objects by the algorithm. Although the layered background subtraction is successful in segmenting the person coming out of the car and the car door (Fig. 3.34(h)), however they are segmented as one object and hence the merge score is reduced. The BE-19 C4 sequence is an indoor scenario and is mostly covered by contextual objects (Fig. 3.34(i)), such as the building door, and hence the scores are

Table 3.4: Comparison of proposed approach with other submission in ETISEO evaluation

	Task	Mean	Var	Min	Max	Proposed
AP-11-C4	Detection	0.59	0.16	0.34	0.95	0.95
	Localisation	0.74	0.10	0.61	0.97	0.97
	Tracking	0.69	0.14	0.44	0.89	0.89
AP-11-C7	Detection	0.63	0.16	0.40	0.99	0.97
	Localisation	0.78	0.80	0.70	0.97	0.94
	Tracking	0.73	0.15	0.39	0.92	0.86
BE-19-C1	Detection	0.71	0.10	0.45	0.84	0.79
	Localisation	0.86	0.50	0.77	0.92	0.89
	Tracking	0.51	0.90	0.38	0.66	0.48
BE-19-C4	Detection	0.34	0.12	0.18	0.56	0.31
	Localisation	0.66	0.70	0.56	0.78	0.69
	Tracking	0.41	0.90	0.26	0.51	0.42
RD-6-C7	Detection	0.63	0.10	0.45	0.86	0.70
	Localisation	0.79	0.5	0.73	0.96	0.81
	Tracking	0.68	0.10	0.52	0.86	0.73

Table 3.5: Average merge and split score on ETISEO evaluation dataset

	AP-11-C4	**AP-11-C7**	**BE-19-C1**	**BE-19-C4**	**RD-06-C7**
Split	1.00	1.00	0.92	1.00	1.00
Merge	0.98	1.00	0.79	0.49	0.89

significantly lowered due to missed detections (Fig. 3.34(b,d,f)). The doors are initially detected by change segmentation as the person steps out of the building (Fig. 3.34(j)), during frames 188 to 291, resulting in higher precision and sensitivity values but a lower merge score. However, as the person walks away from the building, the doors are again misdetected until frame 820, after which the person coming out of the car tries to enter the building. Table 3.5 shows the average merge and split scores for all the 5 sequences of the ETISEO evaluation dataset. The split score is 1 (the maximum possible) in four sequences, which indicates that the approach consistently detects each object as a single blob, instead of multiple splitted blobs. The merge score is also above 0.89 in all except the BE-19 sequences, where there are missed detections due to contextual objects.

Qualitative analysis and failure modes

The qualitative analysis of tracking results on the CLEAR dataset using the CBDH-T features is shown in Fig. 3.35. Detected objects are identified by a colour-coded

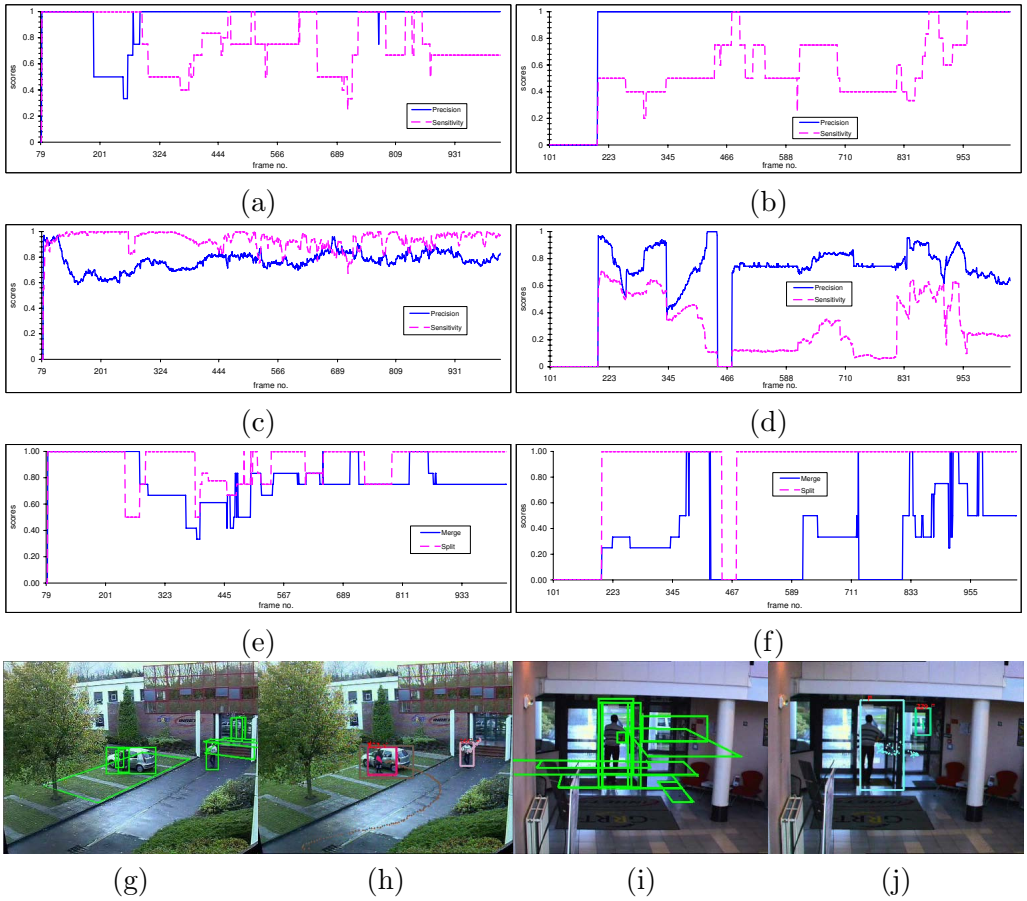


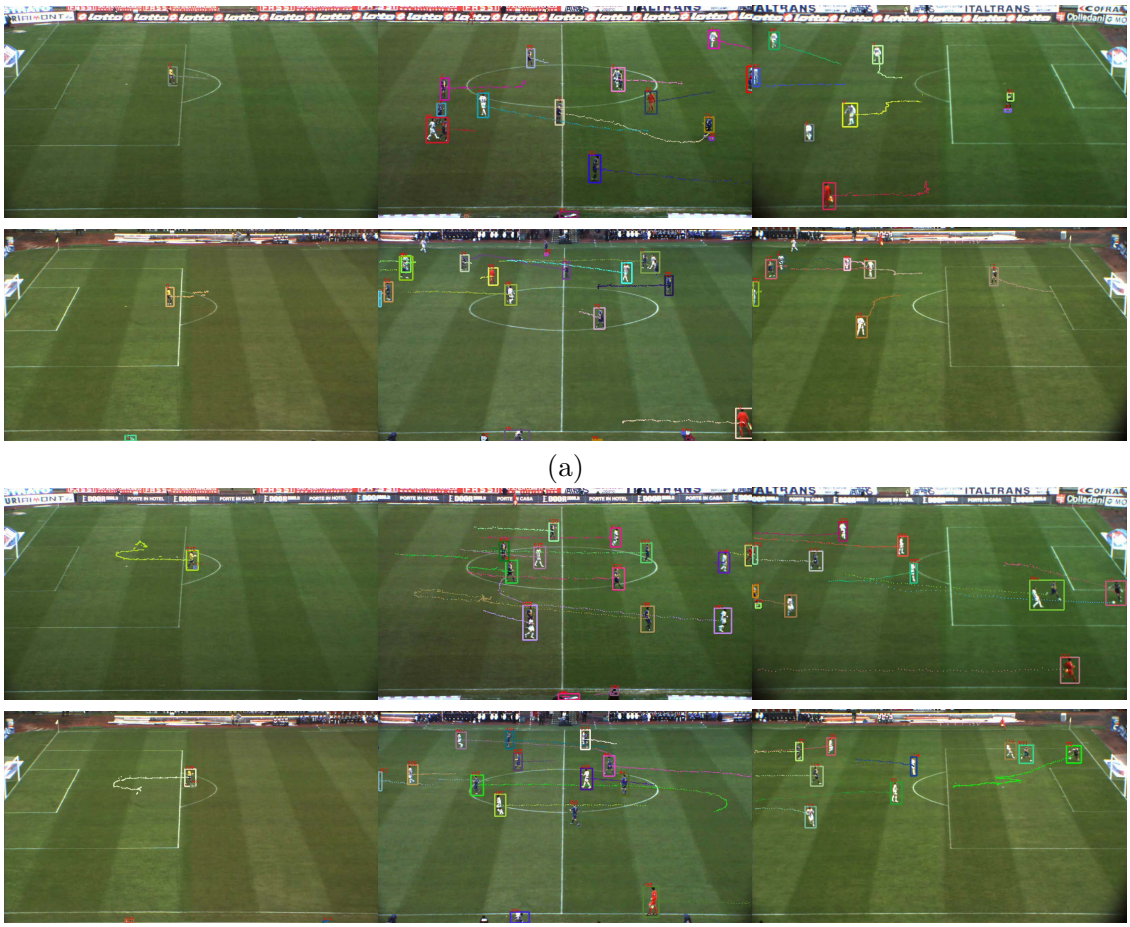
Figure 3.34: Precision, sensitivity, merge and split scores of ETISEO metrics on BE-19 scenario. (Left) BE-19 C1; (right) BE-19 C4. (a-b) Precision and sensitivity of the number of detected objects. (c-d) Precision and sensitivity of the area of detected objects. (e-f) Merge and split scores. (g,i) Ground truth annotations on frame 450 of C1 and 240 of C4. (h,j) Result of proposed approach on frame 450 of C1 and 240 of C4.

bounding box, their respective trajectories and an object ID (top left of the bounding box). The results of the classification into one of the two classes, namely pedestrian (P) and vehicles (V), are shown on the top of the bounding box. The results are shown for all three scenarios of the CLEAR dataset, namely QW, BC and BR under different illumination conditions. The quantitative analysis of detection and tracking results on the ETISEO data can be seen in Fig. 3.34(h,j), in Table 3.5 and in Chapter 6, along with results on other standard datasets, whereas Fig. 3.36 shows the qualitative results on all 6 cameras of the IISA dataset.

Figure 3.37 shows three failure modes of the proposed tracker. In Fig. 3.37 (a-b) two objects are merged and the use of the projection histogram based splitting does not help as the objects are not merged along the horizontal axis. A possible solution could be

Figure 3.35: Sample tracking results using the proposed detection and tracking algorithm (CBDH-T) on the CLEAR dataset. (Row 1) Broadway/Church (BC), (row 2) Queensway (QW) and (row 3) Brannigans (BR).

the use of a body part detector to estimate the number of targets in a blob. Figure 3.37 (a-b) shows missed detections caused by object merging. Figure 3.37(c-d) shows a failure due to small object size and partial occlusion. To overcome this problem, information from multiple cameras could help disambiguating the occlusion.



(a)

(b)

Figure 3.36: Sample single camera tracking results in the IISA dataset showing 6 camera-views of the soccer field. (a) Frame 160. (b) Frame 1275.

(a) (b)

(c) (d)

Figure 3.37: Sample failure modes on the CLEAR Broadway/Church (BC) and Queensway (QW) scenarios (red boxes indicate the areas of the frame where the failures occurred). (a-b) Merged objects. (c-d) Missed detections caused by small objects size and partial occlusion.

Computational cost

The computational cost of each pre- and post-processing block in image based-localization (Fig 3.8) and graph-based tracking is shown in Fig 3.38. The figure shows the cost of a C/C++ implementation for the main algorithmic steps, namely absolute frame differencing, spatio-temporal filtering, edge analysis, contrast adjustment, pixel classification, background learning, morphology and connected component analysis and graph-based tracking. The computational cost is computed per frame in milliseconds using an input video of resolution 960×544 on a Intel Core 2 Quad CPU having speed of 2.39 GHz and 3.25GB RAM. It can be seen that pixel classification takes 363.98 milliseconds (68.15%) for each colour image. The processing time for all the other modules is less then 53 milliseconds with tracking using graph-matching only taking 18.38 milliseconds (3.4% of the time) with approximately 20 objects per frame. This shows that the complete algorithms is working at 2 frames per second (fps). The tracking alone can go up to 55 fps.

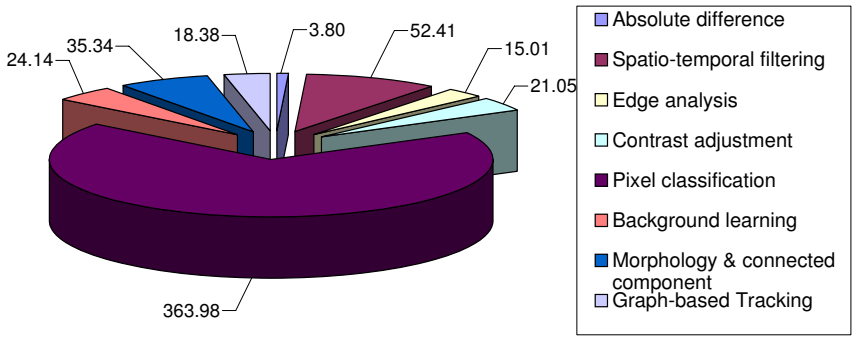


Figure 3.38: Average per frame computation cost in milliseconds for various steps in image-based localisation and graph-based tracking on colour video of resolution 960×544.

3.6.4 Multi-camera tracking

The quantitative evaluation of the proposed multi-target track-before-detect particle filter (MT-TBD-PF) is performed on APIDIS dataset by using two types of detection volume (meta-sensor): (i) using 5 camera fusion and (ii) using 7 camera fusion. The goal of this experiment is to show that similar performance can be achieved without using the top mounted cameras with fish eye lenses, as these types of cameras are generally not available in most real surveillance and sport scenarios. Experiments on each of the two detection volumes were performed using five different sampling periods. This makes a total of 10 experiments, each of which is performed three times and from where a mean precision (Eq. 3.36) and sensitivity (Eq. 3.37) is calculated.

The parameter values used in the experiments were as follows: process noise in target motion and intensity (Eq.(3.15)) were set to be $q_1 = 2.5$ and $q_2 = 0.001$ respectively. These values allow tracking of targets under low SNR values. In this dataset several targets start very close together and cross each other after some interval. To obtain individual non-merged tracks without false detections, the value chosen for the minimum target weight was $\lambda_\omega = 10^{-5}$ whereas the thresholds for mean distance and variance, for cluster merging (Sec. 3.5.2) were $\lambda_\mu = 1$ and $\lambda_A = 2$. The bandwidth chosen for mean-shift (Eq.(3.31)) was $h = 5$, which is appropriate for clustering particles generated around a target that is affected by a blurring (Eq.(3.17)) with $A = 0.3733$ and $(\Delta_x, \Delta_y) = (1, 1)$. The σ value for the likelihood computation was set to 0.3. The tracking was done using 3000 particles per target. The same parameters were used for all the experiments on both APIDIS and IISA datasets. The only exception was that for the IISA dataset we set $q_2 = 0.1$ because in this dataset, there is much less overlap between the cameras hence increase uncertainty in target intensity.

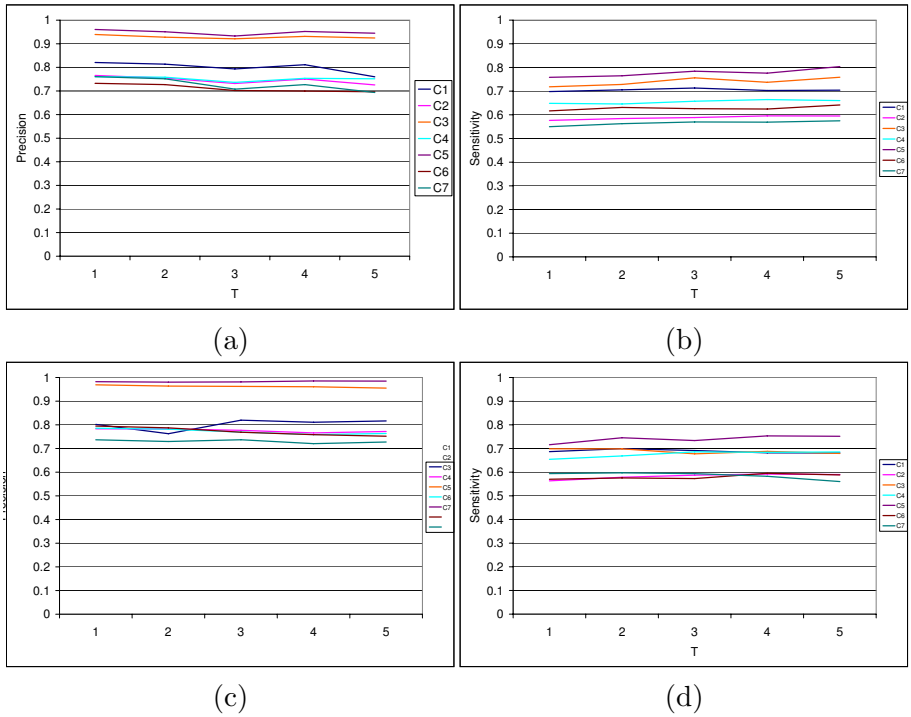


Figure 3.39: Precision and sensitivity scores for cameras C1-C7 of APIDIS dataset. (a-b) Precision and sensitivity of tracking results generated using detection volume obtain from 5 cameras (without 2 top mounted fish eye lenses). (c-d) Precision and sensitivity of tracking results generated using detection volume obtain from 7 cameras including the 2 with fish-eye lens. (τ: sampling interval indicating frame distance).

The evaluation of the proposed approach on the APIDIS dataset is shown in Fig. 3.39. The tracks generated on the top-view are first reprojected onto each camera-view and then evaluated against the ground truth. Precision and sensitivity are computed for results obtained from both 5 camera detection (without 2 top mounted cameras with fish eye lenses) volume and 7 camera detection volume at 5 different sampling intervals (τ). These scores were computed after projecting tracks on each camera-view. It can be seen that both precision and sensitivity have similar values at different sampling intervals. This indicates the stability of the proposed algorithm under lower frame rates, which is due to Eq. 3.15 where the process noise in target motion is defined as a function of sampling interval. The sensitivity is increased by 1.69% for tracking results on 7 camera detection volumes (Fig. 3.39). The slight increase in sensitivity could be due to an increase in information through 2 additional cameras as these cameras are top-mounted and has less perspective distortion. However, this slight increase indicates that similar tracking results can be obtained in cases where the top-view is not available as in most of the real multi-camera setups. The shift in precision and sensitivity for different cameras is due to

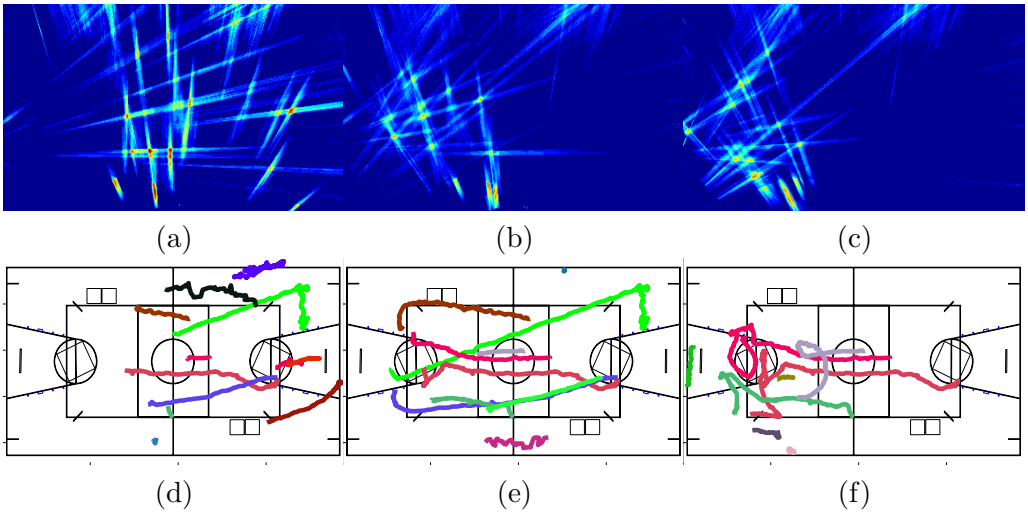


Figure 3.40: Sample fusion and multi-target tracking results on the top view for frames 500, 590 and 765 of the APIDIS dataset. (Row 1) Detection volume on the top-view. (Row 2) Tracking results obtained with the proposed approach.

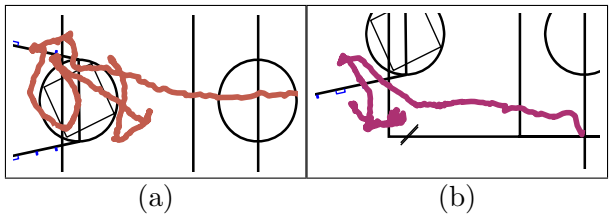


Figure 3.41: Example tracks of high manoeuvring targets. (a) 5 manoeuvres (b) 3 manoeuvres.

difference in error in reprojection on the image plane.

A visualisation of results from the proposed approach on APIDIS data is given in Fig. 3.40 on a schematic of a basket ball court for clarity. The projection of the detection mask on the top-view using multi-layer homography is shown in Fig. 3.40(a-c). Several issues regarding the data can be observed from these results. First, all the targets are not represented by the same level and spread of intensity values (Fig. 3.19(b)). Some targets have very low visibility without a significant amount of spread among neighbouring pixels, whereas others have high intensity values which vary over time. The parallax error can also be easily observed due to which targets have different amount of noisy spread of intensity values in different regions. The shift in intensity values, primarily due to an increase in camera's overlap, is also clearly visible in these projections. These issues make the less visible targets challenging to track.

Furthermore, most of these targets manoeuvre highly as these are players who rapidly change their paths based on the location of the ball and flow of the game (Fig 3.41).

Although we used a constant velocity model, the proposed tracker can still handle manoeuvring targets as we model acceleration with a higher value than in the synthetic experiments. Furthermore, the distribution $p(\mathbf{x}_k|Z_k)$ generates a number of new-born particles proportional to the measurement and also helps coping with manoeuvring targets. The high value of q_1 also allows the tracker to quickly concentrate around new-born targets which usually do not start with initial zero velocity. However, this also increases the spread of particles around the target and results in target merging. This merging was minimised by using the kernel bandwidth $h = 5$ for mean-shift as in the case of synthetic targets. The remaining parameters were the same as for the synthetic data. These parameters are valid for the sub-sampled version of the data having resolution of 388×225. The generated tracks appear smooth due to using 3000 particles per target. Using fewer particles can result in jerky tracks.

The tracking results obtained through the proposed multi-target particle filtering track-before-detect (MT-PF-TBD) technique on the IISA dataset are shown in Fig. 3.42(d-f). It can be seen that most of the targets are tracked over the entire field. The exception being the goal keeper on the left corner of the field. This goal keeper is not tracked initially (Fig. 3.42(d)) despite there being significant information in the detection volume (Fig. 3.42(a)). The reason is that initially this goal keeper is static and hence does not follow the expected motion model. The particle prediction resulted in moving all particles away from the target and hence low weights during the update step, followed by removal of these particles during the resampling step. The track of this target is generated as it starts moving during the attack on the goal (Fig. 3.42(e-f)).

The computational cost of meta-sensor creation and multi-target track-before-detect particle filter (MT-TBD-PF) tracking is shown in Fig 3.43. The figure shows the cost of a C/C++ implementation for image-based localisation and Matlab implementation of projection and fusion and MT-TBD-PF. The computational cost is computed per frame in milliseconds using an 6 colour input video of resolution 960×544 and meta-sensor of resolution 492×288 on a Intel Core 2 Quad CPU having speed of 2.39 GHz and 3.25GB RAM. It can be seen that generation of meta-sensor takes approximately 85% of the processing time whereas MT-TBD-PF takes remaining 15% of the time which is approximately 19 seconds per frame with 20 targets and 3000 particles per target. The major bottle neck in MT-TBD-PF is the update step which takes approximately 17 seconds per frame. This large computation time can be greatly reduced by having an efficient C/C++ implementation on Graphical Processing Units (GPU's) [71].

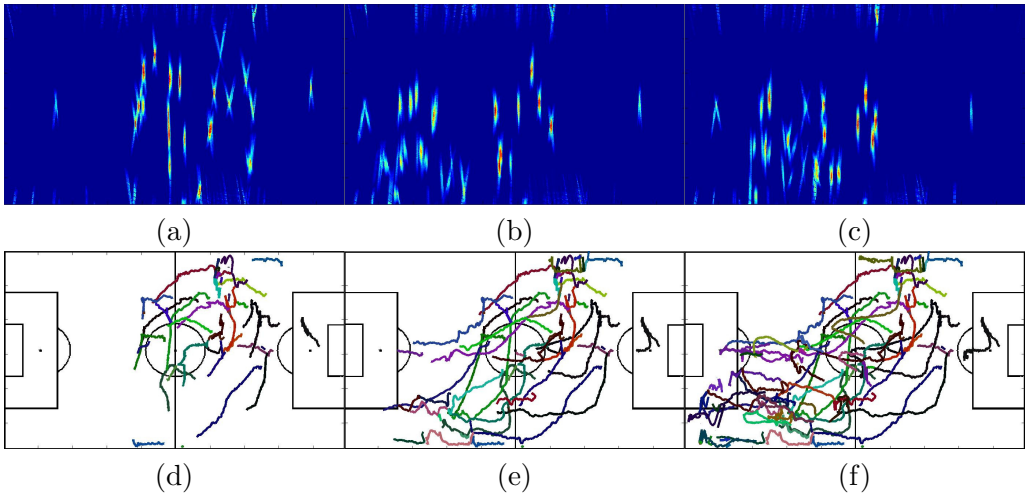


Figure 3.42: Sample fusion and multi-target tracking results on top view for frames 150, 240 and 350 of IISA dataset. (a-c) Detection volume on the top-view. (d-f) Tracking results obtained with the proposed approach.

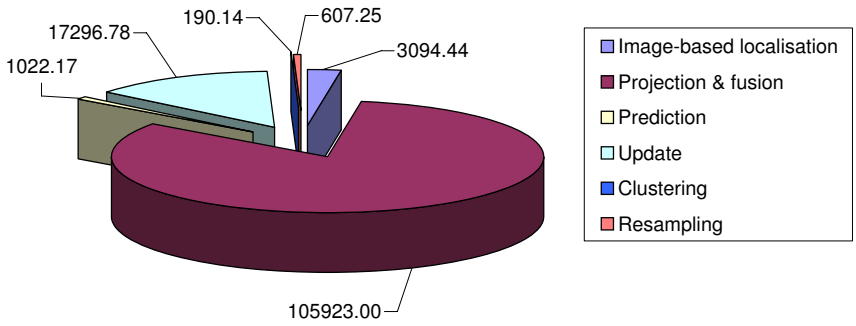


Figure 3.43: Average per frame computation cost in milliseconds for image-based localisation, projection and fusion on top view and track-before-detect tracking on colour video from 6 cameras having resolution 960×544.

3.6.5 Future experiments

In case of image-based localisation the major problem to be addressed is *how can we reduce the number of missed detections?*. The missed detections are caused by objects being (i) small and in the far field of the camera; (ii) having similar appearance as the background; (iii) are under dark regions of the scene (under shadow of a tree or a building); (iv) being partially or fully occluded. The first 3 issues are related to σ of the noise and the learning rate of the adaptive background update. Using a very small value for both parameters can significantly increase the number of false positives. A set of experiments with different set of values for these two parameters and generation of ROC curves can help in better understanding the optimal values. The fourth issue is

partial occlusion resulting in objects that have a different shape. Analysis of shapes of object from different views and during different levels of estimated occlusion and merging can be done to generate a set of training samples. These samples can then be used to train a classifier that can detect different occlusion scenarios and can assist in correctly segmenting multiple merged and partially occluded objects [146].

Multi-camera tracking is resource consuming and an important question to be answered is *how many cameras and in what configuration can provide results similar to the ones obtained by the maximum possible number of cameras.* Considering the configuration of the fixed cameras as in the case of basketball and soccer datasets used in this thesis, the possible set of experiments could be to try all combinations of cameras to find the answer.

Another question to be addressed is *how can we improve the performance of the proposed multi-target track-before-detect particle filtering algorithm.* The proposed MT-TBD-PF has three limitations: (i) it requires large number of particles; (ii) the spread of the particles is affected by the shape of the peaks in the meta-sensor and (iii) the number of targets in the scene is not modelled. The number of particles needed can be reduced by applying a hybrid approach [6] whereby mean-shift is used to sample towards the nearby local maxima. The mean shift based optimisation could be applied for each identified cluster in the resampling step. To analyse the effect of peaks in meta-sensor on the spread of particles it is important to test with different kernel shapes in the mean-shift clustering to understand which kernel achieves the optimal results. It should be noted that the shape of the peaks also depends upon the camera combinations and configurations. For example, in case of tracking on the meta-sensor using 7 cameras of basketball dataset (Fig. 3.13(g)), a circular kernel may be appropriate, whereas in case of soccer where cameras were placed in front of each other, an elliptic kernel may be more appropriate. To model the number of targets in the scene, as discussed in Sec. 3.5.2 the jump Markov Models [147] can be used. The experiments should be performed to evaluate if tuning the target's birth probability in the proposed MT-TBD-PF can reduce the number of false detection and comparison should be performed with the jump Markov Model.

3.7 Summary

In this chapter we have presented an approach for multi-target detection and tracking using single and multiple cameras. We have addressed several challenges in the object detection step and showed that false detections due to local illumination changes can be filtered using edge analysis and spatio-temporal filters. Contrast and brightness

adjustment can be used to eliminate false detections due to rapid global illumination changes, stopped objects can be detected using a layered background model and merged objects can be split using a projection histogram.

We have also proposed an extension to graph-based tracking by using colour histograms. The proposed single camera detection and tracking approach has been evaluated on real sport and surveillance datasets and compared against ground truth and another state of the art approach. The evaluations done for single camera detection and tracking indicate that both approaches, based on motion-based localisation and classifier-based detectors, are comparable in terms of performance. Since the evaluation has been done per class (vehicle and person), the classifier-based algorithm had an implicit advantage. However, it is not general enough to be applied for the detection and tracking of general types of target, including those that may not have any distinguishing appearance information available. The fusion of these two types of localisation techniques can help in improving performance when the goal of the fusion is to validate the detections for identification of false positives and to reduce the number of false negatives.

In the later part of the chapter we proposed a detectionless multi-target tracking algorithm that can track low-visibility targets under noise. The approach does not perform hard thresholding of data, which is one of the major limitations of detection-based tracking algorithms. The proposed technique is applied to multiple sensors where localisation information from each sensor is first fused to obtain a meta-sensor and then only the pixels following a target model are tracked without applying any detection mechanism (*track-before-detect*). This not only eliminates the detection step after data fusion but also implicitly helps in reducing false positives due to noise. It opens a new direction in multi-sensor detection and tracking where the idea being, given a meta-sensor obtained through multi-sensor fusion, we can perform detectionless tracking without any hard thresholding. The algorithm is implemented using sampling importance resampling Particle filtering where we proposed cluster based update and resampling to deal with multiple objects. We have shown that the proposed approach (MT-TBD-PF) can perform at various sampling intervals without readjustment of the parameters.

In the next chapter we will address the problem of multi-modal localisation and tracking, by introducing audio sensors into the problem.

Chapter 4

Multi-modal tracking

4.1 Introduction

Visual modality is likely to fail in situations where there may be prolonged visual occlusions, such as under vegetation or night scenarios. The use of audio can provide additional information in such cases. Most of the techniques found in literature on audiovisual tracking have used large arrays of microphones to track targets in smart meeting rooms [4, 92]. However, in real surveillance scenarios, the use of a large number of microphones is not feasible due to cost and synchronisation issues. In this work we explore the use of stereo microphones for audio source localisation. We call our sensors *Stereo Audio and Cycloptic Vision* (STAC) sensors, each consisting of a camera mounted between two microphones. A single pair of microphones can only provide 1D information about object location, however multiple sets of stereo microphones can provide 2D position information using triangulation.

The problem of multiple audiovisual object tracking can be formalised as a continuous estimation, from audio and video observations, of the state $\mathbf{x}_k$ of each target at time k. At any time k, one of the following conditions is possible: (i) a complete audiovisual observation is available, (ii) only the sound cue is available, or (iii) only the visual cue is available. In case of using STAC sensors, the major issue related to using only 2 microphones as compared to larger arrays is its sensitivity to noise and reverberation. We discuss below how to improve the estimation accuracy of $\mathbf{x}_k$ by fusing audiovisual information using STAC sensors in a particle filtering framework.

The proposed approach first post-processes the audio signal to filter noise prior to estimation of the arrival angle of sound. The arrival angle estimates are then smoothed by applying Kalman filtering. This smoothed audio source localisation estimate is then

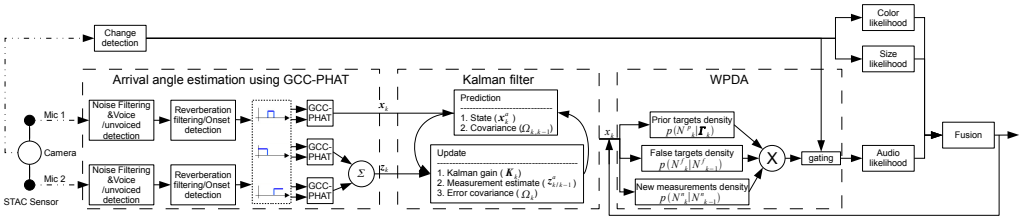


Figure 4.1: Block diagram of the proposed audiovisual tracking algorithm.

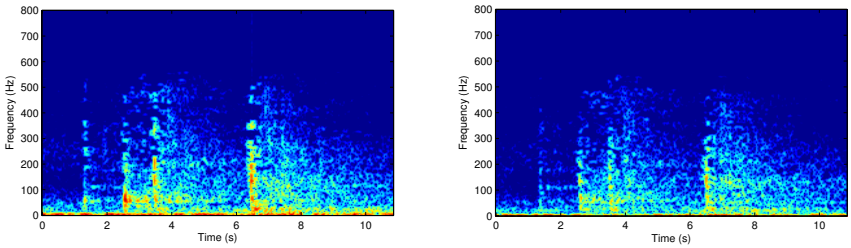
fused with the visual modality under Weighted Probabilistic Data Association (WPDA) and then tracked using particle filtering. Figure 4.1 shows the high level flow diagram of the proposed single STAC tracking algorithm. To perform tracking using multiple STAC sensors, an *audio-audio fusion* mechanism is introduced in the algorithm.

In Sec. 4.2.1 and Sec. 4.2.2 we will discuss our proposed post-processing and audio source localisation. Multi-modal tracking using a single audiovisual sensor is discussed in Sec. 4.3 where we first smooth the estimated arrival angle of sound using a Riccati Kalman filter (Sec. 4.3.1). To further reduce the effect of noise we propose a Weighted Probabilistic Data Association scheme discussed in Sec 4.3. Tracking using multiple STAC sensors is then discussed in Sec. 4.4. Finally, in Sec. 4.5 the proposed approach is evaluated and results are presented followed by the chapter summary in Sec. 4.6.

4.2 Audio source localisation

Consider a single STAC sensor, its two microphones measure the acoustic signals at different time instants as discussed in Sec. 2.3.2 and shown in Fig. 2.9. Let us assume that one target at a certain time emits a sound and that the sound is generated in the direction of the microphones. Then the Generalised Cross-correlation Phase Transform (GCC-PHAT) can be utilised to compute the delay τ of the wave between the reference microphone, M_1, and the second microphone, M_2 (Sec. 2.3.2). However, due to noise and reverberation angular estimates θ (computed using τ) can be erroneous. Hence we pre-process the audio signals before the estimation of the Time Difference of Arrival (TDOA) τ.

The pre-processing involves filtering of the audio frames containing *unvoiced signal segments*, *background noise* or *reflected components* of the source signal. Similar to background modelling in video [38], we assume that the first 200ms of the audio signal contains changes that are due to noise only and use it to compute the statistics of the noise



(a) (b)

Figure 4.2: Sample spectrograms from (a) microphone 1 (M_1) and (b) microphone 2 (M_2). The signal at M_2 is delayed and attenuated.

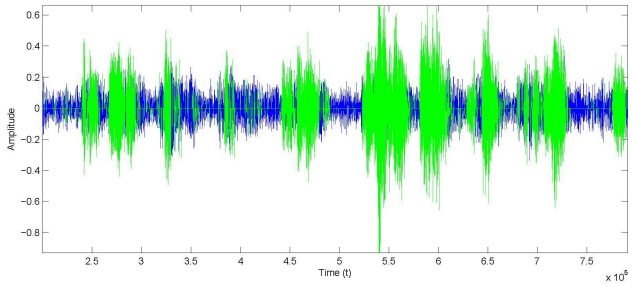


Figure 4.3: Example of voiced/unvoiced segment detection for meeting sequence (M1) using $\alpha = 0.8$. The original signal is shown in solid blue line and the filtered signal is shown in dotted green line.

particularly its mean μ and standard deviation σ. Next we filter *unvoiced segments* by analysing the zero-crossing rate [148]. For each windowed audio segment, zero-crossings are counted and the mean, μ, and the standard deviation, σ, are used to define a high zero-crossing rate as $\varsigma > \mu_{zc} + \omega_{zc}\sigma_{zc}$, where ς is the zero-crossing rate for the windowed segment and ω_{zc} is a weight dependent on the sensor and the environment. Based on this threshold, unvoiced signal segments are filtered, as shown in Fig. 4.3. We then perform reverberation filtering and multi-band analysis to reduce the effect of background noise and reverberation as discussed next in Sec 4.2.1 and Sec. 4.2.2 respectively.

4.2.1 Reverberation filtering

Due to reflections large number of echoes (delayed components of the original signal) exist that decay in amplitude until they can no longer be heard and are called reverberation. To reduce the reverberation effect we estimated the GCCF-PHAT $\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}$ only on an ensemble of frames that are classified as onset. Onset frames are frames containing a significant signal component and a limited or absent reverberation component

caused by the signal itself. These onsets are located at the beginning of a signal audio block (the audio segment between two salient segments of the audio signal). A frame is considered a signal frame if the SNR at both microphones is larger than a noise threshold. Using the same initial 200ms interval, we compute the signal noise level as

$$l_{\mathcal{N}_j} = \frac{1}{N^a \; N^s} \sum_{k=1}^{N^a} \sum_{m=1}^{N^s} \hat{\mathbf{y}}_{jk}[m], \quad with \; j = 1, 2 \tag{4.1}$$

where N^a is the number of audio frames, N^s is the number of samples in a frame and $\hat{\mathbf{y}}_{jk}$ is the audio frame from the j^{th} microphone containing 1764 (0.04 seconds at 44.1KHz) samples. The noise level is then used to detect onset frames with significant signal component without reverberation. The apparent location of a sound source largely depends on the initial onset of the sound, a phenomenon known as the precedence effect or law of the first wavefront [149–151].

Let $l_{\hat{\mathbf{y}}_{jk}}$ be the signal levels computed as:

$$l_{\hat{\mathbf{y}}_{jk}} = \frac{1}{N^s} \sum_{m=1}^{N^s} \hat{\mathbf{y}}_{jk}[m], \quad with \; j = 1, 2. \tag{4.2}$$

The frames are considered as onset frames if $l_{\hat{\mathbf{y}}_{jk}} > \omega_{\mathcal{N}} \times l_{\mathcal{N}_j}$, where $\omega_{\mathcal{N}}$ is the noise weight. After each onset detection, the next $T = 6$ frames are considered as signal component while the rest are assumed to be due to reverberation, and hence ignored until a null frame ($l_{\hat{\mathbf{y}}_{jk}} \leq \omega_{\mathcal{N}} \times l_{\mathcal{N}_j}$) is detected. Assuming that the frame under analysis, $\hat{\mathbf{y}}_k$, is the first frame of an onset y_k^O, the subsequent T frames are processed if identified as signal frames; whereas the signal frames from $\hat{\mathbf{y}}_{k+T}$ to the first *null* frame are considered reverberant frames and therefore discarded.

4.2.2 Multi-band analysis

Some materials have higher absorptivity at high frequencies, whereas others may have higher absorptivity at lower frequencies [152]. This implies that the effects of correlated noise, located in a single frequency band, can be reduced by evaluating the signal in different frequency bands [153]. We estimate the arrival angle using a *multi-band frequency analysis* to further reduce any residual reverberation effect. The two audio signals $\hat{\mathbf{y}}_{jk}, \; j = 1, 2$ are divided into three different frequency bands. The angular estimates are conducted in the low $(0 - 400\text{Hz})$ (B_1), middle $(400 - 960\text{Hz})(B_2)$ and high-frequency $(960 - 1600\text{Hz})(B_3)$ bands. The frequency band division is computed using three different 36-coefficient band-pass linear phase FIR filters, frame-by-frame, for onset frames. The

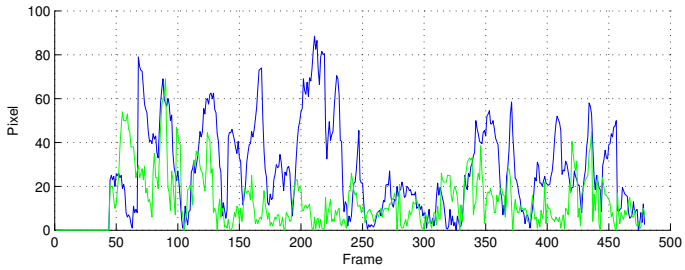


Figure 4.4: Deviation from the ground truth for source localisation using the GCC-PHAT transform (solid blue line) and the proposed method (dotted green line).

cross-correlation function is then estimated for each frequency band. The final estimation of the GCC is obtained by a weighted combination of the three sub-band cross-correlations as

$$\hat{R}_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f) = \sum_{i=1}^{3} \omega_i \frac{G^i_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)}{\gamma|G^i_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)| + (1-\gamma)|\mathcal{N}^i(f)|^2}, \tag{4.3}$$

where $G^i_{\hat{\mathbf{y}}_1\hat{\mathbf{y}}_2}(f)$ is the cross power spectral density function in band B_i, $\gamma \in [0, 1]$ and $\mathcal{N}^i(f)$ is the noise spectral density in band B_i. $\mathcal{N}^i(f)$ is estimated during the initialisation assuming stationary noise. The weights ω_i ($\sum_{i=1}^{3} \omega_i = 1$) are chosen such that higher frequency components contribute less than the low frequency ones. A peak is retained if it is simultaneously located in the same position in the three GCCs. Peaks that appear in a single band only are reduced proportional to the weight associated. The resulting improvements compared to the plain GCC-PHAT is measured by computing the absolute difference between estimated and true x position of the target and can be seen in Fig. 4.4. The green line shows the distance between the ground truth and the results obtained with the proposed approach. The solid blue line shows the distance between the ground truth and the GCC-PHAT (Eq. 2.25) result. It can be seen that error for the proposed system (dotted green line) is much smaller than that of the GCC-PHAT (solid blue line) result.

4.3 Multi-modal tracking using single audiovisual sensor

In this section we will discuss the smoothing of the estimated arrival angle and its fusion with video modality and tracking. The smoothing is applied using a Riccati Kalman filter and is discussed next in Sec. 4.3.1. To further eliminate the effect of noise, Weighted Probabilistic data association is proposed in Sec. 4.3.2 followed by a particle filter for audiovisual trajectory estimation. The flow diagram of multi-modal tracking using single STAC sensors is shown in Fig. 4.5.

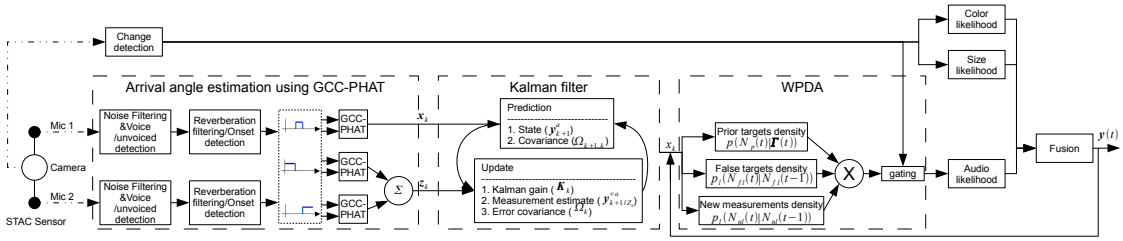


Figure 4.5: Block diagram of the proposed audiovisual tracking algorithm.

4.3.1 Riccati Kalman filter

To determine the audio state $\mathbf{x}_k^a = (\theta_k, \dot{\theta}_k)$ i.e., the arrival angle θ_k and the rate of change of arrival angle $\dot{\theta}_k$ using the estimated delay τ we apply a *Riccati Kalman filter* as it offers better performance in noisy environments [154]. To apply Kalman filtering, we first divide the three frequency bands into two groups, namely Group 1 (middle frequency band) and Group 2 (low and high frequency bands). This division is done as some materials have higher absorptivity at high frequencies, whereas others may have higher absorptivity at lower frequencies [152]. Let the state and the observation vectors represent Group 1 and Group 2, respectively. This means that the audio state vector is obtained from the middle frequency band whereas the observation vector is obtained from the low and high frequency estimates.

The Kalman filter works with a state space model consisting of a process and an observation equation:

$$\begin{cases} \mathbf{x}_k^a = \mathbf{A}\mathbf{x}_{k-1}^a + \mathcal{N}_{1k} \\ \mathbf{z}_k = \mathbf{C}\mathbf{x}_k^a + \mathcal{N}_{2k} \end{cases}, \tag{4.4}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \tag{4.5}$$

and $\mathbf{A}$ is the state transition model and $\mathbf{C} = [1, 0]$ is the observation model. $\mathcal{N}_{1k}$ and $\mathcal{N}_{2k}$ are two noise terms that are assumed to be zero mean Gaussian random vectors with covariance matrices defined by

$$\Psi(\mathcal{N}_{jk}\mathcal{N}_{jk}^T) = \begin{cases} \mathbf{Q}_{jt} & \text{for } k = t \\ 0 & \text{otherwise} \end{cases}, \tag{4.6}$$

where $j = 1, 2$; $\mathbf{Q}_1$ is the covariance matrix of the process noise, and $\mathbf{Q}_2$ is the covariance matrix of the measurement noise. $\mathcal{N}_{1k}$ and $\mathcal{N}_{2k}$ are statistically independent and therefore $\Psi(\mathcal{N}_{jk}\mathcal{N}_{jk}^T) = 0$ for all $t \neq k$.

Let $\hat{\mathbf{x}}_{k|k-1}^a$ be the predicted state estimate of $\mathbf{x}_k^a$ deduced from all observations $Z_{k-1}^a = \mathbf{z}_m^a | m = 1, \cdots, k-1$ up to time $k-1$. The predicted observation is then expressed as

$$\hat{\mathbf{z}}_{k|k-1}^a = \mathbf{C}\hat{\mathbf{x}}_{k|k-1}^a. \tag{4.7}$$

The innovation is the difference between the actual and predicted observations:

$$\beta_k = \mathbf{z}_k^a - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}^a. \tag{4.8}$$

The correlation matrix of the innovation sequence is:

$$\mathbf{P}_k = \mathbf{C}\Omega_{k|k-1}\mathbf{C}^T + \mathbf{Q}_{2k}, \tag{4.9}$$

and the covariance matrix $\Omega_{k|k-1}$ is defined as

$$\Omega_{k|k-1} = \mathcal{E}[(\mathbf{x}_k^a - \hat{\mathbf{x}}_{k|k-1}^a)(\mathbf{x}_k^a - \hat{\mathbf{x}}_{k|k-1}^a)^T]. \tag{4.10}$$

The Kalman gain is defined as

$$\mathbf{K}_k = \mathbf{A}\Omega_{k|k-1}\mathbf{C}^T\mathbf{P}_k^{-1}. \tag{4.11}$$

To compute the Kalman gain, we need to estimate $\Omega_{k|k-1}$, which is

$$\begin{cases} \Omega_{k|k-1} = \mathbf{A}\Omega_{k-1}\mathbf{A}^T + \mathbf{Q}_{1k} \\ \Omega_k = [\mathbf{I} - \mathbf{A}\mathbf{K}_k\mathbf{C}]\Omega_{k|k-1} \end{cases}, \tag{4.12}$$

where $\mathbf{I}$ is the identity matrix. Finally, the state estimate can be updated according to the Kalman gain and innovation [155], that is

$$\hat{\mathbf{x}}_{k|k}^a = \mathbf{A}\hat{\mathbf{x}}_{k|k-1}^a + \mathbf{K}_k\beta_k. \tag{4.13}$$

The audio estimates are fused with visual estimates before multi-modal tracking can be applied (fuse-before-track). The fusion is applied within particle filtering at the likelihood level. The likelihood is computed independently for visual and audio modalities.

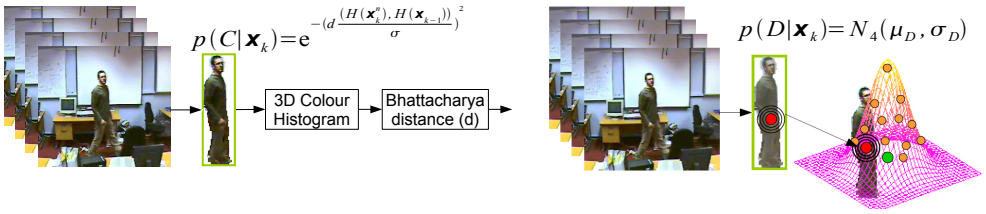


Figure 4.6: Visual likelihood computation (a) Colour likelihood using Bhattacharyya distance (b) Motion likelihood using multi-variate Gaussian.

The visual likelihood is composed of two cues, a colour measurement (Fig. 4.6(a)) and a motion measurement (Fig. 4.6(b)). The *colour likelihood*, $p(\mathcal{C}|\mathbf{x}_k)$, is computed in the *RGB* colour space using 3D colour histograms $\mathcal{H}$, uniformly quantised with $10\times10\times10$ bins, as

$$p(\mathcal{C}|\mathbf{x}_k) = \exp\left(-\frac{d\left(\mathcal{H}(\mathbf{x}_k^n),\mathcal{H}(\mathbf{x}_{k-1})\right)}{\sigma}\right)^2, \tag{4.14}$$

where $\mathbf{x}_k^n$ is the candidate histogram at position defined by n^{th} particle, $\mathbf{x}_{k-1}$ is the reference histogram at position defined by target state at time $k-1$, σ is the standard deviation and $d(.)$ is the distance based on the Bhattacharyya coefficient, computed as

$$d\left(\mathcal{H}(\mathbf{x}_k^n),\mathcal{H}(\mathbf{x}_{k-1})\right) = \sqrt{1-\sum_{u=1}^{N^b}\sqrt{\mathcal{H}_u(\mathbf{x}_k^n)\cdot\mathcal{H}_u(\mathbf{x}_{k-1})}}, \tag{4.15}$$

where $N^b = 1000$ is the number of bins and values for each u^{th} bin are calculated as

$$\mathcal{H}(\mathbf{x}_k^n) = B\sum_i \mathcal{K}\left(\left\|\frac{\mathbf{x}-\mathbf{w}_i}{h}\right\|^2\right)\delta\left(b(\mathbf{w}_i)-u\right), \tag{4.16}$$

where $\mathbf{w}_i$ are the pixels of the target and $b(\mathbf{w}_i)$ associates each $\mathbf{w}_i$ to its histogram bin [109]. The elliptic kernel $\mathcal{K}(.)$ with bandwidth h is used to lower the weight of the pixels that are closer to the border of the target. The normalisation factor B ensures that the sum of the bins is 1.

The *motion likelihood*, $p(\mathcal{D}|\mathbf{x}_k)$, is computed as distance from the results of a change detector. The motion likelihood from the detection is finally computed by applying a multi-variate Gaussian comprising of 4 dimensions as

$$p(\mathcal{D}|\mathbf{x}_k^n) = \mathcal{N}_4(\mathbf{x}_k^n,\mu_\mathcal{D},\sigma_\mathcal{D}), \tag{4.17}$$

where $\mu_{\mathcal{D}}$ is the position and size of the detected target and $\sigma_{\mathcal{D}}$ is the standard deviation.

The audio, $\mathcal{A}$, likelihood, $p(\mathcal{A}|\mathbf{x}_k)$, is computed using a univariate Gaussian:

$$p(\mathcal{A}|\mathbf{x}_k) = \frac{1}{\sigma_{\mathcal{A}}\sqrt{2\pi}} \exp\left(-\frac{\hat{\mathbf{x}}^a_{k|k-1} - \mathbf{x}^a_{k-1}}{2{\sigma_{\mathcal{A}}}^2}\right). \tag{4.18}$$

The audio and visual cues are fused in the particle filter as product of the audio and visual likelihoods [93]. The overall likelihood is computed as

$$p(\mathbf{z}_k|\mathbf{x}_k) = p(\mathcal{D}|\mathbf{x}_k)p(\mathcal{C}|\mathbf{x}_k)p(\mathcal{A}|\mathbf{x}_k), \tag{4.19}$$

where c is a constant, $\mathbf{z}_k$ is the observation, $p(\mathcal{D}|\mathbf{x}_k)$ is the motion likelihood, $p(\mathcal{C}|\mathbf{x}_k)$ is the colour likelihood, and $p(\mathcal{A}|\mathbf{x}_k)$ is the audio likelihood. When one modality is unavailable, its likelihood is set to 1.

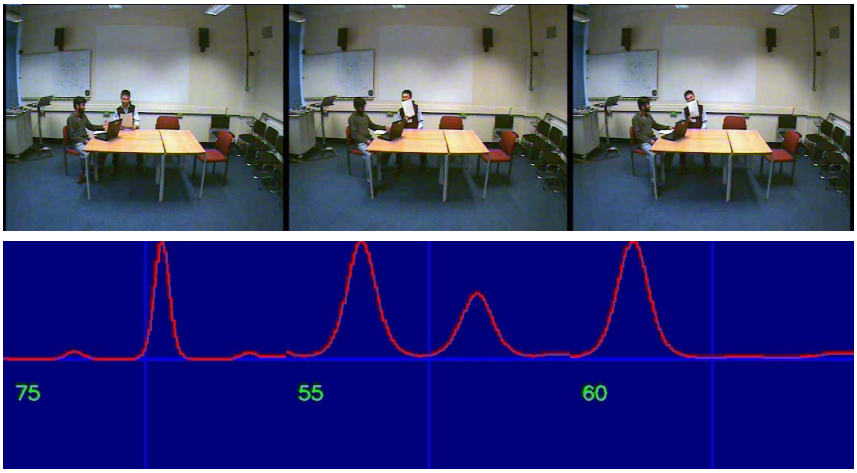
Once $p(\mathbf{z}_k|\mathbf{x}_k)$ is computed, the weights are set proportional to the likelihood (Eq. 3.27). The final estimation of the state $\mathbf{x}_k$ at time k is computed based on the discrete approximation of Eq. 3.20 using the Monte Carlo approximation of the expectation:

$$\mathbf{E}(\mathbf{x}_k|\mathbf{z}_k) \approx \frac{1}{N^{pt}} \sum_{i=1}^{N^{pt}} \omega^i_k \mathbf{x}^i_k, \tag{4.20}$$

where N^{pt} is the number of particles.

4.3.2 Weighted probabilistic data association

Classical particle filters [156] attempt to solve the tracking problem by finding an approximate state $\mathbf{x}_k$ on the basis of the previous and current observations Z_k, which contains visual and audio features. Although the Kalman filter reduces the localisation discrepancy, the estimated GCC peaks can deviate from the real audio source positions due to various noise components. As a result, the performance of the entire multi-modal tracker will deteriorate, especially in the presence of adjacent objects (Fig. 4.7). To minimise the discrepancy between the real and the estimated positions, we propose a strategy that associates the hypotheses and the measurements with a real target, using a Weighted Probabilistic Data Association (WPDA) algorithm. Unlike PDA and Joint-PDA, WPDA takes into account a weighted probability of the detections in each iteration to increase the importance of reliable audiovisual measurements, based on the prior estimates and on validation data, and further weaken the unreliable hypotheses. The correspondence between the audio and the video modality is done using a Gaussian reliability window.



(a) (b) (c)

Figure 4.7: Examples of audio source mislocalisation. (a): The speaker on the left is talking, but the peak indicates the person on the right. (b) and (c): The speaker on the right is talking, but the peak indicates the person on the left.

Only the measurements falling within this region are considered to be valid.

Let $\mathbf{p}_k$ denote the probability of the prediction $\mathbf{\Gamma}_k$, given the measurements Z_k up to time k:

$$\mathbf{p}_k \propto p(\mathbf{\Gamma}_k|\mathbf{z}_k). \tag{4.21}$$

The prediction $\mathbf{\Gamma}_k$ can be obtained based on the prior $\mathbf{\Gamma}_{k-1}$ and the association hypotheses Ξ_{k-1} for the current measurements [62]. Ξ_{k-1} associates each measurement $\mathbf{z}_k$ with a target. $\mathbf{p}_k$ is intractable due to the unknown association. Instead, we can estimate $p(\mathbf{\Gamma}_k, \Xi_k|\mathbf{z}_k)$ using the Bayes' rule as

$$p(\mathbf{\Gamma}_k, \Xi_k|\mathbf{z}_k) = c_1 p(\mathbf{z}_k|\mathbf{\Gamma}_k, \Xi_k) p(\Xi_k|\mathbf{\Gamma}_k) p(\mathbf{\Gamma}_k), \tag{4.22}$$

where c_1 is a normalising factor and $p(\mathbf{z}_k|\mathbf{\Gamma}_k, \Xi_k)$ is the likelihood of the measurements and can be expressed assuming independence as

$$p(\mathbf{z}_k|\mathbf{\Gamma}_k, \Xi_k) = cp(\mathcal{D}|\mathbf{x}_k) p(\mathcal{C}|\mathbf{x}_k) p(\mathcal{A}|\mathbf{x}_k), \tag{4.23}$$

The second term of the right hand side of Eq. 4.22, $p(\Xi_k|\mathbf{\Gamma}_k)$, is the probability of a current data association hypotheses, given the previous prediction and estimation. Let N_k^p, N_k^f, and N_k^n be the number of measurements associated with the prior, false and new targets respectively. Considering a binomial distribution for N_k^p and the positive side

of a Gaussian distribution for N_k^f and N_k^n, we can express $p(N_k^p, N_k^f, N_k^n|\mathbf{\Gamma}_k)$ as

$$\begin{aligned} p(N_k^p, N_k^f, N_k^n|\mathbf{\Gamma}_k) = & \\ = \; & p(N_k^p|\mathbf{\Gamma}_k)p(N_k^f, N_k^n|N_k^p, \mathbf{\Gamma}_k) \\ = \; & p(N_k^p|\mathbf{\Gamma}_k)p(N_k^f|N_k^p, N_k^n, \mathbf{\Gamma}_k) \times \\ & \times p(N_k^n|N_k^p, N_k^f, \mathbf{\Gamma}_k), \end{aligned} \tag{4.24}$$

where

$$p(N_k^p|\mathbf{\Gamma}_k) = \begin{pmatrix} N_k^t \\ N_k^d \end{pmatrix} p_k^{d,N_k^d}(1-p_k^d)^{N_k^t - N_k^d}, \tag{4.25}$$

where N_k^t and N_k^d are the numbers of previously known and currently detected targets, respectively. p_k^d can be determined using its current probability p_k^{md} and prior probability p_{k-1}^{md}:

$$p_k^d = \begin{cases} p_k^{md} & \text{if } p_{k-1}^{md} \leq p_k^{md} \\ p_{k-1}^{md} & \text{otherwise} \end{cases}, \tag{4.26}$$

$$p_k^{md} = \frac{p(\mathcal{D}^m|\mathbf{x}_k^m)p(\mathcal{C}^m|\mathbf{x}_k^m)p(\mathcal{A}^m|\mathbf{x}_k^m)}{\sum_{m=1}^{N_k^M} p(\mathcal{D}^m|\mathbf{x}_k^m)p(\mathcal{C}^m|\mathbf{x}_k^m)p(\mathcal{A}^m|\mathbf{x}_k^m)}, \tag{4.27}$$

where N_k^M is the number of the measurements obtained by different sensors. The monotonic increasing nature of Eq. 4.26 ensures that once the target is selected as the speaking target it remains such (as the probability does not decrease) unless the other target has higher probability of detection as speaker.

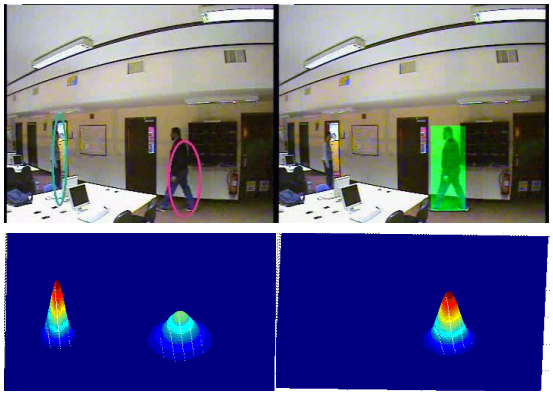
This strategy considers the probabilities of the previous and current measurements in addition to a normalised likelihood for the contribution of the different sensors. The target with the highest probability in a group of candidates will be the one associated with the track. Due to the contribution of previous measurements, this strategy can minimise the identity switches when the available measurements are inaccurate due to noise or errors.

The second and third terms of the right hand side of Eq. 4.24 can be expressed as

$$p(N_k^f|N_k^p, N_k^n, \mathbf{\Gamma}_k) \propto p(N_k^f|N_{k-1}^f) \tag{4.28a}$$

Algorithm 1 WPDA Algorithm

1: Create samples for the target states $\mathbf{x}_k^m$;
2: Compute the posterior distributions $p(\mathbf{z}_k|\mathbf{\Gamma}_k, \Xi_k)$ and $p(\Xi_k|\mathbf{\Gamma}_k)$ using Eqs. (4.23)-(4.24);
3: Compute the joint association probability $p(\mathbf{\Gamma}_k, \Xi_k|\mathbf{z}_k)$ using Eq. 4.22;
4: Calculate the marginal association probability as $\gamma = \sum_{m=1}^{N_k^M} p(\mathbf{\Gamma}_k^m, \Xi_k^m|N_k^m)$;
5: Generate the target likelihood: $p(\mathbf{z}_k|\mathbf{x}_k) = \Pi_{m=1}^{N_k^M} \gamma_m p(N_k^m|\mathbf{x}_k^m)$;
6: Update the particle weights using Eq. 3.21;
7: Apply resampling for each target to avoid the degeneracy of the particle sets.



(a) (b)

Figure 4.8: Sample images from the proposed audiovisual tracker. (Row 1): position estimation using (a) video and (b) audio features; (Row 2): Likelihood of the measurements: (a) visual measurements (two persons), (b) audio detection showing the speaker under the green patch associated to the change detection bounding box. The horizontal and vertical axes of the graphs in the second row correspond to the width and the height of the image, respectively. (Note that the images are tilted for improved visualisation).

and

$$p(N_k^n|N_k^p, N_k^f, \mathbf{\Gamma}_k) \propto p(N_k^n|N_{k-1}^n), \tag{4.28b}$$

where

$$p(N_k^f|N_{k-1}^f) \approx \Pi_{m=1}^{N^M} p^m(N_k^{f,m}|N_{k-1}^{f,m}). \tag{4.29}$$

The right hand side of Eq. 4.28(a-b) is modelled as a Gaussian distribution. The mean and variance of these distributions are computed based on N^f and N^n, respectively such that its mean is at the previous estimate (N_{k-1}^n or N_{k-1}^f). The variance of these distribution is set empirically. The main steps of the WPDA algorithm for each frame within the particle filter framework are summarised in Algorithm 1 and a sample output is shown in Fig. 4.8.

4.4 Multi-modal tracking using multiple audiovisual sensor

Consider an environment partially covered by cameras such as a networks of non-overlapping cameras. In such networks the trajectories in regions outside the cameras' field of view can only be estimated using a learned motion model and contextual information. However, these estimates could be erroneous especially in case of non-availability of contextual information. Sensors with a wider field of observance, such as microphones, can be used to address this problem (Fig. 4.9). In this section we extend the localisation and tracking using multi-modal sensors to multiple multi-modal sensors for extended tracking. Here again we consider that the multi-modal network is composed of multiple STAC sensors such that each c^{th} camera be equipped with a microphone pair, with $M = \{M_1, \ldots, M_N\}$ being the set of N microphone pairs, where $M_i = (M_{i1}, M_{i2})$. We assume that the microphones' sound field is wider than the corresponding cameras' field of view and that the sound field of multiple microphone pairs M_i overlap each other (Fig. 4.9).

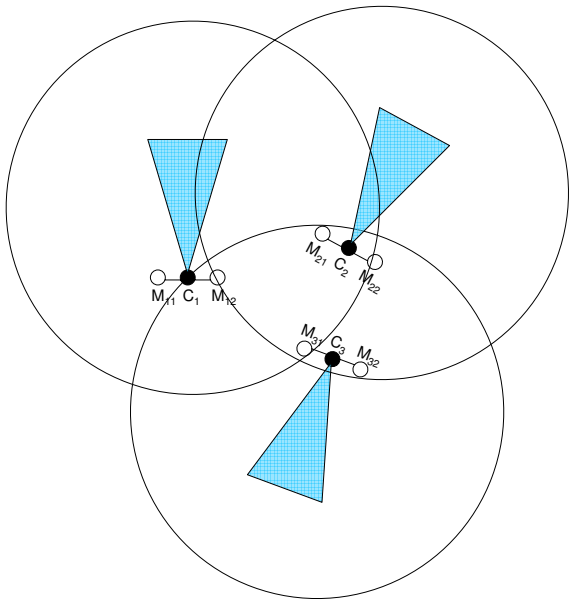
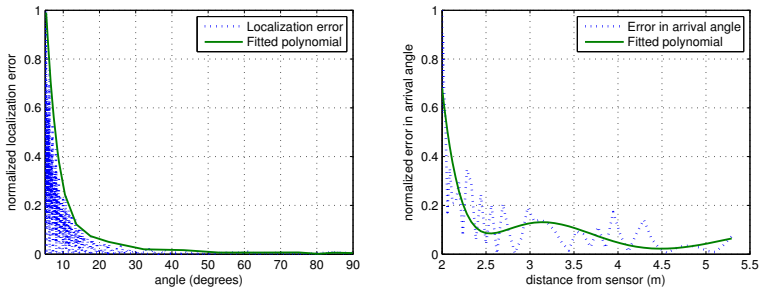


Figure 4.9: Illustration showing audiovisual sensor network with overlapping sound field of observance.

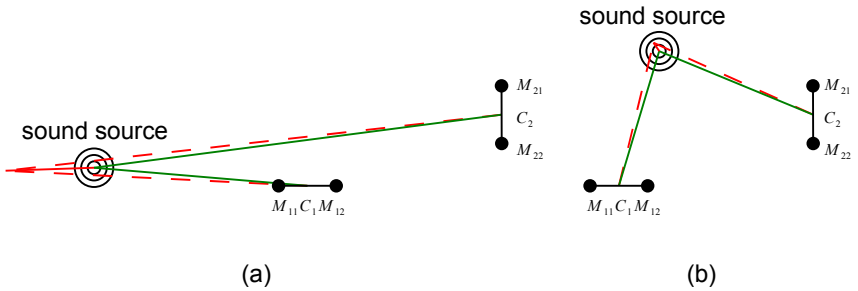
Here the problem is to perform *audio-audio fusion* in regions unobserved by cameras in order to localise target position. The localisation using stereo pairs from multiple STAC sensors can be performed using triangulation. After the estimation of the arrival angle θ^i, a line is projected from the mid-point of the two microphones in the direction θ from each STAC sensor and the intersection of these lines from multiple STAC

sensors gives the target position (Fig. 4.12). However, this localisation can be erroneous and the error increases as $\rho \to 0$ (Fig. 4.10(a), Fig. 4.11) or as $\rho \to 180$, where ρ is the angle between the two intersecting lines. The minimum localisation error is achieved at $\rho = 90$ (Fig 4.11). The estimation done by the pair of STAC sensors for $147^o < \rho$ or $\rho < 33^o$ is ignored and the information from other STAC pairs is used. The audio performance also decreases as the target moves closer than 5m from the sensor as the assumption of parallel sound waves in TDOA estimation will no longer be valid (Fig. 4.10(b)). In case no STAC sensor is able to provide the localisation information, we apply trajectory estimation using the first order motion model as $\mathbf{x}_{k+1} = \mathbf{x}_k + U\nu_k + \mathcal{N}(\mu, \Sigma)$ where $\nu_k = (0, \nu_x, 0, \nu_y)$.



(a) (b)

Figure 4.10: Arrival angle and localisation error analysis. Blue dotted line: error; green solid line: fitted polynomial. (a) Example of increase in the localisation error with the decreasing of the angle between the intersecting lines from two STAC sensors. (b) Example of increase in the error in the arrival angle when the target moves closer than 5m from the sensor. This error is due to the violation of the parallel line propagation assumption.



(a) (b)

Figure 4.11: Example localisation using triangulation showing change in localisation error with angle between intersecting lines. (a) Large localisation error due to noise and interline angle. (b) Small localisation error due to same noise but interline angle close to 90^o. (solid green line: true triangulation; dashed red line: estimated triangulation).

The *audiovisual fusion* is then performed within Kalman filtering by taking a weighted sum of the two measurements as $\gamma\mathbf{z}_i^\nu + (1-\gamma)\mathbf{z}_i^a$ where $\mathbf{z}_i^\nu$ is the measurement from video

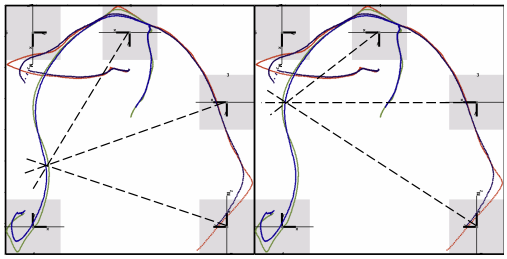
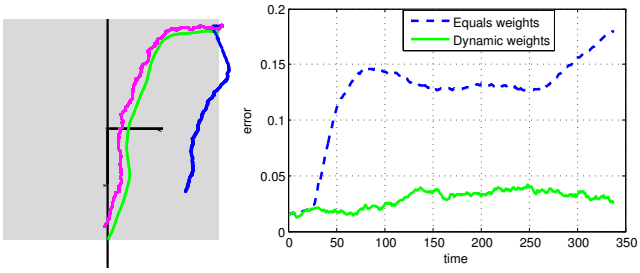


Figure 4.12: Target localisation using TDOA with multiple STAC sensors. Red and green lines: ground truth; blue and black line: estimated trajectories. Grey squares: overlapping regions; black dashed lines: audio source localisation using arrival angles with 3 STAC sensors.



(a) (b)

Figure 4.13: Example and evaluation of audiovisual fusion. (a) Example of audiovisual fusion; (b) Error graph for audio localisation using equal weights and dynamic weights. Grey square: field of view of a camera; green: ground truth; blue: audiovisual fusion with equal weights (i.e. $\gamma = 0.0$ in Eq. 4.30); Magenta: audiovisual fusion with dynamic weights computed using Eq. 4.30.

modality, $\mathbf{z}_i^a$ is the measurement from audio modality and γ is computed as

$$\gamma = \begin{cases} 1 & \text{video only} \\ 0 & \text{audio only} \\ 0.5 + 0.25\left(\psi_d(d_e) + \psi_p(\rho)\right) & \text{otherwise} \end{cases}, \tag{4.30}$$

where $\psi_d(d_e)$ is a 25^{th} order polynomial fitted over the normalised error in the estimation of the arrival angle θ with respect to the Euclidean distance d_e between the target and the microphone pairs (Fig. 4.10(b)) and $\psi_p(\rho)$ is a 9^{th} order polynomial fitted over the normalised error in localisation based on ρ (Fig. 4.10(a)). This weighting mechanism will only penalise audio detections in overlapping regions and will give a weight of at least 0.5 to the video modality, if available. In the cases where these error graphs may not be available, another method of dynamic weight assignment could be through utilising the

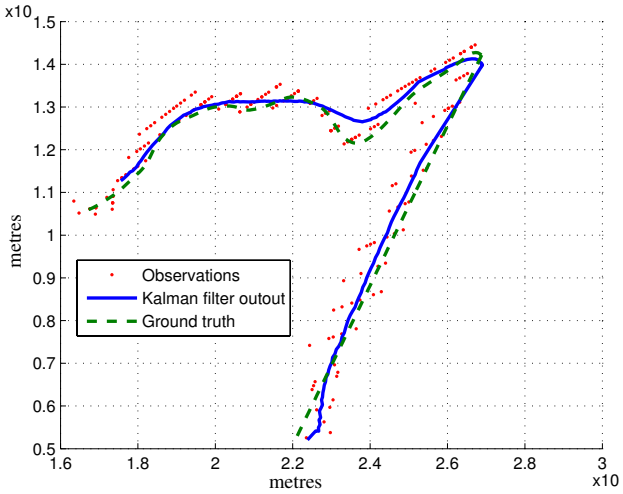


Figure 4.14: Example of audio only trajectory estimation using TDOA followed by correlation and fusion within Kalman filter. Red dots: estimated target position using audio; green dashed line: ground truth; blue solid line: Kalman filter output.

covariance of the Kalman filter. The higher covariance due to uncertainty in tracking can be used to assign lower weights to audio estimates and vice versa.

This weighting has contributed to a 13.17% error reduction where error is measured as the Euclidean distance between the ground truth and the estimated track. Moreover, the error standard deviation has also decreased by approximately 1 decimal place (when evaluated on 50 randomly generated synthetic trajectories, each consisting of 1500 points and a total of 2928 points in the visible region of a single sensor in a network of 3 STAC sensors). Figure 4.13 shows an example of the obtained improvement using this dynamic weighting technique compared to using equal weights for both modalities. The audiovisual fusion dynamic has improved performance as audio estimates are weighted (Eq. 4.30) given the expected error in estimation using the polynomial for arrival angle and localisation. Figure 4.14 shows an example of the results from the proposed audio estimation method.

4.5 Results

4.5.1 Evaluation metrics

For a quantitative evaluation of the tracking using a single audiovisual sensor, we use two scores: ϵ, the one-dimensional *Euclidean distance* between the detected x-coordinates and the ground truth, and N^{LT+IS}, the number of lost tracks N^{LT} and identity

switches N^{IS} over the entire sequence. This score is computed as $N^{LT+IS} = (N^{LT} + N^{IS})/N^{TF}$ where N^{TF} is the total number of frames in the sequence. Hence the lower N^{LT+IS}, the better the performance.

The approach is also compared with six other strategies: (1) vision only by PF; (2) vision only by graph matching (GM) [157]; (3) estimation of arrival angle only before and after Kalman filtering (the former: AB-KF; the latter: AA-KF); (4) GCC-PHAT arrival angle estimation and particle filter based audiovisual tracker (GP-PF) [117]; (5) Kalman filtering audio detection and the particle filter-based audiovisual tracker with PDA (KF-PF-P) [158]; (6) the proposed arrival angle estimation using Kalman filtering and particle filter based audiovisual tracker with WPDA (KF-PF-WP).

The evaluation of tracking using multiple audiovisual sensors is performed by computing the 2D Euclidean distance (ϵ) with the ground truth. To analyse the benefits of audiovisual tracking, we performed the comparison with (i) audio only tracking (TDOA), (ii) video only tracking (CLUTE [159]), (iii) audiovisual fusion using dynamic weighting (AVDW), and (iv) audiovisual fusion using dynamic weighting with trajectory smoothing using Kalman filter (AVKF). The CLUTE algorithm [159] computes the trajectories in unobserved regions by employing forward and backward estimation using Kalman filtering and linear regression. To evaluate the robustness of each algorithm we further performed the test with missing audio observations.

4.5.2 Experimental set-up

The proposed multi-modal detection and tracking algorithm is evaluated using data collected with a STAC sensor composed of two Beyerdynamic MCE 530 condenser microphones and a KOBI KF-31CD camera. The image resolution is 360×288 pixels (25Hz) and the audio is sampled at 44.1 KHz. The audiovisual synchronisation is performed using a VisioWave Discovery 300 Series recorder. The tracker is tested on seven scenarios without occlusions, with video occlusion, and with single and multiple targets (Table 4.1,Fig 4.15). The data consisted of a total of $14,052$ frames and were collected in a reverberant room with significant audiovisual background noise. In real scenarios such as those used in this work the assumptions of parallel line propagation and that the sound is produced in the direction of the microphones (speaker facing the microphones) may not be true at all times. The data used here does have intervals where these assumptions are violated and we will discuss its effect on tracking during such situations.

We present two types of set-up: the first type consist of sequences OC (Fig 4.15(a)), SD (Fig 4.15(b)), L1 and L2 (Fig 4.15(c-d)) in which the distance between the microphones

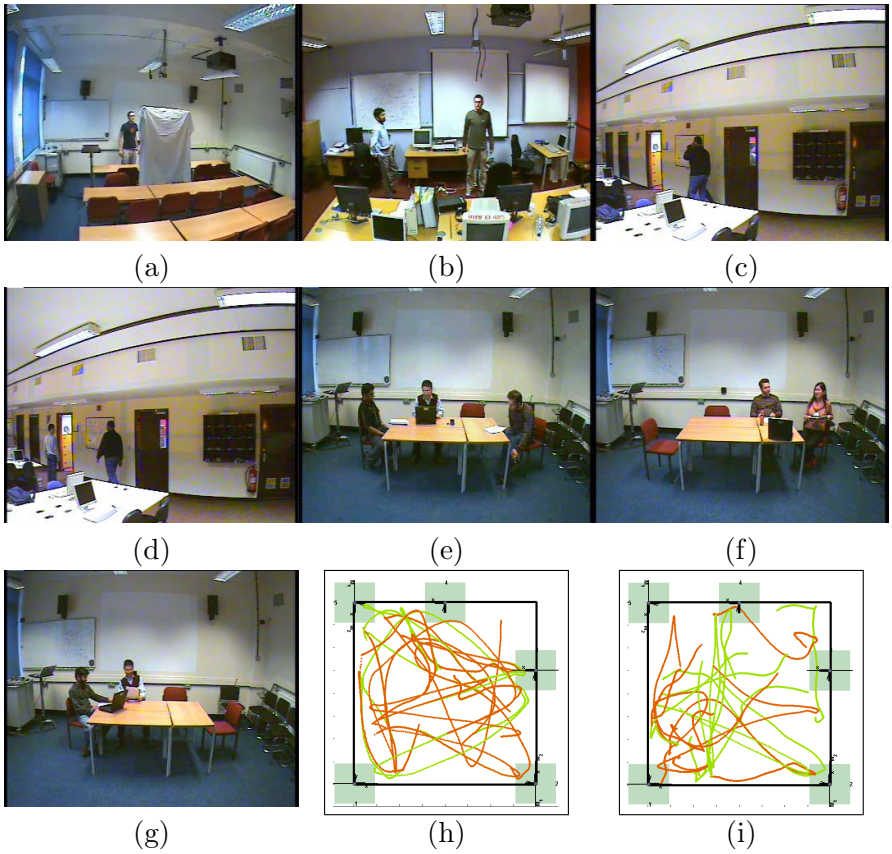


Figure 4.15: The evaluation datasets for multi-modal tracking. (a) Room scenario with visual occlusion (OC); (b) Office scenarios (SD); (c-d) Computer laboratory (L1 and L2); (e-g) Room scenario (M1, M2 and M3); and (h-i) Sample synthetic trajectories for 2 targets (green and red lines) in a network of 5 STAC sensors (grey squares).

is 95 cm (OC, SD) or 124 cm (L1, L2) and the video camera is located in the middle. In second type, sequences M1, M2 and M3 (Fig 4.15(e-g)) were used in which the distance between the microphones is 124 cm and the camera is placed 200 cm in front of the microphones. The camera and the microphones have the same height from the floor (170 cm). The distance between the sensors and the speaker is larger than 500 cm.

The values of the parameters used in the experiments are: the zero crossing weight is set to $\omega_{zc} = 0.9$. The noise weight for onset frame detection is set to $\omega_{\mathcal{N}} = 2.5$. The normalised frequency bands are set to $B_1 = [0, 0.25]$, $B_2 = [0.25, 0.6]$, and $B_3 = [0.6, 1]$ with the maximum frequency of sound $f_{max} = 6000\ Hz$. The weight of the three frequency bands are set to be $\omega_1 = 0.5,\ \omega_2 = 0.3,\ \omega_3 = 0.2$. The number of particles used by PF is $N^{pt} = 200$. The onset interval is of $T = 6$ frames.

The evaluation of multi-STAC tracking algorithm is done using synthetic trajectories. The trajectory estimation was performed with 4 different sensor configuration

Table 4.1: Summary of the datasets used in the multi-modal tracking experiments obtained using a camera and 2 microphones

Dataset	**Seq.**	**Resol.**	**No. of video frames**	**Frame rate (Hz) video/audio**	**Distance between microphones**
Occlusion	OC	360x288	1077	25/44100	95
Speaker detection	SD	360×288	945	25/44100	85
Lab	L1	360×288	1857	25/44100	124
	L2	360×288	1884	25/44100	124
Meeting	M1	360×288	2733	25/44100	124
	M2	360×288	4881	25/44100	124
	M3	360×288	675	25/44100	124
Total number of frames			14052		

consisting of 2, 3, 4 and 5 STAC sensors, respectively. The data consisted of 181 trajectories containing approximately 2200 points each (Fig. 4.15(h-i)). All the trajectories pass through the FOV of each STAC sensor to have fair comparison with CLUTE [159]. These trajectories are generated using synthetic audiovisual signals.

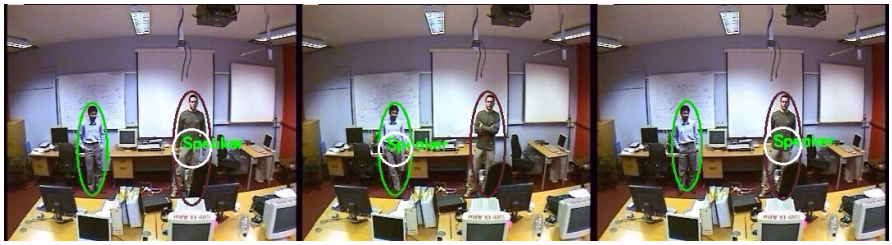
4.5.3 Multi-modal tracking using single audiovisual sensor

Figure 4.16 shows sample audiovisual target tracking results on the sequence OC during a visual occlusion. In this sequence, a person walks, talks and hides himself behind a barrier for about 1 second. The changes in the colour of the ellipse correspond to the identity switches of a target. It is possible to notice that the audio-only tracker (row 1) is capable of tracking the target *during* and *after* the occlusion and that there are no identity switches, although the accuracy is low. The low accuracy could be due to the violation of the assumptions of parallel line propagation and that the sound is produced in the direction of the microphones as well as due to the presence of a significant background noise however the target is tracked consistently. The video-only trackers (PF and GM, row 2 and row 3 respectively) correctly localise the object only when it is observable, however they fail during the visual occlusion and generate an identity switch when the target reappears. The audiovisual tracker correctly follows the target during occlusion and also improves the localisation accuracy compared to the audio-only tracker. The improvement in the tracking accuracy is summarised in Table 4.2: an error reduction of 12-24 pixels is obtained when using audiovisual fusion compared to audio only. As the video-only tracker fails due to a track loss, its results are not considered in this comparison. Table 4.2 also shows the error reduction when using the reverberation filtering (RF vs. Plain) and the multi-band frequency analysis (RF and MB vs. RF).

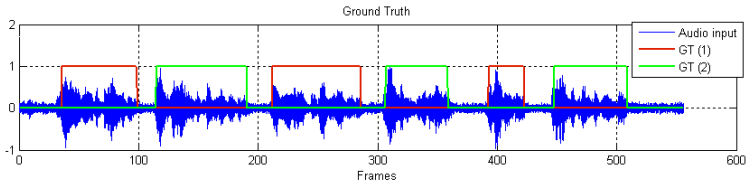
Figure 4.16: Comparison of tracking results (sequence OC) using audio-only tracking (first row), video-only tracking (second row) and audiovisual tracking (third row), and the computed correlation after reverberation filtering (fourth row). Frames: (a) 804; (b) 922; (c); 996.

Table 4.2: Comparison of tracking accuracy results (sequence OC). Absolute location estimation errors (ϵ: average in pixels) reduction between audio-only tracking, audiovisual tracking with reverberation filtering (RF) and with RF and multi-band analysis (MB)

GCC-PHAT	Plain	with RF	with RF and MB
Audio only	28.63	23.35	15.36
Audiovisual	4.47	3.93	3.47



(a) (b) (c)



(d)

Figure 4.17: Detection and tracking of alternating speakers using audiovisual cues for the sequence SD. (a) Frame 50, (b) Frame 313, (c) frame 425. (d) Ground truth of speaker detection: the green and the red lines represents the speaking activity of the two people.

Figure 4.17 shows an example of application of the proposed multi-modal tracker to an active speaker detection and tracking scenario, with two people moving freely in a room and alternately speaking. By comparing the sample results with the ground truth, it is possible to notice that the algorithm accurately detects the active speaker (white circle).

The sequence L1 and L2 are lab scenarios and have a similar set-up. In these experiments two persons walk from right to left and then meet. The results are shown in Table 4.3 and Table 4.4. They have more identity switches as compared to OC. This is again due to violation of the assumptions of parallel line propagation and that the sound is produced in the direction of the microphones (Sec. 4.2), however the proposed approach has the smallest number of identity switches and localisation error as compared to other approaches. This indicates that in cases where assumptions are violated, WPDA helps in correctly associating audio with visual measurements. Table 4.3 shows that KF-PF-P

Table 4.3: Performance comparison of the trackers under analysis. Absolute location estimation errors (ϵ: average in pixels)

	Seq.	PF	GM	AB-KF	AA-KF	KF-PF-P	KF-PF-WP
ϵ	OC	6.4	6.2	18.1	14.5	5.2	4.9
	L1	5.7	5.9	23.6	21.3	5.2	4.7
	L2	7.3	6.8	25.2	23.1	5.5	5.4

Table 4.4: Performance comparison of the trackers under analysis. Lost tracks and identity switches (N^{LT+IS}: percentage over an entire sequence)

	Seq.	PF	GM	GP-PF	KF-PF-P	KF-PF-WP
$N^{LT+IS}(\%)$	OC	5.76	5.76	5.29	4.83	4.51
	L1	14.49	14.55	12.39	12.23	11.95
	L2	11.36	11.26	10.99	9.19	8.54

and KF-PF-WP have lower average errors compared to AB-KF and AA-KF. Table 4.4 shows that KF-PF-P and KF-PF-WP have the smallest lost tracks/identity switches in the test sequences.

We also evaluate the performance of the proposed detection and tracking algorithm (KF-PF-WP) in three sequences (M1, M2 and M3) of a meeting scenario and show how metadata generated automatically (object position and their sound activity) can be transferred to other sensors or multimedia receivers (e.g., mobile phones) with limited bandwidth requirements.

The first sequence (M1) has three subjects (sample frames and the corresponding results are shown in Fig. 4.18, row 1 and 2) who initially are sitting and having pair-wise conversation. Next, the person sitting in the middle stands up, moves and talks to the person on the left. This results in a difficult audio detection, as he keeps changing the direction of his face. The proposed tracking algorithm enables us to effectively detect and track the speakers: for example, the second column of row 1 and row 2 shows that the speaker sitting in the middle is correctly detected and tracked. However, in row 1 column 4 the audiovisual tracker does not detect the real speaker (the speaker on the left), due to a biased estimation of the audio GCC estimates when the person faces away from the microphones. In row 5 and 6, the proposed audiovisual tracker correctly identifies the speaker despite large measurement errors (3^{rd} column of row 5 and 6). Although the audio detection deviates from the correct position, the final audiovisual result is accurate as the estimation of the speaker's position in the previous image frame is correct. In fact, this leads to a larger posterior probability of detection of the speaker on the left than that of the one on the right in the current frame, and hence the estimated position settles on the person on the left. Figure 4.19 shows sample animations generated using the automatically

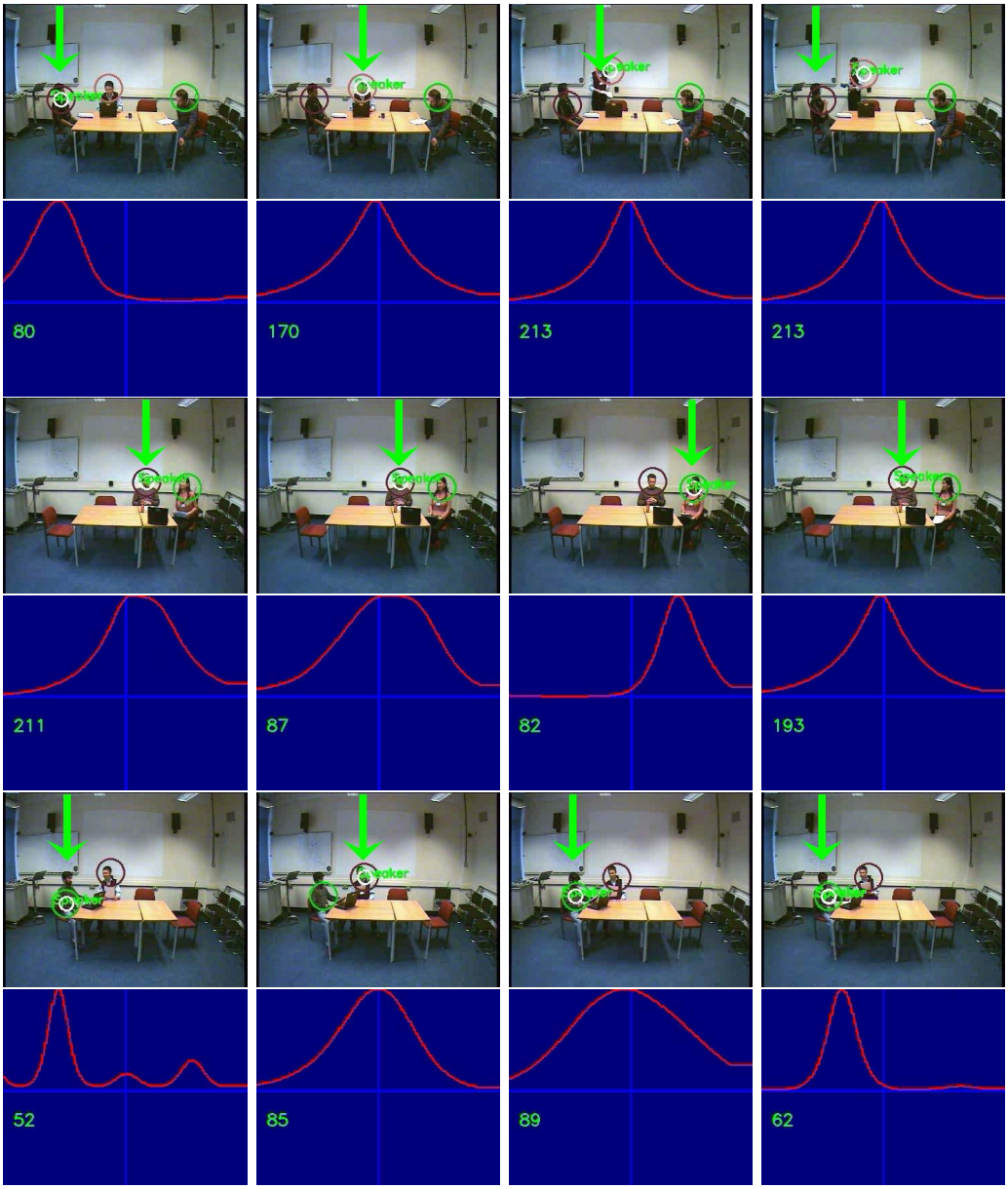


Figure 4.18: Audiovisual speaker detection and tracking for sequence M1 (row 1 and 2), sequence M2 (row 3 and 4) and sequence M3 (row 5 and 6). Row 1, 3 and 5: green arrows indicates ground truth, white circles with "speaker" indicate the detected audio source location and circles in other colours denote the visual detection and tracking. Row 2, 4 and 6 show the audio GCC estimates.

extracted metadata. The comparison of the bandwidth requirement when using different coding methodologies is shown in Table 4.5 for (1) MPEG-1, (2) MPEG-2, (3) MPEG-4 and (4) the metadata generated by the proposed multi-modal tracker. The size of all the four formats also contains the audio file size. These bandwidth requirements correspond

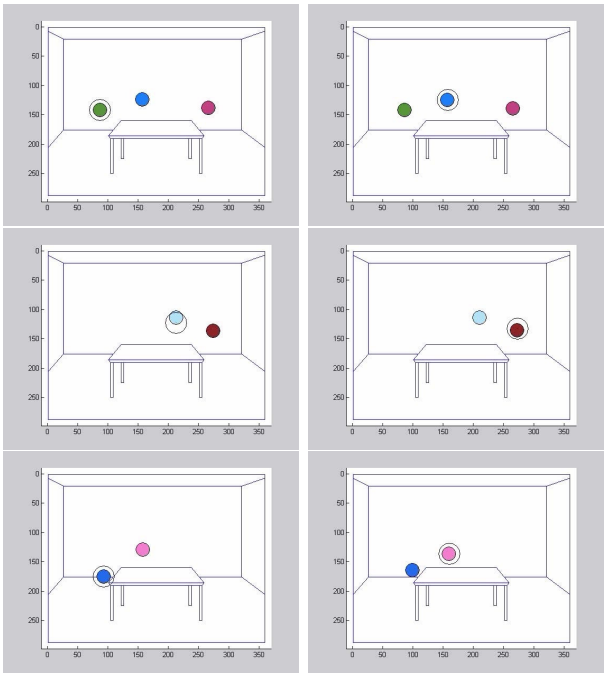


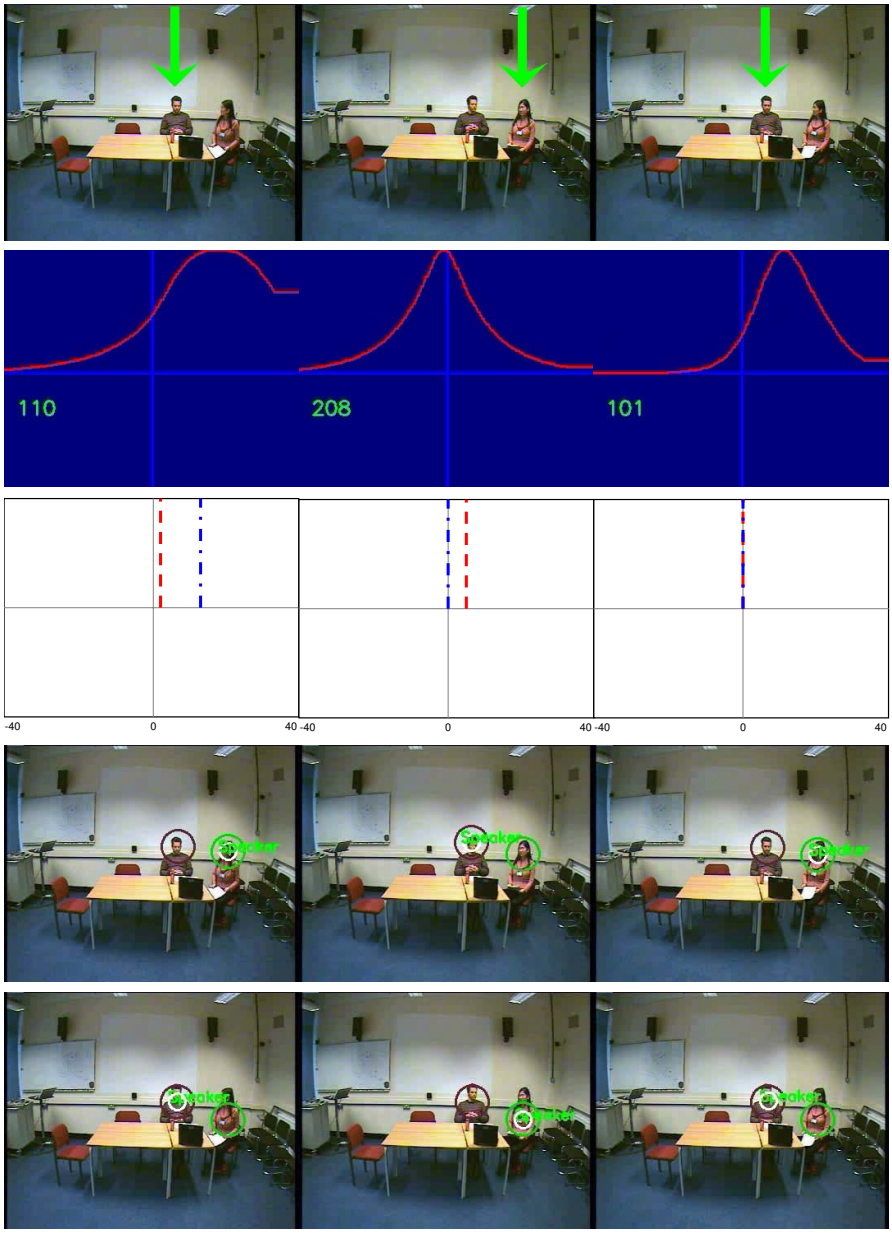
Figure 4.19: Sample object animations and speaker detection created using the generated metadata. (Row 1): sequence 1, (Row 2): sequence 2, (Row 3): sequence 3. Coloured circles denote the visual detection, white circles represent the audio detection, and the axes show the original image size.

Table 4.5: Bandwidth estimates of different methodologies on sequence M1, sequence M2 and sequence M3. Units: Kilobytes per frame.

Seq.	MPEG-1	MPEG-2	MPEG-4	Metadata	Metadata with audio
M1	7.61	7.82	6.39	0.21	1.15
M2	6.74	7.26	5.75	0.48	1.42
M3	7.76	8.16	7.36	0.25	1.19

to the information to be transmitted when multiple multi-modal sensors exchange the position and the activities of the observed objects.

Figure 4.20 compares sample results from different object tracking strategies on sequence M2 of the meeting scenario. It is possible to notice that the Kalman filter leads to smaller errors in audio source localisation (Fig. 4.20(a-b)). The PDA results (Fig. 4.20 row 4) are most affected due to the biased audio GCC estimates shown on row 2, the WPDA locates the speaker due to the correct likelihood estimation of the visual and the audio locations.



(a) (b) (c)

Figure 4.20: Performance comparison of different tracking algorithms: (a) frame 157 (b) frame 435 and (c) frame 723. (Row 1): ground truth marked with green arrows; (Row 2): GCC noisy audio estimates; (Row 3): estimated arrival angles of speakers with the Kalman filter (dash lines) and without the Kalman filter (dash dots); (Row 4): Particle filter and PDA-based audiovisual tracking; (Row 5): Particle filter and WPDA-based audiovisual tracking.

4.5.4 Multi-modal tracking using multiple audiovisual sensor

To evaluate the tracking using multiple audiovisual sensors we generate synthetic data using 2, 3, 4 and 5 STAC sensors. The audio data are generated by transmitting an audio signal (impulse train) from the position of the target and then recording it at the sensor location after applying environmental constraints [82]. In case of synthetic data the attenuation Γ is calculated using the Beer-Lambert law[1], to mimic real world signals, as

$$\Gamma_i = \Gamma_0 \exp(-\alpha L_{M_{ij}}). \tag{4.31}$$

where Γ_i is the attenuation for i^{th} microphone, Γ_0 is the initial sound intensity, $L_{M_{ij}}$ is the path length between the i^{th} microphone and j^{th} object and α is the attenuation coefficient.

Synthetic video data are generated using a first-order motion model defined as

$$\mathbf{x}_{k+1} = \mathbf{U}\mathbf{x}_\mathbf{k} + \mathcal{N}(\mu, \mathbf{\Sigma}), \tag{4.32}$$

where $\mathbf{U}$ is the observation model which ensures smooth transformation of the target state at time k to the next state at time $k+1$ and is defined as

$$\mathbf{U} = \begin{bmatrix} 1 & 0.35 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.35 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.33}$$

where 0.35 is chosen to maintain slow target speed. $\mathcal{N}(\mu, \Sigma)$ is a zero-mean Gaussian noise and serves as a process noise to introduce small variation in the motion. The covariance Σ of this process noise is defined as

$$\Sigma = diag[10^{-10}, 10^{-6}, 10^{-10}, 10^{-6}]. \tag{4.34}$$

Table 4.6 shows the mean (μ) and standard deviation (σ) of the trajectory estimation error (ϵ) for TDOA, AVDW, AVKF and CLUTE. The algorithm is also evaluated with approximately 50% randomly missing audio observation as in real data the targets will not be producing continuous audio signal. The error after applying smoothing using Kalman filtering is increased compared to TDOA and AVDF only. This is mainly because Kalman filtering estimation deteriorates when the target exhibits sharp turns. The error

[1] http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/attenuation.htm, last accessed: 10 Nov, 2009

Table 4.6: Comparison of localisation error (ϵ) in trajectory estimation using method A (TDOA), B (AVDW), C (AVKF) and D (CLUTE) (see text for definitions) using 2, 3, 4 and 5 STAC sensors without(W)/with missing(M) audio observations

		number of sensors							
		5		4		3		2	
		W	M	W	M	W	M	W	M
A	μ	0.0373	0.0580	0.0414	0.0624	0.0560	0.0766	0.4607	0.4819
	σ	0.0662	0.0961	0.0780	0.1052	0.1115	0.1374	0.5242	0.5422
B	μ	0.0278	0.0462	0.0319	0.0509	0.0434	0.0619	0.4078	0.4257
	σ	0.0459	0.0673	0.0562	0.0769	0.0762	0.0956	0.4812	0.4965
C	μ	0.5181	0.5347	0.5183	0.5352	0.5191	0.5361	0.7681	0.7832
	σ	0.2925	0.3020	0.2922	0.3020	0.2921	0.3021	0.5319	0.5430
D	μ	4.4561	4.4561	5.3760	5.3760	6.1361	6.1361	6.4242	6.4242
	σ	4.0786	4.0786	5.5655	5.5655	6.4181	6.4181	9.7167	9.7167

in trajectory estimation with audiovisual data is due to the approximation in computing delay in samples. The delay is initially estimated in seconds and is converted into a discrete number of samples and this rounding error introduces a quantisation effect (Fig. 4.14) and creates an error of maximum 0.2233° (considering a rounding error of 0.5 samples at 44.1kHz) in the arrival angle estimation. This error can be reduced by increasing the sampling frequency. Table 4.6 also shows that the mean error difference between with and without missing audio observations for 5 STACs in case of AVKF is $0.5347 - 0.5181 = 0.0166$, whereas in case of AVDW is $0.0462 - 0.0278 = 0.0184$. This increase of a small value indicates that audio estimation can also be used for complete path estimation in case of non-continuous audio observations. Note that the error for CLUTE remains the same as it does not depends on audio observations.

4.5.5 Future experiments

The source localisation using audiovisual fusion is highly sensitive to noise in the signal for which this thesis proposes to apply multi-band analysis for time difference of arrival estimation. The multi-band analysis uses three different frequency bands the output of which is fused using a weighted sum. However, it is not very clear what are the optimal ranges of these frequency bands, and so is the case for the most appropriate order of filter. In future work the results can be improved by performing experiments with ranges of frequency bands for targets of different categories such as person or vehicle and with filters of different orders. This should be aimed towards generating a bank of filters from which the desired set of filters may be automatically selected by recognising the type of input signal. These experiments should also aim towards identifying the relationship

between the absorption coefficients of the various materials in the scene and the filter bands and order that is most appropriate for scenes with specific type of materials (e.g., wood, glass and concrete).

This work shows real tracking results using a single Stereo Audio Cycloptic Vision (STAC) sensor. Experimental results using multiple sensors are limited to synthetic data. The natural extension of the work is to test the working of the proposed multi-modal tracking using multiple audiovisual sensors on real data. The experiments can be performed on the datasets acquired using sensors where each camera may be equipped with 2 or more microphones arranged in different configurations such as in linear, circular or T-shape arrays. The goal of these experiments would be to compare which array configuration is best suited for the task of localisation and to understand the amount of information gain using more than 2 microphones. The fused signal obtained from these different microphone configurations as well as from multiple STACs may have several false detections due to reflection and reverberation. If the fusion between multiple microphones or from multiple STAC is performed using the steered beamformer, then the signal similar to the meta-sensor obtained through multi-camera multi-level homography (Chapter 3) can be generated. This will allow us to perform experiments with MT-TBD-PF in tracking audio targets in noisy scenarios.

4.6 Summary

This chapter discussed multiple modalities and how they can be fused to improve tracking results. Furthermore, the use of sensors with a large coverage area, such as microphones, in a multiple non-overlapping camera setting helps estimating track information in regions unobserved by visual sensors, has been discussed.

In order to obtain a solution feasible for large scale deployment we have investigated the use of a simple Stereo Audio Cycloptic vision (STAC) sensor consisting of only a camera and a microphone pair. Due to having a small number of microphones, such a solution is very sensitive to noise and reverberation. To solve this problem, the audio signal is filtered by utilising the precedence effect and multi-band analysis. To further reduce the effects of noise and reverberations we used a Riccati Kalman filter that automatically updates the audio measurements using the historic estimates in a least squares sense as well as a Weighted Probabilistic Data Association scheme to associate the audio detections with the visual measurements.

To this extent a particle filter based tracking algorithm has been proposed that

integrates measurements from heterogeneous sensors and it has been demonstrated on audio and video signals. The audio and visual cues are fused at the likelihood level (Eq. 4.19). This makes the framework extensible and additional features can be added directly at the likelihood level. Experimental results have demonstrated that the proposed strategy can improve classical audio or video approaches in terms of tracking accuracy and performance.

In Part II we will show how to use trajectory information for recognising interactions.

Part II

Interaction recognition

Chapter 5

State of the Art

5.1 Introduction

Event recognition and behaviour analysis are desirable yet demanding functionalities of video analytics. These functionalities can help CCTV operators to focus more effectively on cameras that are observing activities of interest. They can also help in automated analysis of large volumes of videos for indexing and retrieval applications, such as multimedia databases and video surveillance.

Object behaviour can be categorised as either interaction with static objects or with other dynamic objects in the scene. In this chapter we will discuss these two categories of interaction in videos in Sec 5.2. In Sec 5.2.3 we discuss the Bayesian networks for interaction recognition and finally the chapter is summarised in Sec 5.3.

5.2 Recognising interactions in video

Video activity detection can be decomposed into three main steps: (i) extraction of the *features* that best describe the activities of interest; (ii) learning *models* that describe the various activities given the static and dynamic contextual information and, finally, (iii) *recognising* the events of interest. Figure 5.1 shows a generic block diagram for activity recognition.

Depending upon the type of activity to be detected, different types of *features* can be extracted. For activities such as crowd behaviour that affect the entire scene (scene-level activities), motion vectors can be used [160]. For activities such as road crossing or running that may be related to single objects only (object-level activities), information about an object location and its temporal evolution can be used [161,162]. For object-level

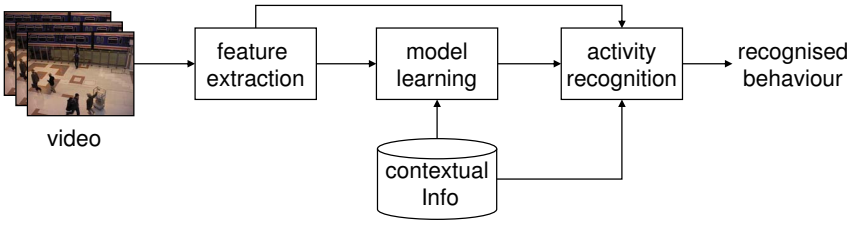


Figure 5.1: Block diagram of a generic activity recognition system.

activities, object detection and tracking information form part of the features. Derived features such as velocity or spatio-temporal cuboids of patches inside the object detection area over a certain number of frames can also be used [162–164].

An activity can be decomposed into a set of atomic events or states through which an object passes over time. An event can be defined as an observable occurrence or a distinguished occurrence that can be explained using a set of rules. For example, a running event can be defined as a detected dynamic object whose speed is larger than the average walking speed, where the average walking speed is defined either statistically, using the labelled interval of a walking event, or based on domain knowledge. Given the rules, an event can be recognised using thresholds on the values of the extracted features. For example, a common event in video analytics is a *tripwire* [165] event. A tripwire event occurs whenever an object crosses a line or a boundary defined by the user.

In probabilistic terms, an event can be defined as a set of outcomes to which a probability is assigned. Under this definition, an observable measurement or underlying state can be represented by a distribution such as a Gaussian, a Multi-variate Gaussian or a Mixture of Gaussians. Most approaches [162, 163] involving a *learning step* require the estimation of the parameters of these distributions [166] or the boundaries employed by the set of rules defining the event or state [162, 163].

Once an atomic event or state is defined, the next step is *to model a certain activity*. An activity may be represented using a 3D shape such as motion history volumes [161], an action cylinder [167], a quantised vocabulary of local spatio-temporal (ST) volumes (or cuboids), or a quantised vocabulary of spin-images [164]. However, these approaches are generally computationally expensive, have high memory requirements and are sensitive to noise in the extracted features. Alternatively, an activity can be modelled as an evolution of states over time. Certain activities can be represented as a temporal template that define the occurrence of various states or events in a specific order where the goal is to recognise the temporal template in the extracted features accumulated over time [168, 169].

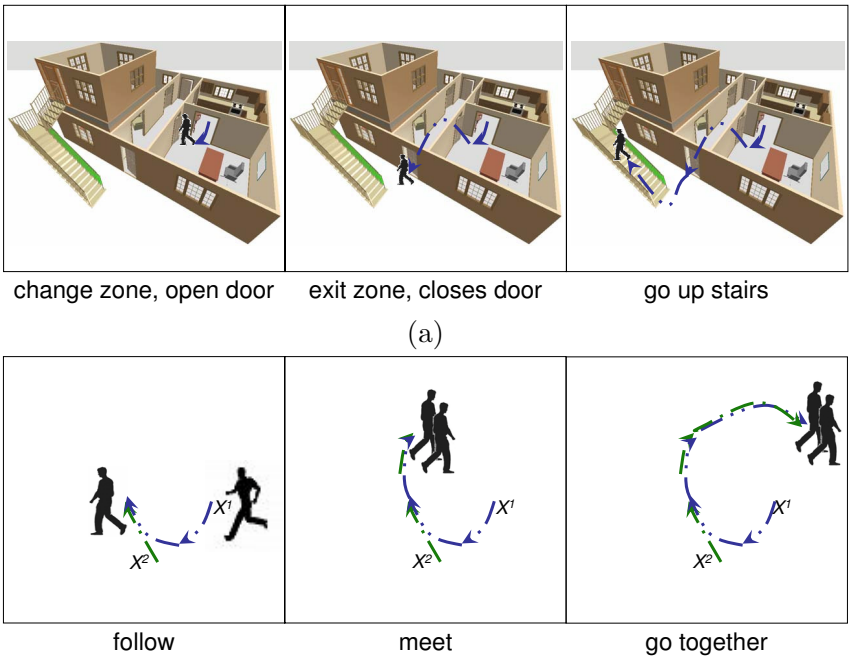


Figure 5.2: Illustrations showing two different types of interaction events. (a) Dynamic-static interaction. (b) Dynamic-dynamic interaction.

The observed objects interact with the environment as well as with other objects. The problem of interaction event detection can therefore be divided into two categories of state estimation, namely dynamic object with respect to static objects and dynamic objects with respect to each other. Graphical methods are well suited to represent such structures as they have the inherent capability of encoding the various rules that govern the transition between states [170]. Examples of graphical models used for event detection are Petri-nets [168], Bayesian Networks [171] and Hidden Markov Models (HMMs) [46, 172].

This section focuses on video event detection techniques using HMMs and in particular on the recognition of object interactions. The next two sub-sections discuss various interaction detection techniques particularly those that involve interaction between dynamic and static objects (Sec. 5.2.1)(Fig 5.2(a)). The techniques that involve recognition of interactions between multiple dynamic objects are then discussed in sub-section 5.2.2 (Fig 5.2(b)). Finally, in sub-section 5.2.3, the formulation of various Bayesian networks for interaction event detection is discussed.

5.2.1 Interaction among dynamic and static objects

An activity that involves interaction of a dynamic object with static objects requires estimation of a sequence of atomic states of either a static or dynamic object with respect to the interacting object. In some cases the state of the static object is estimated with respect to the dynamic object. For example, in the case of unattended baggage detection, the baggage state goes from one of *attended*, *unattended*, *abandoned*, *retrieved* or *stolen* [143]. On the other hand in activities such as unauthorised access, the dynamic object state goes from one of *approaching*, *entering* and *inside zone* [140]. These kinds of interaction activities, involving dynamic-static objects, are common in surveillance and domestic scenarios. In recent years there has been a significant amount of interest in detecting such activities. This can be seen from various efforts from governments and research communities through the availability of various real datasets and evaluation campaigns [139,140,142,173–175] in order to find potential solutions to the problems. The literature available is also biased by the datasets and the activities targeted in them.

One of the most important dynamic-static interaction is that of abandoned baggage detection. Several attempts have been done to detect this interaction [51, 176–190]. However, many of these approaches [51, 176, 185–190] do not model the problem as interaction between static and dynamic objects. In fact they solve the problem of detecting static objects based on thresholding of time intervals, classifying them simply as *abandoned*. Many of these approaches merely detect the stopped object [185–189], while others classify objects as person or non-person. Non-person stopped objects [176, 181] or those who match certain shape models [190] are then classified as bags. These approaches tend to fail as stationary objects can be anything (e.g. stopped vehicle) and several other classes of objects can be misclassified as baggage. In [182, 183] a baggage classifier is proposed to resolve this issue. However, this method is also based on heuristics for event detection. The major reason behind failure in the modelling of such events is that it relies on the tracking of owners, detection of baggage and recognition of activities. The detection and tracking is itself a difficult problem in real scenarios which involve dense crowds, especially in public transportation scenarios. Most of these approaches try to solve the detection and tracking problem instead.

The interaction detection applied in some of these approaches is based on the detection of drop-off events (blob splits). One of the split objects is considered as owner, whereas the other that becomes static is classified as baggage [177, 179, 184]. In [184] Bhattacharya coefficients were used to track the owner and the interactions were detected

Table 5.1: Summary of dynamic-static interaction algorithms (Key: BM=Bayesian method, EM=Expectation Maximisation, HU=Heuristics, FSM=Finite State Machines, VL=Viterbi-like algorithm)

Ref.	**Algo.**	**Features**	**Application**
[191]	BM, EM	spatio-temporal patches	Domestic activities
[192]	BM	Histogram of gradients	Domestic activities, Sport, surveillance
[193]	BM	Action threads related by temporal templates	Surveillance
[194]	DBN, VL	position	Domestic activities
[171]	BM	height, width, speed, motion, direction, distance, contextual information	Surveillance
[179]	BM	speed, direction and relative distance	Surveillance
[178]	FSM	size, shape, position, motion, class, stay-time, inter-object relation	Surveillance
[184]	HU	colour, relative position	Surveillance
[182, 183], [180, 181]	HU	relative position	Surveillance
[177]	HU	size and speed	Surveillance

using relative distances. Similarly, in [180], relative distances were used, whereas, in [177], the top-view generated from multiple cameras is also utilised to remove any false or missed detections due to camera perspective. A few methods however proposed to model the problem instead of using the heuristics, such as in [179], abandoned baggage is detected by utilising tracking information to compute features such as speed, direction and inter-object distances. This method defines four hypotheses based on the relative distance between the static object (bag) and the dynamic object (owner). These hypotheses are tested based on evidence using Bayesian inference to detect the drop-off event. To better model the transition from state to state between interacting objects, in [178] a finite state machine (FSM) is used. Features such as size, shape, position, motion, object class, stay-time, and relations with other objects are used to drive the state machine. The use of a state-machine allows better control of the transition between states.

Recently, effort has been done to obtain a general solution for modelling the problem of dynamic-static object interaction using *probabilistic graphical models* [191,192,194]. *Probabilistic graphical models* [170] provide a simple way to visualise the structure of a probabilistic model, insight about their conditional dependence and a way to solve complex computations as graphical manipulations. A common type of graphical model is the *directed acyclic graph* (DAG) [195]. In a DAG, one can move from one node to another along the links but cannot reach the initial node again. DAGs of stochastic processes are called Dynamic Bayesian Networks (DBN) [196]. The simplest types of DAGs are

HMMs [170], which are characterised by one discrete hidden node (state) and one discrete or continuous observed node (output or emitting symbol) at each time-step. In [192], a Bayesian approach to model interactions between dynamic and static objects is used. The activities are modelled as graphs and discrete HMMs are used to estimate the sequence of states performed by the object. One of the problems in recognising interactions with static objects is having to classify the objects. In this method, an Adaboost classifier trained on histogram of gradients (HOG) is used to detect objects of interest such as phones, cups, bells etc. Hand trajectories are then estimated using the difference of histograms as a similarity measure for tracking. The track and object class information along with contextual information is then used to compute spatial and contextual constraints to recognise interactions which are coherent with the semantics of the scene to improve the results. In [191], joint probability distributions to represent both actor and object appearances as well as their intrinsic spatio-temporal configurations are employed. This method uses features such as spatio-temporal cuboids to train the probabilistic framework. The trained clusters are then used to estimate the state of the interacting objects. This method does not require tracking information nor does it rely on contextual information; hence it can only detect simple interactions such as *grasp fork*, *grasp cup*, *push toy car* etc. in constrained environments. Complex events can be detected by incorporating contextual information and then using any appropriate algorithm to evaluate object trajectories, such as, for example, Bayesian methods [171]. In [194], Dynamic Bayesian Networks (DBN) are used to encode prior knowledge about the activities such as expected action and ordering constraints. A complex activity is defined as one containing sub-activities. DBN also allows the use of contextual disambiguations that provide additional cues for activity recognition. In this method, the state estimation is performed using a Viterbi-like algorithm. This method also proposes the Erlang distribution as a comprehensive model of the idle time between actions and frequency of observing new actions. In Sec. 6.3 we propose our method for detecting such interactions based on HMM and show its applicability both for unattended baggage detection as well as for several other interactions. A summary of state of the art techniques for video event analysis is shown in Table 5.1.

5.2.2 Interaction among dynamic objects

As objects move simultaneously based on their intentions, their actions depend also upon the behaviours of other objects. For this reason, monitoring a single target separately [171, 197, 198] may not provide the complete information about its state.

Target interactions can be modelled using either predefined heuristics (rules) [199]

or by using graphical models [48, 166, 172, 200]. *Heuristics-based methods* generally define rules for interactions based on each target's spatio-temporal features, such as speed and distance, to compute the probability of interaction [199]. Events can be assumed to be composed of sub-events and graphs can be used to model the conditional dependency between them. Normalised cuts can be used to partition such a dependency graph to extract complex events as a highly correlated chain of sub-events.

Interaction detection can be modelled as a random process that is segmental in nature, as the piecewise stationarity assumption of HMMs [46] is well suited for time-series analysis. HMMs and their variants [201] are used to address the problem of Interaction Event Recognition (IER) due to their inherent capability of modelling uncertain temporal information. Interactions can happen between a static object and a dynamic object or between dynamic objects. Static objects can either be temporally static (such as a bag or a car) or permanently static such as structural objects. HMMs were used to model interactions with such static objects in [202] where object trajectories were used to obtain a 4D feature vector accounting for object position and size. This feature vector is then used inside a continuous distribution HMM using multi-variate Gaussians for estimation of emission probabilities for detection of interactions associated to static objects. The approach is further extended by modelling the duration of each state which imposes a practical constraint that objects should take a transition from one state to another after a certain interval. Hidden-Semi Markov models (HSMMs) were used to model such duration-related transitions.

HSMMs have been shown to better model temporal evolution of object behaviour. However, they are limited in modelling interactions with static objects only. Other methods for modelling such interactions include Multi-Observation-Mixture+Counter Hidden Markov Models (MOMC-HMM), which allows representation of multiple observations of different objects at each state [203] similar to Variable-length HMM (VLHMM) [204]. The extension to VLHMM is proposed in [169]. This method is made immune to noise by abstracting the continuous variables (tracks) into discrete space while preserving the underlying behavioural patterns and applying them in modelling interaction among cars on highways. However, it should be noted that such discretisation may result in losing some relevant detail and therefore may not be suitable for some applications.

Coupled Hidden Markov Models (CHMMs) have received significant attention [48, 166,172,200], to model group activity among multiple dynamic objects, as they allow modelling the full coupling between the processes and can be solved in polynomial time using dynamic programming [200]. Events like *walk*, *approach* and *chat* have been detected using

Table 5.2: Summary of Interaction Event Recognition (IER) algorithms (Key: Coupled Hidden Markov model (CHMM); Maximum Likelihood (ML); Maximum a posteriori (MAP); Dynamically Multi-Linked HMM (DHL-HMM); Variable Length HMM (VLHMM); Multi-observation-mixture+counter (MOMC) HMM; Vector Quantisation (VQ); Dynamic Bayesian Network (DBN); Fractional Spectral Radius (FSR); Functional Magnetic Resonance Imaging (fMRI))

Ref.	**Algorithm**	**Model features**	**Applications**
[207]	Belief Networks	Tracks	American football
[199]	Supervised learning	Speed and distance	Surveillance
[208]	Normalised cuts	Tracks	Surveillance
[200]	CHMM-ML	3D hand tracks	Hand gesture recognition
[48]	CHMM-ML	Relative distance, velocity, angle sign	Surveillance Surveillance
[205]	CHMM-ML	FSR, reflection coefficients	Medical
[206]	CHMM-ML	3D position	Hand tracking
[166]	CHMM-MAP	FSR, reflection coefficients	Medical
[209]	DHL-HMM	4D state-space, filling ratio, pixel change history	Surveillance
[203]	MOMC HMM	4D observing flow vector	Surveillance
[169]	VLHMM VLHMM	3D rotations of 19 major joints	3D motion capture data
[204]	VQ and VLHMM	Velocity, relative distance	Traffic monitoring
[210]	DBN DBN	Head pose, body, hand and head position	Meeting
[201]	DBN	fMRI	Medical
[211]	DBN	Tracks, signal power	Medical

CHMM [48, 172] with synthetic data. CHMMs have also been applied to medical data to detect changes in heart beats during sleep [166, 205] and for gesture recognition [200, 206]. A summary of state of the art techniques for interaction event analysis is shown in Table 5.2.

These state of the art approaches are limited to the camera's FOV only and naturally fail when targets exit the observed regions. This thesis overcomes that limitation by using heterogeneous sensors with a combination of audio and visual sensors, and performs event analysis as well as interaction recognition in regions unobserved by visual sensors (Sec. 6.4).

5.2.3 Bayesian networks for interaction event modelling

Let an object detector generate at each time k a set of $N^{\mathbf{x}}$ objects $X_k = \{\mathbf{x}_k^1, \mathbf{x}_k^2, \cdots, \mathbf{x}_k^{N^{\mathbf{x}}}\}$. Let a tracker associate object instances between consecutive frames to

establish the track $X_k^r = \{\mathbf{x}_1^r, \mathbf{x}_2^r, \ldots, \mathbf{x}_k^r\}$, up to time k, for each r^{th} object (for simplicity the superscript r and the subscript k will be dropped whenever the discussion is related to a track of a single object up to the total number of time steps K).

Let $\lambda = \{A, B, S, s_0\}$ be a continuous distribution first-order Hidden Markov Model, where $S = \{s_1, s_2, \cdots, s_{N^S}\}$ represents a set of N^S discrete events (states) to be detected (the actual state at time k is denoted as a time indexed discrete variable $s_k \in S$), $\mathbf{A} = \{a_{ij}\} = \{P(s_k = s_j | s_{k-1} = s_i)\}$ represents the state transition probabilities where $P(.)$ represents the probability of transition from state s_i to state s_j from time $k-1$ to k; $\mathbf{B} = \{b_{jk}\}$ are the emission probabilities, with $b_{jk} = P(\mathbf{x}_k | s_k = s_j)$ and s_0 is the known initial state at time $k = 0$. The emitting symbols of each state are provided by the features extracted from the objects. The associated optimal state sequence can be obtained by applying Bayes' rule

$$\hat{s} = \arg\max_S P(S|X) = \arg\max_S \frac{P(X|S)P(S)}{P(X)}, \tag{5.1}$$

where $\hat{s}$ is the estimated state. To estimate the posterior probability $P(S|X)$, the likelihood $P(X|S)$, the prior $P(S)$ and the probability $P(X)$ need to be calculated. $P(X)$ is a normalising constant that can be estimated as

$$P(X) = \sum_{c=1}^{N^C} \sum_{i=1}^{N^S} P(\mathbf{x}^c | s_i) P(s_i), \tag{5.2}$$

where N^C is the total number of interacting chains and $\mathbf{x}^c$ is the state of the target belonging to c^{th} chain. Using the Markov property, the likelihood $P(X|S)$ is given as

$$\begin{aligned} P(X|S) &= P(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K | s_1, s_2, \cdots, s_K) \\ &= P(\mathbf{x}_1|s_1) P(\mathbf{x}_2|s_2) \cdots P(\mathbf{x}_K|s_K) \\ &= \prod_{k=1}^{K} P(\mathbf{x}_k|s_k), \end{aligned} \tag{5.3}$$

where K is the total number of time steps. Similarly, the prior $P(S)$ can be computed

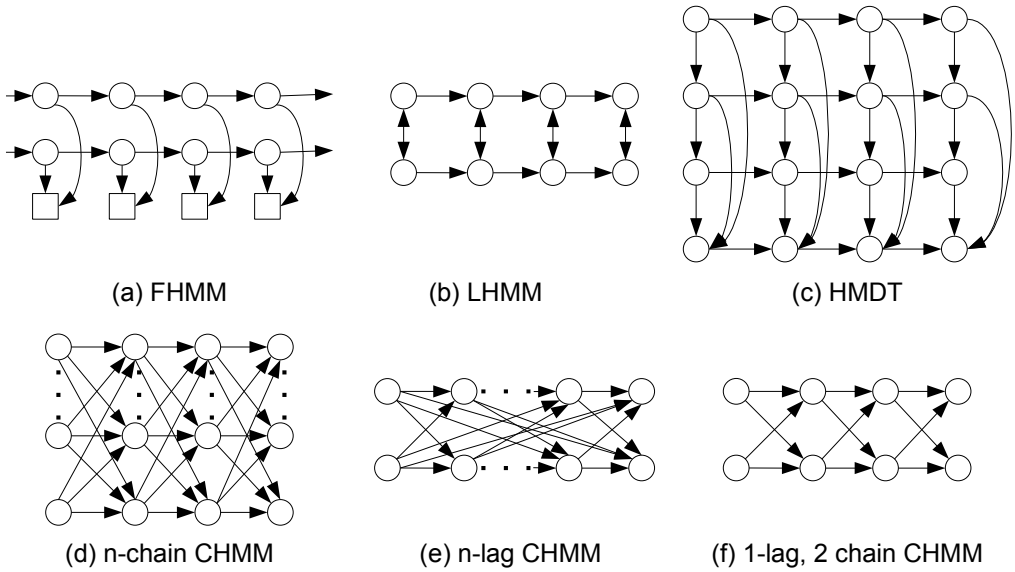


Figure 5.3: Graphical representation of HMM variants. (a) FHMM, (b) LHMM, (c) HMDT, (d) n-chain CHMM, (e) n-lag CHMM and (f) 1-lag, 2 chain CHMM. FHMM models independent processes while the rest model dependent processes. (Key: Circles: state variables; squares: output variables).

using the Markov property as

$$\begin{aligned} P(S) &= P(s_1, s_2, \cdots, s_k) \\ &= P(s_1)P(s_2|s_1)\cdots P(s_K|s_1, s_2, \cdots, s_{K-1}) \\ &= P(s_1)\prod_{k=2}^{K} P(s_k|s_{k-1}). \end{aligned} \tag{5.4}$$

In standard HMM models, the causal relationship within a single process and each HMM state causes an effect on the next state. However, in many real-world scenarios there may be more than one process interacting with each other.

The variants of HMM existing to model such inter–process causal relationships (or couplings) can be divided into two major groups based on whether the processes are dependent or independent. In the case of *independent processes*, the coupling is at the output level only and is called Factorial HMM (FHMM, Fig. 5.3(a)). A number of HMM variants exist for modelling *dependent processes*, based on their degree of dependence: Linked HMM (LHMM, Fig. 5.3(b)), Hidden Markov Decision Tree (HMDT, Fig. 5.3(c)) and Coupled HMM (CHMM, Fig. 5.3(d)).

In general, for a graph of N^G nodes, the joint distribution is given by [170]:

$$P(S) = \prod_{n=1}^{N^G} P(s_n|pa_k), \tag{5.5}$$

where pa_k denotes the set of parents of s_n. In Factorial HMM, signals are factored as component processes which are modelled independently. The full posterior of the N^C-chain FHMM can be computed as

$$P(S^{N^C}|X) = \frac{1}{P(X)} \prod_{c}^{N^C} \left(P(s_1^c) P(\mathbf{x}_1^c|s_1^c) \prod_{k=2}^{K} P(s_k^c|s_{k-1}^c) P(\mathbf{x}_{k-1}^c|s_{k-1}^c) \right). \tag{5.6}$$

In LHMM the current state of a process is dependent on its previous state and the current state of the neighbouring process. The full posterior of the LHMM can be computed as

$$P(S^{N^C}|X) = \frac{1}{P(X)} \prod_{c}^{N^C} \left(P(s_1^c) P(\mathbf{x}_1^c|s_1^c) \times \right.$$
$$\left. \times \prod_{k=2}^{K} \left(P(\mathbf{x}_k^c|s_k^c) P(s_k^c|s_{k-1}^c) \prod_{d}^{N^C} P(s_k^c|s_{k-1}^d) \right) \right). \tag{5.7}$$

In contrast to LHMM, in HMDT the current state of any process is dependent on the current state of all the interaction processes as well as the previous state of the process. The full posterior in this case can be computed as

$$P(S^{N^C}|X) = \frac{1}{P(X)} \prod_{k=2}^{K} \left(P(s_k^1|s_{k-1}^1) P(s_k^2|s_{k-1}^2, s_k^1) \cdots \right.$$
$$\left. \cdots P(s_k^{N^C}|s_{k-1}^{N^C}, s_k^1, \cdots, s_k^{N^C-1}) \right) \prod_{c}^{N^C} P(\mathbf{x}_1^c|s_1^c). \tag{5.8}$$

CHMMs are an example of the full coupling and the process state does not only depends upon its previous state, but also on the previous state of all the interacting processes. The full posterior of the N^C-chain CHMM $P(S^{N^C}|X)$ can be computed as

$$P(S^{N^C}|X) = \frac{1}{P(X)} \prod_{c}^{N^C} \left(P(s_1^c) P(\mathbf{x}_1^c|s_1^c) \prod_{k=2}^{K} \left(P(\mathbf{x}_k^c|s_k^c) \prod_{e}^{N^C} P(s_k^c|s_{k-1}^e) \right) \right), \tag{5.9}$$

where $P(s_t^c|s_{t-1}^e)$ is the state transition probability at time k for object c to state s_k^c given the state s_{k-1}^e of the interacting object e at time $k-1$. In CHMMs, each discrete hidden

node is coupled also with all the previous nodes of all the N^C chains. Such a model is called an n-lag C-chain model.

To reduce the complexity while keeping the interaction modelling, the 1-lag model is considered, the posterior of which, $P(S^{N^C}|X)$, is given in Eq. 5.9. In the case of only two objects p and q, only a 2-chain CHMM is required, for which the posterior can be further simplified to

$$\begin{aligned} P(S|X) = & \frac{P(s_1^p)P(\mathbf{x}_1^p|s_1^p)P(s_1^q)P(\mathbf{x}_1^q|s_1^q)}{P(X)} \times \\ & \times \prod_{k=2}^{K} \Big(P(s_k^p|s_{k-1}^p)P(s_k^q|s_{k-1}^q)P(s_k^p|s_{k-1}^q) \times \\ & \times P(s_k^q|s_{k-1}^p)P(\mathbf{x}_k^p|s_k^p)P(\mathbf{x}_k^q|s_k^q) \Big), \end{aligned} \tag{5.10}$$

where s_k^p is the state of chain p and s_k^q is the state of chain q at time k. The problem of interaction event modelling involves analysing jointly the states of multiple objects to model their influence on each other. This implies the need for accounting for the influence of the current state of the interacting processes while estimating the next. The problem of estimating the most probable next state of the interaction is similar to the single process case (Eq. 5.1), except that the posterior $P(S|X)$ is estimated as in Eq. 5.10 to take into account the coupling.

5.3 Summary

This chapter categorises objects' interactions to be either with static objects (*dynamic-static interactions*) or between dynamics objects (*dynamic-dynamic interactions*). In case of *dynamic-static interactions* the state of either dynamic or static objects needs to be estimated with respect to the other interacting object. This could be the state estimation of the temporarily stopped object, such as an abandoned baggage, with respect to that of its owner or it could be state estimation of dynamic object with other structural objects. Approaches based on Bayesian networks, particularly Hidden Markov Models (HMM) and their variants are found to be well suited for recognising these kinds of interactions.

Similarly, in case of recognising *dynamic-dynamic interactions* graphical models such as the dynamic Bayesian network and variants of HMM are used. This is due to their inherent capability of state estimation and modelling inter-process coupling. This chapter gives the formulation for a wide range of HMM variants to model coupling between the

interacting process, some of which may be used in future for recognising interactions in video while others are widely in use such as Coupled Hidden Markov Models (CHMM). The state of art on interaction recognition is limited to camera field of view only. In the next chapter we will discuss the proposed (*dynamic-static interactions*) and (*dynamic-dynamic interactions*) recognition technique. Furthermore we will extend it to recognise interactions in regions uncovered by cameras by utilising trajectories estimated using multi-modal tracking.

Chapter 6

Recognising Interactions

6.1 Introduction

The event recognition problem can be decomposed into three main steps: (i) the *extraction* of objects of interest, (ii) the *tracking* of the objects, and (iii) the *recognition* of events generated by the tracked objects. Given the detection and tracking information using algorithms discussed in Chapter 4, in this chapter we discuss the proposed event recognition technique for each object as well as for interactions between them. The use of multiple vision sensors for detecting events in wide areas is of great interest for applications such as surveillance and sports analysis. Because many scenes to be monitored cannot be covered completely by a single sensor, multiple cameras are used to observe the behaviour of the objects [27, 29]. However, in many cases even multiple cameras cannot cover the whole scene, thus reducing the number of available observations. The missing information to overcome this problem, can be estimated either by a prediction based on the objects state in the cameras' fields of view and their motion dynamics [159, 212] or by coupling the cameras with sensors having a wider field of observation. Microphones are examples of sensors with a wider field of observations (the sound field), as discussed in Sec 4.3. In this chapter we will discuss the interaction recognition within the camera field of view (FOV) as well as outside using multi-modal trajectories. The overall block diagram of the system is shown in Fig. 6.1.

This chapter is organised as follows: in Sec 6.2 the problem of interaction recognition is formalised as a state estimation problem. This state estimation aims to recognise either interaction between dynamic and static objects or between multiple dynamic objects. The former is discussed in Sec. 6.3 whereas later is discussed in Sec. 6.4. The results and evaluation for both these categories of interaction are discussed in Sec. 6.5. Finally,

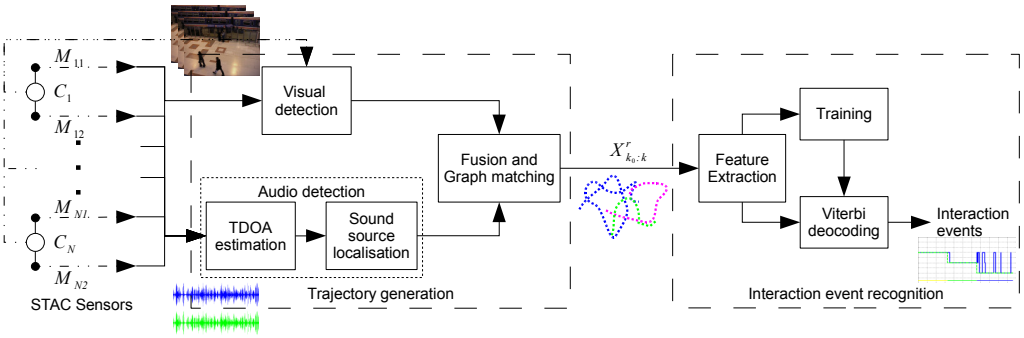
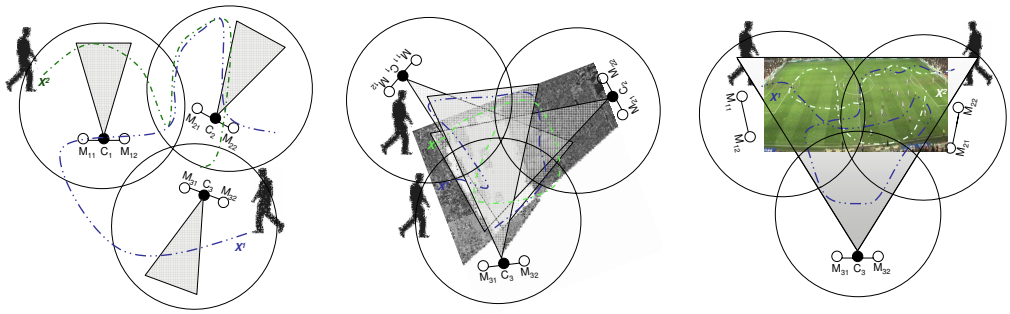


Figure 6.1: System block diagram showing multi-modal detection, tracking and interaction recognition.



(a) (b) (c)

Figure 6.2: Sample environment configurations showing 2 targets, X^1 (green single dotted line) and X^2, (blue double dotted line) moving under the coverage area of multiple heterogeneous sensors (M_{i1} and M_{i2} are the pair of microphones with camera C_i). (a) Non-overlapping field of view and overlapping field of sound. (b) Overlapping field of view and sound field with common ground plane. (c) Single wide-range visual sensor and multiple microphones with overlapping sound field.

the chapter is summarised in Sec. 6.6.

6.2 Problem formulation

Let a wide-area be monitored by a set $C = \{C_1, \ldots, C_N\}$ of N cameras with non-overlapping FOV. Let each camera be equipped with a microphone pair, with $M = \{M_1, \ldots, M_N\}$ being the set of N microphone pairs, where $M_i = (M_{i1}, M_{i2})$. We assume that the microphones' sound field is wider than the corresponding cameras' field of view and that the sound field of multiple microphone pairs M_i overlap each other (Fig. 6.2). This heterogeneous sensor network enables extended tracking and allows estimation of trajectories in regions within and outside camera FOV. Given these extended tracks, the

goal is to perform the activity recognition in the environment including regions obscured by cameras.

As discussed in Sec. 5.2.3, let $X^r_{1:k} = \{\mathbf{x}^r_1, \mathbf{x}^r_2, \ldots, \mathbf{x}^r_k\}$ be the track associated with the r^{th} object up to time k and $\lambda = \{A, B, S, s_0\}$ be a continuous distribution first-order Hidden Markov Model. The observed objects interact with the environment as well as with other objects. The problem of interaction event detection can therefore be divided into two categories of state estimation, namely dynamic object with respect to static objects (see Fig. 6.5 and Fig. 6.4) and dynamic objects with respect to each other (see Fig. 6.6). More formally, the problem of interaction event detection can be defined as follows.

- **Dynamic-static interaction**. Given the model λ and the tracks $X^p_{1:k}$ and $X^q_{1:k}$ from a dynamic and a static object, it is desired to generate the sequence in which they interact and have gone through various states in S.
- **Dynamic-dynamic interaction**. Given the model λ, the tracks $X^p_{1:k}$ and $X^q_{1:k}$ for two interacting objects p and q, and set of interaction event templates $\{E_1, \cdots, E_N\}$, it is desired to find the interaction between the objects.

Interaction event analysis will be performed using the track information and the available contextual information. The next section discusses proposed approaches for solving these problems.

6.3 Interaction among dynamic and static objects

In case of one static and another dynamic object, it is enough to estimate the state of one object only. This implies generating the optimal state sequence observed by one of the objects with respect to the other. Given $s_1, \cdots, s_{k-1}, s_k$ are the most probable states from time 1 to k. To find the single most probable state sequence, S_k, for the given observation sequence $X_k = \{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$, a quantity δ_{ik} needs to be defined [46] as

$$\delta_{ik} = \max_{s_1, \cdots, s_{k-1}} P(s_1, \cdots, s_{k-1}, s_k = s_i, \mathbf{x}_1 \cdots \mathbf{x}_k | \lambda), \tag{6.1}$$

i.e., δ_{ik} is the highest probability along a single path, at time k, with s_k being the state s_i. According to the Markovian assumption, the conditional probability distribution of future states depends on the current state only and not on past states, hence using the Forward Viterbi [46] we have

$$\delta_{jk+1} = \max_{1 \leq i \leq N} [\delta_{ik} a_{ij}] b_{j\mathbf{x}_{k+1}}. \tag{6.2}$$

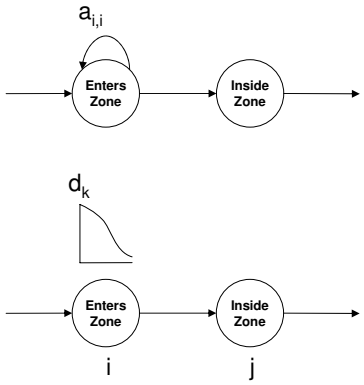


Figure 6.3: Examples of self-transition modelling for a Hidden Markov Model: (top) self-transition probability (a_{ii}); (bottom) self-transition replaced with a state occupancy duration *pdf*.

Finally, the most likely hidden state s_{k+1} up to time $k+1$ is computed as

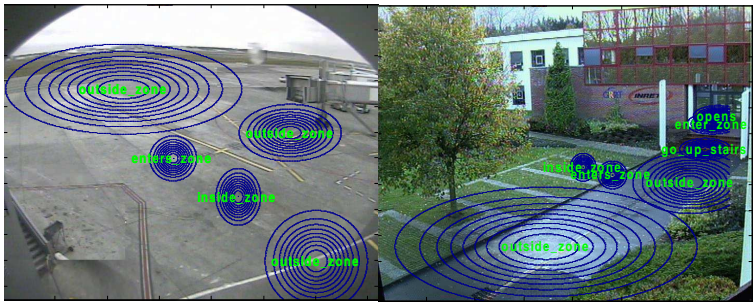
$$s_{k+1} = \varphi(\arg\max_{1 \leq i \leq N}[\delta_{ik+1}]), \tag{6.3}$$

where function $\varphi(.)$ returns the i^{th} state from the set of discrete states S.

This simple Hidden Markov Model is unable to completely model certain events due to the duration distribution of the observation sequence for their states. The Markovian assumption constrains the state occupancy distribution to be exponential [213]. The estimation of the most likely path S_K is problematic, because a state with high self-transition probability a_{ii} can cause the algorithm to stay in this state for a longer interval. To avoid such self-transitions, we use Hidden Semi-Markov Models (HSMM) [214] to enable the explicit modelling of duration probability distribution d_k. The duration probability distribution is the probability of staying at least for a duration τ in the state s_j, with $1 \leq \tau \leq D_j$ (Fig. 6.3). To compute the most likely state sequence S_K using the durational distribution, we use the forward Viterbi algorithm and solve Eq. 6.2 as

$$\delta_{jk+1} = \max_{k_e \leq \tau \leq D_j + k_e} (\max_{1 \leq i \leq N} (\delta_{ik} a_{ij}) d_{j\tau} b_{k,\mathbf{x}_{k+1}}). \tag{6.4}$$

Given the model λ and the duration probability distribution d_{jk}, we can now use Eq. 6.4 to compute the best state sequence by performing the HMM decoding using the Viterbi algorithm. The state transition probabilities a_{ij} can be defined empirically or, if there is sufficient training data, they can be calculated using the Baum-Welch algorithm [215]. In order to use the Viterbi algorithm we need first to model the duration probability distribution d_{jk} and the observation sequence.



(a) (b)

Figure 6.4: Scene-centric distribution model showing states represented as multivariate Gaussians. (a) Sequence AP-11 C4; (b) Sequence BE-19 C1.

6.3.1 Duration probability distribution

The duration probability distribution $d_{j\tau}$ can be modelled using different parametric duration distributions. We evaluate two distributions, namely the *half-normal distribution* and the *triangular distribution*, which are well adapted to the problem at hand. The half-normal distribution, $d_{j\tau}$, can be expressed as

$$d_{j\tau} = \frac{1}{\sigma}\sqrt{\frac{2}{\pi}} \exp\left(-\frac{1}{2}\left(\frac{\tau-\mu}{\sigma}\right)^2\right), \tag{6.5}$$

where σ is the standard deviation, computed as $3\sigma = \tau$, and μ is the mean. The mean is the time k_e when the object transits into the state and $k_e \leq \tau \leq D_j + k_e$ and D_j is the duration of the state s_j. The triangular distribution, $d_{j\tau}$, can be expressed as

$$d_{j\tau} = \frac{(-\tau + k_e + D_j)}{D_j}. \tag{6.6}$$

In case of events with high self-transitions a uniform distribution can be used which implicitly converts HSMM to HMM. The selection of the appropriate distribution, for the specific event or activity, can be done using the Chi-square test.

6.3.2 Object-centric and scene-centric models

The estimation of the emission probabilities b_{jk} depends on the possible states of the object. These states can either be associated with the static object with respect to other interacting objects in the scene or can be associated with the dynamic object with respect to the static object with which it is interacting. We propose and evaluate two models to estimate b_{jk}, namely a scene-centric and an object-centric model. In the

scene-centric approach, the b_{jk} are modelled as a multivariate Gaussian. For each j^{th} state we use a multivariate Gaussian $\mathcal{N}_j(\mu_j, \Sigma_j)$ with mean μ and covariance Σ as

$$b_{jk} = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2}(\mathbf{x}_k^r - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_k^r - \mu_j)\right), \tag{6.7}$$

where $\mathbf{x}_k^r$ is the position and size of the r^{th} object at time k, $n = 4$, $\mu_j = \{\mu_x, \mu_y, \mu_w, \mu_h\}$ is the mean of the state where (μ_x, μ_y) represents the (x, y) position of the state on the image and (μ_w, μ_h) is the mean of the objects (w, h) in that region. $|\Sigma_j|$ is the determinant of the covariance matrix Σ_j, which we assume to be a diagonal matrix: $\Sigma_j = diag[\sigma_x^2, \sigma_y^2, \sigma_w^2, \sigma_h^2]$. The values for μ_j and Σ_j are set based on the contextual information specific to the task at hand (Fig. 6.4). In scene-centric model, states are associated to structural objects and hence fixed mean location. At each time k the emission probability b_{jk} is computed for each moving object with respect to each of these states given the position and size of the object.

In the *object-centric* approach, we model b_{jk} as a multivariate distribution composed of a mixture of a normal and a uniform distribution $\mathcal{N}_j(\mu, \Sigma, \rho, F, G)$ with mean μ, covariance Σ, weight ρ and range of uniform distribution $[F, G]$:

$$b_{jk} = \frac{\rho}{(2\pi)^{\frac{K}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left(-\sum_{i=1}^{K}\left[\frac{(\theta_i - \mu_{\theta_i})^2}{2\sigma_{\theta_i}^2}\right]\right) + \frac{(1-\rho)}{\pi}\prod_{i=1}^{K}\left[\frac{\psi_{\theta_i}}{\sigma_{\theta_i}}\right], \tag{6.8}$$

where K=2; θ_1=x and θ_2=y. Therefore σ_x and σ_y are the standard deviations, respectively. The second term accounts for rapid change in probability after σ so that the HMM can quickly move to the next state. The functions ψ_i are piecewise binary and defined as

$$\psi_x = \begin{cases} 1 & \text{for } F_x < x < G_x \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

and

$$\psi_y = \begin{cases} 1 & \text{for } \zeta(F_y) < y < \zeta(G_y) \\ 0 & \text{otherwise} \end{cases}, \tag{6.10}$$

where $[F_x, G_x]$ and $[F_y, G_y]$ are the ranges of the uniform distribution along (x, y) axis, $\zeta(x) = \pm\sigma_y\sqrt{1 - (\frac{x-x_c}{\sigma_x})^2} + y_c$, with (x_c, y_c) representing the object centroid around which the model is built, and $\pi\prod_{i=1}^{2}\sigma_{\theta_i}$ is the area of an ellipse. $|\Sigma_j|$ is the determinant of the covariance matrix, with $\Sigma_j = diag[\sigma_x^2, \sigma_y^2]$ and therefore $|\Sigma_j| = \sigma_x\sigma_y$ in Eq. 6.8. The values of the elements in Σ_j depend on the state to be modelled, whereas the value of μ is

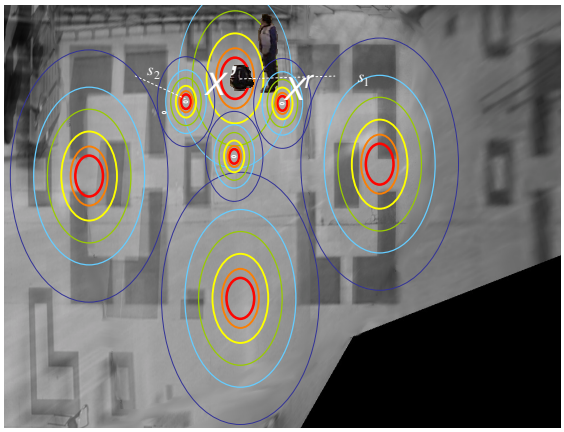


Figure 6.5: Multivariate object-centric distribution model. The means μ_{s_1} and μ_{s_2} of the distributions of the states is placed on the line joining the centroids of the objects $X^{'}$ and X^{r}.

assigned dynamically. This is the key point of the proposed *object-centric* modelling. The value of μ of the first state is set as the centroid of the reference object $X_k^{'}$ (Fig. 6.5). The remaining state distributions are then placed around $X_k^{'}$ to estimate the possible state of $X_k^{'}$ with respect to the objects X_k^r. The μ of the other states are positioned on the line passing through the centroid of the two objects ($X_k^{'}$ and X_k^r) at a distance that is a function of the variances of the states to be detected. The rationale for using Gaussian functions instead of hard boundaries and fixed threshold is to increase the flexibility of the algorithm in order to detect several different events in different scenarios.

Moreover, as the behaviours of objects in real scenarios are generally characterised by fuzzy boundaries between different states, a progressive transition from one state to another is preferred to a fixed threshold-based transition [216]. If computational time is an issue, it is possible to use in the proposed framework a uniform distribution to model the states with equal state transition probabilities among the states.

The estimation of the emission probabilities b_{jk} using the proposed object-centric approach completes the computation of the HMM parameters. These parameters are now used to compute the most likely state sequence S_K for each object r by applying the *Forward Viterbi algorithm* every N^V observations. The last state s_k of the state sequence is then used as the initial state s_0 for next computation. The event detection algorithm using the *forward Viterbi algorithm* for HSMM is summarised in Algorithm 2.

Table 6.1 shows the performance comparison between the proposed algorithm (with the two different duration distributions) and the HMM-based algorithm without state duration modelling [202]. The comparison was done on the ETISEO [140] sequences

Algorithm 2 Event Detection

$S = \{s_1, \cdots, s_{N^S}\}$: events (states that an object can acquire)
a_{ij} : state transition probabilities between state i to j
μ_j : mean for each state j; Σ_j : covariance matrix for each state j
$X^r_{1:K}$: trajectory for object r upto time K ; *count* : counter

for $k = 1$ to K **do**
  for $j = 1$ to N_T **do**
    Compute $b^r_{j,k}$:

$$b^r_{j,k} = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2}(X^r_k - \mu_j)^T \Sigma_j^{-1} (X^r_k - \mu_j)\right)$$

  end for
  $count = count + 1$
  if $count = n$ **then**
    Initialise initial state s^r_0
    if $s^r_0 = -1$ **then**

$$s^r_0 = \zeta(\max_{j=1\ldots l} b^r_{jk})$$

      where ζ returns s_j corresponding to b^r_{jk}
    end if
    Apply Forward Viterbi Algorithm:

$$\delta^r_{jk+1} = \max_{k_e \le \tau \le D_j + k_e} (\max_{1 \le i \le N} (\delta_{ik} a_{ij}) d_{j\tau} b_{k\mathbf{x}_{k+1}})$$

$$s^r_{k+1} = \arg\max_{1 \le j \le N} [\delta_{jk+1}].$$

$$s^r_0 = s^r_{k+1}$$

  end if
end for

Table 6.1: Performance comparison between the proposed HSMM algorithm with half-normal and triangular distribution for state occupancy duration and event detection using HMM without state duration modelling.

	HMM	**HSMM-TRI**	**HSMM-HN**
AP	0.882	0.980	0.956
BE	0.790	0.966	0.980
Total	0.847	0.965	0.975

AP-11 (C1 and C4) and BE-19 (C1) using the CREDS evaluation metrics [217] in which the score associated with the correct detections is defined as

$$score(k, AD_\tau, GT_\tau) = \begin{cases} 0 & k \in]-\infty, k_b[, k, \in]D, \infty[; \\ \frac{score_{max}}{k_a - k_b} k - k_b & k \in [k_b, k_a] \\ score_{max} & k \in]A, 0[; \\ \frac{score_{max}}{k_d}(k_d - k) & k \in [0, k_d]; \end{cases}, \tag{6.11}$$

where $k_b < k_a < k_e < k_d$ are the thresholds on time for considering an event to be an anticipated, correct or delayed event, AD_τ and GT_τ are the event durations in automatic

detected results and ground truth and $score_{max}$ is computed as

$$score_{max} = \begin{cases} 50\left[2 - \left(1 - \frac{AD_\tau}{GT_\tau}\right)\right] & \frac{AD_\tau}{GT_\tau} \in [0, 2] \\ 50 & \frac{AD_\tau}{GT_\tau} \in]\,, \infty[\end{cases}. \tag{6.12}$$

It is possible to notice that the duration modelling in HMM improves the results. In summary, the HSMM model with triangular distribution performed at 96.5%, the HSMM model with half-normal distribution obtained a score of 97.5% and the HMM scored 84.7%. In the next section we will discuss the proposed approach for recognising interactions among multiple dynamic objects.

6.4 Interaction among dynamic objects

6.4.1 Problem definition

When multiple interacting objects are dynamic, the state sequences for all such objects need to be estimated. Furthermore, since their states are dependent on the states of other interacting objects it is important to model the coupling between them. Here we first consider a few examples of interaction events where for simplicity only two objects are considered. Methods for modelling these interactions will be discussed in Sec. 6.4.3. Let the objects p and q be at positions (x^p, y^p) and (x^q, y^q) at time k_0 with a certain velocity on a random path. Let at a certain time $k_1 (> k_0)$, the two targets starts with approaching a meeting point (x^m, y^m) such that at time $k_1 + n_r$, with $r \in \{p, q\}$, they arrive at the meeting point either together ($n_p = n_q$) or one after another ($n_p \neq n_q$). After staying at the meeting point for a duration n_d, the two targets start following a common or separate random path. Our goal is to model the interaction between these targets by considering their joint states.

Let there exist a full coupling between the target states modelled as 1-lag, 2-chain Coupled Hidden Markov model (Fig. 5.3(f)). Let X^p and X^q represent the set of observations; S^p, S^q be the set of states for the two chains and $P(s^i_{k+1}|s^j_k)$, $i, j \in \{p, q\}$ be the transition probabilities among these states. The observation $\mathbf{f}^{pg}_k$ is the feature vector computed using the current and the previous positions, of the interacting objects, obtained from their trajectories $X^r_{k_0:k}$ up to time k for object $r \in \{p, q\}$.

Let us define the five types of interactions, namely *follow-reach-go together*, *approach-meet-go separately*, *approach-meet-go together*, *change direction-approach-meet-go separately* and *change direction-approach-meet-go together*. Figure 6.6 shows the illus-

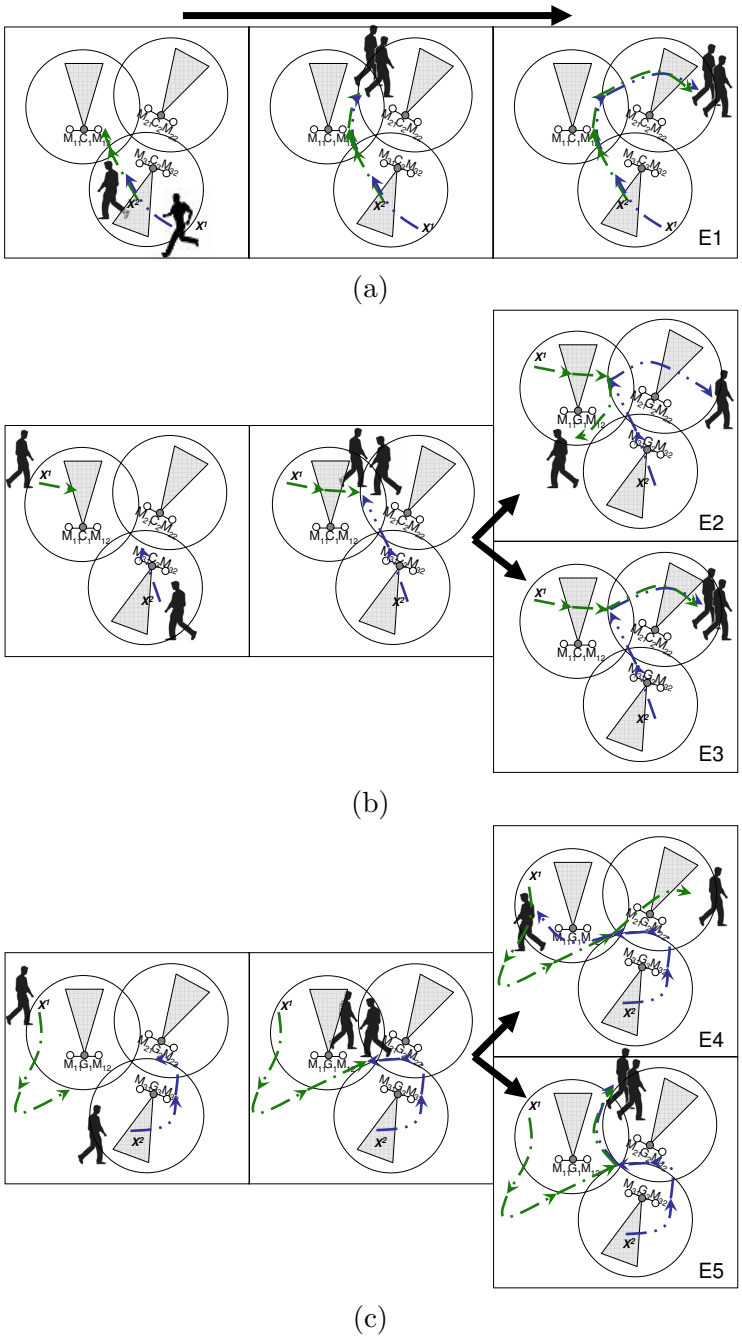


Figure 6.6: Illustration showing different stages of five interactions (M_{i1} and M_{i2} are the pair of microphones with camera C_i, green single dotted line represents object X^1 and blue double dotted line represents object X^2). (a) follow-reach-go together (E1), (b) approach-meet-go separately (E2) and approach-meet-go together (E3), (c) change direction-approach-meet-go separately (E4) and change direction-approach-meet-go together (E5).

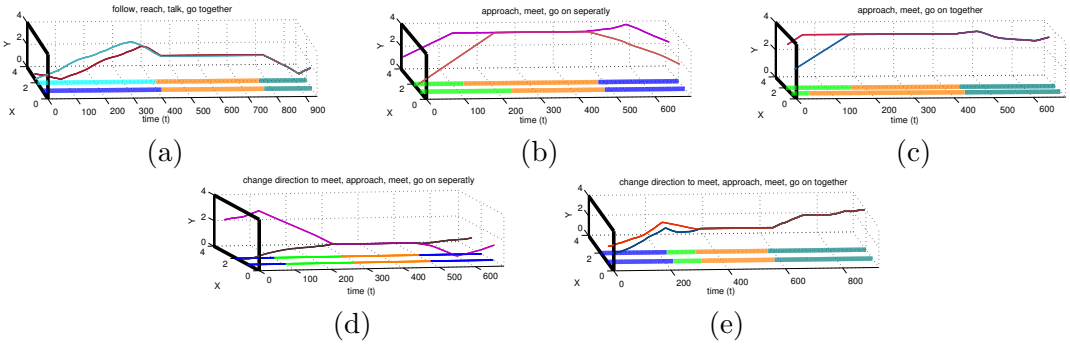


Figure 6.7: Sample trajectories for each interaction (E1 to E5) between 2 targets. (a) follow-reach-go together (E1), (b) approach-meet-go separately (E2), (c) approach-meet-go together (E3), (d) change direction-approach-meet-go separately (E4) and (e) change direction-approach-meet-go together (E5). The horizontal coloured lines along the time axis shows the change in states (dark blue: walking/going separately; green: approaching; brown: meeting and waiting to meet; teal: going together; cyan: follow).

tration of these interactions.

In the *follow-reach-go together* interaction (E1), two objects X^p and X^q follow a random path. At time k, X^q starts approaching the current position of X^p. At $k+n_1$, X^q approaches X^p, i.e. $X^q_{k+n_1}(x,y) = X^p_k(x,y)$. X^q then starts following X^p's path with higher speed and meets X^p after n_2 time steps at time $k+n_1+n_2$. From time $k+n_1+n_2$ both objects move together (Fig. 6.6(a)).

In the *approach-meet-go separately* interaction (E2), two objects X^p and X^q start with moving on a random path. At time k, both objects start approaching a meeting point and wait for each other. After arriving, both objects first wait and then go separately on different paths. The *approach-meet-go together* interaction (E3) is similar to E2, with the difference that after meeting the two objects move together (Fig. 6.6(b)).

In the *change direction-approach-meet-go separately* interaction (E4), objects X^p and X^q start moving on a random path as in case of E2. At time k, X^q starts approaching X^p with an increased speed and changing direction continuously to reach X^p. At $k+n$, X^q meets X^p, i.e., $X^q_{k+n}(x,y) = X^p_{k+n}(x,y)$. After arriving, both objects first wait and then move separately on some random path (Fig. 6.6(c)).

Finally, the *change direction-approach-meet-go together* interaction (E5) is similar to E4, with the difference that after meeting the two objects move together. The corresponding sample trajectories are shown in Fig. 6.7.

6.4.2 Interaction features

The interactions could be identified by analysing if the two objects coexist at a certain time in the same region. However, this would make the approach heavily dependent on contextual information (e.g., the presence of a kiosk). The five interactions E1 to E5 can be divided into the following sub-events: *following*, *approaching*, *meeting*, *going together* and *walking/going separately*. These sub-events have similar values for features (such as speed, distance from scene boundary, and direction), to other non-interacting objects. To identify such interactions, the combination of relative features among objects offers a better representation of the states.

For example in both the *following* and the *approaching* states the objects will be getting closer to each other and therefore the relative distance will be a continuously decreasing (Fig. 6.8(c)). However, the relative direction will have distinguishing properties as in the case of *following*, and the two objects will have the similar directions. Similarly, in the case of *meeting* and *going together*, the relative distance will be approximately zero; and the magnitude of the speed will be the differentiating factor (Fig. 6.8(a)). The relative direction together with the relative distance helps in differentiating between *going together* and *going separately* (Fig. 6.8(b-c)), as two objects going in the same direction with same speed can be far from each other. The relative distance helps in solving such ambiguities as its value will be high in such situations. The clustering is therefore performed on a 5-dimensional feature space $\mathbf{f}$, represented as

$$\mathbf{f}_k^{pq} = \left(\nu_k^p,\ \nu_k^q,\ \phi_k^{pq},\ \varrho_k^{pq},\ \dot{\varrho}_k^{pq}\right), \tag{6.13}$$

where ν_k^p and ν_k^q are the magnitudes of speed; i.e.,

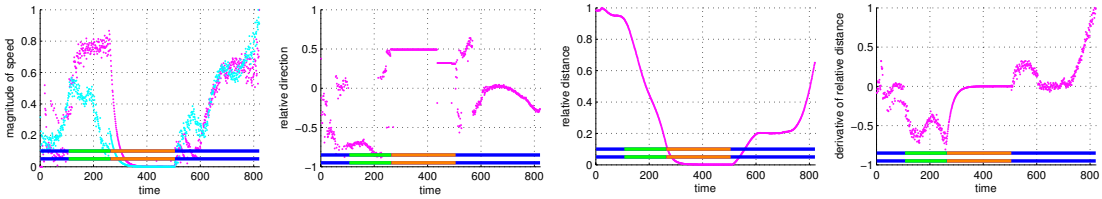
$$\nu_k^p = \sqrt{\nu_{x_k^p}^2 + \nu_{y_k^p}^2}, \tag{6.14}$$

with $\nu_{x_k^p}$ and $\nu_{y_k^p}$ representing the horizontal and vertical components of the velocity and ϕ^{pq} is the relative direction, defined as

$$\phi_k^{pg} = \arctan\left(\frac{y_k^p - y_{k-1}^p}{x_k^p - x_{k-1}^p}\right) - \arctan\left(\frac{y_k^q - y_{k-1}^q}{x_k^q - x_{k-1}^q}\right), \tag{6.15}$$

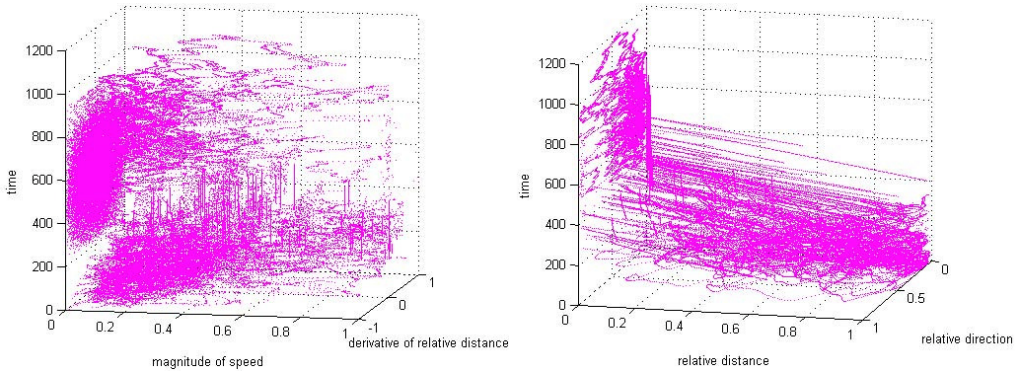
with ϱ_k^{pg} being the relative distance, given as

$$\varrho_k^{pq} = \sqrt{(x_k^p - x_k^q)^2 + (y_k^p - y_k^q)^2}, \tag{6.16}$$



(a) (b) (c) (d)

Figure 6.8: Sample features for a 2-target interaction computed on synthetic trajectories. The coloured bar at the bottom of each graph is the ground truth of the targets individual states for interaction E2 (dark blue: walking/going separately; green: approaching; brown: meeting and waiting to meet). (a) the magnitude of speed for both targets, (b) the angle between the two targets, (c) relative distance between the targets and (d) derivative of the relative distance between the targets.



(a) (b)

Figure 6.9: Sample extracted features from 50 synthetic trajectories for the interaction E3. (a) Magnitude of speed vs. relative distance over time. (b) Relative direction vs. derivative of relative distance over time (2 clusters).

and $\dot{\varrho}^{pq}$ is the derivative of the relative distance, computed as

$$\dot{\varrho}_k^{pq} = \frac{\varrho_k^{pq} - \varrho_{k-1}^{pq}}{\Delta k}. \tag{6.17}$$

Note that the relative direction and distance are computed based on the current position of two interacting targets and not from a fixed reference point. Figure 6.9 shows the projection of the features computed on 50 trajectory pairs (E3). For visualisation, we present 2 features per graph only. It can be seen that these features form certain clusters in the space.

6.4.3 Interaction event sequence estimation

We perform Interaction event recognition (IER) with a trained CHMM [166,218]. The model λ consists of the initial state, the state transition and the emission probabilities. The states are discrete random variables (both the initial state and the state transition will result in one of the possible states). A distribution suited to model this discrete random variable is a multinomial distribution, as each trial results in exactly one out of a fixed finite number of possible outcomes. Hence both state transition and initial state probabilities of each chain are chosen to be multinomial distributions. On the other hand, the emitting symbols are continuous random variables modelled using a continuous distribution. We choose this distribution to be a multivariate Gaussian, representing the projection of the trajectory onto the 5-dimensional feature space $f : X^p_{k_0:k} \times X^q_{k_0:k} \rightarrow \mathbb{R}^5,\ \forall\ p, q$. We train the CHMM using the Expectation Maximisation (EM) algorithm [219] and the Maximum a Posterior (MAP) approach as, unlike Maximum Likelihood (ML), MAP incorporates the prior distribution over the quantities to be estimated to achieve a better optimisation [166]. We therefore aim to maximise

$$\mathcal{Q}(\lambda) = \int Q(S|\lambda) \log P(S, X|\lambda) dS + P(\lambda), \tag{6.18}$$

where $Q(S|\lambda)$ is the probability of a state given the model λ. λ consists of the state transitions, the initial state probabilities and the emission probabilities and $P(\lambda)$ is the probability of the model parameters. The *E-step* aims to identify the states S given the data and model parameters from the previous step, while maximising Eq. 6.18. In the *M-step*, Eq. 6.18 is maximised given the states S and the model λ. The integral is solved by expanding Eq. 6.18, for each chain, into the initial state probability, the state transition probabilities, the observation model and the log prior $\log P(\lambda)$ of each chain. This results in update, for each chain, of mean, covariance, initial state probabilities and state transition probabilities [166].

The initialisation of the EM algorithm will have significant effect on the final trained CHMMs. A Gaussian Mixture Model (GMM) is applied to estimate the initial clusters, since in a GMM, the probability of a cluster u given the data point $P(u|\mathbf{f})$, in our case representing the features, plays a key role in the soft cluster membership. The total number of different states in the interaction event is defined by the number of clusters obtained by the GMM clustering. For example, in interaction E3 there are three possible states (i.e., *approaching*, *staying at a meeting point*, and *going together*), and hence we perform clustering with 3 clusters.

The convergence of EM is also prone to local maxima. Since the function we want to optimise (Eq. 6.18) will have several local maxima, a post-processing step is performed to solve this problem. The problem can be avoided using two solutions. The first solution is post-processing the EM results by applying the algorithm directly and analysing the obtained clusters based on their population and compactness (covariance) [220]. However, in case of multiple sub clusters with similar densities and compactness, the problem will remain unsolved. The second solution is to improve the initialisation of the EM algorithm to eliminate the problem.

The EM algorithm is applied iteratively to learn the CHMM model parameters until the change in log-posterior is less than a tolerance $\eta = 10^{-4}$. The value of this threshold is selected empirically in order to avoid over–training and to ensure termination of the algorithm after achieving the desired accuracy.

The recognition of the interactions is performed by applying the CHMM Viterbi decoding [221] using the trained model parameters. The posterior $P(S, X|\lambda)$, in Viterbi decoding, is calculated using Eq. 5.10. The decoding strategy is preferred here over the evaluation, as it does not require the event templates to be recognised but allows the generation of a sequence of activities performed by the targets. The steps for IER are summarised in Algorithm 3.

6.5 Results

6.5.1 Evaluation metrics

To evaluate the event detection results, we estimate the *precision*, the *sensitivity* and the *accuracy* of the algorithms. Let FP be the number of false positive detections, TP the number of true positive detections, and FN the number of false negative detections. The *precision* is defined as $TP/(TP+FP)$ and the *sensitivity* is defined as $TP/(TP+FN)$. The accuracy is measured both at event level and sequence level. The event-level accuracy is defined as

$$\gamma_{s_i} = \left(1 - \frac{|GT^{s_i} - AD^{s_i}|}{\tau_{s_i}}\right) \times 100, \tag{6.19}$$

with τ_{s_i} representing the mean duration of an event s_i over the entire dataset, GT^{s_i} and AD^{s_i} are the start and end frame numbers of the event in the ground truth and automatic detection results respectively. Note here we used normalising factor τ_{s_i} to be the mean duration of an event s_i instead of a fixed value as in case of [222]. This makes the event-level accuracy to be more sensitive for events with shorter mean duration and less sensitive

Algorithm 3 Interaction Event Recognition

λ : model parameters vector containing state transition probabilities, initial state probabilities and observation model parameters
μ_{S^i} : mean for each state i;
Σ_{S^i} : covariance for each state i;
μ^i_{GMM} : mean of i^{th} cluster of GMM;
Σ^i_{GMM} : covariance of i^{th} cluster of GMM;
n : total number of states;
η : threshold of log-posterior;
$X^r_{1:k}$: trajectory for object r upto time k;
$\mathbf{f}$: feature vector;

Compute: $X^r_{1:k}$
Compute:

$$\mathbf{f}^{pg}_k = (\nu^p_k,\ \nu^q_k,\ \phi^{pg}_k,\ \varrho^{pg}_k,\ \dot{\varrho}^{pg}_k)\,,$$

Train CHMM:
Initialise CHMM using GMM clustering on $\mathbf{f}$
for $i = 1$ to n **do**
    $\mu_{S^i} \leftarrow \mu^i_{GMM}$
    $\Sigma_{S^i} \leftarrow \Sigma^i_{GMM}$
end for
Apply EM to train CHMM by maximising

$$\mathcal{Q}(\lambda) = \int \mathcal{Q}(S|\lambda) \log P(S, X|\lambda) dS + P(\lambda)$$

repeat $\kappa = \kappa + 1$
    E-step compute: $\mathcal{Q}(\lambda; \lambda_\kappa)$
    M-step :

$$\lambda_{i+1} = \arg\max_{\lambda} \mathcal{Q}(\lambda; \lambda_\kappa)$$

until $\mathcal{Q}(\lambda_{\kappa+1}; \lambda_\kappa) - \mathcal{Q}(\lambda_\kappa; \lambda_{\kappa-1}) \leq \eta$
$\hat{\lambda} = \lambda_{\kappa+1}$
Apply CHMM Viterbi Algorithm using $\hat{\lambda}$ to estimate the state sequence.

for events with longer duration. This helps in better estimating the accuracy of the system for each event.

The sequence level accuracy is computed by creating two matrices GT and AD both of dimensions $N^E \times N^F$ where N^E and N^F are the total number of possible events and number of frames in the sequence. Each column indicates a frame and each row indicates an event. Each entry in GT indicates if the event represented by the row is occurring in the frame represented by the column or not. Similarly, each entry in AD indicates if the event represented by the row is detected in the frame represented by the column or not. The sequence level accuracy is then computed as

$$\gamma = \left(1 - \frac{1}{N^F} \sum_{k=1}^{N^F} \gamma_k\right) \times 100, \tag{6.20}$$

where

$$\gamma_k = \begin{cases} 1 & \sum_{e=1}^{N^E} |GT_k^e - AD_k^e| > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{6.21}$$

The event-level accuracy γ_{s_i} indicates the performance for each type of event whereas, the sequence-level accuracy indicates the overall performance for the entire sequence.

The evaluation of dynamic-interaction event recognition is also performed by participating in ETISEO evaluation cycle (Table 6.3). In the next sub-section (Sec. 6.5.2) the interactions recognised and the datasets used in this work are introduced. The discussion on scores obtained using these performance measure along with subjective evaluation is then discussed in Sec 6.5.3 and Sec 6.5.4.

6.5.2 Experimental set-up

We demonstrate the performance of the proposed algorithm on standard event detection sequences from real surveillance scenarios as well as on other sports and synthetic datasets. The real surveillance scenarios are from ETISEO [140] (including sequences from i-Lids [174] dataset), CAVIAR (leaving bags behind) [142] and PETS 2006 [143] datasets. The sports scenarios are from two Football matches from MediaPro [144] and IISA [141]. These sequences (Table 6.2) include indoor and outdoor scenarios with pedestrians, vehicles, objects and their interactions. The datasets mentioned in Table 6.2 are used for the testing of the algorithms whereas the training for the computation of mean duration of each event is performed using ground truth for the remaining part of the ETISEO dataset consisting of annotated videos from Airport (AP) (12min 4sec), Road (RD) (5min 3sec) and Building entrance (BE) (1min 47sec) of scenarios.

Three scenarios from ETISEO datasets are used to evaluate the performance of dynamic-static event recognition, namely airport (Fig. 6.10(a-b)), road (Fig. 6.10(c)) and building entrance (Fig. 6.10(d-e)). These scenarios contain interactions such as activity of dynamic objects with (i) a marked zone (such as parking area), (ii) steps (outside entrance) and (iii) doors and door control. The CAVIAR (Fig. 6.10(j)) and PETS 2006 (Fig. 6.10(g)) datasets also contain dynamic-static interactions, particularly abandoned baggage scenarios. Unlike the CAVIAR dataset, PETS 2006 is recorded in a real metro station scenario and hence it also contains dynamic-dynamic interactions such as *approach-meet-go together*. Similarly, the two soccer match scenarios also contains dynamic-dynamic interactions such as *approach-meet-go separately*, *follow-reach-go together* and *change direction-approach-meet-go together*.

Table 6.2: Summary of the test datasets used in the experiments

Dataset	**Seq.**	**Cameras**	**Resolution**	**No. of frames**	**Frame rate**
ETISEO	AP-11	C4, C7	720×576	805 , 805	12.5
	BE-19	C1, C4	768×576	1025 , 950	25
	RD-6	C7	720×576	1201	25
PETS	S1	C3	720×576	3022	25
	S2	C3	768×576	2551	25
	S3	C3	768×576	2372	25
	S5	C3	720×576	3402	25
	S6	C3	720×576	2802	25
CAVIAR	CL1	NA	384×288	1441	25
	CL2	NA	384×288	1357	25
Soccer	S7	C1, C3	1920×1080	6002 , 6002	25
	S8	NA	1440×537	277	25
Total number of frames				34014	-

The real datasets are limited in instances of each interaction, and they are not sufficient for training as well as for testing. In the case of a dynamic-dynamic interaction we use synthetic data to overcome this limitation. The synthetic dataset (D1) consists of 100 trajectory pairs for each dynamic-dynamic interaction event and approximately 750 data points per trajectory.

The synthetic events consist of trajectories generated using the constraints and the estimation mechanism defined in Sec. 4.3. The resulting data better mimic real-life scenarios and real tracker outputs as compared to the data used in other approaches [48]. The synthetic data is equally divided into training and test sets. The results on these datasets are discussed next.

6.5.3 Interaction among dynamic and static objects

In this section we will discuss various results obtained by applying the proposed dynamic-static interaction recognition approach. First we will discuss results of object-centric model and apply it for abandoned baggage detection. Then we will employ the scene-centric model in several real surveillance scenarios including the building entrance and the airport for recognising various other interactions between a dynamic object and various static objects.

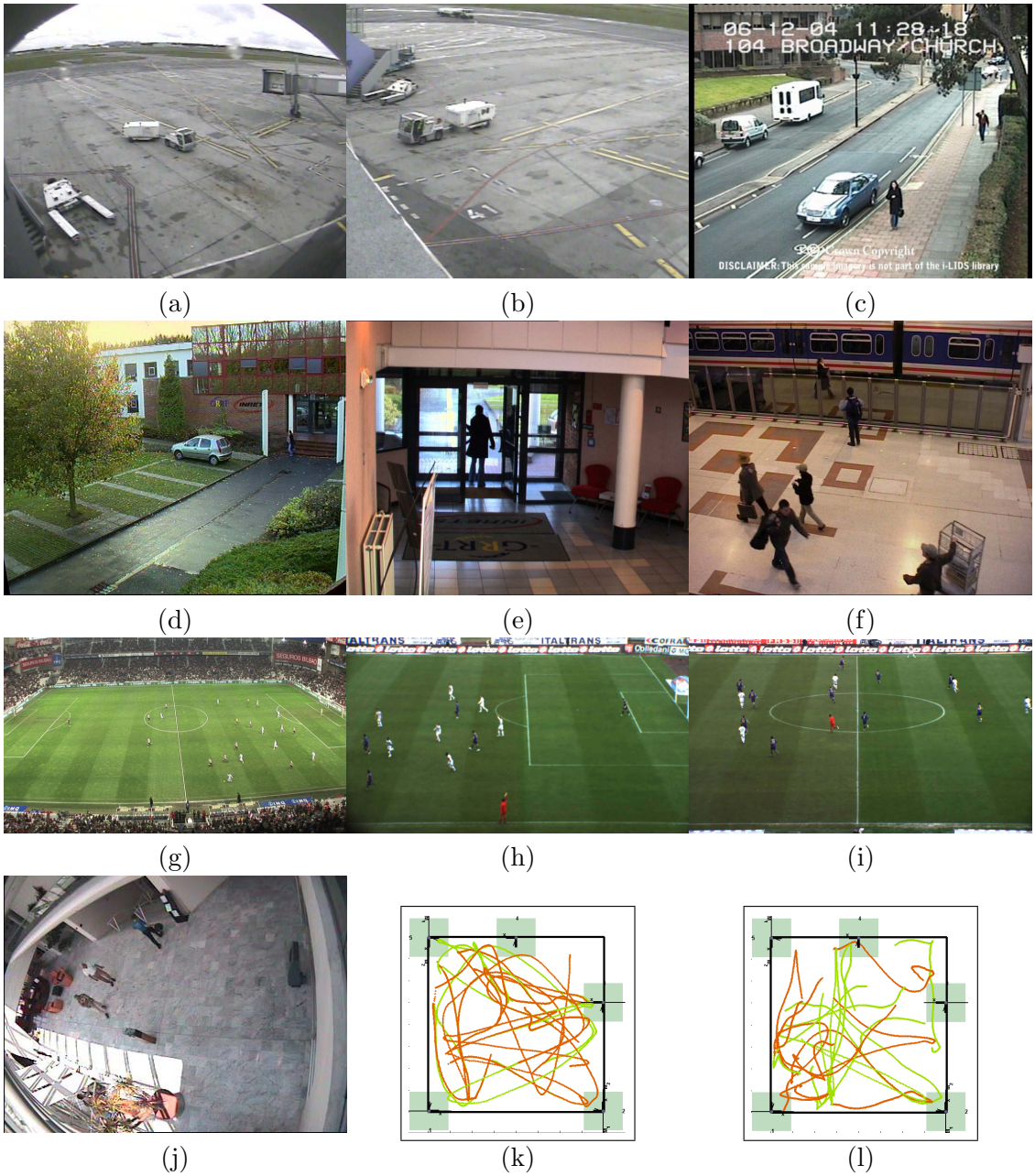


Figure 6.10: The evaluation datasets for interaction recognition. (a-b) ETISEO airport scenario (AP-11 C4 and C7) (c) ETISEO/iLids road scenarios (RD-6-C7) (d-e) ETISEO building entrance scenario (BE-19 C1 and C4) (f) PETS 2006 metro station (C3 of sequence S1,S2, S3, S5 and S6) (g) MediaPro soccer match (S8) (h-i) IISA soccer match (C1 and C3 of sequence S7) (j) CAVIAR scenarios (CL1 and CL2) (k-l) Sample synthetic trajectories for 2 targets in a network of 5 STAC sensors.

Object-centric model

The *object-centric* HMM model is used for detecting abandoned baggage scenarios where the states are *attended*, *unattended* and *abandoned*. Figure 6.11 shows the HMM

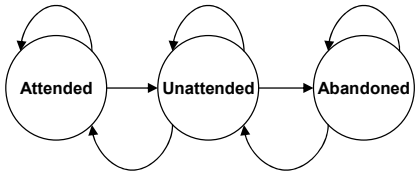


Figure 6.11: HMM model for baggage detection on the PETS and CAVIAR datasets. Each state represents an event. The initial state is selected as the state with the maximum emission probability b_{jk} at time k.

model used to detect these states. The object is considered *attended* when the owner is within a 2 metres distance of the baggage. The warning of baggage being *unattended* is raised when the object moves further than 3 metres from the baggage. The baggage is considered *abandoned* when it is unattended for more then 30sec. These event definitions were provided with the PETS dataset and were used to detect events on real surveillance scenarios from both the PETS and the CAVIAR datasets.

Here each object at each time k is represented by a 4D state-space consisting of position (x, y) and size (w, h). For the PETS sequences, the baggages are detected based on their size and aspect ratio (ranging between 1 and 1.8, set empirically on the test dataset). The μ and σ values for each event were set using the event definitions provided for the PETS dataset. For the *attended baggage (s_1)* event, $\sigma_x = \sqrt{2 \times g_x}$ and $\sigma_y = \sqrt{2 \times g_y}$ respectively, whereas for the *unattended baggage (s_2)* and the *abandoned baggage (s_3)* events the values are $\sigma_x = \sqrt{g_x/2}$ and $\sigma_y = \sqrt{g_y/2}$ (see Eq. 6.7). The values $g_x = 36$ and $g_y = 96$ are the distances along x and y coordinates on the ground planes. These values are based on the calculation that, for this scenario, 1 metre in world-coordinates corresponds to 36 pixels along the x-axis and to 96 pixels along the y-axis on the ground plane. A baggage is considered *unattended* when its related object (the *owner*) is 2 metres away. A baggage is considered *abandoned* when its related object is 3 metres away, for at least 30 seconds. For the CAVIAR sequences, the baggages are detected in a similar fashion and the parameters of the events are defined as follows. For the *attended baggage (s_1)* event, $\sigma_x = \sqrt{2 \times g_x}$ and $\sigma_y = \sqrt{2 \times g_x}$ respectively, whereas for *unattended baggage (s_2)* the values are $\sigma_x = \sqrt{36}$ and $\sigma_y = \sqrt{48}$ and for *abandoned baggage (s_3)* the values are $\sigma_x = \sqrt{24}$ and $\sigma_y = \sqrt{24}$. The probabilities of possible transitions a_{ij} (Fig. 6.11) between these states are set to be equally likely.

Figure 6.12 shows sample event detection results on the sequences S1 and S5 of the PETS 2006 dataset. The images show the detection of the object around which the HMM model is built (the bag) and the subsequent sequence of events, namely a *warning*

Figure 6.12: Sample event detection results for the PETS 2006 dataset. (Row 1): Sequence S1, frames 1955, 2004, 2754 and 2790. (Row 2): Sequence S5, frames 2020, 2083, 2833 and 2890. The evaluation of the event detection accuracy is discussed in the text.

(unattended baggage) and an *alarm* (abandoned baggage). The computed event-level and sequence-level accuracy values are shown in Table 6.4. The sequence-level accuracy is above 90.00% for all four videos. The event initialisation and termination accuracy (Eq.(6.19) is also above 90.00% for all except PETS-S5 scenarios. The overall event-initialisation accuracy for PETS-S5 is 86.00% whereas the overall event termination is 91.60%. The lower event-initialisation accuracy in this case is due to occlusion at frame 2222 which delayed the recognition of unatttended baggage event. The reason for having accuracy of around 90% and not above is, the detection of baggage is associated with a drop-off (object split) event which is delayed by some frames due to morphology. Both the precision and sensitivity scores for the PETS dataset are unity as the object-centric approach selects events associated with detected objects only.

Figure 6.16 shows sample event detection results on the sequences CL1 and CL2 of the CAVIAR dataset. Figure 6.16(Row 1) shows the detection of the *abandoned* and *attended baggage* events, which are generated as the person first abandoned the baggage and then reappears and approaches the baggage. Figure 6.16(Row 2) shows that the person has left the baggage at the end of the stairs moving toward the kiosk machine and hence the *attended* and then *unattended baggage* events are generated. The computed event-level and sequence-level accuracy values are also shown in Table 6.4. In accordance with the ground truth available for the CAVIAR dataset, we compute the accuracy of the detection of the activities related to the baggage rather than the accuracy for the alarm and the warning events. For the sequence CL1, the event initialisation accuracy is 94.33% and the event termination accuracy is 86.31%. The event initialisation accuracy for the sequence

CL2 is 74.51% and the termination accuracy is 72.55%. The reason for lower event-level accuracy values is the merging of the blob of the baggage with that of the person when the bag is placed on the floor. This results in a delayed detection of the event by 28 frames and 23 frames for CL1 and CL2 respectively. Similarly, when the baggage is picked up, the two objects are merged thus resulting in an anticipation of the event. However, it should be noted that the frame-level accuracy is 98.01% in case of CL2. This indicates that although the algorithm performed delayed initialisation and termination of the event, it does not have false detections in the rest of the sequence. Improvements in the object detection accuracy will help in further enhancing the event detection accuracy. Similarly to the PETS dataset, the precision and sensitivity scores for the CAVIAR dataset are unity as all events are detected.

Scene-centric model

The ETISEO dataset contains several interactions between dynamic and contextual objects and is used here to demonstrate the performance of the *scene-centric* approach. The HMM model used for activity monitoring for the ETISEO dataset is shown in Fig. 6.14. In this case we model ten events, namely *enter zone*, *inside zone*, *exit zone*, *change zone*, *opens*, *closes*, *go up stairs*, *go down stairs*, *empty area*, and *stopped object*. The definition of the areas of interest is part of the contextual information provided with the dataset. The values for all the parameters for these events were defined using the ground truth of the training data. The value for duration τ for each event is computed by taking the mean duration for all the instances of a particular event in the ground truth. The contextual objects such as open space for the outside zone state are identified manually whereas the location and size information of remaining contextual objects (zone, door, door control, stairs) were taken from the ground truth and were used to set the parameters of Gaussians in the case of scene-centric model.

The evaluation scores (Table 6.5) are computed during the 2^{nd} ETISEO evaluation cycle by the evaluator on the entire length of ground truth. Despite these challenging scenarios the precision in all except BE-19-C1 is very high. The lower precision in case of BE-19-C1 is due to the glass doors which have a reflection of a car on it as the car comes closer to the building. This generates FP event of opens. The sensitivity is relatively low in most of the scenarios of ETISEO dataset because of large number of false negatives. Some of these false negatives are due to events not modelled in this work and others due to missed events associated with the building entrance. The ETISEO dataset contains several challenging situations such as the entrance to the building which has a

zone between two double glass doors with a door control in between. This kind of situation (BE-19-C4) is difficult to handle using change detection information (Fig 6.13). This results in large number of false negatives and hence lower sensitivity. The sequence-level accuracy is also low again due to missed detections. However, the event-level accuracy in activity initialisation and termination is high in 25 out of 30 cases (Table 6.4 and Table 6.5). The lower accuracy is in the case of very short events such as *enters zone*, *opens* and *closes*. These events are of very short interval and for e.g. in case of AP-11-C7 the accuracy of initialisation of *enters zone* event is 43.33% where the delay in recognition by the system is just half a second. Similarly in case of *enters zone* termination the delay in recognition by the system is 1 sec however the accuracy has gone down to just 16.67%. This is because the *enters zone* event has a shortest mean duration of 30 (1.2 sec) frames only. Similarly, *opens* and *closes* events has also very short mean duration of 50 (2 sec) and 86 (3.44 sec) frames respectively. This is the major cause of lower accuracy as well as sensitivity because a slight delay can result in an event being classified as a misdetection if it does not overlap with the ground truth. For the remaining events such as *inside zone* and *stopped* the event-level accuracy is above 98.00% in AP-11-C7 and BE-11-C4 and is almost 90.00% in the case of the AP-11-C4 scenario.

Table 6.3 shows the minimum and maximum score obtained on ETISEO dataset during the 2nd evaluation campaign in which there were 16 participants. It also shows the mean and the variance in score among participants. The last column of Table 6.3 indicates the scores for the proposed approach. The scores mentioned in Table 6.3 are the average over various precision and sensitivity values. The details about the 19 different evaluation matrix used to compute these scores can be found in [140]. The table indicates that the proposed approach has obtained either maximum or close to maximum score for task of detection, localisation and tracking. In the event recognition task the scores for the AP-11 scenario are 78.00% and 71.00% with a mean score of 54.00% and 56.00% and maximum score obtained is 87.00% and 90.00% among the participants. This indicates that the proposed approach is comparable with other approaches. These scores are also due to FN in the proposed approach as we do not model events such as the opening of a door of a car. The significant difference here is in the case of the RD-6 scenario where the sequence-level accuracy (Table 6.5 (last row)) is 93.17% however the score in Table 6.3 is just 38.00%. This is due to a low sensitivity of 0.25 as compared to high precision of 1.00.

Figure 6.15 shows detection results on the ETISEO dataset for the *enter zone, inside zone, stopped* and *empty area* events. To demonstrate the flexibility of the proposed

Table 6.3: ETISEO evaluation scores for event recognition and its comparison with the proposed approach

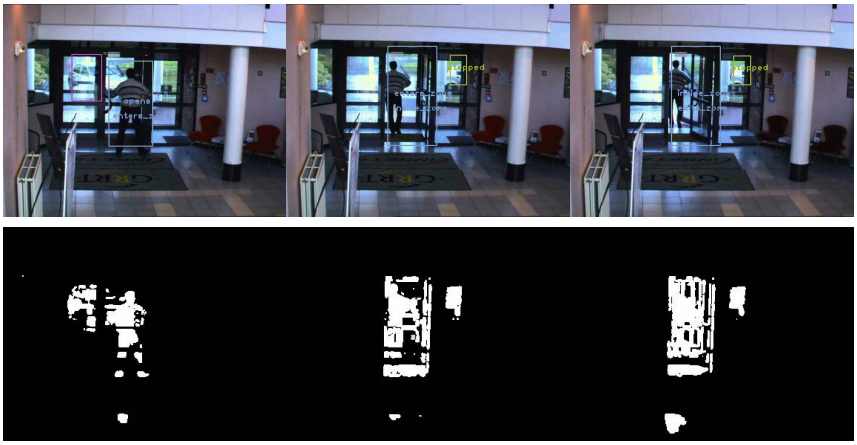
	Mean	Var	Min	Max	Proposed
AP-11-C4	54.00	28.00	6.00	87.00	77.78
AP-11-C7	56.00	27.00	4.00	90.00	71.25
BE-19-C1	32.00	20.00	5.00	68.00	33.00
BE-19-C4	33.00	11.00	14.00	44.00	29.05
RD-6-C7	40.00	22.00	11.00	69.00	38.00

Table 6.4: Event detection for 6 test sequences of the PETS and CAVIAR dataset. (Key: GT = Ground Truth; AD = Automatic Detection)

	Start frame			**End frame**		
	GT	**AD**	**Accuracy (%)**	**GT**	**AD**	**Accuracy (%)**
PETS-S1 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 94.14%)						
Warning	2754	2825	90.53	2789	2842	92.93
Alarm	2790	2843	92.93	3021	3021	100.00
All			**91.73**			**96.47**
PETS-S3 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 100.00%)						
Warning	0	0	100.00	0	0	100.00
Alarm	0	0	100.00	0	0	100.00
All			**100.00**			**100.00**
PETS-S5 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 90.12%)						
Warning	2833	2749	88.80	2889	2763	83.20
Alarm	2890	2764	83.20	3401	3401	100.00
All			**86.00**			**91.60**
PETS-S6 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 97.18%)						
Warning	2414	2403	98.53	2455	2421	95.47
Alarm	2456	2422	95.47	2801	2801	100.00
All			**97.00**			**97.73**
CAVIAR-CL1 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 96.60%)						
unattended	979	975	92.16	1002	989	74.51
abandoned	1003	990	96.51	1291	1284	98.12
unattended	1292	1285	86.27	1315	1310	90.20
All			**94.33**			**86.31**
CAVIAR-CL2 (Precision: 1.00, Sensitivity: 1.00, Seq. accuracy: 98.01%)						
unattended	713	700	74.51	930	916	72.55
All			**74.51**			**72.55**

Table 6.5: Event detection precision and sensitivity for 5 test sequences of the ETISEO dataset. (Key: GT = Ground Truth; AD = Automatic Detection; Acc. = Accuracy)

	Start frame			**End frame**		
	GT	**AD**	**Acc. (%)**	**GT**	**AD**	**Acc. (%)**
AP-11-C4 (Precision: 1.00, Sensitivity: 0.56, Seq. accuracy: 86.34%)						
empty area	1	12	98.63	689	664	96.88
enters zone	675	664	63.33	720	728	73.33
inside zone	690	731	89.70	804	803	99.75
stopped	1	2	99.80	804	803	99.80
stopped	1	3	99.60	804	803	99.80
All			**90.21184**			**93.91173**
AP-11-C7 (Precision: 1.00, Sensitivity: 0.50, Seq. accuracy: 64.60%)						
empty area	1	187	76.75	689	653	95.50
enters zone	675	658	43.33	720	695	16.67
inside zone	690	696	98.49	804	803	99.75
stopped	1	2	99.80	804	803	99.80
All			**79.59**			**77.93**
BE-19-C1 (Precision: 0.65, Sensitivity: 0.65, Seq. accuracy: 20.98%)						
closes	335	371	58.14	453	450	96.51
opens	258	250	84.00	320	300	60.00
opens	366	395	42.00	400	407	86.00
stopped	270	283	97.41	1025	1024	99.80
All			**70.39**			**85.58**
BE-19-C4 (Precision: 0.87, Sensitivity: 0.35, Seq. accuracy: 21.79%)						
inside zone	185	180	98.74	245	338	76.63
opens	77	101	52.00	150	180	40.00
opens	737	717	60.00	780	776	92.00
stopped	170	206	92.83	950	1048	80.48
All			**75.89**			**72.28**
RD-06-C7 (Precision: 1.00, Sensitivity: 0.25, Seq. accuracy: 93.17%)						
stopped	570	559	97.81	710	743	93.43
All			**97.81**			**93.43**



(a) (b) (c)

Figure 6.13: Detection, dynamic-static interaction recognition and tracking results for the ETISEO BE-19-C4 scenario. (a) Frame 75, *opens* event; (b) frame 160, *inside zone* and *enter zone* events; (c) frame 190, *exit zone*, *inside zone* events.

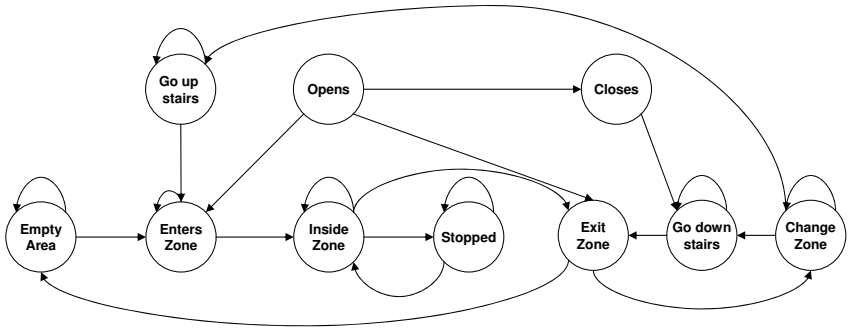


Figure 6.14: Scene-centric HMM model for activity monitoring on the ETISEO dataset. Each state represents an event. The initial state is selected as the state with the maximum emission probability b_{jk} at the time of object birth.

framework, in this case event detection is performed on the image plane. The green rectangle drawn on the tarmac (Fig. 6.15) is the zone considered for triggering the events *enter zone, inside zone* and *empty area*. The *stopped* event is detected anywhere in the scene. Table 6.5 shows the accuracy for the detected events in all ETISEO sequences. Finally, a high-level summarisation of the sequences is shown in Fig. 6.17 that visualises the metadata generated with the proposed event analysis framework. The tracks of the objects and the corresponding labels of the various events occurring at different time instants are visualised for a sequence. The blue spheres mark the start and end of the events. These events are shown on the track of the object they are associated, with the exception of the *empty area* event that is not associated with any object and the bar attached to the start and end of this event shows its time span.

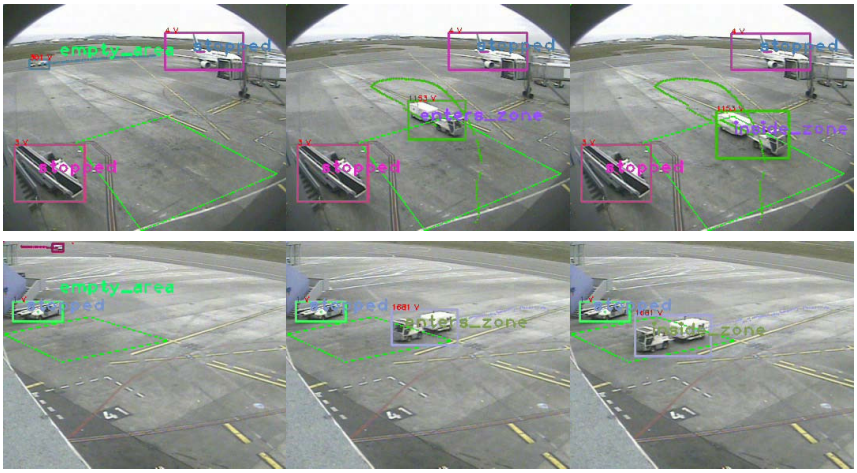


Figure 6.15: Sample tracking and event detection results for the ETISEO dataset using the scene-centric event modelling (from top to bottom: frame 23, 690 and 750). The detected events are *stopped*, *empty area*, *enter zone* and *inside zone*. (Row 1): ETISEO AP-11-C4. (Row 2): ETISEO AP-11-C7.

Computational cost

The computational cost of interaction recognition using the proposed object-centric and scene-centric models utilising Hidden Semi-Markov Model for state estimation is shown in Fig 6.18, together with the cost of image-based localisation and graph-based tracking. The figure shows the cost of a C/C++ implementation for all the modules. The computational cost is computed per frame in milliseconds using a colour input video of resolution 960×544 on a Intel Core 2 Quad CPU having speed of 2.39 GHz and 3.25GB RAM. It can be seen that interaction recognition between dynamic and static objects takes only 1.05 milliseconds per frame and the major computational burden is due to image-based localisation which takes 515.74 milliseconds (96.37% of the processing time). This indicates that given tracking information the events can be detected at approximately 950 frames per second however the overall algorithm works at 2 frames per second.

6.5.4 Interaction among dynamic objects

The proposed dynamic object interaction recognition algorithm is evaluated both on real as well as on synthetic data and is compared with ground truth as well as with four other algorithms: a Baseline, a DBN, an HMM and a CHMM Maximum Likelihood (CHMM-ML). These methods work on the same feature space used by CHMM-MAP. In DBN the states are defined as a set of discrete and continuous random variables and the transition and observation models are defined as a product of the conditional probability

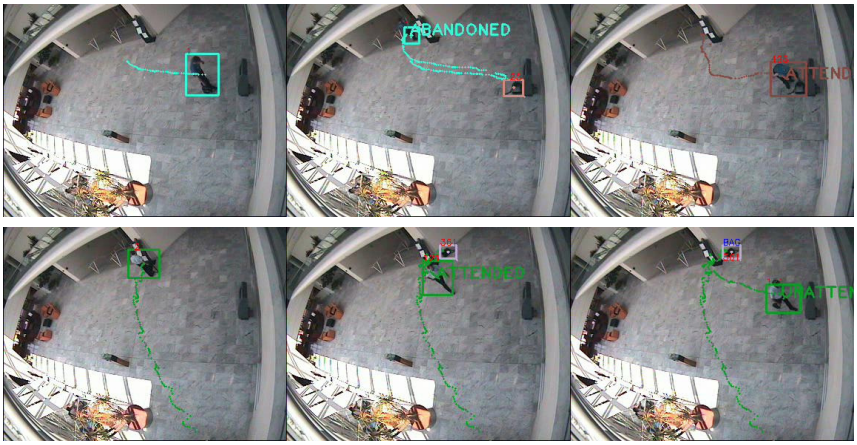
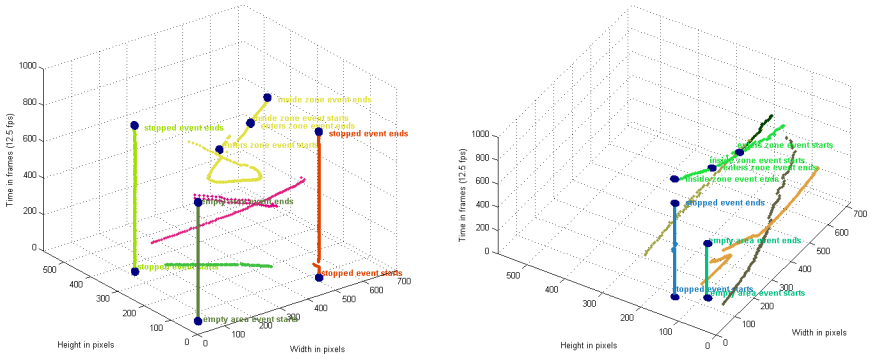


Figure 6.16: Example of left baggage detection on the CAVIAR dataset using the object-centric event modelling. (Row 1) *Abandoned* and *attended* baggage event in sequence CL1 (frame 914, 1014, 1070 and 1334); (Row 2) *attended* and *unattended* baggage event in sequence CL2 (frame 441, 548, 670 and 721).



(a) (b)

Figure 6.17: Example of high-level summarisation using the metadata generated with the proposed event analysis for the sequence ETI-VS2-AP-11. (a) Visualisation of the object tracks and their associated detected events for C4; (b) Visualisation of the object tracks and their associated detected events for C7.

distributions (CPDs) [196]. The Baseline method is a rule-based approach in which the interactions are subdivided into 5 sub-events: *random walking* (E_w)(*going separately*), *approaching* (E_p), *staying* (E_s), *going together* (E_g), and *following* (E_f). We define four thresholds ς_ν, ς_ϕ, ς_d and $\varsigma_{\dot{d}}$ for speed, relative direction, relative distance and its derivative. The *staying* sub-event (E_s) is detected if the magnitude of the speed of the objects is almost zero, i.e. if $\nu^i < \varsigma_{\nu^i},\ \forall\ i \in \{p, q\}$, then it is detected as E_s. The *going together* sub-event (E_g) requires the targets to have similar speed, going in the same direction and spatially

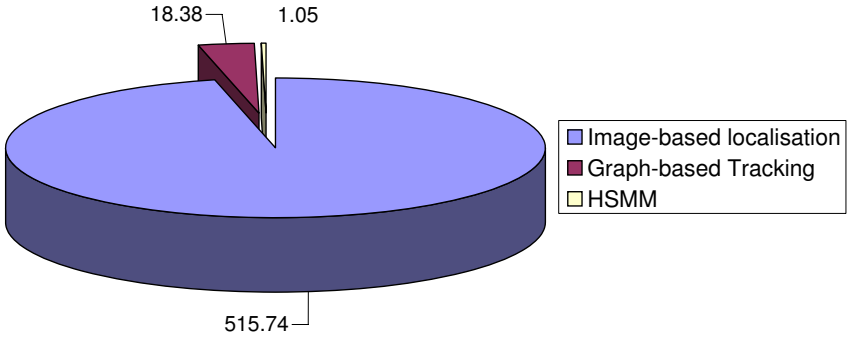


Figure 6.18: Average per frame computation cost in milliseconds for image-based localisation, graph-based tracking and event detection using Hidden Semi-Markov Model (HSMM) on colour video of resolution 960×544.

Table 6.6: Accuracy comparison on synthetic test data for interaction detection using a Baseline method, a DBN, an HMM, a CHMM-ML and a CHMM-MAP

		E1	E2	E3	E4	E5
Baseline	μ	0.5461	0.7218	0.6066	0.5952	0.5156
	σ	0.0986	0.1078	0.0899	0.0924	0.1312
DBN	μ	0.8069	0.7084	0.7706	0.4702	0.4403
	σ	0.1300	0.1416	0.1045	0.1248	0.1481
HMM	μ	0.8585	0.7791	0.6967	0.7987	0.7555
	σ	0.0717	0.0743	0.2210	0.0546	0.0808
CHMM-ML	μ	0.8563	0.7881	0.5419	0.7976	0.7527
	σ	0.0798	0.0717	0.1861	0.0574	0.0851
CHMM-MAP	μ	0.8665	0.8688	0.7650	0.8376	0.8049
	σ	0.0682	0.0575	0.1175	0.0557	0.0804

close during the event span. Therefore, it depends upon four features (i.e., relative speed, relative direction, relative distance and derivative of relative distance) and is detected if $|\nu^p - \nu^q| < \varsigma_s$ & $\phi^{pq} < \varsigma_\phi$ & $d^{pq} < \varsigma_d$ & $\dot{d} < \varsigma_{\dot{d}}$. The *following* sub-event (E_f) also depends upon these four features, but it is differentiated from the sub-event E_g based on the relative distance and its derivative as the target following another target must be at a certain distance. E_f is detected if $|\nu^p - \nu^q| < \varsigma_s$ & $\phi^{pq} < \varsigma_\phi$ while the remaining features are greater than their respective thresholds. The *approaching* sub-event (E_p) implies that the relative distance between interacting targets will be decreasing, i.e. the derivative of the relative distance will have negative value, and it is detected when $\dot{d} < 0$. If the target's activity is not classified as any of the sub-events then it is considered as random walk (E_w).

Data and performance measures

The HMMs and DBN were trained for five interactions and then tested on real data without retraining. The sport scenarios (Soccer-S7 and Soccer-S8) give a wide view of a soccer match where targets are in the far-field of the camera and can be considered as point targets as in training data. This is not the case in surveillance data where targets are in the near-field of the camera. To use the HMMs and DBN trained on synthetic data on the real trajectories we normalised them by projecting in the environment where synthetic data is generated (i.e., a 4×4 units area where 1 unit can be considered as 2 metres). These real datasets inherently contain the interactions that are modelled by our proposed approach. In real datasets (Soccer-S7, Soccer-S8 and PETS-S2) targets are likely to merge while staying together during an interaction. The generated trajectories were post-processed to make them smooth and to resolve the identity-switches [10]. The merging of the targets can be solved by using a bipartite graph [223] in order to have complete trajectories for event analysis. The soccer data (Soccer-S7 and Soccer-S8) contained a large number of objects and although in such cases the complexity of CHMM is combinatorial, we reduced the computational cost through gating. Gating eliminates objects that do not lie within 3σ of the Gaussian windows centred around the mid-point of the base of the bounding box of each object. The improvement in performance depends upon the covariance of the Gaussian window, which depends upon the average size and speed of the targets and is set empirically to 100 and 125 pixels along the horizontal and vertical directions. An audio signal was added to only those trajectories pairs from real data that contained desired interactions (identified empirically), followed by the re-estimation of trajectories using audiovisual data.

Analysis of the results

The evaluation of the IER via state estimation using Viterbi decoding is shown in Table 6.6. Table 6.6 shows the comparison of the proposed approach with the four alternative methods. CHMM-MAP trained on the selected feature set extracted from synthetic trajectories achieves the highest mean μ accuracy. DBN has a very low accuracy for interaction E4 and E5 as these interactions have an initial interval of random walk at the beginning and again at the end of the interaction. On the other hand, DBN has the highest accuracy for interactions E3 as it has a least amount of random walk. The standard deviation σ of CHMM-MAP is the lowest among all approaches except for interaction E3 and E4. For interaction E3, the baseline method has a lower standard deviation, whereas

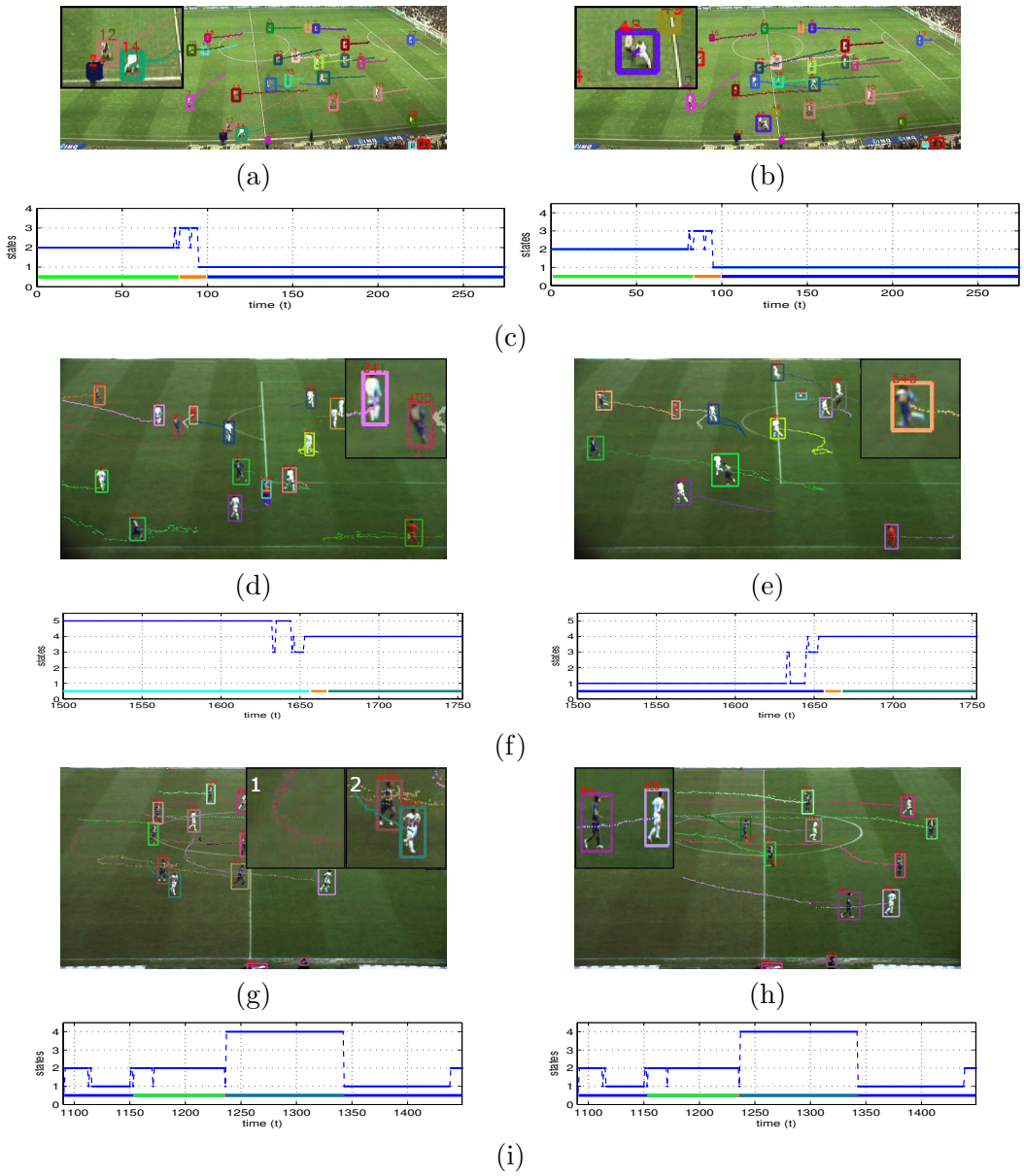


Figure 6.19: Sample detection, tracking and interaction event recognition results on Soccer match sequences S7 and S8. (a-b) Frame 50 and 85 of soccer match with approach-meet-go separately interaction (E2) among targets shown on the magnified area. (c) CHMM-MAP generated sequence of interactions and the ground truth. (d-e) Frames 1640 and 1720 of a soccer match show a follow-reach-go together interaction (E1). (f) CHMM-MAP generated sequence of interactions with the ground truth. (g-h) Frames 1228 and 1400 showing change direction-approach-meet-go together interaction (E5) and also showing a go-separately state after an interval of go together (i.e., an interaction similar to E4). (i) CHMM-MAP generated sequence of interactions and the ground truth (light green: approaching; brown: meeting and waiting to meet; blue: walking/going separately cyan: follow; dark green: going together; dark blue: going separately).

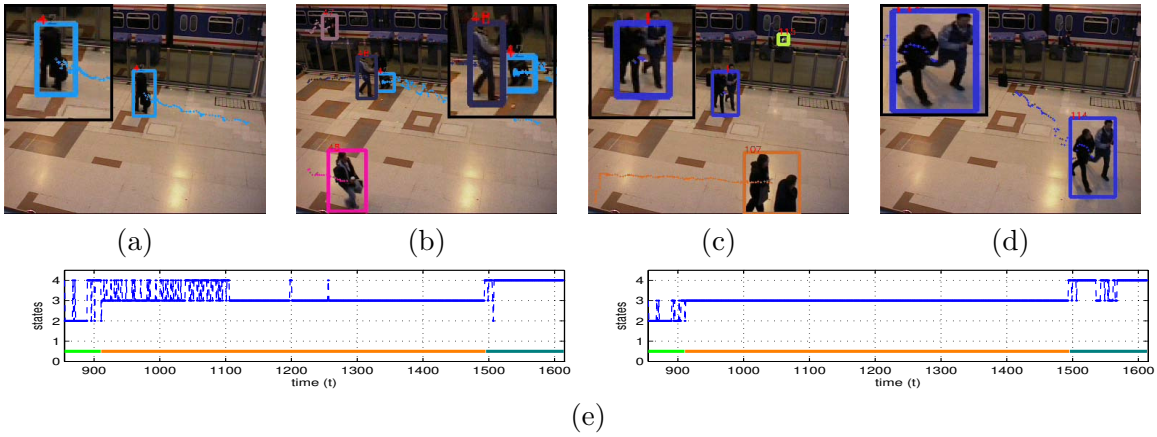


Figure 6.20: Sample detection, tracking and interaction event recognition results on a real surveillance scenario. (a-d) Frames 805, 902, 1535 and 1600 of the sequence S2-T3-C3 from the PETS2006 dataset, showing an approach-meet-go together interaction (E3). (j) CHMM-MAP generated sequence of interactions and the ground truth (light green: approaching; brown: meeting and waiting to meet; dark green: going together).

for E4 the HMM has a lower σ. For interactions E1, E2, E4 and E5 both CHMMs have the highest accuracy.

Figure 6.19 (a-b) shows the *approach-meet-go separately* interaction (E2) where two players from opposite teams approach the ball almost simultaneously, stay together while tackling the ball, and then the ball is kicked away and the two players go separately. The detection and tracking results are shown in Fig. 6.19(a-b) where the 22 players are detected and tracked. The two players interacting with each other while trying to get the ball are shown in a magnified section on the top-left corner of the image. The ground truth of the resulting state sequence is shown in Fig. 6.19(c). The horizontal colour bar indicates state 2 (*approaching*) in light green, state 3 (*meeting*) in brown and state 1 (*going separately*) in blue. The state sequence generated by CHMM-MAP is shown with a dotted and dashed lines which coincides with the ground truth by 97.45% (Table 6.7). The slight flickering during state 3 are due to the fact that targets never stopped completely and the state of *staying* is not a constant state. This can be further analysed by investigating the features generated by these two trajectories. The features are shown in Fig. 6.21(b), where it can be seen that during the interval of state 3, the magnitude of the speed of the two targets is around 0.2 and not zero and that the features are more noisy than those of the synthetic data (Fig. 6.8). However, the correct state was estimated due to them having the same relative direction and almost zero relative distance supported by low speed magnitudes, as can be seen in Fig. 6.21(b). Table 6.7 also shows the event-

Table 6.7: Event detection for 4 test sequences of the football and PETS datasets. (Key: GT = Ground Truth; AD = Automatic Detection)

		Start frame			**End frame**		
		GT	**AD**	**Accuracy (%)**	**GT**	**AD**	**Accuracy (%)**
Soccer-S8 (Seq. Accuracy: 97.45%)							
E2	approach	1	1	100.00	80	83	95.89
	meet	84	84	100.00	89	99	93.46
	go separately	95	100	94.44	274	274	100.00
	All			**98.15**			**96.45**
Soccer-S7-C2 (Seq. Accuracy: 97.95%)							
E1	follow	1500	1500	100.00	1632	1656	84.62
	meet	1647	1657	93.46	1652	1667	90.20
	go together	1653	1668	84.38	1753	1753	100.00
	All			**92.61**			**91.60**
Soccer-S7-C3 (Seq. Accuracy: 97.17%)							
	go separately	1116	1091	72.22	1150	1152	97.78
E3	approach	1172	1153	73.97	1235	1234	98.63
	meet	1236	1235	99.35	1236	1236	100.00
	go together	1237	1237	100.00	1342	1343	98.96
	go separately	1343	1344	98.89	1438	1449	87.78
	All			**99.41**			**95.58**
PETS-S2 (Seq. Accuracy: 97.71%)							
E3	approach	872	855	76.71	890	910	72.60
	meet	912	911	99.35	1493	1495	98.69
	go separately	1508	1496	86.67	1517	1517	100.00
	All			**87.58**			**90.43**

level accuracy for each state of the interaction event. In case of interaction events among dynamic objects the longest duration without transition for each state is considered for computation of event-level accuracy. The event initialisation accuracy for interaction E2 for Soccer-S8 is 98.15% and termination accuracy is 96.45% on average with accuracy being 100% in initialisation of *approach* and *meet* states and termination of *go separately* state. The *follow-reach-go together* interaction E1 is shown in Fig. 6.19(d-e). The player from team 1 (navy blue (dark)) follows the player of team 2 (white) who is following the ball. The player of team 1 increases speed and comes closer to the player of team 2 and then they go together toward the ball. This interaction sequence is detected with 97.95% similarity with ground truth and is shown in Fig. 6.19(f) and in Table 6.7 (Soccer-S7-C2). The *meeting* state is detected earlier due to blob merging and is a very short state of just 10 frames as the targets hardly stop and continue chasing the ball. The event-level accuracy for various states in this interaction is on average 92.61% and 91.60% for initialisation and termination respectively (Table 6.7).

The combination of *change direction-approach-meet-go together* (E5) and *change direction-approach-meet-go separately* (E4) interactions is shown in Fig. 6.19(g-h). The detected interactions with ground truth are shown in Fig. 6.19(i) and have sequence level accuracy of 97.17%. It shows that the two players tried to approach the ball but then it was kicked again in a different direction. The two players then changed direction (shown in magnified section on the top right corner of Fig. 6.19(g)) and started moving towards the new location of the ball; however it was again kicked far away so the two targets approached a point and started moving together for sometime after which one of them slowed down while the other continued to move forwards. Note that the scenario in Fig. 6.19(h) is not detected as a *following* interaction because the lagging target is slowing down, whereas the leading target is maintaining the same speed. This is opposite to the definition of the *following* state. Furthermore, the *random walk/going separately* state in the beginning and the end of the generated sequence is possibly misclassified as an *approaching* state due to the fact that in these sport sequences almost all movements are goal directed. This also results in lower event initialisation and termination accuracies of 72.22% and 87.78% respectively for *go separately* state (Table 6.7). Moreover, this kind of interesting sequence is generated due to the use of the Viterbi decoding strategy instead of the evaluation strategy.

In PETS-S2, two persons meet and then leave the scene together (Fig. 6.20(a-d)) resulting in detection of interaction sequence E2. The example frame of the sequence with ground truth is shown in Fig. 6.20. The sequence level accuracy for this recognition is 97.71% (Table 6.7). The event-level accuracies of *approach* state are 76.71% and 72.60%. This is due to late initialisation and early termination of the state caused by flickering due to camera perspective. It can be seen that the relative distance (Fig. 6.21(d)) between the two targets remain relatively high compared to the synthetic data and soccer scenarios as targets are in the close field of the camera. In such cases, even when the targets are close to each other, there exists a distance between the mid-point of the bases of their bounding boxes, which is much larger than in the case of point targets. This problem could be solved by defining a normalising function based on the targets 4D state space (position and size), as opposed to the target positions only. The event-level initialziation and termination accuracy for *meet* is 99.35% and 98.69% whereas for *go together* states is 86.67% and 100%. The slightly lower accuracy for intialisation of *go together* state is, as the two targets starts to move on the left in Fig. 6.20(c), they first come close to the other target and then they move together towards the right in the scene. This initial movement does not produce a significant increase in the target's speed and hence delayed

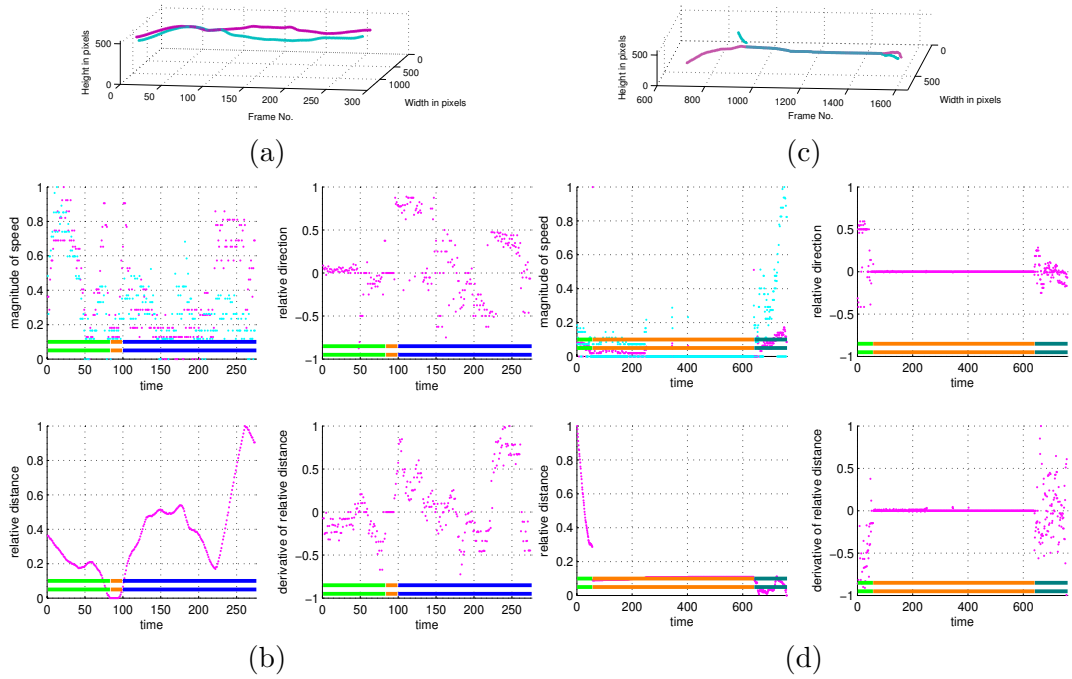


Figure 6.21: Sample interacting trajectories and their features for real data. (a) Trajectories of two players from the soccer data, showing the interaction E2. (b) Normalised features for the trajectories of interacting soccer players (note: time 0 represents frame 855 of the sequence) (c) Trajectories of two persons in the scene S2-T3-C3 of the PETS2006 data, showing interaction E3. (d) Normalised features for the trajectories of two persons.

the initialisation of the state. The accuracy for the initialisation and termination of *meet* states is 99.35%, 98.69% respectively and the accuracy for the termination of *go together* state is 100% which indicates good performance of the proposed approach.

The computational cost of interaction recognition between dynamic objects using the Coupled Hidden Markov Model is shown in Fig 6.22 together with cost of image-based localisation and graph-based tracking. The figure shows the cost of a C/C++ implementation for detection and tracking and Matlab implementation for Coupled Hidden Markov Model [166, 218]. The computational cost is derived per frame in milliseconds using a colour input video of resolution 960×544 on a Intel Core 2 Quad CPU having speed of 2.39 GHz and 3.25GB RAM. It can be seen that interaction recognition between dynamic objects takes only 13.09 milliseconds for 11 interactions per frame and the major computational burden is due to image-based localisation, which takes 515.74 milliseconds (94.25% of the processing time). This indicates that, given tracking information, the events can be detected faster than real-time.

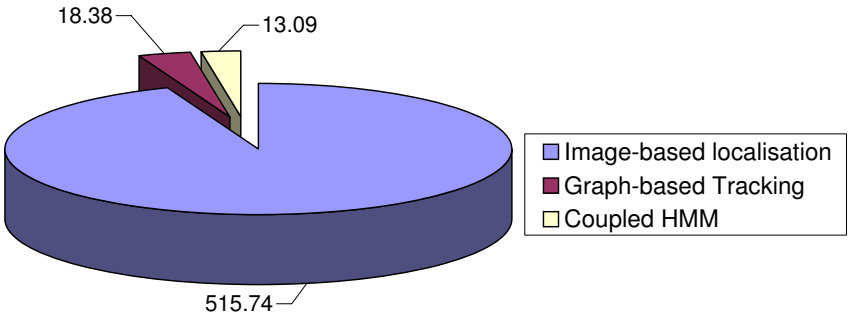


Figure 6.22: Average per frame computation cost in milliseconds for image-based localisation, graph-based tracking and event detection using Coupled Hidden Markov Model (CHMM) on colour video of resolution 960×544 for up to 11 interactions in per frame.

6.5.5 Future experiments

The interaction recognition between multiple dynamic objects is performed using five relative features between the objects. The question to address is now *which features and what combination of features are most appropriate in recognising the interactions?*. To answer this question experiments can be performed using subsets of the features used in this work in different combination. The experiments should also be performed on crowded scenarios where there may be a large number of possible interactions. However only a few of them are related to real world interactions and the rest is due to the presence of a crowd. The training and testing of the algorithm on such data can also help in better understanding the contribution from each feature and can help in improving the results.

This work focused on interaction detection between a dynamic or static object with another dynamic or static object (i.e., only 2 objects are considered at a time). The question is *how can it be extended to more than two objects?*. Experiments with a dataset where there are interactions between more than 2 objects could be performed. To enable recognition of such interaction, a new set of interactions should be defined other than interactions $E1$ to $E5$. The same features defined in this thesis can be used to enable recognition of interaction involving more than 2 objects. Experiments should be performed with 3 or more chain CHMMs.

6.6 Summary

In this chapter we demonstrated that several important activities in sports and surveillance videos can be modelled as interactions. Interactions can either be between dynamic objects and static objects (dynamic-static), or between two dynamic objects.

Modelling event detection as interaction recognition simplifies the problem to that of state sequence estimation based on detection and tracking information.

Such state estimation can be modelled using time-series analysis and graphical methods, in particular Hidden Markov Models and their variants, are well suited for this analysis with spatio-temporal independence. In case of dynamic-static interaction the state sequences of only one object need to be estimated. We proposed *object-centric* and *scene-centric* models to detect such interactions. In the *object-centric* model, the state of the temporarily static object is estimated with respect to the position of the dynamic objects. In the *scene-centric* model, the collective state of the dynamic objects is estimated with respect to static objects. For improving the event detection accuracy, explicit duration modelling is performed using Hidden-Semi Markov Models to avoid prolonged self-transitions. The detection of interactions among multiple dynamic objects requires the state estimation for each object. To this end, the full coupling between multiple object states is modelled using Coupled HMMs on relative features among the objects under analysis. Unlike activity recognition using only visual cues, the interaction recognition using audiovisual tracking information allows activity monitoring in camera-views as well as in regions outside the cameras' field of view. These tracks obtained via multi-modal tracking enable *wide-area interaction recognition.*

Chapter 7

Conclusions

7.1 Summary of achievements

We have extended an algorithm based on the track-before-detect strategy initially developed for tracking targets with low observability to visual multi-sensor multi-target tracking. We use the *fuse-before-track* approach to perform multi-camera tracking with multi-level homography employed to fuse information from each camera on the top-view. Next, we have extended the target state with signal strength and used it as the intensity signal as in the track-before-detect approach. The tracking algorithm is applied using a Sampling Importance Resampling Particle filter. To enable multi-target tracking we have proposed a distributed weighting scheme that enables tracking of newly born targets with low confidence within the multi-modal mixture of particles. The distributed weight update is applied by first clustering the particles using mean-shift and then updating the weight of surviving particles. To further ensure consistent multi-target tracking, the resampling is also applied in a distributed manner. This distributed resampling not only avoids degeneracy but also ensures fair allocation of particles across multiple targets. The framework also reduces the computational load by eliminating the detection step. The approach is general and could be applied to other domains such as tracking in beamforming data in wireless networks. We have showed that reducing the number of sensors in the overlapping regions has a very minimal effect on the performance of the algorithm (Fig. 3.39) provided the total coverage area remains the same. This may not be the case in other state of the art algorithms where either a camera mounted at a significant height is needed [73] or where multiple overlapping cameras are used for occlusion reasoning [71].

We proposed a multi-modal tracking framework for Stereo Audio and Cyclop-tic vision (STAC) sensors, consisting of a camera mounted between a microphone pair.

We have improved the localisation accuracy by post-processing with multi-band analysis and reverberation filtering, and with a multi-modal tracking algorithm based on Weighted Probabilistic Data Association (WPDA). WPDA takes into account a weighted probability of the detections in each iteration, to increase the importance of reliable audiovisual measurements, based on the prior estimates and on validation data, and further weaken the unreliable hypotheses. The approach can be applied to overcome failures in video modalities due to video only occlusion. The comparison of the proposed approach with algorithm performing trajectory estimation in regions outside cameras' fields of view without using audio modality [159] shows that using audio has resulted in improvements in localisation estimates (Table 4.6). In regions within cameras' fields of view the proposed dynamic fusion between audio and video modality has contributed in decreasing localisation errors, particularly when audio estimates are unreliable due to being generated from few STACs (Table 4.6). Since the proposed approach also performs reverberation and noise filtering it can be applied in indoor scenarios such as meeting rooms. Moreover since the STAC sensor is simple, this makes the approach scalable and can be made applicable in wide area surveillance such as in underground stations.

We have modelled the interaction recognition as a state estimation problem. The problem is modelled as either recognising interactions between dynamic and static (dynamic-static) objects or between multiple dynamic objects. For dynamic-static interaction recognition, object-centric and scene-centric models are proposed. In these models, the state of either the static object or the dynamic object is estimated with respect to the other interacting targets. In contrast with existing approaches, the state estimation is performed using Viterbi decoding, which eliminates the dependency on having the event sequence known a priori. The dynamic-dynamic interactions were modelled using a Coupled Hidden Markov Model (CHMM) that allows modelling of the full coupling between the interacting targets. It has been demonstrated that CHMM-MAP (using maximum a posterior approach) performs better than both HMM and CHMM-ML (using maximum likelihood approach) (Table 6.6). The proposed interaction recognition utilises the trajectories estimated using multi-modal sensors. This enables interaction recognition within the fields of view of cameras as well as in regions uncovered by the cameras. The algorithm is demonstrated on scenarios from real surveillance and sports scenarios where, given that tracking can be performed, the interactions can be recognized using the proposed approach.

7.2 Future work

- One of the limitations of the proposed multi-target track-before-detect particle filtering algorithm (MT-TBD-PF) is that it requires a large number of particles in order to estimate the posterior distribution. The performance can be improved by applying a hybrid approach [6] whereby mean-shift is used to sample towards the nearby local maxima. The mean shift based optimisation could be applied for each identified cluster in the resampling step.

- The modelling of the number of targets in the scene is not performed in the proposed MT-TBD-PF. This results in sporadic birth of new targets in the case of noisy data. Jump state Markov Models [147] can be used to model the number of targets as part of the state. This would introduce a smoothing effect on the number of targets and further help to reduce the number of false detections.

- We have observed by analysing the steered beamformer output, from multiple stereo audio cycloptic vision (STAC) sensors that it generates a similar intensity signal to that of the detection volume obtained by multi-camera multi-level homography discussed in Chapter 3. This will enable us to explore MT-TBD-PF in tracking audio targets in noisy scenarios.

Bibliography

[1] H. G. Jung, Y. H. Lee, P. J. Yoon, I. Y. Hwang, and J. Kim. Sensor fusion based obstacle detection/classification for active pedestrian protection system. In *Springer LNCS 4292*. 2006.

[2] Y.-C. Chiou and W.-C. Li. Flaw detection of cylindrical surfaces in pu-packing by using machine vision technique. *Measurement*, 42(7):989–1000, 2009.

[3] N. Inamoto and H. Saito. Intermediate view generation of soccer scene from multiple videos. In *Proc. of IEEE Conf. on Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 2002.

[4] S. Sylvain, R. Stiefelhagen, and J. McDonough. *Computers In the Human Interaction Loop*. CHIL, ELDA, 2005.

[5] E. Maggio, F. Smeraldi, and A. Cavallaro. Adaptive multifeature tracking in a particle filtering framework. *IEEE Trans. on Circuits System and Video Technology*, 17(10):1348–1359, 2007.

[6] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Philadelphia, PA, USA, 2005.

[7] S. Zhou, R. Chellappa, and B. Moghaddam. Appearance tracking using adaptive models in a particle filter. In *Proc. of the Asian Conf. on Computer Vision*. Melbourne, AU, 2002.

[8] A. Bhattacharyya. *On a measure of divergence between two statistical populations defined by probability distributions*, volume 35. Bulletin of the Calcutta Mathematical Society, 1943.

[9] M. Deans, C. Kunz, R. Sargent, E. Park, and L. Pedersen. Combined feature based and shape based visual tracker for robot navigation. In *Proc. of IEEE Int. Conf. on Aerospace*. Big Sky, MT, USA, 2005.

[10] S. Karlsson, M. Taj, and A. Cavallaro. Detection and tracking of humans and faces. *EURASIP Journal on Image and Video Processing*, (1):1–9, 2008.

[11] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. of the European Conf. on Computer Vision*. Prague, CZ, 2004.

[12] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *Proc. of the European Conf. on Computer Vision*. Marseille, FR, 2008.

[13] J. Chen, J. Benesty, and Y. Huang. Time delay estimation in room acoustic environments: An overview. *EURASIP Journal on Applied Signal Processing*, 2006.

[14] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. on Image Processing*, 14:294–307, 2005.

[15] M. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Kauai, HI, USA, 2001.

[17] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. of Int. Conf. on Computer Vision Systems*. Nice, FR, 2003.

[18] D. Smith and S. Singh. Approaches to multisensor data fusion in target tracking: A survey. *IEEE Trans. on Knowledge and Data Engineering*, 18(12):1696–1710, 2006.

[19] J. Czyz, B. Ristic, and B. Macq. A particle filter for joint detection and tracking of color objects. *Elsevier Journal of Image and Vision Computing*, 25:1271–1281, 2006.

[20] M. Hadzagic, H. Michalska, and E. Lefebvre. Track-before detect methods in tracking low-observable targets: A survey. *On-line magzine: Sensors and Transducers, special issue on Multisensor Data and Information Processing*, 7(2), 2005.

[21] N. Anjum and A. Cavallaro. Trajectory association and fusion across partially overlapping cameras. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. Genova, IT, 2009.

[22] W. Du and J. Piater. Multi-camera people tracking by collaborative particle filters and principal axis-based integration. In *Proc. of the Asian Conf. on Computer Vision*. Tokyo, JP, 2007.

[23] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Madison, WI, USA, 2003.

[24] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Madison, WI, USA, 2003.

[25] V. Morariu and O. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. NY, USA, 2006.

[26] N. Pham, W. Huang, and S. Ong. Probability hypothesis density approach for multi-camera multi-object tracking. In *Proc. of the Asian Conf. on Computer Vision*. Tokyo, JP, 2007.

[27] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *IEEE Int. Workshop on Motion and Video Computing*. Orlando, FL, USA, 2002.

[28] W. Qu, D. Schonfeld, and M. Mohamed. Distributed Bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. *EURASIP Journal on Applied Signal Processing*, (1), 2007.

[29] Q. Cai and J. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(11):1241–1247, 1999.

[30] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, 2003.

[31] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl. Autonomous multicamera tracking on embedded smart cameras. *EURASIP Journal on Embedded Systems*, 2007.

[32] B. Wu, X. Song, V. Singh, and R. Nevatia. Evaluation of usc human tracking system for surveillance videos. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[33] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. of IEEE Int. Conf. on Computer Vision*. Washington, DC, USA, 2005.

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

[35] S. Munder and D. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, 2006.

[36] X. Song and R. Nevatia. Robust vehicle blob tracking with split/merge handling. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[37] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:747–757, 2000.

[38] A. Cavallaro and T. Ebrahimi. Interaction between high-level and low-level image analysis for semantic video object extraction. *EURASIP Journal on Applied Signal Processing*, 6:786–797, 2004.

[39] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 2005.

[40] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[41] A. Pnevmatikakis, L. Polymenakos, and V. Mylonakis. The AIT outdoors tracking system for pedestrians and vehicles. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[42] Y. Zhai, P. Berkowitz, A. Miller, K. Shafique, A. Vartak, B. White, and M. Shah. Multiple vehicle tracking in surveillance video. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[43] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proc. of the European Conf. on Computer Vision*. Springer-Verlag, London, UK, 2000.

[44] B. Han, D. Comaniciu, and L. Davis. Sequential kernel density approximation through mode propagation: applications to background modeling. In *Proc. of the Asian Conf. on Computer Vision*. Jeju, KR, 2004.

[45] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht. Neural network approach to background modeling for video object segmentation. *IEEE Trans. on Neural Networks*, 18(6):1614–1627, 2007.

[46] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Kaufmann, San Mateo, CA, USA, 1990.

[47] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Buhmann. Topology free hidden markov models: application to background modeling. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 1. Vancouver, CA, 2001.

[48] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:831–843, 2000.

[49] L. Latecki, R. Miezianko, and D. Pokrajac. Motion detection based on local variation of spatiotemporal texture. In *Proc. of IEEE Int. Workshop on Computer Vision and Pattern Recognition.* Washington, DC, USA, 2004.

[50] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report 7694, California Institute of Technology, 2007, online: http://authors.library.caltech.edu/7694/1/CNS-TR-2007-001.pdf, last accessed: 12 July, 2009.

[51] R. Miezianko and D. Pokrajac. Detecting and recognizing abandoned objects in crowded environments. *Springer LNCS 5008*, 5008:241–250, 2008.

[52] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. on Image Processing*, 14(3):294–307, 2005.

[53] J. Gao, A. Kosaka, and A. Kak. A multi-kalman filtering approach for video tracking of human-delineated objects in cluttered environments. *Elsevier Journal of Computer Vision and Image Understanding*, 102(3):260–316, 2006.

[54] Y.-S. Yao and R. Chellappa. Tracking a dynamic set of feature points. *IEEE Trans. on Image Processing*, 4(10):1382–1395, 1995.

[55] J. Vermaak, A. Doucet, and P. Perez. Maintaining multimodality through mixture tracking. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 2. Nice, FR, 2003.

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

[57] Z. Ding, H. Leung, and L. Hong. Decoupling joint probabilistic data association algorithm for multiple target tracking. *IEEE Trans. on Radar, Sonar Navigation*, 146(5):251–254, 1999.

[58] J. Vermaak, S. J. Godsill, and P. P. Pérez. Monte carlo filtering for multi-target tracking and data associtation. *IEEE Trans. on Aerospace and Electronic Systems*, 41(1):309–332, 2005.

[59] T.-L. Liu and H.-T. Chen. Real-time tracking using trust-region methods. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):397–402, 2004.

[60] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, London, UK, 2004.

[61] S. Herman. *A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification*. Ph.D. thesis, University of Illinois at Urbana Champaign, 2005.

[62] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Contr.*, AC-24:843–854, 1979.

[63] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle PHD filter for multi-target visual tracking. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Honolulu, HI, USA, 2007.

[64] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:51–65, 2005.

[65] C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[66] M. Rowan and F. Maire. *An Efficient Multiple Object Vision Tracking System using Bipartite Graph Matching*. Federation of Int. Robot-soccer Association, Busan, KR, 2004.

[67] H. L. H. Chen and T. Liu. Multi-object tracking using dynamical graph matching. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Kauai, HI, USA, 2001.

[68] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Ft. Collins, CO, USA, 1999.

[69] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):1–45, 2006.

[70] C. Shen, C. Zhang, and S. Fels. A multi-camera surveillance system that estimates quality-of-view measurement. In *Proc. of IEEE Int. Conf. on Image Processing*, volume 3. San Antonio, TX, USA, 2007.

[71] S. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2009.

[72] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[73] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008.

[74] K. Kim and L. S. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *Proc. of the European Conf. on Computer Vision*. Graz, AT, 2006.

[75] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. of the European Conf. on Computer Vision*. Graz, AT, 2006.

[76] D. Delannay, N. Danhier, and C. D. Vleeschouwer. Detection and recognition of sports (wo)man from multiple views. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*. Como, IT, 2009.

[77] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94, 1996.

[78] P. Moghaddam, H. Amindavar, and R. Kirlin. A new time-delay estimation in multipath. *IEEE Trans. on Signal Processing*, 51(5):1129–1142, 2003.

[79] J. Fuchs. Multipath time-delay detection and estimation. *IEEE Trans. on Signal Processing*, 47(1):237–243, 1999.

[80] T. Manickam, R. Vaccaro, and D. Tufts. A least-squares algorithm for multipath time-delay estimation. *IEEE Trans. on Signal Processing*, 42(11):3229–3233, 1994.

[81] J. Ianniello. Large and small error performance limits for multipath time delay estimation. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(2):245–251, 1986.

[82] J. Allen and D. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4), 1979.

[83] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. Signal Processing*, 24:320–327, 1976.

[84] Y. Rui and D. Florencio. Time delay estimation in the presence or correlated noise and reverberation. Technical report: Msr-tr-2003-01, Microsoft Research, 2003, online: http://research.microsoft.com/pubs/69982/tr-2003-01.pdf, last accessed: 10 July, 2009.

[85] J. Vermaak, M. Gangnet, A. Blake, and P. Pérez. Sequential monte carlo fusion of sound and vision for speaker tracking. In *Proc. of IEEE Int. Conf. on Computer Vision*. Vancouver, CA, 2001.

[86] R. Chellappa, G. Qian, and Q. Zheng. Vehicle detection and tracking using acoustic and video sensors. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Montreal, CA, 2004.

[87] V. Cevher, A. C. Sankaranarayanan, J. H. McClellan, and R. Chellappa. Target tracking using a joint acoustic video system. *IEEE Trans. on Multimedia*, 9(4):715–727, 2007.

[88] R. Cutler and L. S. Davis. Look who's talking: Speaker detection using video and audio correlation. In *Proc. of IEEE Int. Conf. on Multimedia and Expo (III)*. NY USA, 2000.

[89] D. Gatica-Perez, G. Lathoud, I. McCowan, J.-M. Odobez, and D. Moore. Audio-visual speaker tracking with importance particle filters. In *Proc. of IEEE Int. Conf. on Image Processing*. Barcelona, ES, 2003.

[90] B. Kapralos, M. Jenkin, and E. Milios. Audio-visual localization of multiple speakers in a video teleconferencing setting. *Int. Journal of Imaging Systems and Technology*, 13(1):95–105, 2003.

[91] H. Asoh, F. Asano, T. Yoshimura, K. Yamamoto, Y. Motomura, N. Ichimura, I. Hara, and J. Ogata. An application of a particle filter to bayesian multiple sound source tracking with audio and video information fusion. In *Proc. of the Seventh Int. Conf. on Information Fusion*. Stockholm, SE, 2004.

[92] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan. Audiovisual probabilistic tracking of multiple speakers in meetings. *IEEE Trans. on Audio, Speech, and Language Processing*, 15:601–616, 2007.

[93] N. Checka, K. Wilson, and M. S. T. Darrell. Multiple person and speaker activity tracking with a particle filter. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Cambridge, MA, USA, 2004.

[94] Y. Rui and Y. Chen. Better proposal distributions: object tracking using unscented particle filter. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. HI, USA, 2001.

[95] M. J. Beal, H. Attias, and N. Jojic. Audio-video sensor fusion with probabilistic graphical models. In *Proc. of the European Conf. on Computer Vision*. Copenhagen, DK, 2002.

[96] T. Hospedales and S. Vijayakumar. Structure inference for bayesian multisensory scene understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(12):2140–2157, 2008.

[97] X. Zou and B. Bhanu. Tracking humans using multi-modal fusion. In *IEEE Int. Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*. San Diego, CA, USA, 2005.

[98] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. of IEEE*, 92:495–513, 2004.

[99] T. Gehrig, K. Nicel, H. K. Ekenel, U. Klee, and J. McDonough. Kalman filters for audio-video source localization. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. NY, USA, 2005.

[100] N. Strobel, S. Spors, and R. Rabenstein. Joint audio-video object localization and tracking. *IEEE Signal Processing Magazine*, 18(1):22–31, 2001.

[101] A. Abad, C. Canton-Ferrer, C. Segura, J. L. Landabaso, D. Macho, J. Casas, J. Hernando, M. Pardas, and C. Nadeu. UPC audio, video and multimodal person tracking systems in the CLEAR evaluation campaign. In *CLEAR, Springer LNCS 4122*. Southampton, UK, 2006.

[102] A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. A decision fusion system across time and classifiers for audio-visual person identification. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[103] D. Gatica-Perez, G. Lathoud, J. Odobez, and I. McCowan. Audio-visual probabilistic tracking of multiple speakers in meetings. *IEEE Trans. Audio, Speech and Language Processing*, 2006.

[104] R. Brunelli, A. Brutti, P. Chippendale, O. Lanz, M. Omologo, P. Svaizer, and F. Tobia. A generative approach to audio-visual person tracking. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[105] D. N. Zotkin, R. Duraiswami, and L. S. Davis. Joint audio-visual tracking using particle filters. *EURASIP Journal on Applied Signal Processing*, 2002:1154–1164, 2001.

[106] Y. Chen and Y. Rui. Speaker tracking using particle filter sensor fusion. *Proceedings of the IEEE*, 92(3), 2004.

[107] C. Stauffer. Automated audio-visual analysis. Technical Report MIT-CSAIL-TR-2005-057, Computer Science and Artificial Intelligence Laboratory, MIT, 2005.

[108] O. Lanz. Approximate bayesian multibody tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, 2006.

[109] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[110] K. Zia, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1805–1819, 2005.

[111] D. B. Ward and R. C. Williamson. Particle filter beamforming for acoustic source localisation in a reverberant environment. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Orlando, FL, USA, 2002.

[112] J. Vermaak and A. Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Salt Lake City, UT, USA, 2001.

[113] E. Lehmann, D. Ward, and R. Williamson. Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Hong Kong, CN, 2003.

[114] D. B. Ward, E. A. Lehmann, and R. C. Williamson. Particle filtering algorithms for acoustic source localization. *IEEE Trans. on Speech and Audio Processing*, pages 826–836, 2003.

[115] A. Blake, M. Gangnet, P. Perez, and J. Vermaak. Integrated tracking with vision and sound. In *Proc. of IEEE Int. Conf. on Image Analysis and Processing*. Palermo, IT, 2001.

[116] B. D. O. Anderson and J. B. Moore. *Optimal filtering*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1979.

[117] M. Bregonzio, M. Taj, and A. Cavallaro. Multi-modal particle filtering tracking using appearance, motion and audio likelihoods. In *Proc. of IEEE Int. Conf. on Image Processing*. San Antonio, TX, USA, 2007.

[118] K. Wilson, N. Checka, D. Demirdjian, and T. Darrell. Audio-video array source separation for perceptual user interfaces. In *Workshop on Perceptive user interfaces*. 2001.

[119] U. Bub, M. Hunke, and A. Waibel. Knowing who to listen to in speech recognition: visually guided beamforming. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1. 1995.

[120] B. Kapralos, M. Jenkin, and E. Milios. Audio-visual localization of multiple speakers in a video teleconferencing setting. *Int. Journal of Imaging Systems and Technology*, pages 95–105, 2003.

[121] M. Fallon and S. J. Godsill. Multi target acoustic source tracking using track before detect. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY, USA, 2007.

[122] I. Garvanov and C. Kabakchiev. Sensitivity of track before detect multiradar system toward the error measurements of target parameters. *Cybernetics and Information Technologies*, 7(2), 2007.

[123] M. G. S. Bruno and J. M. F. Moura. Multiframe detector/tracker: optimal performance. *IEEE Trans. on Aerospace and Electronic Systems*, 37(3):925–945, 2001.

[124] D. J. Salmond and H. Birch. A particle filter for track-before-detect. In *Proc. of the American Control Conference*. Arlington, VA, USA, 2001.

[125] Y. Boers and J. Driessen. Multitarget particle filter track before detect application. *IEE Proc.-Radar Sonar Navig.*, 151(6):1271–1281, 2004.

[126] O. Nichtern and S. R. Rotman. Parameter adjustment for a dynamic programming track-before-detect-based target detection algorithm. *EURASIP Journal on Advances in Signal Processing*, 2008.

[127] M. Rutten, N. Gordon, and S. Maskell. Recursive track-before-detect with target amplitude fluctuations. *IEEE Trans. on Radar, Sonar Navigation*, 152(5):345–352, 2005.

[128] T. Yang, Q. Pan, S. Z.Li, and J. Li. Multiple layer based background maintenance in complex environment. In *Int. Conf. on Image and Graphics*. Hong Kong, CN, 2004.

[129] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[130] W. Hu, M. Hu, X. Zhou, and J. Lou. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):663, 2006.

[131] D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(1):22–30, 1999.

[132] J. Hopcroft and R. Karp. An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Computing*, 2(4):225–230, 1973.

[133] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media Inc., Sebastopol, CA, USA, 2008.

[134] G. Kayumbi and A. Cavallaro. Multiview trajectory mapping using homography with lens distortion correction. *EURASIP Journal on Image and Video Processing*, (1), 2008.

[135] N. Anjum and A. Cavallaro. Unsupervised fuzzy clustering for trajectory analysis. In *Proc. of IEEE Int. Conf. on Image Processing*. San Antonio, TX, USA, 2007.

[136] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[137] K. Punithakumar, T. Kirubarajan, and A. Sinha. A sequential monte carlo probability hypothesis density algorithm for multitarget track-before-detect. In *Proc. of SPIE*. 2005.

[138] Z. Jia, A. Balasuriya, and S. Challa. Vision based data fusion for autonomous vehicles target tracking using interacting multiple dynamic models. *Elsevier Journal of Computer Vision and Image Understanding*, 109(1):1–21, 2008.

[139] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009.

[140] D. Cher. *ETISEO Metrics Definition*. Silogic, Toulouse Cedex 1, FR, 2006, https://www-sop.inria.fr/orion/ETISEO/iso_album/eti-metrics_definition-v2.pdf, last accessed: 30 June, 2009 .

[141] Raw videos courtesy of Institute of Intelligent Systems for Automation - C.N.R., Bari, IT. http://www.issia.cnr.it, last accessed: 26 June, 2008.

[142] R. Fisher. Caviar: Context aware vision using image-based active recognition, 2001–2005, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/caviar.htm, last accessed: 30 June, 2009.

[143] J. Ferryman. Performance evaluation of tracking and surveillance, 2006, http://www.cvg.rdg.ac.uk/PETS2006/data.html, last accessed: 30 June, 2009.

[144] Raw videos courtesy of mediapro, ES. http://www.mediapro.es/eng/futbol.htm, last accessed: 18 July, 2009.

[145] http://www.apidis.org/dataset/ Last accessed: 14 April 2009.

[146] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1200–1207. Miami, FL, USA, 2009.

[147] W. R. Gilks. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1995.

[148] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signal*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1978.

[149] R. Y. Litovsky, H. S. Colburn, W. A. Yost, and S. J. Guzman. The precedence effect. *The Journal of the Acoustical Society of America*, 106:1633–1654, 1999.

[150] K. Wilson and T. Darrell. Improving audio source localization by learning the precedence effect. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Philadelphia, PA, USA, 2005.

[151] W. A. Yost and G. Gourevitch. *Directional Hearing, chapter The precedence effect*. Springer-Verlag, 1987.

[152] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, Inc., 4^{th} edition, 2000. Ch 12 pp. 341, ISBN 0-471-84789-5.

[153] T. Sullivan and R. Stern. Multi-microphone correlation-based processing for robust speech recognition. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Minneapolis, MN, USA, 1993.

[154] R. Nikoukhah, A. S. Willsky, and B. C. Levy. Generalized riccati equations for two-point boundary-value descriptor systems. In *Proc. of the IEEE Int. Conf. on Decision and Control*. LA, CA, USA, 1987.

[155] S. M. Bozic. *Digital and Kalman Filtering*. Edward Arnold, London, UK, 1979.

[156] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998.

[157] M. Taj, E. Maggio, and A. Cavallaro. Multi-feature graph-based object tracking. In *CLEAR, Springer LNCS*, volume 4122. Southampton, UK, 2006.

[158] H. Zhou, M. Taj, and A. Cavallaro. Audiovisual tracking using STAC sensors. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*. Vienna, AT, 2007.

[159] N. Anjum, M. Taj, and A. Cavallaro. Relative position estimation of non-overlapping cameras. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Honolulu, HI, USA, 2007.

[160] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *Proc. of IEEE Conf. on Pattern Recognition*. Hong Kong, CN, 2006.

[161] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Elsevier Journal of Computer Vision and Image Understanding*, 104(2-3), 2006.

[162] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008.

[163] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008.

[164] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008.

[165] S. Velipasalar, L. Brown, and A. Hampapur. Specifying, interpreting and detecting high-level, spatio-temporal composite events in single and multi-camera systems. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. NY, USA, 2006.

[166] I. Rezek, M. Gibbs, and S. J. Roberts. Maximum a posteriori estimation of coupled hidden Markov models. *Journal of VLSI Signal Processing Systems*, 32(1-2):55–66, 2002.

[167] T. S. Mahmood, A. Vasilescu, and S. Sethi. Recognizing action events from multiple view points. In *Proc. of IEEE Workshop on Detection and Recognition of Events in Video*. Madison, WI, USA, 2001.

[168] N. Ghanem, D. DeMenthon, D. Doermann, and L. Davis. Representation and recognition of events in surveillance video using Petri nets. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Washington DC, USA, 2004.

[169] Y. Wang. The variable-length hidden Markov model and its applications on sequential data mining. Technical report, Tsinghua University, Beijing, CN, 2006, online: http://learn.tsinghua.edu.cn:8080/2001315444/VLHMM/icdm-techreport.pdf, last accessed: 09 June, 2008.

[170] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[171] G. G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):873–889, 2001.

[172] P. Natarajan and R. Nevatia. Coupled hidden semi Markov models for activity recognition. In *IEEE Int. Workshop on Motion and Video Computing*. Austin, TX, USA, 2007.

[173] J. M. Ferryman. Performance evaluation of tracking and surveillance. In *Proceedings of the Ninth IEEE Workshop on PETS*. NY, USA, 2006.

[174] i-LIDS Team. Imagery library for intelligent detection systems (i-lids); a standard for testing video based detection systems. In *Proc. of IEEE Int. Carnahan Conf. on Security Technology*. 2006.

[175] J. Fiscus and T. Rose. *Straw Man Proposal for the TRECVid 2008 Evaluation*. NIST, Gaithersburg, USA, 2008.

[176] M. Spengler and B. Schiele. Automatic Detection and Tracking of Abandoned Objects. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. Nice, FR, 2003.

[177] S. Guler and M. K. Farrow. Abandoned object detection in crowded places. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[178] L. Li, R. Luo, R. Ma, W. Huang, and K. Leman. Evaluation of an ivs system for abandoned object detection on pets 2006 datasets. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[179] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia. Left-luggage detection using bayesian inference. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[180] K. Smith, P. Quelhas, , and D. Gatica-Perez. Detecting abandoned luggage items in a public space. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[181] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, and R. Collins. Multi-view detection and tracking of travelers and luggage in mass transit environments. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[182] M. Bhargava, C. Chen, M. Ryoo, and J. Aggarwal. Detection of abandoned objects in crowded environments. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. London, UK, 2007.

[183] M. Bhargava, C. Chen, M. Ryoo, and J. Aggarwal. Detection of object abandonment using temporal logic. *Springer Journal of Machine Vision and Applications*, 20(5), 2009.

[184] S. Ferrando, G. Gera, M. Massa, and C. Regazzoni. A new method for real time abandoned object detection and owner tracking. In *Proc. of IEEE Int. Conf. on Image Processing*. Atlanta, GA, USA, 2006.

[185] T. Martiriggiano, A. Caroppo, M. Leo, P. Spagnolo, and T. DOrazio. An innovative approach for abandoned or removed objects detection. In *Proc. of the Int. Symposium on Communications, Control and Signal Processing*. Marrakesh, MA, 2006.

[186] F. Porikli. Detection of temporarily static regions by processing video at different frame rates. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. London, UK, 2007.

[187] F. Porikli, Y. Ivanov, , and T. Haga. Robust abandoned object detection using dual foregrounds. *EURASIP Journal on Advances in Signal Processing*, 2008.

[188] N. Bird, S. Atev, N. Caramelli, R. Martin, O. Masoud, and N. Papanikolopoulos. Real time, online detection of abandoned objects in public areas. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*. 2006.

[189] Y. Tian, R. Feris, and A. Hampapur. Real-time detection of abandoned and removed objects in complex environments. In *Int. Workshop on Visual Surveillance*. Marseille, FR, 2008.

[190] J. San Miguel and J. Martinez. Robust unattended and stolen object detection by fusing simple algorithms. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. 2008.

[191] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008.

[192] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2009.

[193] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *Proc. of IEEE Int. Conf. on Computer Vision*. Vancouver, CA, 2001.

[194] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. 2007.

[195] G. Chartrand. *Introductory Graph Theory*. Dover Publications, NY, USA, 1985. Ch. 1, Directed Graphs as Mathematical Models, pp. 16–19.

[196] K. Murphy. *Dynamic Bayesian networks: Representation, inference and learning*. Ph.D. thesis, Department of Computer Science, UC Berkeley, 2002.

[197] G. Wu, Y. Wu, L. Jiao, Y. Wang, and E. Y. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. In *Proc. of ACM Int. Conf. on Multimedia*. NY, USA, 2003.

[198] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. of IEEE Int. Conf. on Computer Vision*. Beijing, CN, 2005.

[199] R. J. Morris and D. C. Hogg. Statistical models of object interaction. *Int. Journal of Computer Vision*, 37(2):209–215, 2000.

[200] M. Brand. Coupled hidden Markov models for modeling interacting processes. MIT media lab perceptual computing / learning and common sense technical report 405, Massachusetts Institute of Technology, 1997, online: http://citeseer.ist.psu.edu/7422.html, last accessed: 30 December, 2008.

[201] L. Zhang, D. Samaras, N. A. Klein, N. Volkow, and R. Goldstein. Modeling neuronal interactivity using dynamic bayesian networks. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1593–1600. MIT Press, Cambridge, MA, USA, 2006.

[202] M. Taj and A. Cavallaro. Multi-camera scene analysis using an object-centric continuous distribution hidden Markov model. In *Proc. of IEEE Int. Conf. on Image Processing*. San Antonio, TX, USA, 2007.

[203] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:844–851, 2000.

[204] A. Galata, A. Cohn, D. Magee, and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length Markov models. In *Proc. of European Conf. on Artificial Intelligence*. Lyon, FR, 2002.

[205] I. Rezek and S. J. Roberts. Estimation of coupled hidden Markov models with application to biosignal interaction modelling. In *IEEE Int. Workshop on Neural Networks for Signal Processing*. 2000.

[206] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*. San Francisco, CA, USA, 1996.

[207] S. Intille and A. Bobick. Recognizing planned, multi-person action. *Elsevier Journal of Computer Vision and Image Understanding*, 81(3):414–445, 2001.

[208] A. Hakeem and M. Shah. Learning, detection and representation of multi-agent events in videos. *Elsevier Journal of Artificial Intelligence*, 171:586–605, 2007.

[209] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 2. Washington, DC, USA, 2003.

[210] P. Dai, H. Di, L. Dong, L. Tao, and G. Xu. Group interaction analysis in dynamic context. *IEEE Trans. Syst., Man, Cybern. B*, 38(1):275–282, 2008.

[211] D. Chen, R. Malkin, and J. Yang. Multimodal detection of human interaction events in a nursing home environment. In *Proc. Int. Conf. on Multimodal Interfaces*. Penn State University, State College, PA, USA, 2004.

[212] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1. Washington, DC, USA, 2004.

[213] E. Marhasev, M. Hadad, and G. A. Kaminka. Non-stationary hidden semi Markov models in activity recognition. In *Proc. of the AAAI Workshop on Modeling Others from Observations*. Boston, MA, USA, 2006.

[214] M. Russell and R. Moore. Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*. Tampa, FL, USA, 1985.

[215] D. Burshtein. Robust parametric modeling of durations in hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, 4(3):240–242, 1996.

[216] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier. Left-luggage detection using homographies and simple heuristics. In *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. NY, USA, 2006.

[217] F. Ziliani, S. Velastin, F. Porikli, L. Marcenaro, T. Kelliher, A. Cavallaro, and P. Bruneaut. Performance evaluation of event detection solutions: the creds experience. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. 2005.

[218] M. toolbox for Coupled Hidden Markov Modelling using Maximum A posteriori EM. http://www.robots.ox.ac.uk/ parg/software.html. Last accessed: 26 November, 2009, toolbox last updated: 20 july 2001.

[219] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*, volume 2. John Wiley & Sons, NY, USA, 1996.

[220] V. Roth and T. Lange. *Feature Selection in Clustering Problems*. MIT Press, Cambridge, MA, USA, 2004.

[221] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of IEEE*, pages 267–296, 1990.

[222] M. Taj and A. Cavallaro. Object and scene-centric activity detection using state occupancy duration modeling. In *Proc. of IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*. Santa Fe, NM, USA, 2008.

[223] S. W. Joo and R. Chellappa. A multiple-hypothesis approach for multiobject visual tracking. *IEEE Trans. on Image Processing*, 16:2849–2854, 2007.