Algorithms for trajectory integration in multiple views

Kayumbi-Kabeya, Gabin-Wilfried

Algorithms for trajectory integration in multiple views

A thesis presented to the University of London
by

Gabin-Wilfried Kayumbi-Kabeya

for the degree of
Doctor of Philosophy
in
Electronic Engineering

Queen Mary University of London
Mile End Road
London, E1 4NS, UK
December 2009

Thesis supervisor
Andrea Cavallaro

Author
Gabin Kayumbi

Algorithms for trajectory integration in multiple views

Abstract

This thesis addresses the problem of deriving a coherent and accurate localization of moving objects from partial visual information when data are generated by cameras placed in different view angles with respect to the scene. The framework is built around applications of scene monitoring with multiple cameras. Firstly, we demonstrate how a geometric-based solution exploits the relationships between corresponding feature points across views and improves accuracy in object location. Then, we improve the estimation of objects location with geometric transformations that account for lens distortions. Additionally, we study the integration of the partial visual information generated by each individual sensor and their combination into one single frame of observation that considers object association and data fusion. Our approach is fully image-based, only relies on 2D constructs and does not require any complex computation in 3D space. We exploit the continuity and coherence in objects' motion when crossing cameras' fields of view. Additionally, we work under the assumption of planar ground plane and wide baseline (i.e. cameras' viewpoints are far apart). The main contributions are: i) the development of a framework for distributed visual sensing that accounts for inaccuracies in the geometry of multiple views; ii) the reduction of trajectory mapping errors using a statistical-based homography estimation; iii) the integration of a polynomial method for correcting inaccuracies caused by the cameras' lens distortion; iv) a global trajectory reconstruction algorithm that associates and integrates fragments of trajectories generated by each camera.

Table of Contents

Previously Published Work

[J1] G. Kayumbi and A. Cavallaro. Multi-view trajectory mapping using homography with lens distortion correction. *EURASIP Journal on Image and Video Processing*, Article ID 145715, doi:10.1155/2008/145715, Volume 2008, 2008.

[C1] G. Kayumbi, N. Anjum, A. Cavallaro. Global trajectory reconstruction from distributed visual sensors. *Proc. of ACM / IEEE Int. Conference on Distributed Smart Cameras (ICDSC)*, Stanford, California (USA), 7-11 September, 2008.

[C2] G. Kayumbi, P.L. Mazzeo, P. Spagnolo, M. Taj, A. Cavallaro. Distributed visual sensing for virtual top-view trajectory generation in football videos. *Proc. of ACM International Conference on Image and Video Retrieval (CIVR)*, Niagara Falls, Canada, July 7-9, 2008.

[C3] G. Kayumbi and A. Cavallaro. Robust homography-based trajectory transformation for multi-camera scene analysis. *Proc. of ACM / IEEE Int. Conference on Distributed Smart Cameras (ICDSC)*, Vienna (AUSTRIA), 25-28 September, 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 and made me learn many lessons beyond that frame. I am grateful to Dr. Carlo Regazzoni for having stimulated my interests in computer vision.

All my gratitude to Nadeem Anjum for his attention and collaboration, Dr. Divna Djordjevic for the unreserved encouragements and faithful friendship, Dr. Emilio Maggio and Dr. Murtaza Taj for the valuable input to my work. I would like to thank all the *ACGroup* guys, in particular Ioannis Tziakos, Fahad Daniyal and Dr. Prathap Nair for their company and help.

Finally, I am indebted to everyone who has contributed in different ways to my work: Dr. Colette Gordon, Simone Messina, Patrizia and Costantino Malatto, Marco Nocetti, Maria Pina Usai, Barthélemy Cador, Pascal Maison and all of you that I could not name here.

To my parents,

Cécile Mujing and Grégoire Kabeya.

~

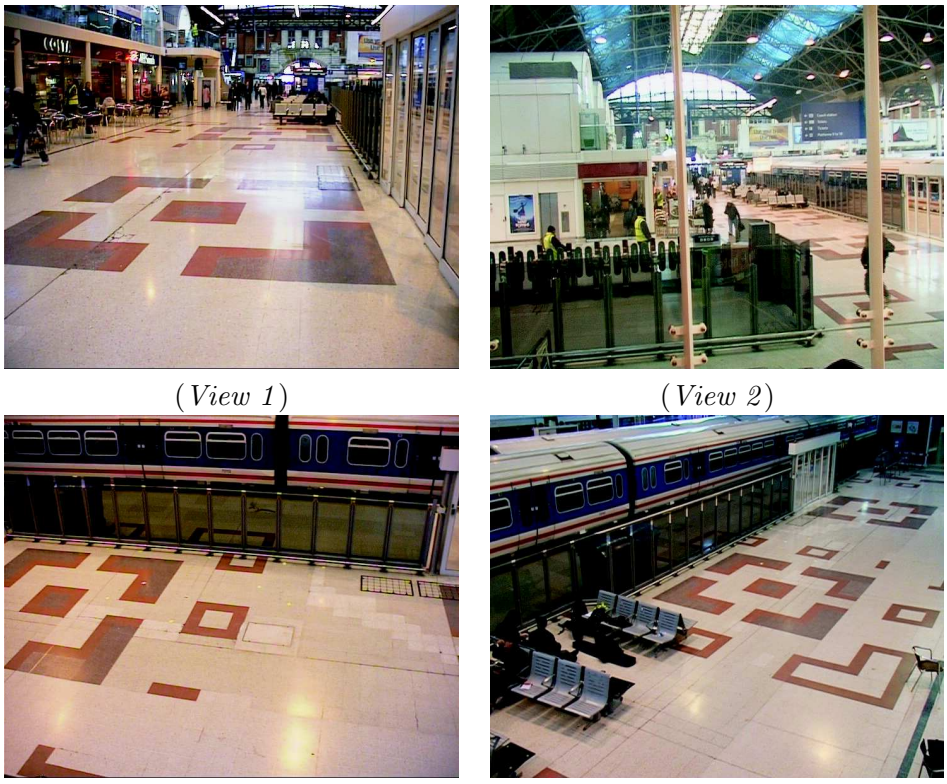
To Kathryn Louise Judd.

Chapter 1

Introduction

1.1 Motivation

Enabling a computer with the capability to perform vision tasks is a remarkably challenging problem. Geometry plays an important role in this context by providing the laws that govern and model the relationship between multiple images of a given scene. Any vision system is intrinsically connected to geometry and thus it is natural that a domain of computer vision makes considerations about geometric viewangle approaches. The application of projective geometry transformations to the fundamental elements of a 2D image (the pixels), for vision task purpose, gives the general context of the area of research the present work belongs to. By and large, these tasks include various domains such as image synthesis, camera calibration, remote sensing, autonomous or aided-navigation and surveillance. Monitoring large areas such as airports, underground stations and sensitive facilities requires a set of distributed cameras. Figure 1.1 is an illustration of such a situation. The station floor is imaged in four cameras, each of them viewing a different portion of the area. These cameras are endowed with a capability to capture common patterns of activities and detect unusual events or anomalous behaviours. The integration of information across the cameras, essential to a complete understanding of the scene,



(*View 1*) (*View 2*)

(*View 3*) (*View 4*)

Figure 1.1: Multi-camera system to monitor a train station. Each camera provides a partial view of the scene. The integration of the information passes through object correspondence across cameras. (*Copyright ISCAPS consortium. Permission of the PETS 2006 workshop.*)

requires the ability to relate the four views in one single global frame of observation. An example of a centralised multiple view system is illustrated by Fig. 1.3. Such systems will recover the visual information and perform automatic scene analysis [2, 3]. The aforementioned situations often include occlusion (objects' visibility being partially or totally hidden by another object or a structure in the scene) and require the estimation of accurate locations in challenging environments. Throughout this thesis, we consider the widely used pinhole camera model [4]. The choice of this model in these works is due to the characteristics of the cameras themselves and the type of information being considered. The pinhole model is simple and sufficient to model the most commonly used cameras in surveillance and monitoring [5]. This model represents the camera's image plane by the projective plane $\pi = P^2$, which is defined as the real Euclidean plane augmented by the line

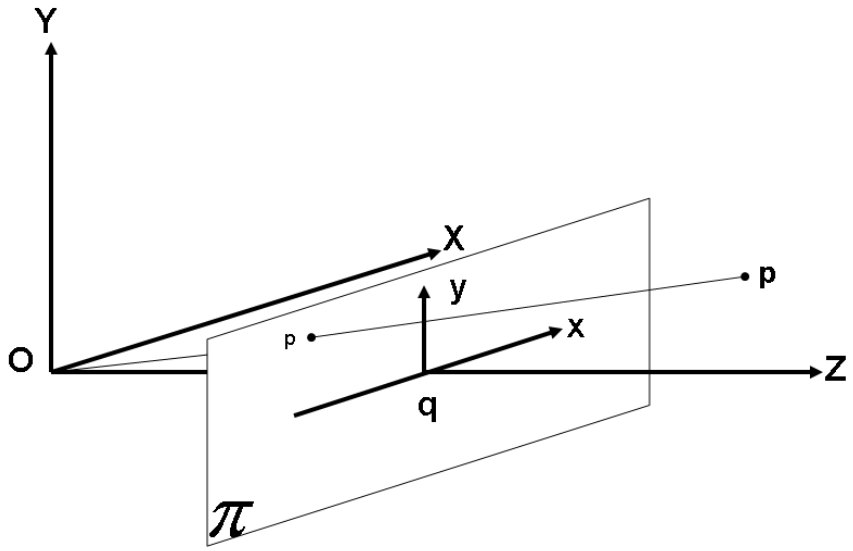


Figure 1.2: The projective camera model. Here π represents the projective plane, $\mathbf{P}$ is a generic point in the Euclidean space and p is its representation on the image plane.

at infinity (Fig. 1.2). Geometric points in a projective space are defined by *homogeneous coordinates*, which means, they are defined up to a non-zero scale factor. Figure 1.2 is an illustration of the aforementioned model. π is the image plane, located at distance $Z = \ell$ from the camera centre O. The line from the camera centre, perpendicular to the image is the principal axis. Under this model, a point $P = (X, Y, Z)^T$ in the Euclidean space $\Re^3$, is mapped to the point $p = (\ell X/Z, \ell Y/Z)^T$ on the image plane. This latter is the Euclidean space $\Re^2$, augmented with the line to infinity [6]. The projective nature of visual sensors, in a context of multi-view monitoring, raises several challenges in the field of Computer Vision and its numerous applications. Surveillance, robot vision, intelligent transportation and video summarization to name a few, require the development of algorithms to perform tasks such as object detection and tracking, object reconstruction and recognition. For the multi-view set, the mutual knowledge of cameras' individual output and the geometry of the ensemble is crucial for an effective maximization of the information collected and processed.

A set of cameras often presents overlapping views that prove to be useful in

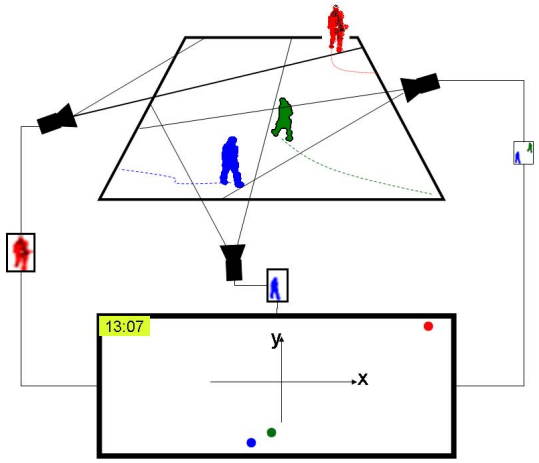


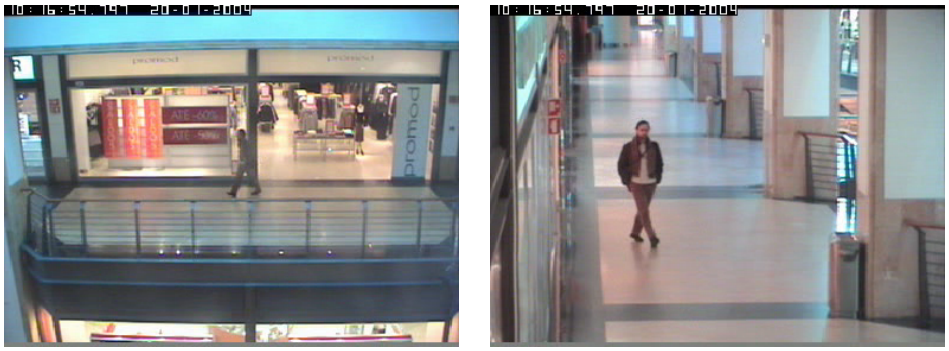
Figure 1.3: Example of a multiple views object tracking system. Data from each camera is integrated into one single framework describing the global visual information of the scene. Adapted from [1]

(View 1) (View 2)

Figure 1.4: Redundancy in multiple view sensing contributes to solve the occlusion occurring in *View 1* by the integration of the information provided by *View 2*.

providing selective observations of specific parts of a scene and delivering elements of redundancy. This will help minimise ambiguities of occlusions (Fig. 1.4), increase accuracy over the position estimate of objects, and extend a site coverage (Fig. 1.5).

The geometry of the scene and the individual configuration of each sensor raise



(front) (corridor)

Figure 1.5: The monitoring of a shop entrance (left) is extended with the addition of a corridor view (right).

the issues of:

- how to combine the observations from each sensor into one consistent view.
- how to perform scene understanding based on features extracted from the views.

This thesis is to be placed within the context of issues related to the first category. In this thesis, we address the problem of visual sensing from multiple views by answering the following questions:

- How to relate partial and concurrent information generated by sensors monitoring a scene from different viewpoints?
- How to reduce errors in the information mutually conveyed by sensors to enhance accuracy in object localization and correspondence across views?

1.2 Main contributions

The main contributions of this thesis are as summarised below.

- We have created a framework for distributed visual sensing that accounts for inaccuracies in the geometry of multiple views [C3, C2]. The reduction of the trajectory mapping errors is achieved by applying a statistical-based homography estimation [7] on points sampled from objects' trajectories, as opposed to linear-based methods [8].

- We have integrated a polynomial method for correcting inaccuracies caused by the lens distortion in the homography-based image-to-image correspondence and thus lowered registration errors in trajectory mapping [J1].
- We have proposed a global trajectory reconstruction algorithm that associates and integrates fragments of trajectories generated by each camera monitoring different portions of a scene [C1]. Our approach is fully image-based, only relies on 2D constructs and does not require any complex computation in 3D.

1.3 Outline of the thesis

This thesis focuses on the use of multiple view relations and the sensors' accuracy in achieving a global objects' location. The contributions mentioned above are presented in Chapter 3 preceded by an analysis of the state-of-the-art and followed by conclusions and directions for future work.

Chapter 2 describes both monocular and multiple views approaches. Motion segmentation as well as pure single view approaches to object tracking are presented. The limitations of these approaches are discussed. Next, multiple view-based algorithms are introduced and their solutions to overcome the limitations of single-view techniques are presented. Finally, a discussion on limitations of conventional multiple views approaches is presented.

Chapter 3 elaborates on the proposed approach, detailing planar homographic constructs, particularly in the case of overlapping fields of view. Two aspects involving accuracy in the correspondence across views are expanded: first, the statistical homography estimation technique and then the embedded lens distortion correction.

Chapter 4 describes the dataset used and the metrics adopted and presents the experiments using surveillance and sport scenarios and discusses results. An evaluation of the proposed approach is also presented.

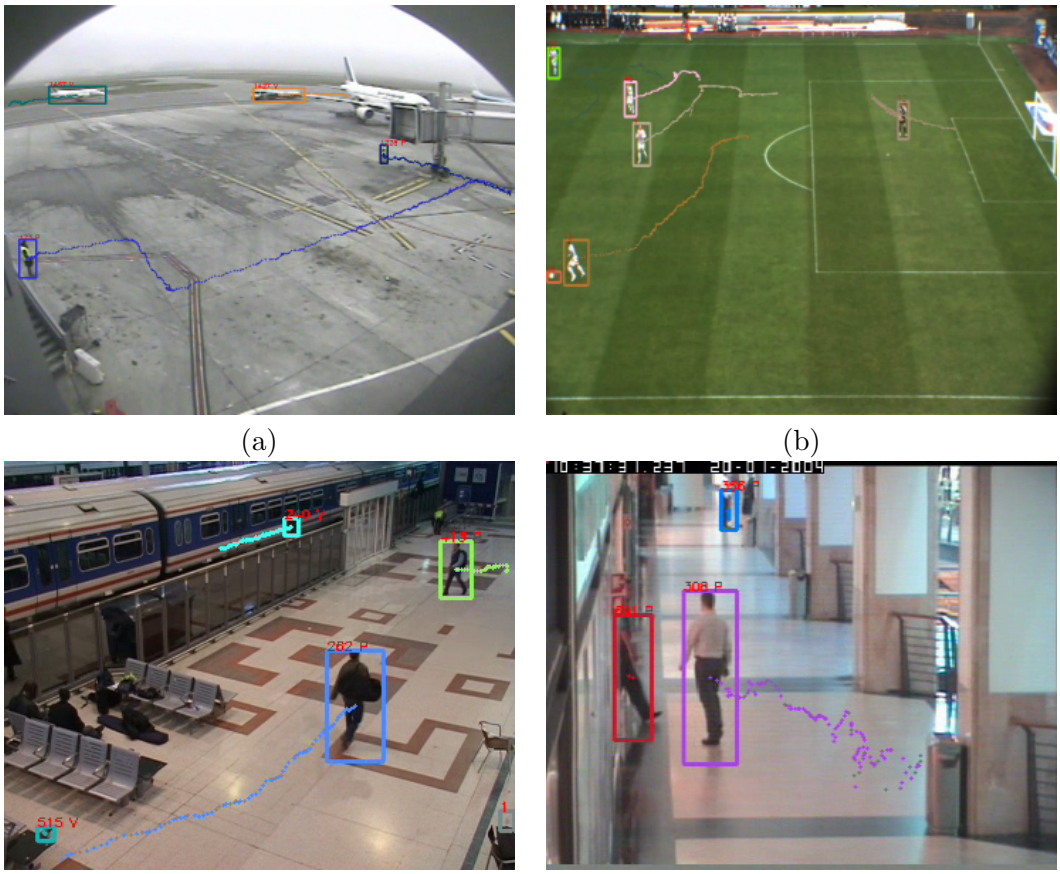
In **Chapter 5** we comment on and summarize the achievements of this thesis and we discuss possible extensions of this work.

Chapter 2

State of the art

2.1 Introduction

In this chapter, we present existing approaches in object tracking to describe the context of the present thesis. The first stage in visual sensing can be characterised by object detection that consists in extracting objects and performing motion segmentation over time. However, because this is not part of the current study, we refer the reader to the body of research presented in [2, 9, 10]. Once objects of interest have been extracted, there is need to track them over time. We highlight the limits of single-view methods, and then we discuss different solutions based on multiple view approaches to overcome these limits. Multi-camera algorithms involved in object tracking can be classified into three main categories: appearance-based, geometry-based and hybrid approaches. These categories are related to the type of features extracted from images to establish the correspondences between objects across views and, build and integrate complementary information in a coherent fashion.



(a) (b)

(c) (d)

Figure 2.1: Examples of object tracking in different scenarios

2.2 Single-view object tracking

Following the detection of objects of interest, there is need to estimate a global spatio-temporal record of their locations (object tracking, Fig.2.1 and 2.2). An overview on object tracking is presented by Yilmaz's work [11]. Tracking is often hampered by changes in appearance, irregular and discontinuous motion and occlusions. In this section, we group the different methods according to their underlying characteristics: deterministic or statistical or the type of features used.

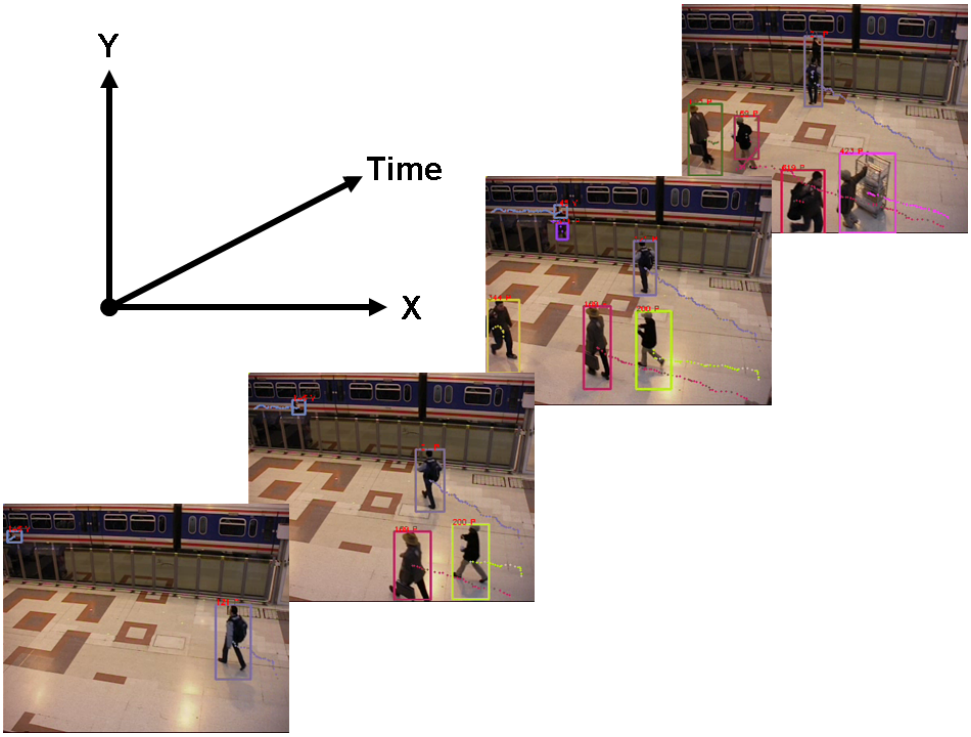


Figure 2.2: Object tracking: spatio-temporal information over time.

2.2.1 Feature point-based approaches

An object is represented by one or more support points that are being tracked along images of the sequence. This point can represent, for example: the centre of gravity, the ground location or the top of the head in the case of humans. These approaches work under the assumption of the availability of this feature at every instant of time and a correspondence is computed between the different observations. The drawback of these methods is that the observations can be affected by noise and corrupt the measurements. The occurrence of occlusion between features can generate a partial or total loss of some feature points.

Deterministic techniques

These methods exploit the idea of temporal continuity constraint to maintain and connect object's tracks and tackle occlusion. An approach consisting of global trajectory

optimization is presented in [12, 13]. The first uses the tensor voting methodology and the second uses dynamic programming to follow individual trajectories. However, these approaches only deal with a limited number of simultaneous trajectories with occluding objects. In order to tackle the issue of occlusion in single-view methods, several techniques have been proposed. A typical approach would detect the occurrence of occlusion by a blob merger [2, 14]. Feng et *al.* [15] propose a graph-based object representation and use SIFT features to describe the object. Features are related as edges in a graph. The computation of likelihood is expressed as a graph matching problem and they use relaxation labelling as a solution. Although this method handles relevant variations of an object's appearance, pose and occlusion, it is heavily dependent on the feature stability and the performance can dramatically decrease if features are not stable. In the graph matching algorithm [16], objects are treated as nodes of a bi-partitioned digraph (i.e., a directional graph), whereas edges are determined by all possible object combinations in adjacent frames and weighted using multiple object features namely position, direction and size. The graph is formed by iteratively creating new edges from the detected targets. Edges represent all possible track hypotheses, including miss-detections and occlusions. The best set of tracks is generated by computing the maximum weight path cover of the graph. Given that the gain function is dependent on the backward correspondences (i.e., the speed at the previous step), a greedy suboptimal version of the graph matching algorithm is used. Since graph matching links nodes based on the highest weights, to avoid connecting two trajectory points far from each other, a gating window is used [16].

Statistical techniques

A probabilistic approach is adopted by Brostow [17] et *al* who propose a probabilistic framework in which they cluster feature points in the trajectories to detect individual pedestrians in a crowd. Other similar approaches [18, 19, 20, 21] rely on appearance models that are actually trained for specific unoccluded views of their respective objects

in the monitored scenes. Therefore, they fail in the case of full occlusion and appearance changes. Wu et *al* [22] incorporate an additional hidden process for occlusion into a dynamic Bayesian network and rely on the statistical inference of the hidden process to reveal occlusion relations. Senior et *al.* [23] use appearance models to localize objects and use ambiguous pixels to resolve their depth ordering when objects occlude each other. However, this algorithm does not maintain the subject's identity after the occurrence of occlusion. Other works [24, 25] have used Expectation Maximization to derive object's appearance and motion, having initially modeled videos as a layered composition of objects.

Hybrid techniques

Okuma et *al.* adopt a synergy of different techniques [26]. They combine both Adaboost, to extract foreground objects and successively, a particle filter to perform multiple-object tracking. This approach reduces the algorithm failures as compared to either one on its own. Additionally, their framework addresses both detection and consistent building of object's tracks in a comprehensive way. A two-stage algorithm has been presented by Perera et *al* [27], which first establishes a one-to-one correspondence and then a split and merge module to maintain object identities and connect trajectories' parts, segmented by occlusions.

2.2.2 Region-based approaches

These approaches either track boundaries of objects or the appearance of the area within these boundaries. The first category focuses on the shape of the objects while the second operates on models of appearance features.

Contour

A curve fitting is used to detect features of a region by means of energy minimization functions. These functions pull the curves towards the detected features. Ricquebourg

et al. [28] exploit the spatio-temporal slices from the image sequence volume to track human motion. Later works propose tracking techniques exploiting object's contours [29, 30] and appearances [31, 22]. They use hidden variables of depth ordering of objects toward the camera, to describe and estimate occlusion relationships between moving objects.

Appearance

In this direction, Zhao et *al* [32, 33] proposed a method where they used articulated ellipsoids to model contour of bodies, colour histograms to model their appearance, and an augmented Gaussian distribution to model the background for segmentation. Once moving head pixels are detected as foreground objects, a principled MCMC approach is used to maximize the posterior probability of a multi-person configuration. Another example is given by the BraMBLe system [34]. It is a multi-blob tracker where the likelihood of each blob is generated using an available background model and the appearance models of the objects in the scene. However, this tracker's performance decreases in situations where several objects, because they are located in each other's vicinity, merge into one blob. To overcome this limit, other methods have considered maintaining the object's state. The object's spatio-temporal location, the dynamics of its motion as well as features related to its appearance (e.g., colour), are constantly maintained and updated.

2.2.3 Graph Matching algorithm

Tracking can be performed by applying a Graph Matching (GM) algorithm on the detected objects [16]. Objects are treated as nodes of a bi-partitioned digraph (i.e., a directional graph), whereas edges are determined by all possible object combinations (track hypotheses) in adjacent frames and weighted using multiple object features, namely position, direction and size. The graph is formed by iteratively creating new edges from the detected targets. The optimal set of tracks is generated by computing the maximum weight path cover of the graph. Since graph matching links nodes based on the highest

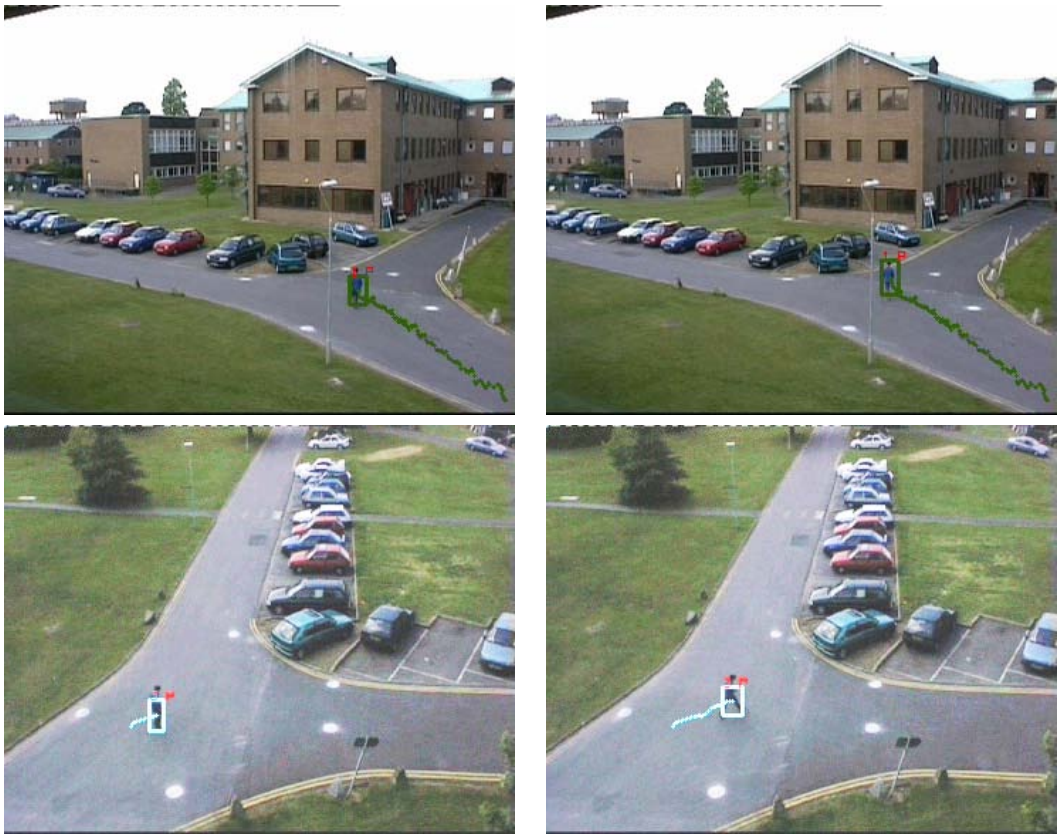


Figure 2.3: Example of detection and tracking results in multiple views from the PETS dataset. (Top) camera1. (Bottom) camera2. (Left) frame 331. (Right) frame 351.

weights, a gating window is used to avoid connecting two trajectory points far from each other. Figure 2.3 shows an example of object detection and tracking results using GM algorithm. Foreground segmentation is performed by a statistical colour change detector [35]: a model-based algorithm that assumes additive white Gaussian noise on each frame. The noise amplitude is estimated for each colour channel. Important local illumination changes are dealt with by performing an edge-based post-processing using selective morphology that filters out misclassified foreground regions by dilating strong foreground edges and eroding weak foreground edges. Next, 8-neighbour connected components analysis is performed to generate a foreground mask. The frequent local illumination changes in real-world sequences affect the estimation of an object shape. A model-based shadow removal approach is employed that assumes that shadows are cast on the ground. The

result of the object detection step is a bounding box for each blob. The next step is to associate subsequent detections of the same object over time, as explained in the next section. Data association has to be verified throughout several frames to validate the correctness of the tracks.Data association is a challenging problem due to track management issues such as the appearance and disappearance of objects, occlusions and false detections due to clutter and noisy measurements.

2.2.4 Limitations

Single-view based approaches often rely on partial observations from spatially restricted views, resulting in difficulties to handle full occlusions. To deal with occlusion, the aforementioned approaches assume small and consistent motions to allow the prediction of objects' motion patterns. However, that assumption results in problems when dealing with extended periods of occlusions of an object under unpredictable motions. Therefore, it clearly appears that single view approaches are still nonetheless limited in their ability to deal with coherent global scene observation. In fact, despite the remarkable progress made, there are difficulties in handling situations where multiple objects occlude each other because the single view-angle platform is intrinsically unable to observe the hidden regions.

Occlusion is a phenomenon that occurs in dynamic scenes when the visibility of an object is partially or totally covered by other moving or static objects, considered as foreground or part of the background.

- *Dynamic occlusions*

  These occur when, during a given interval of time, a second object that entered the field of view of a camera, finds itself in the line of sight between the camera centre and the first object. The blobs of two objects merge and are consequently segmented as one single object. In the case where the occlusion occurs after the two objects

were initially distinguishable, mechanism of occlusions detection can be used to solve it. However, it is difficult to identify the event of occlusion in a single-view sensing if the two objects were occluded from the time they start being visible on the camera. Crowds, movements of pedestrians and team-sports are cases where occlusion are very frequent and difficult to solve with one view only.

- *Static occlusions*

  These take place when an object is hidden from a camera by an object that is part of the background. This includes a building, a tree or any static structure in the scene. The visibility of the object simply ceases and the decision on whether the object is still present or not is made difficult from one view only.

In conclusion, because of the ambiguity generated by changes in the object's appearances, the coverage of a site being limited by a single camera's field of view and the monitoring of groups being hindered by occlusions, it appears that there is need for a wider multiple view approach to overcome these limits.

2.3 Multi-view object tracking

The use of multiple sensors turns out to be a necessity for systems aiming to accurately detect and track multiple objects in real life scenarios. Visual sensing using multiple cameras has been the subject of increasing interest in computer vision in recent years. On the ground, the idea also meets practical requirements. Since conventional cameras have limited fields of view but are getting cheaper, the use of lower resolution commercial off-the-shelf cameras, deployed in a networked system, would achieve the flexibility and scalability that would otherwise be more costly with high-resolution cameras with wide fields of view [36]. Research in this field has spawned different approaches, in different environments to demonstrate the large range of applications and the advantages of cooperative sensing.

A multi-sensor environment is not just a collection of devices that perform their respective tasks independently. Indeed, there is the opportunity to coordinate and integrate information on activities across all cameras in order to improve the performance of the overall system. Multiple view approaches require, in general, the consideration of the relationship between the devices to obtain a global understanding of scene dynamics. Data originating from individual cameras can be brought into one coherent common frame of observation so that the automated surveillance system can infer a global analysis to understand the scene and accordingly allow appropriate decisions. For example, it is important to know whether an object, simultaneously visible in two views, is the same or not and to reconstruct the path followed by the object across cameras' fields of view without ambiguity. The reconstruction of trajectories generated by objects moving across cameras can allow the extraction of information on their global behaviour for various applications, such as sport events analysis, remote sensing, surveillance and monitoring. Multi-view tracking methods also aim to decrease the hidden regions by providing large coverage of monitored areas. This is achieved by exploiting the redundant information in the scene and, representing objects of interest from different viewpoints [37, 38, 39, 40, 41, 42, 43, 44].

The model of projective cameras brings new challenges for information integration in multiple views. Real world constraints are exploited to tackle issues related to creating a coherent and consistent framework for multiple views:

- The existence of a planar surface, the *ground plane*, on which objects move.
- The temporal continuity of motion across cameras.
- The appearance properties of objects.

Figure 1.1 shows an example of a multiple-view monitored site. The station floor is visible in four cameras, each of them viewing a different portion of it. The integration of information across the cameras, essential to a complete understanding of the scene, requires the ability to relate the four views in one single global frame of observation.

The use of multiple cameras raises several issues, enumerated by Weiming [3]:

1. *Image registration*: the process concerns the alignment of corresponding images of the same scene, observed from different viewpoints [45, 46, 9, 47, 48, 1].
2. *Data Fusion*: integration of features from the different sensors [49, 50].
3. *Camera calibration*: determines cameras' intrinsic and extrinsic parameters. It estimates those parameters by using projected images of planar calibration patterns [51, 52, 53].
4. *Camera switching*: finding the camera that gives the best view of an object that enters and exits different fields of view. The system should minimise the switching [54, 55].
5. *Data association*: finding correspondences between the objects in different image sequences from different cameras [46, 54, 1, 44].
6. *Camera installation*: optimum coverage of a scene with minimum number of cameras [56].

We can categorize multiple view approaches in two groups, namely non-overlapping and overlapping configurations.

2.3.1 Non-overlapping fields of view

These methods perform object tracking in areas that are not fully covered by cameras' fields of view. This raises several issues, including: the difficulty to use spatio-temporal proximity given that observations can be widely a part; the high variance in the object's appearance due to difference in pose with respect to two different sensors, different exposure to light and camera properties. Javed et *al.* [57] propose a method that uses the most commonly followed paths of objects to establish correspondences and learn the inter-camera relationships as multivariate probability density distributions of space and time variables using kernel density estimation. They solve the variation in objects' appearances across views, in object matching, by resorting to the principal subspace of brightness

transfer functions that is learned by using principal component analysis. The work by Rahimi et *al.* [58] aims to reconstruct an object's global trajectory across views with unobservable regions, in order to derive the ground plane calibration of the sensors. The objects' dynamics are modelled as Markovian process and, using a non-linear minimization method, they extract the most likely path taken by the object. Kettnaker et *al.* [59] transform the Bayesian problem of trajectory reconstruction, into a linear program to solve the matching between objects across views. However, their method requires the manual input of the paths that are allowed. Therefore, this approach fails to deal with changes in the objects' usual paths. Quaritsch et *al.* present a method for decentralized handover mechanisms between neighbouring cameras [60] where there is no need for a centralised coordination. A single view tracking is initiated and trackers use Camshift algorithm to follow supervised objects across views. The camera handoff is achieved by using a mobile agent system available on the intelligent camera network.

2.3.2 Overlapping fields of view

The relationships between cameras may be extracted in two main ways: using a prior information on knowledge about the scene static features, either through camera calibration or by extracting corresponding points in areas of overlap. Camera calibration makes use of the relative positions of static features. They present a high level of accuracy. However, particularly in cases of wide baseline and outdoors scenes, this often implies the use of sophisticated equipment such as GPS devices or geodetically aligned elevation maps of the ground. Calibration can also be a daunting task as a slight change in one device will require the entire process to be repeated. Other methods are those based on video data and extract feature points from objects' trajectories. According to the degree of overlap, there exist two cases, the *small baseline* and the *wide baseline*.

Small baseline

They present small perspective distortion hence linear methods are applicable to matching process. Feature based cross-correlation methods easily find correspondences between the images. However, in the case of significant repetitions of structures of features, there is the risk of high numbers of false positives. When it comes to monitoring large areas, configurations that use small baselines would be impractical as they would require a considerable number of devices. Several works have addressed the problem of stereo matching in multiple view by proposing the integration of stereo pairs as the main approach. In their work, Krumm et *al.* [61] utilise stereo cameras and integrate visual data from several stereo cameras in the world coordinate system. First, a background subtraction is carried out and then the detection of people's shapes is performed in 3D space. They model each individual with a colour histogram which is successively used to identify and track people across views. In the same fashion, Mittal et *al.* [62] propose to integrate information from a stereo image pair. They consider areas in the image pair and compare them with each other. The back projection into world coordinates is conducted so that corresponding 3D points eventually lie inside the object.

Wide baseline

The process of matching features across views, particularly static features, is made difficult by the large perspective deformation. Automatic matching processes often rely on dynamic features. Cai et *al.* [54] extend a monocular tracking system and start by a single view tracking and then perform a camera handoff when the algorithm anticipates that the current sensor is about to stop presenting a good view of the moving object. The matching process is conducted by computing the Euclidean distance between a point and its corresponding epipolar line. Khan et *al.* propose a method for object correspondence with multiple cameras [63]. A homography between views is calculated and, for the

correspondence, a training phase is conducted during which one person enters and exits cameras fields of view. A limitation of this approach is presented by scenarios in which a person enters from the bottom of the image. The ground location cannot be simply computed by taking the bottom point of a detected blob. Since the ground location of objects is extracted at the camera handoff, such a scenario will generate false correspondences because the second assumption of the method is not verified anymore.

2.3.3 Appearance-based approaches

They use colour to match objects across cameras. Kang et al. [42] use colour information and a joint probability data association filter for tracking football players. Nummiaro et al. [40] propose a coloured-based object tracking approach with a particle filter implementation in a multi-camera environment. Kim et al. [64] use TV-broadcasted images and a tracking method based on template matching and on histogram back-projection to solve the occlusion problem. Orwell et al. [65] propose a multiple object tracking algorithm in multiple views using appearance models (colour). In their approach, they connects blobs extracted with background subtraction based on colour histogram techniques and use these blobs to match and track objects across views. Appearance-based methods generally suffer from illumination variations that undermine colour effectiveness as a cue. Also, colour information alone does not suffice to disambiguate elements of a group such as members of a team in a sport scene

2.3.4 Geometry-based approaches

These methods establish correspondences between objects appearing simultaneously in different views. These approaches generally exploit epipolar geometry, homography and camera calibration. Junejo et *al.* [66] propose a method that rectifies trajectories and models people's paths. Using a non-linear approach, cameras are calibrated during an unsupervised training and trajectories are rectified. Prototype path models are built

from these trajectories and a similarity measure is used to match input trajectories to path models. Iwase et *al.* [67] use eight cameras covering the penalty area and integrate the tracking data of football players from multiple cameras by using homography and a virtual ground image. An extension of point distribution models is proposed by Meneses et *al.* [68] to analyse object's motion in their temporal, spatial and spatio-temporal dimensions. Motions are expressed in terms of modes and associated to particular behaviour. Methods based on pure geometric constraints rely heavily on the accuracy during the correspondence process. For example, epipolar geometry suffers from ambiguity generated by the point-to-line correspondence [6].

Other works have adopted geometric approaches in object correspondence by using homographic transformations or epipolar geometry. Particularly, techniques using grids of space occupancy have recently been the object of a series of works [69, 62, 12, 70, 71]. Although these approaches rely on strong geometric constraints, thus allowing more accuracy, they are limited by the need for camera calibration because the information fusion takes place in 3D space.

In their work, Khan et *al.* [72] propose a homographic-based technique to fuse data from multiple sources and address the occlusion problem by localizing humans on multiple scenes. Basically, they localize, on every image plane, the locations of points that are likely to be occupied by objects. Successively, the locations found are used to solve occluded scenes. The advantage of this method resides in its synergetic approach where evidence is gathered from all cameras into one framework and only then detection and tracking is performed simultaneously. The results of these processes are then back-projected to each view. However, this approach fails when a human does not belong to the high foreground likelihood areas in the image plane (single view) as this situation causes missed detections (false negatives). One typical case is that of a static occlusion where a person is occluded by a part of the background itself (a building or a tree for example). A second case is when a region of the scene is occluded in all views by moving objects,

resulting in false positives. A dramatic drop of performance is noticed with an increased number of people in the scene. This approach then has difficulties in handling cases of merges and splits and would require a higher number of views to detect the empty spaces between people.

Kelly et al. use voxels to build a 3D environment model [73]. People are modelled as collections of voxels to handle the camera-handoff problem. Jain et *al.* [74] resort to calibrated cameras to extract objects' 3D locations in an environment model for the Multiple Perspective Interactive Video. These works, like [75], characteristically use environment models and calibrated cameras.

2.3.5 Hybrid methods

These methods use multiple features to integrate the information in the camera network. Sheikh et *al.* [38] present a statistical approach to associate trajectories across multiple views from airborne cameras. They assume the availability of a ground plane and a minimum duration during which at least one object is observed by two cameras. Taking as input the timestamped trajectories from each view, the algorithm estimates the inter-camera transformations, the objects' associations across views and the canonical trajectories. These are considered to be the best estimates in the maximum likelihood sense. Figueroa et *al.* [76] present a multi-camera tracking algorithm that uses a graph representation to find the positions of the football players on the pitch. Misu et *al.* [77] propose the integration of multiple features from multiple views (e.g texture, colour, region and motion). The trajectories are updated by back-projecting the 2D observations of the features and weighting them adaptively to their self-evaluated reliability. An algorithm that combines particle filtering and belief propagation in a unified framework is presented by Wei et *al.* [78]. Local particle filtering trackers interact with each other via belief propagation and compensate for poor individual observations. This algorithm is restricted to overlapping areas and relatively short time duration of occlusions. Busnell et *al.* [79]

investigate multiple object tracking and associate trajectories in a deterministic way. The distributed environment is built as a sparse connected graph and each vertex is a binary sensor. The connection between two nodes, hence the presence of an edge between two sensors, is established if an object can pass from one sensed area to another without triggering the activation of any other remaining sensor. Kim et *al.* [80] adopt both the appearance and homography approach to segment and track multiple persons. They use humans' centre vertical axes to recover their ground location. They solve the increase in state space, they incorporate an iterative segmentation-searching into a particle-filtering framework. However, occurrences of occlusions and similarity in the appearances of the people involved are difficult to handle.

Nakazawa et *al.*build a state transition map linking areas that belong to one or more sensors' field of view, along with a set of action rules to integrate information between different cameras. In their work [42], Kang et *al.*propose an approach where the tracking algorithm is performed in both image plane and the top view scene in a synergy of appearance and motion models.

Eshel et *al.* present a multi-view tracking in dense crowd [81]. They detect people's heads and the tracking is performed using assumptions on consistency of motion direction and velocity. The support point candidates are detected by plane image alignment with homography and intensity correlation. However, this method fails with the growing size of the crowd.

Other approaches use Bayesian networks to deal with object detection and tracking in multiple views. Chang's work [82] adopts a hybrid approach using Bayesian networks that combine a purely geometric approach (epipolar geometry, planar homographies and scene landmarks) with appearance (height and appearance) based modalities to match objects across different views. Falling in the same category of Bayesian networks, Dockstader et *al.*[50] propose an object tracking method that resolves occlusions across multiple calibrated cameras.

2.4 Homography estimation

Several applications have used homographic transformations to solve the integration of information from multiple views. Park et *al.* [83, 84] propose a framework that combines multiple view information with contextual domain knowledge with the purpose of analysis and query of person and vehicle interactions. This is applied to intelligent transportation to enhance pedestrian safety and situation awareness. Other applications are listed below:

- Object Tracking and Event Detection with multiple cameras: homography is used to infer transformations between planes to tackle occlusion, obtain an extended field of view and increase accuracy measurements [42, 44, 2, 46].
- Metric Rectification: homography is applied to obtain the fronto-parallel view of objects from a projectively distorted image. This relates to applications in modelling structured environments and recognition of road billboards with known patterns [85].
- 3D reconstruction: in this application, a projective reconstruction of the imaging geometry is obtained by applying projective transformation [86].
- Image registration and mosaicing: this applications involves creating a large view from images aligned by a homography [87].
- Automotive: these applications include vision-based control for cars in driving assistance and visual servoing. These involve computing the relative positions of vehicles with respect to a fixed object or a leading moving car. A camera captures the transformed image of the perimeter contour of a road sign. The homography relates a known contour in template image to the same contour in the captured image [88, 89, 90].
- Automatic system in disaster prevention: the activities and interactions of people and vehicles are analysed for situational awareness. These applications exploit planar homography constraints to extract footage area and objects velocity [91].

Algorithm 1 Homography estimation from textural information

Goal : Compute the homography between the two transformed views of the same planar texture.

1: Compute the DFT of both images.
2: Transform the images to the $\alpha\beta$ space.
3: Recover the translation by looking for an impulse in the inverse Fourier Transform of the two cross power spectrum of the $log\alpha\beta$ space representation of the two images which gives us the non-linear scale factors ψ_1 and ψ_2
4: In the Fourier domain recover the upper 2×2 minor of H^{-1} in a linear least squares manner.
5: Recover the translation component from the location of the impulse in the inverse Fourier transform of the cross power spectrum of the original image $f(.)$ and $g_t(.)$.
6: Use the Levenberg-Marquardt algorithm to compute the projective components of H.

To estimate homography, algorithms are mainly categorized in two methods [92]. Namely, texture-based and geometric approaches.

2.4.1 Texture-based methods

The texture-based approach [93, 94, 95] exploits the image intensities and works in the Fourier domain to compute the image-to-image transformation. The identification, selection and extraction of corresponding primitives in two images limits the applicability of spatial-based approach [96]. Instead, texture-based methods make use of image intensities for computing the image-to-image transformation and therefore present the advantage of exploiting the information over the whole image. However, the drawback is that most methods related to this last group are often limited to affine transformations or do not operate on wide baseline views.

Texture-based methods transform an image into the frequency domain and have the advantage of exploiting information provided in the intensities of the whole image. Unlike their spatial counterparts, they do not require an explicit point-to-point correspondence. Therefore, they avoid the critical issue of primitives identification and extraction in the corresponding images. Images are transformed to the Fourier domain and represented in a coordinate system where the homography is reduced to an anisotropic scaling. These methods work under the hypothesis of homographies being approximated by a subgroup of affine transformations, as illustrated by **Algoritm 1**. This subgroup present in-plane

rotation, translation and scaling. Most texture-based techniques are limited to this case of similarity transformations. Kruger et *al.* [96] perform image registration using multiresolution approach under affine transformations. They use the Fourier Transform of image patches and apply affine transformation while performing minimal line correspondences and affine homographies. Lucchese et *al.* present an affine transformation between images from a planar scene [93, 95]. They make use of radial projections of the energy and then to estimate the transformations, they resort to a series of non-linear optimizations techniques. In techniques that are based on tonal information in corresponding views, the anisotropic scale factors that characterize subgroup of affine transformations are computed using cross correlation methods. The algorithms work backwards from these methods until the entire affine transformation is computed. The relationship between textures is used to recover planar homographies since it only depends on the matching points and their effect in the frequency domain.

2.4.2 Geometric-based methods

These methods ([7, 97, 98]) are represented by the spatial-based approach. Techniques related to this approach solve the problem of correspondence by extracting and matching primitives (points, lines, conics or algebraic curves) across views.

The homography between two views is defined up to scale and can be estimated by finding sufficient constraints to fix the degrees of freedom that determines the transformation matrix. Homography is estimated by identifying and extracting geometrical primitives (points, lines, conics or algebraic curves) or conics present in corresponding views. Most methods make use of the Direct Linear Transformation (DLT) or other similar algorithms [6, 99]. The renormalization technique [100, 101, 102] is proposed as a step forward to enhance the numerical stability since most linear methods are particularly sensitive to the accuracy of the correspondence as well as to the condition numbers of the matrices. Robustness is also introduced by using standard techniques like Maximum

Likelihood Estimates and RANSAC [6]. These statistical methods improve the robustness of those algorithms against noise in the correspondence process. Techniques based on geometrical primitives such as polygons are also used [103, 104]. They employ projective invariants such as cross-ratios to approximate a contour by growing a polygon that is identified in two corresponding views. The polygonal approximation is then used in order to estimate the homography. Here below are reported some techniques that resort to the spatial approach and is followed by applications in related works:

- Points

Algorithm 2 Direct Linear Transformation on points correspondence

Goal : Given $n \geq 4$ $2D$-to-$2D$ point correspondences $x_i \leftrightarrow x_i^{'}$, determine H the $2D$ homography matrix such that $x_i^{'} = Hx_i$.

1: For each correspondence $x_i \leftrightarrow x_i^{'}$ compute L_i Only two first rows needed.
2: Assemble n $2x9$ matrices L_i into a single $2nx9$ matrix L_i
3: Obtain SVD of L as UDV^T diagonal with positive diagonal entries, arranged in descending order down the diagonal, then h is last column of V.
4: Determine H from h.

Homography is recovered as a relationship on point features. This method is applied to recover homography induced by the plane using at least four corresponding points to fix eight degrees of freedom. It uses simple and fundamental features such as points representing objects' ground location or the tip of pedestrians' heads. The method, illustrated in **Algorithm 2**, is subject to inaccuracies due to noise [6, 7, 105].

- Points with additional cues

  This methods use weak calibration and employ projective transformation. They need additional transformation (Fundamental Matrix) that in turn needs correspondence in its estimation [6]

- Lines

  Similarly to algorithms on points, these methods solve linear equations by numerical methods to perform projective transformation (DLT algorithm). They require four

Algorithm 3 Direct Linear Transformation on Lines correspondence

Goal : Given 4 or more corresponding lines $l_i \leftrightarrow l_i^{'}$, determine H the $2D$ homography matrix such that $l_i^{'} = \left(H^{-1}\right)^T l_i$.

1: For each correspondence $l_i \leftrightarrow l_i^{'}$ compute L_i. Only two first rows needed.
2: Assemble n $2x9$ matrices L_i into a single $2nx9$ matrix L_i
3: Obtain SVD of L as UDV^T diagonal with positive diagonal entries, arranged in descending order down the diagonal, then h is last column of V.
4: Determine H from h.

line correspondences to be found, thereby solving eight degrees of freedom. The advantage of this approach is that lines are easier to detect than points [6] and more robust to noise than points. However, it should be pointed out that the performance of these approaches depends on the image coordinate system. Particularly, situations where a detected line is located close to the origin of the selected image coordinate system, tend to generate instability in the homography estimation. Conversely, a point-based estimation still performs well in those cases. **Algorithm 3** illustrates the line correspondence method.

- Mixture Points and Lines

Algorithm 4 Direct Linear Transformation of Points and Lines

Goal : Given 4 or more corresponding elements (points or lines) compute the homography between the two images.

1: For each correspondence $l_i \leftrightarrow l_i^{'}$ or $x_i \leftrightarrow x_i^{'}$ compute L_i. Only two first rows needed.
2: Assemble n $2x9$ matrices L_i into a single $2nx9$ matrix L_i
3: Obtain SVD of L. Solution for h is last column of V.
4: Determine H from h.

These methods solve linear equations by numerical techniques in projective transformation (DLT algorithm). The homography is computed from three points and one line or three lines and one point, thereby solving eight degrees of freedom. The drawback is that these methods require a high level of accuracy in the correspondence. However, they allow more flexibility in feature extraction. It is noteworthy to point out that a two lines and two points combination leads to a degenerate case [6, 105]. This approach is shown in **Algorithm 4**

- Conics

Algorithm 5 Homography from pair of conics

Goal : Given 2 corresponding conics, compute the homography between the two images.

1: Obtain the equations of the conics in both views.
2: Rectify both views assuming that the conics are images of circles. The results are correct even if this is not so. Let H_1 and H_2 be the rectifying homographies.
3: Calculate the similarity transform H_s between the two rectified views using two point correspondences obtained by finding the centers of the two circles.
4: The homography between the two views is obtained as $H_1 H_s H_2^{-1}$

This approach uses projective invariants. The homographic transformation can be computed from two pairs of corresponding conics as shown in **algorithm 5** by presenting additional geometric constraints, making the correspondence more robust [106, 104, 103].

- Planar Algebraic curves

To compute the homography, first the intersection between the curves with the Hessian curve is computed. This latter is given by the determinant $\mathcal{H}$:

$$\mathcal{H} = \left| \frac{\partial^2 f}{\partial x_i \partial x_j} \right| = 0, \tag{2.1}$$

where $f(x_1, x_2, x_3) = 0$ is the equation of the curve. Solutions are normalised and points minimising the Hausdorff distance are selected as presented in **algorithm 6**. Particularly useful in scenes rich with man-made objects [107]

Algorithm 6 Homography from algebraic curves

Goal: Given a pair of cubic or higher order curves, compute the homography between them.

1: Compute the Hessian curves in both images following Eq.2.1.
2: Compute the intersection of the curve with its Hessian in both images. The output is the set of inflexion and singular points.
3: Discriminate between inflexion and singular points by the additional constraint for each singular point $\nabla f\,(a)$.
4: Separate the real points from the complex points.
5: Find the solution to $\mathcal{H}$ that makes S the closest to zero or minimizes the Hausdorff distance between the sets of points.

- Non-algebraic curves

Algorithm 7 Homography from non-algebraic curves

Goal: Estimates the homography between 2 views given the Fundamental Matrix and the perspective images of the plane curve and single corresponding image points.

1: Compute curvature and tangents at x and $x^{'}$.
2: Compute the one-parameter family of the homography by the tangents of the curve at x and $x^{'}$.
3: Choose $A = \left[e^{'}\right] \times F$.
4: The parameter μ can be determined using the curvature at points x and $x^{'}$ and A
5: Compute H.

Algorithm 7 shows the estimation of homographic relationship from non-planar curves. The Euclidean curvature is mapped by a homography. The transformation is uniquely defined from corresponding tangents and curvature at one point. The method requires the use of epipoles and can present two types of degenerate cases: epilolar tangents and inflections [108, 109].

2.4.3 Trajectory mapping using homography transformation

The homography maps trajectory points lying on a plane in one view onto points on the same plane imaged on another view, from a set of known corresponding control points in the two images. Based on these correspondences, the homography matrix H, is estimated as

$$\mathbf{x}' = H\mathbf{x}, \tag{2.2}$$

where $\mathbf{x} = (x, y, 1)$ is a point in the first view in homogeneous coordinates and $\mathbf{x}' = (x', y', 1)$ is the corresponding point in the second view. Eq. (2.2) can therefore be expressed as

$$\begin{cases} x' = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}} \\ y' = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}} \end{cases}, \tag{2.3}$$

where the unknown, h_{ij}, are entries of the matrix

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}. \tag{2.4}$$

H is therefore the matrix whose parameters are to be estimated from the set of corresponding control points $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$ in two images. Most works [8, 110, 63] compute H using linear methods using the Direct Linear Transformation (DLT), a widely used linear method based on **Algorithm 2**. To derive a linear solution for H, Eq. (2.2) is rewritten under a vector cross product form as

$$\mathbf{x}' \times H\mathbf{x} = 0, \tag{2.5}$$

If the j-th row of the matrix H is indicated as $\mathbf{h^{j\mathcal{T}}}$ then

$$H\mathbf{x_i} = \begin{pmatrix} \mathbf{h^{1\mathcal{T}}x_i} \\ \mathbf{h^{2\mathcal{T}}x_i} \\ \mathbf{h^{3\mathcal{T}}x_i} \end{pmatrix}. \tag{2.6}$$

The superscript $\mathcal{T}$ indicates the transpose. Writing $\mathbf{x'_i} = (x'_i, y'_i, z'_i)^{\mathcal{T}}$, the cross-product can be written as

$$\mathbf{x}' \times H\mathbf{x} = \begin{pmatrix} y'_i\mathbf{h^{3\mathcal{T}}x_i} - \mathbf{h^{2\mathcal{T}}x_i} \\ \mathbf{h^{1\mathcal{T}}x_i} - x'_i\mathbf{h^{3\mathcal{T}}x_i} \\ x'_i\mathbf{h^{2\mathcal{T}}x_i} - y'_i\mathbf{h^{1\mathcal{T}}x_i} \end{pmatrix}. \tag{2.7}$$

Because $\mathbf{h^{j\mathcal{T}}x_i} = \mathbf{x_i^{\mathcal{T}}h^j}$ for $j = 1, ..., 3$, this yields a set of three equations in the

parameters of H, that can be expressed as

$$\begin{bmatrix} \mathbf{0}^{\mathcal{T}} & -\mathbf{x_i}^{\mathcal{T}} & \mathbf{y'_i}\mathbf{x_i}^{\mathcal{T}} \\ \mathbf{x_i}^{\mathcal{T}} & \mathbf{0}^{\mathcal{T}} & -\mathbf{x'_i}\mathbf{x_i}^{\mathcal{T}} \\ -\mathbf{y'_i}\mathbf{x_i}^{\mathcal{T}} & \mathbf{x'_i}\mathbf{x_i}^{\mathcal{T}} & \mathbf{0}^{\mathcal{T}} \end{bmatrix} \begin{pmatrix} \mathbf{h^1} \\ \mathbf{h^2} \\ \mathbf{h^3} \end{pmatrix} = 0. \tag{2.8}$$

These equations presents the form $A_i\mathbf{h} = 0$. The key idea in this method is that given pairs of corresponding pixels, $A_i\mathbf{h} = 0$ is linear in the unknown $\mathbf{h}$, whereas the entries of A_i are quadratic in the known coordinates of the points. This means that given enough equations, it is possible to implement linear algebra methods to compute the coefficients of H. A_i is a $3x9$ matrix defined up to a scale with 8 degree of freedom and $\mathbf{h}$ is a 9-vector made up of entries of H. Each point corresponding pair $\mathbf{x_i}, \mathbf{x_i'}$ account for two constraints because for every x_i in the first image the two degrees of freedom of the point in the second image has to correspond to the transformed point $H\mathbf{x_i}$. A point on the plane has two degrees of freedoom that correspond to its components, (x, y). Despite Eq. 2.8 presents three equations, only two among them are linearly independent. That is because the third row is obtained, up to a scale, from the sum of x_i' times the first row and y_i' times the second). Therefore, every corresponding pair generates two equations in the entries of H. In estimating H, the third row can be omitted and the set of equations in 2.8 is then written as

$$\begin{bmatrix} \mathbf{0}^{\mathcal{T}} & -\mathbf{x_i}^{\mathcal{T}} & \mathbf{y'_i}\mathbf{x_i}^{\mathcal{T}} \\ \mathbf{x_i}^{\mathcal{T}} & \mathbf{0}^{\mathcal{T}} & -\mathbf{x'_i}\mathbf{x_i}^{\mathcal{T}} \end{bmatrix} \begin{pmatrix} \mathbf{h^1} \\ \mathbf{h^2} \\ \mathbf{h^3} \end{pmatrix} = 0. \tag{2.9}$$

Now, A_i in the $A_i\mathbf{h} = 0$, is a $2x9$ matrix in Eq. 2.9. Each corresponding pair generates two equations in the entries of H. With a set of four corresponding pairs, we have a set of equations $A\mathbf{h} = 0$. A is the matrix of equations coefficients built from the matrix rows A_i contributed from every correspondence. $\mathbf{h}$ the vector of unknown entries of H such as

$$\begin{pmatrix} \mathbf{h^1} \\ \mathbf{h^2} \\ \mathbf{h^3} \end{pmatrix} \tag{2.10}$$

In general, more than 4 four corresponding pairs are used to estimate the transformation matrix and this leads to the expression $A\mathbf{h} = \mathbf{0}$ being an over-determined system. In the event of the location of the corresponding points being exact, then A is of rank 8, a 1-dimensional space and there exists an exact solution for h. However, there is no exact solution, when the measurement of the image coordinates is not exact, that is, noisy. Therefore, instead of an exact solution, solving for h will lead to an approximated solution that minimises a appropriate cost function. The goal is to determine a non-zero solution $\mathbf{h}$ up to a scale factor. In order to avoid the null solution, linear estimation methods based on DLT algorithm add an additional constraint on the norm by setting $\|\mathbf{h}\| = 1$. However, because there is no exact solution to $A\mathbf{h} = 0$, the norm $\|A\mathbf{h}\|$ is minimised instead of the constraint $\|\mathbf{h}\|$. Research [6, 111] showed that this yields to the same as minimising the algebraic residuals in the expression $\|A\mathbf{h}\| \,/\, \|\mathbf{h}\|$. The solution is the unit eigenvector corresponding to the smallest eigenvalue of $A^{\mathcal{T}}A$. This eigenvector can be obtained directly by the Singular Value Decomposition (SVD) of A. If the vector $\xi = A\mathbf{h}$ is the residual vector, then its components are derived from the individual correspondences that generate

Algorithm 8 Homography estimation using the Direct Linear Transformation (DLT) algorithm

Goal: Given $n \geq 4$ $2D$ point correspondences $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$, determine the $2D$ homography matrix H such that $\mathbf{x'_i} = H\mathbf{x_i}$.

1: For each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$, compute the matrix A_i. Only the first two rows need to be used, in general.
2: Assemble the n 2 x 9 matrices A_i into a single $2n$ x 9 matrix A .
3: Obtain the SVD of A. The unit singular vector corresponding to the smallest singular value is the solution h.
4: The matrix H is determined from h.

each row of A. Each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$ contributes a partial error vector ξ_i (algebraic error vector), toward the full error vector ξ, whose norm is the algebraic distance.

Although the DLT algorithm (**Algorithm 8**), based on SVD, has the advantage of easy implementations, due to its linearity and simplicity, it is quite sensitive to noise. Additionally, one other cause of error in the matrix computation lies in the value being minimized as it does not account for the noise in the geometry of the corresponding points.

2.5 Summary

Despite considerable efforts in addressing problems related to object tracking, single-view approaches show limits in tackling occlusion, maintaining continuity of trajectories and achieving accuracy in the measurements of objects' spatial location. Multiple view approaches may help overcome these limitations. However, these approaches still suffer from other hindrances: they heavily rely on features such as shapes and appearance templates, whose integrity is not always preserved during the various stages of extraction processes. Cases where, for example, subjects wear similar colours tend to confuse these algorithms. The use of objects' appearance (shape, contour or colour distributions) can produce poor segmentation and detection. Additionally, the main idea behind occlusion solving techniques is essentially built upon the assumption of temporal consistency of the adopted motion model, namely, Kalman filtering or the family of Markov models. Geometric approaches, exploiting the relationship between objects' trajectories from multiple views, rely on the spatio-temporal constraints that remain more stable than the features extracted in appearance-based models. However, the linear method, commonly used in estimating the homographic transformations between trajectories presents limitations. Although this method yields non-iterative computation methods that are easy to implement using linear algebra packages, it is found to be sensitive to noise even with numerous corresponding points [4]. To address this issue, we present in the next chapter, a robust

algorithm that transforms trajectories from multiple views into one common global view, while reducing homography errors.

Chapter 3

Multi-view trajectory transformation and fusion

3.1 Introduction

In this chapter, we present the proposed approach and demonstrate the different techniques we use to generate the final integrated information on one global view. This latter is the synthesis of the partial information provided by the cameras. The proposed approach is presented here in its main parts which constitutes the contribution of the present work:

- The generation of control points obtained by sampling the time series consisting of the spatio-temporal locations of objects on the image plane of each camera.
- The trajectory transformation and the embedded lens distortion correction that map the trajectory points into the common view.
- The trajectory reconstruction that gives the final complete tracks of moving objects across the different views.

Hereinafter, we assume the availability of the motion segmentation results from the video sequences. To this end, we use background subtraction [112] to extract fore-

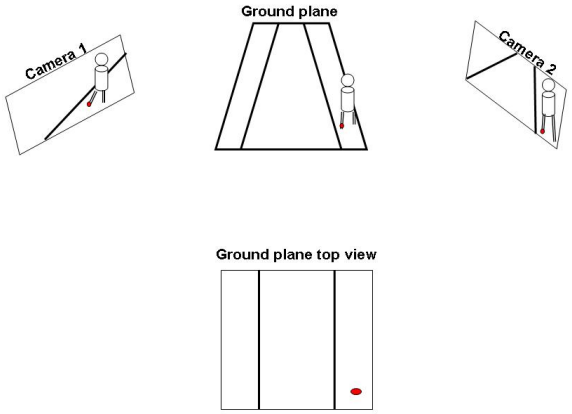


Figure 3.1: An object of interest is observed in two cameras. The ground location is calculated and the mapped into a top view scene by homographic transformations.

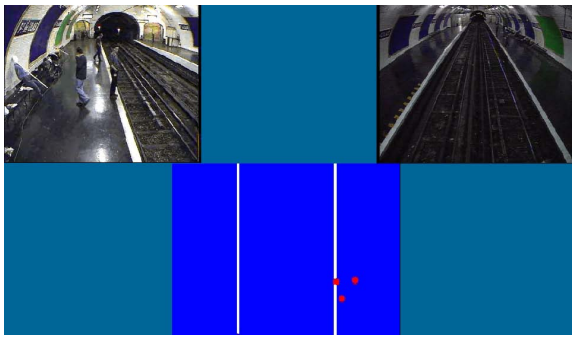


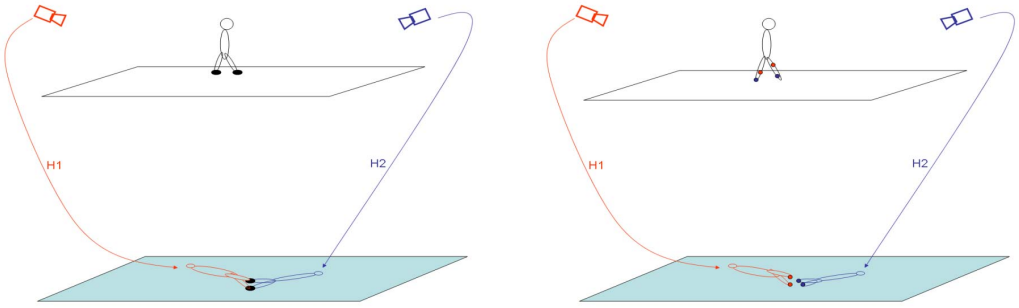
Figure 3.2: Example of a homography-based transformation on real data in an underground scene. The red dots indicate people's spatial locations on a scene top view.

ground objects and then graph matching [16] to obtain the related tracks.

3.2 2D homography-based trajectory transformations

3.2.1 Control points extraction for homography estimation

The algorithm of trajectory transformation from an image plane to a common view, considers planar homographies as presented in the previous chapter (Sect. 2.4). In this work, we assume a wide baseline multi-view set-up (typical in surveillance and sport scenes scenarios) and suppose the distance between moving objects and cameras is far enough to assume that the objects move on a dominant plane. This is often the case in surveillance systems and moreover in sports scenes where the dominant plane is the

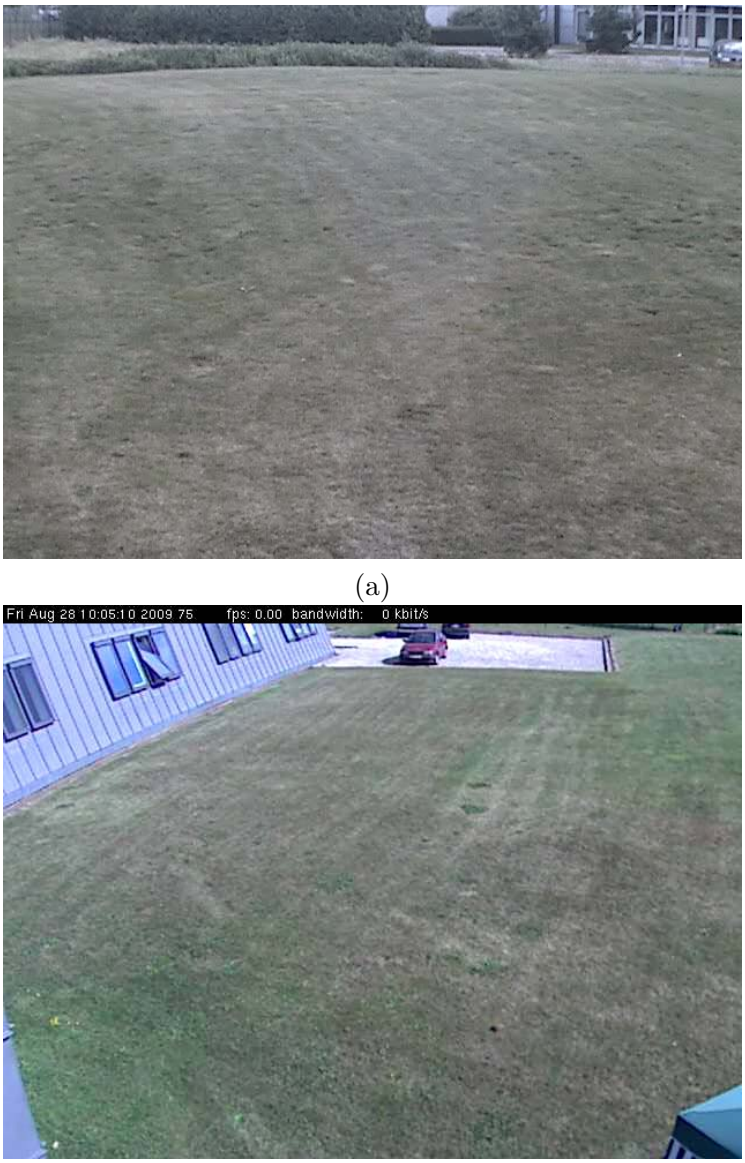


(a) (b)

Figure 3.3: Example of the mapping of an object ground location being affected by off-plane errors due to faulty object detection. (a) Correspondence that satisfies homography constraint; (b) off-plane error in the correspondence due to errors in computing the object's ground location.

ground plane; the football pitch on which players move is approximated with a geometric plane. When an object lying on that plane is simultaneously visible from multiple views, these views are related by a unique homographic transformation. Figure 3.1 sketches a scene where the ground location of a human (red dot), is represented as a pixel which we consider as a geometric point on the planar surface the object is moving on. The object is observed in two different cameras and its ground location is mapped onto an image model of the scene top view by homographic transformations. An example of this mapping, on a real data, is shown in Fig. 3.2 which shows an underground scene with three people on a platform. Their positions are mapped onto a scene top view. In practice, the primitives detected in images are likely to be too noisy to get good solutions when only using the strict minimum numbers indicated. It is often necessary to obtain a larger number of features in order to make the solution more robust [6]. Additionally, the mapping of an object's ground location is often affected by off-plane errors due to faulty object detection as sketched in Fig. 3.3. Ideally, the ground location from the two cameras should match as in Fig.3.3(a). However, as shown in Fig.3.3(b), the ground location computed from each camera image plane can lead to misalignment after mapping on a common view.

One problem remaining is to determine the control points from which to estimate the

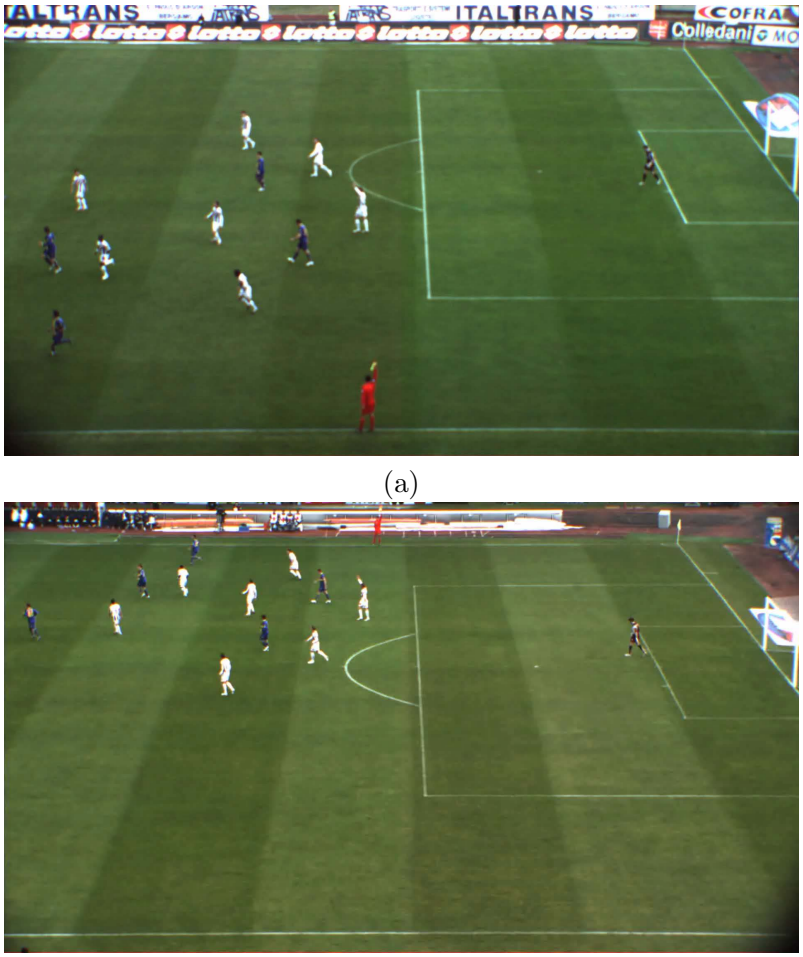


(a)

(b)

Figure 3.4: Example of corresponding views with no landmarks in the overlapping area.

homographies. These are corresponding points across multiple views -minimum number of four- and are used as parameters in the homography estimation. There are two categories of algorithms in feature extraction for homography estimation: static and dynamic feature-based. The first category includes methods which extract features such as Harris Corners [113], SIFT [114], or MOPS [115]. Features from the first image are matched to features from the second image and methods like Ransac [116], are used to select the



(a)

(b)

Figure 3.5: Example of corresponding views with wide baseline. The use of SIFT fails in this case.

inliers before fitting a homography. However, one main problem with these approaches is that while they are applicable to rotation, translation, small perspective distortions and in general with small baseline. They fail in the case of cameras with a wide baseline and accentuated perspective distortions. Additionally, the exhaustive search to find all possible corresponding pairs with Ransac can lead to high computational costs because of the brute force approach. Figures 3.4 and 3.5 are examples where static feature-based algorithms fail to extract control points. Figure 3.4 presents no easily detectable landmark on the ground plane, making difficult the process of extracting static features. Figure 3.5 instead presents wide baseline views and the use of SIFT fails in this case. To circumvent these

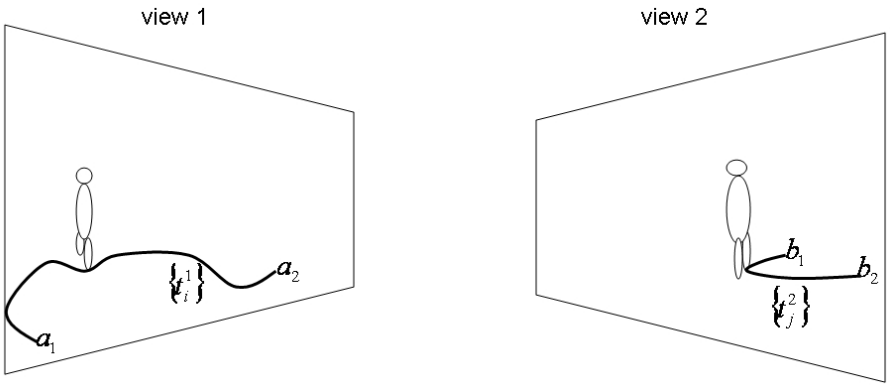


Figure 3.6: Control points from objects' trajectories in multiple views.

limitations, the second category of algorithms [117, 118] uses dynamic features. That is, extracting features that form the set of control points, from the objects' trajectories.

Our method is closely related to this last category and uses time line constraint to obtain the control points. The control points are extracted from the rich set of information provided by the positions of the moving objects over time. To achieve this, we assume constant the frame rates of the video sequences and consider the trajectory of one object simultaneously visible in the multiple views.

The steps that describe the method we proposed for control points extraction are summarised as follows:

- We determine the segment of trajectories which belong to the overlapping area. If a_1a_2 and b_1b_2 are the trajectories of an object visible in two views (Fig. 3.6), where the subscripts indicate the ends of the segments. We extract $a_{t_m}a_{t_n}$ and $b_{t_m}b_{t_n}$, the sub-segments of a_1a_2 and b_1b_2, respectively. For each segment, the first subscript represents the time at which the object appears in the overlapping area of the images and the second is when it disappears from it.

- We subsample points from the segments of the trajectories above to avoid collinearity in the set of control points as this might lead to a degenerate case during the homography estimation.

(a) (b)

Figure 3.7: Examples of control points extraction in ETISEO dataset. The control points are sampled from the object's trajectory points in the overlap between (a) *view*1 (yellow) and (b) *view*2 (red).

3.2.2 Homography estimation

As pointed out in Chap. 2, Sect. 2.4.3, the widely used linear methods of estimating homography transformation suffers from two main hindrances [7, 4]:

- it is quite sensitive to noise
- the value being minimized in the matrix computation does not account for the noise in the geometry of the corresponding points. In fact, it does not have any geometric interpretation.

To overcome these problems, we use the theory behind the renormalization technique to attain the theoretical accuracy bound in geometry fitting [119]. In fact, it is demonstrated that higher order errors give more accurate estimates [119].

Control points uncertainty

We start by considering the trajectory points as random variables. The uncertainty of data points (x_α, y_α) and (x'_α, y'_α) is described by their covariance matrices Σ_α and

$\Sigma_\alpha^{'}$. It follows that the vectors $\mathbf{x}_\alpha$ and $\mathbf{x}_\alpha^{'}$ have the following singular covariance matrices:

$$V\left[\mathbf{x}_\alpha\right] = \frac{1}{f^2}\begin{pmatrix} \Sigma_\alpha & 0 \\ 0^\mathcal{T} & 0 \end{pmatrix} \tag{3.1}$$

and

$$V\left[\mathbf{x'}_\alpha\right] = \frac{1}{f^2}\begin{pmatrix} \Sigma_\alpha^{'} & 0 \\ 0^\mathcal{T} & 0 \end{pmatrix}, \tag{3.2}$$

where f is a scale factor. Because it is difficult to predict the uncertainty of every single point in advance, let us assume only the relative tendency of noise occurrences is known. This means, the covariance matrices are known up to a scale such that $V\left[\mathbf{x}_\alpha\right] = \epsilon^2 V_o\left[\mathbf{x}_\alpha\right]$ and $V\left[\mathbf{x}'_\alpha\right] = \epsilon^2 V_o\left[\mathbf{x}'_\alpha\right]$, where ϵ is the noise level. The normalized covariance matrices $V_o\left[\mathbf{x}_\alpha\right]$ and $V_o\left[\mathbf{x}'_\alpha\right]$ indicate the relative dependence of noise occurrence on positions and orientations. Assuming isotropy and homogeneity leads to the default values $V_o\left[\mathbf{x}_\alpha\right] = V_o\left[\mathbf{x}'_\alpha\right] = diag\left(1,1,0\right)$. Let $\hat{H}$ be an estimate of the homography matrix and $\bar{H}$ the true value. Therefore, we can measure the covariance tensor that describes the uncertainty of the estimate $\hat{H}$ by

$$V\left[\hat{H}\right] = E\left[T\left(\left(\hat{H}-\bar{H}\right)\otimes\left(\hat{H}-\bar{H}\right)\right)T^\mathcal{T}\right], \tag{3.3}$$

where $E\left[.\right]$ indicates the expectation, T is the tensor and $\otimes$ the tensor product. The homography matrix, made of 9 elements, is normalised to have unit norm and therefore can be represented as a point lying on a sphere S^8 of dimension 8 in the 9-dimensional parameter space $\mathcal{R}^9$. The tensor T projects the deviation $\hat{H}-\bar{H}$ onto the tangent space $Tg_{\bar{H}}(S^8)$ at $\bar{H}$. The $(ijkl)$ element of tensor T is $T_{ijkl} = \delta_{ik}\delta_{jl} - \hat{H}_{ij}\hat{H}_{kl}$, where δ_{ij} is the Kronecker delta. The root mean error e_{rms}, $(0 \leq e_{rms} \leq 1)$, over the estimation of $\hat{H}$ is

$$0 \leq e_{rms}\left(\hat{H}\right) = \sqrt{E\left[\left\|T\left(\hat{H}-\bar{H}\right)\right\|^2\right]} \leq 1. \tag{3.4}$$

Theoretical bounds in the homography estimation

The problem can be now formulated as estimating H such that

$$\bar{\mathbf{x}}'_\alpha \times H\bar{\mathbf{x}}_\alpha = 0 \tag{3.5}$$

from noisy control points $\{\bar{\mathbf{x}}_\alpha\}$ and $\{\bar{\mathbf{x}}'_\alpha\}$.

By modeling the uncertainties in geometric inference suggested in [7], the theoretical accuracy bound is obtained as:

$$V\left[\hat{H}\right] \succ \epsilon^2 \left(\sum_{\alpha=1}^{N} \sum_{k,l=1}^{3} \bar{W}_\alpha^{(kl)} \left(e^{(k)} \times \bar{\mathbf{x}}'_\alpha\right) \otimes \bar{\mathbf{x}}_\alpha \otimes \left(e^{(k)} \times \bar{\mathbf{x}}'_\alpha\right) \otimes \bar{\mathbf{x}}_\alpha \right)_8^-, \tag{3.6}$$

$$\bar{W}_\alpha = \left(\bar{\mathbf{x}'}_\alpha \times \bar{H} V_o\left[\mathbf{x}_\alpha\right] \bar{H}^T \times \bar{\mathbf{x}'}_\alpha + \left(\bar{H}\bar{\mathbf{x}}_\alpha\right) \times V_o\left[\mathbf{x}_\alpha\right] \times \times \left(\bar{H}\bar{\mathbf{x}}_\alpha\right)\right)_2^-. \tag{3.7}$$

In Eq. (3.6), $S \succ C$ indicates that $S - C$ is a positive semi-definite symmetric tensor. The operator $(.)_q^-$ denotes the Moore-Penrose generalized inverse of rank q. The product $\mathbf{u} \times \mathbf{M} \times \mathbf{u}$, where $\mathbf{u}$ is a vector $\mathbf{u} = u_i$ and $\mathbf{M}$ a matrix $\mathbf{M} = M_{ij}$, is a matrix whose elements are $m_{ij} = \sum_{k,l,m,n=1}^{3} \epsilon_{ikl}\epsilon_{jmn}a_k a_m M_{ln}$. ϵ_{ijk} is the Eddington espilon and equals 1 or -1 if (ijk) is an even or odd permutation of (123) and 0 otherwise. The vectors $e^{(s)}$ are defined as $e^{(1)} = (1,0,0)^\mathcal{T}$, $e^{(2)} = (0,1,0)^\mathcal{T}$ and $e^{(3)} = (0,0,1)^\mathcal{T}$. The e_{rms} in the estimation is bounded by $e_{rms}\left(\hat{H}\right) \geq \sqrt{trV\left[\hat{H}\right]}$, where tr denotes the tensor trace.

Maximum likelihood in homography estimation

The optimal method, in the statistical sense of "maximum likelihood" attains the theoretical accuracy bound in its first order by minimising the squared Mahalanobis

Algorithm 9 Homography estimation with renormalization

Goal: Given two overlapping views, find the homography that attains the accuracy bounds derived from image noise model.

1: $c \leftarrow 0$
2: $W_\alpha \leftarrow I$; $\alpha = 1, \ldots, N$.
3: Compute the tensor M

$$M = \frac{1}{N} \sum_{\alpha=1}^{N} \sum_{k,l=1}^{3} W_\alpha^{(kl)} \left(e^{(k)} \times \mathbf{x}'_\alpha \right) \otimes \mathbf{x}_\alpha \otimes \left(e^{(l)} \times \mathbf{x}'_\alpha \right) \otimes \mathbf{x}_\alpha. \tag{3.8}$$

4: Compute the tensor $N = N_{ijkl}$ as:

$$N_{ijkl} = \frac{1}{N} \sum_{\alpha=1}^{N} \sum_{m,n,p,q=1}^{3} \epsilon_{imp} \epsilon_{knq} W_\alpha^{(mn)} \left(V_o \left[\mathbf{x}_\alpha \right]_{jl} \mathbf{x}'_{\alpha(p)} \mathbf{x}'_{\alpha(q)} + V_o \left[\mathbf{x}'_\alpha \right]_{jl} \mathbf{x}_{\alpha(j)} \mathbf{x}_{\alpha(l)} \right) \tag{3.9}$$

5: Compute the 9 eigenvalues, $\lambda_1 \geq \ldots \geq \lambda_9$, related to tensor $\hat{M} = M - cN$ and the corresponding orthonormal system of eigenmatrices $H_1, \ldots, H_9$ of unit norm.
6: **if** $\lambda_9 \approx 0$ **then**
7: stop
8: **else**
9: $c \leftarrow c + \frac{\lambda_9}{(H_9; N H_9)}$ and

$$W_\alpha \leftarrow \mathbf{x}'_\alpha \times H_9 V_o \left[\mathbf{x}_\alpha \right] H_9^T \times \mathbf{x}'_\alpha + (H_9 x_\alpha) \times V_o \left[\mathbf{x}'_\alpha \right] \times V_o \left[\mathbf{x}'_\alpha \right] \times ({}_9\mathbf{x}_\alpha) \tag{3.10}$$

10: **end if**
11: Back to step 3.

distance

$$J = \sum \left(\mathbf{x}_\alpha - \bar{x_\alpha}, V_o \left[\mathbf{x}_\alpha \right]_2^- \left(\mathbf{x}_\alpha - \bar{x_\alpha} \right) \right) + \sum \left(\mathbf{x}'_\alpha - \bar{x'_\alpha}, V_o \left[\mathbf{x}'_\alpha \right]_2^- \left(\mathbf{x}'_\alpha - \bar{x'_\alpha} \right) \right), \tag{3.11}$$

subject to the constraint Eq. (3.3). Using the Lagrange multipliers and retaining the first order approximation yields:

$$J = \sum \left(\mathbf{x}'_\alpha \times H\mathbf{x}, W_\alpha \left(\mathbf{x}'_\alpha \times H\mathbf{x}_\alpha \right) \right), \tag{3.12}$$

where W_α denotes the matrix

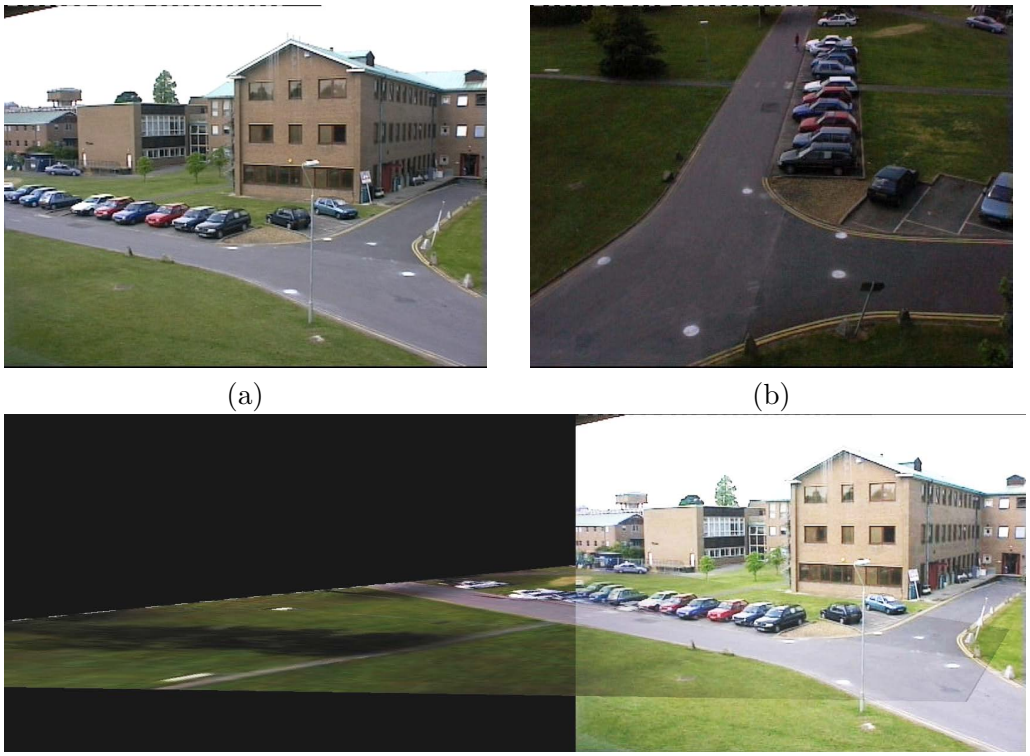
$$W_\alpha = \left(\mathbf{x}'_\alpha \times HV_o\left[\mathbf{x}_\alpha\right]H^T \times \mathbf{x}'_\alpha + (H\mathbf{x}_\alpha) \times V_o\left[\mathbf{x}_\alpha\right] \times \times (H\mathbf{x}_\alpha)\right)_2^- . \tag{3.13}$$

Let U denote the eigenmatrix of the resulting covariance tensor $V\left[\hat{H}\right]$ for the maximum eigenvector λ. U shows the orientation in the $9D$ space in which the error is most likely to occur. The solution H is perturbed along U in both directions such that

$$\begin{cases} H^+ = N\left[\hat{H} + \sqrt{\lambda}U\right] \\ H^- = N\left[\hat{H} - \sqrt{\lambda}U\right]. \end{cases} \tag{3.14}$$

Equation (3.14) indicates the deviation pair associated to the homography matrix H, where $N[.]$ is the normalization operator to a unit norm. The overall renormalization procedure is illustrated in **Algorithm 9**.

To visualise an application of the above algorithm, we generate the mosaics from overlapping views and create a larger cameras' field of view that represents the common coordinate frame where we map the objects' trajectories (Fig. 3.8). A mosaic allows uninterrupted observations of objects that enter and exit individual camera's fields of view. We proceed in a pairwise mode by first aligning the two images then by applying image stitching to composite the two images [120]. The alignment is obtained by warping one image onto the other, considered as reference view, using the estimated homography transformation. Although aligned, a simple juxtaposition of the two images would create visible photometric artifacts such as inconsistencies in pixel colours in the resulting mosaic. We apply image stitching by pixel selection and center-weighting [120]. We blend a pixel's colours in the overlapping area by interpolating the pixels' intensities in that region. Since we pursue a seamless merging, the colours of the pixels in the overlapping areas are weighted through averaging. For this purpose, we calculate the centers of the images and use them as coefficients to weight pixels' intensities in the overlap. Let I_1 and I_2 represent



(a) (b)

(c)

Figure 3.8: (a)-(b) Corresponding views with control points on overlapping area; (c) mosaic generated after homography estimation.

the pixel intensity in the first and the second image, respectively. Furthermore, let a_1 be the Euclidean distance between the center of the first image to the pixel [121]. The same computation is carried out for a_2 with respect to the second image. The intensities of pixels in the overlapping area are weighted through averaging. The resulting pixel intensity, I, of the composed image is given by

$$I = \left(\frac{a_1}{a_1 + a_2}\right) I_1 + \left(\frac{a_2}{a_1 + a_2}\right) I_2, \tag{3.15}$$

An example of homography estimation and the resulting mosaic after image alignment and stitching is shown in Fig. 3.8. The view illustrated in Fig. 3.8(b) is warped onto Fig. 3.8(a) which is the reference view. Colors in the overlap are blended and the mosaic is shown in Fig. 3.8(c).

Figure 3.9: Examples of radial lens distortions, where the image peripheries are particularly affected.

3.2.3 Homography estimation with lens distortion correction

The geometric constraints in the homography estimation presented in the section above, are valid when the model considered is the linear pinhole camera (Section 3.1). However, lens distortions represent a hindrance in computer vision applications because they introduce nonlinearity in the pinhole camera models. An extensive review on these distortions is presented by Slama [122]. There are different types of distortions, mainly the tangential and the radial distortion [123]. Radial distortion is the severest one for most cameras [124, 125]. Although the phenomenon does not impact on the quality of the image itself, it has however, an influence on the image geometry. Figure 3.9 illustrates this phenomenon.

In multiple views, matching control points across views to estimate homography can be undermined by lens distortions, in particular at the image periphery [105]. To solve this problem, some works on online distortion estimation (*plumb line*) often use straight

lines in the scenes to provide constraints on the distortion parameters [123, 126]. These methods assume the availability of such lines in the scene. In cases of unstructured scenes, or lines that are not easy to detect in a scene, the application of the aforementioned approach is difficult. Such methods work under the assumption that a straight edge, bent by lens distortion, will deviate from a fitted segment. An optimization is performed on the distortion parameters to minimise the deviation of edges from straight lines. A caveat for this method is the presence of real-world curves in the scene that might be wrongly straightened. Unlike this approach, Stein [127] requires neither the 3D location of points nor the camera calibration and uses point correspondences in multiple views to recover epipoles and epipolar lines considering lens distortion. A cost function, defined as the root mean square of the distances, is computed from the feature points to the epipolar lines. Zhang [128] describes the epipolar geometry between two images with lens distortions by matching a point to its correspondent on the other image. The corresponding point is considered as lying on a curve rather than a straight line as it is the case in a distortion-free camera. Swaminathan *et al.* [124] derive a metric to measure distortions in multi-viewpoint images, but this method requires scene priors such as spheres, cylinders or planes to be defined. Tardif et *al.* propose a calibration method [125] that estimates the distortion centre, opening angles of viewing cones and the optical centres. They adopt a double approach, using geometric constraints in linking viewing cones with calibration planes, and a homography-based method. However, their method requires a prior knowledge of the scene's Euclidean structure. Radial lens distortion can be a significant factor introducing errors typically in the range of 10-100 pixels at the edges of the image [127]. To overcome this problem, we take into account the radial distortion introduced by the camera lenses when estimating the point-to-point correspondence between views. The general mathematical formulation we adopted is the division model for distortion presented by Fitzgibbon [105]. As shown by [4], when more than one view is available, it is possible to use the homographic multi-view constraints on the corresponding pairs to recover the

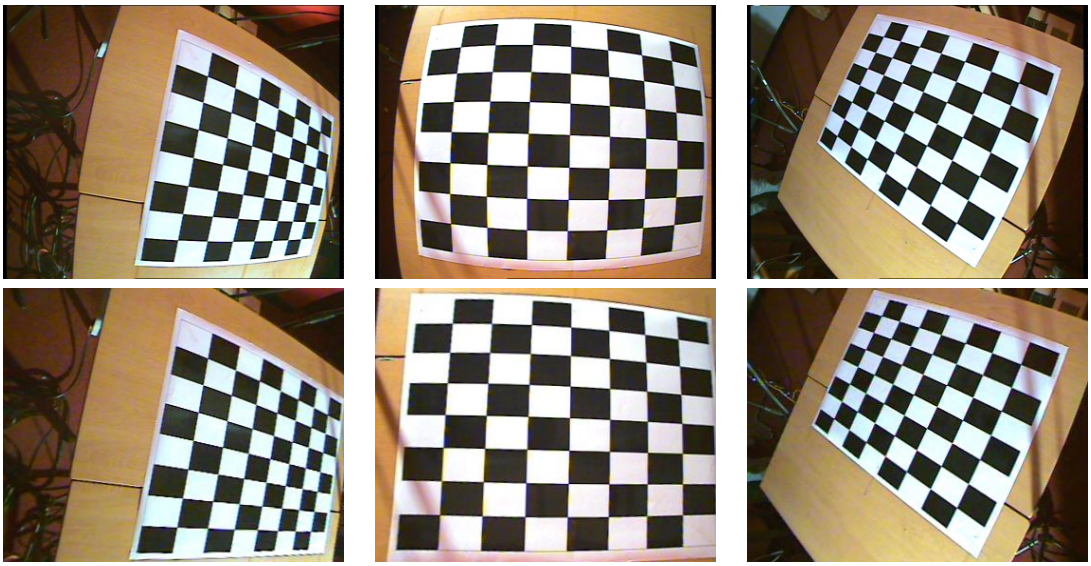


Figure 3.10: Three-view of a chessboard affected by radial lens distortion (top row). Images after distortion correction (bottom row).

distortion.

The estimation of the homography as presented in Sect. 3.2.2 does not remove lens distortions. Therefore, we propose to augment this model with a correction factor. Let an undistorted image point $(x_u, y_u, 1)$ be subject to a radial distortion and let $(x_d, y_d, 1)$ be the resulting distorted point. Generally, the lens distortion model is described by infinite series as follows:

$$\begin{cases} x_u = x_d + x_d(k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \\ y_u = y_d + y_d(k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \end{cases}, \tag{3.16}$$

where $r = \sqrt{(x_d)^2 + (y_d)^2}$ and k_i are coefficients of the radial distortion. It has been demonstrated that approximating this series with its low order elements corrects for more than 90% the radial distortion on the image [124]. As including more coefficient increases the risk of numerical stability in the distortion model, we consider only the first term of the radial distortion. We embed the division distortion model [105] in the correspondence algorithm. The geometric constraints in the homography matrix estimation (Eq. (2.2 in

Algorithm 10 Undistorted homography estimation

Goal: Given corresponding points in two views, estimate the homography relating the undistorted images.

1: Compute the set of distorted corresponding control points pairs $X_d \leftrightarrow X'_d$.
2: Scale the control points by subtracting the center and then normalizing by the sum of the image width and height.
3: Compute $[V, A^{-1}] = polyeig(D_1^T D_3, D_1^T D_2, D_1^T D_1)$, where V is the matrix of eigenvectors and A^{-1} the corresponding inverse eigenvalues.
4: Discard imaginary and null eigenvalues and select the median value from the above remaining eigenvalues.
5: Compute corresponding pairs of undistorted control points $X_u \leftrightarrow X_u^{'}$.
6: Minimise the squared Mahalanobis distance J:

$$J = \sum \left((\mathbf{x}'_\alpha)_\mathbf{u} \times H\mathbf{x}_\mathbf{u}, W_\alpha \left((\mathbf{x}'_\alpha)_\mathbf{u} \times H(\mathbf{x}_\alpha)_\mathbf{u} \right) \right) \tag{3.20}$$

updating W_α using the renormalization technique.
7: Obtain the estimated homography H and the deviation pair $H_{(+)}, H_{(-)}$ from above.

Chap. 2, Sect. 2.2) is augmented to include the first term, k_1, of the radial lens distortion

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \\ 1 + k_1 \left({x_d}^2 + {y_d}^2 \right) \end{pmatrix}, \tag{3.17}$$

where $P = (x_u, y_u, 1)$ is the distortion-free point, $X = (x_d, y_d, 1)$ the distorted point and k_1 the distortion parameter. Thus

$$P = X + k_1 Z, \tag{3.18}$$

where $Z = \left(0, 0, \left(x_d^2 + y_d^2\right)\right)$. The homography constraint in Eq. (2.2) (Chap. 2, Sect. 2.2, can be expressed in terms of vector cross product for each corresponding pair $(P_i, P_i^{'})$ as

$$\mathbf{P'} \times H\mathbf{P} = 0. \tag{3.19}$$

Considering the distorted point as in Eq. (3.18) yields to

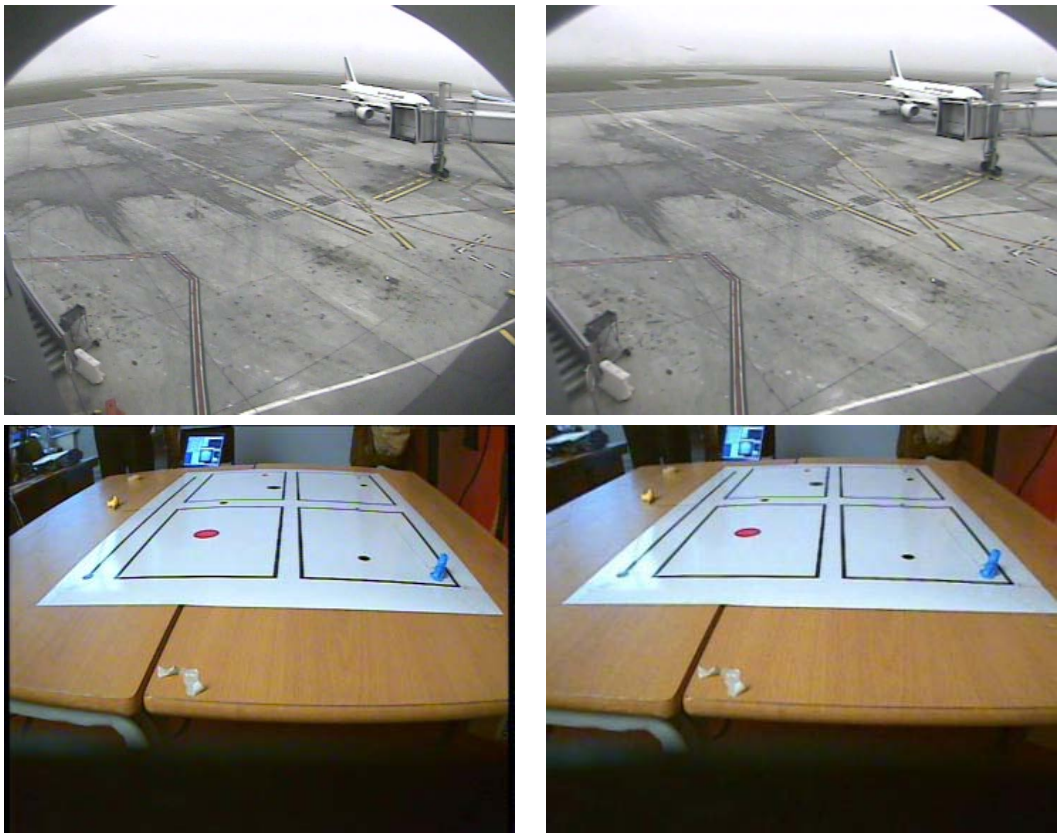


Figure 3.11: Example of radial lens distortions on a two-view scene. The distortions are particularly visible on the image periphery (left column) as compared to corrected images (right column).

$$(\mathbf{x}'_{\mathbf{d}} + k_1 Z') \times H(\mathbf{x}_{\mathbf{d}} + k_1 Z) = 0, \tag{3.21}$$

which is quadratic in k_1 and linear in H. Expanding with the coordinates we obtain

$$(D_1 + k_1 D_2 + k_1^2 D_3)h = 0, \tag{3.22}$$

where h is the vector in Eq. (2.5) and the coefficients D_r are such that

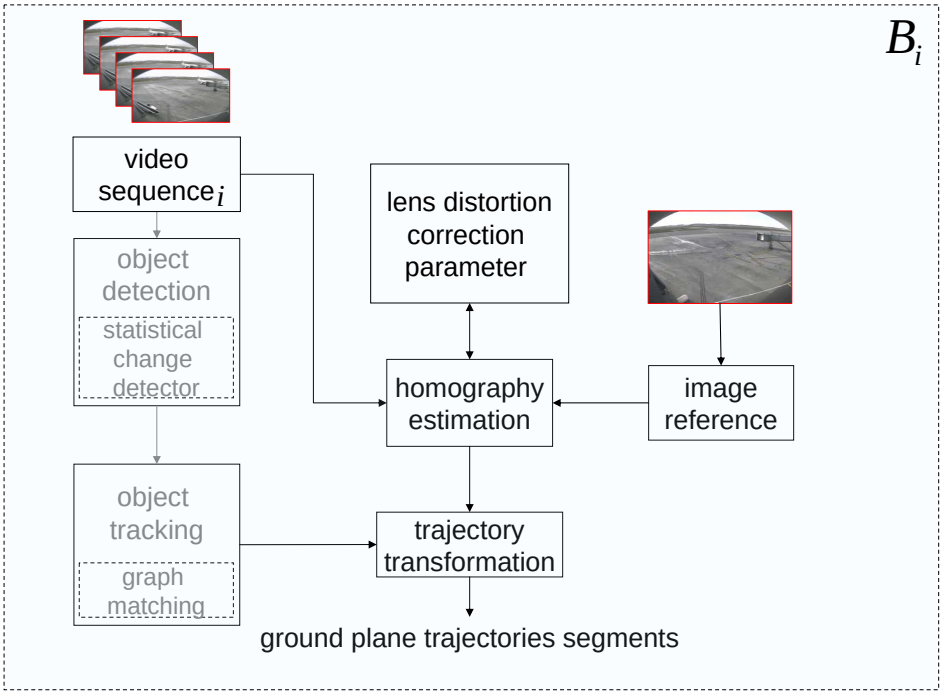
$$D_1 = \begin{pmatrix} 0 & 0 & 0 & -x'_d & -y'_d & -1 & y_d x'_d & y_d y'_d & y_d \\ x'_d & y_d & 1 & 0 & 0 & 0 & -x_d x'_d & -x_d y'_d & -x_d \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 0 & 0 & -r x'_d & -r y'_d & -r - r' & 0 & 0 & y_d r' \\ r x'_d & r y'_d & r' + r & 0 & 0 & 0 & 0 & 0 & -x_d r' \end{pmatrix}$$

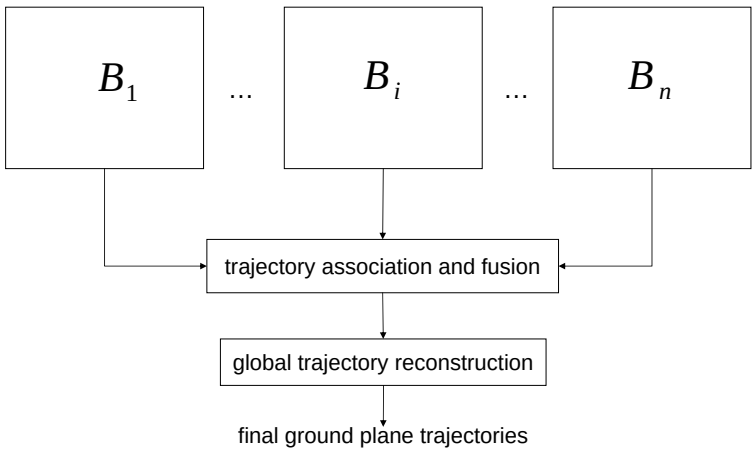
$$D_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -r' r & 0 & 0 & 0 \\ 0 & 0 & r' r & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Equation (3.22) is a Quadratic Eigenvalue Problem (QEP). The solution of this equation yields 4-6 non-imaginary, non-null values. The best values of k_1 have been determined as corresponding to the median value of the set of solutions. Figure 3.11 shows an example of the application of the algorithm embedding lens distortion correction in homography estimation.

The method we proposed is now illustrated in Fig. 3.12 where each single module B_i, in Fig. 3.12(a), gives a simplified view of the mapping from one view to the second that represents the image reference. Corresponding pairs are used in a homographic constraint, first to compute the lens distortion correction parameter , then to estimate the homogaphic transformation itself. Figure 3.12(b) shows the overall block diagram where the segments of trajectories are transformed onto the common ground plane and the global trajectories reconstructed from the segments generated in each initial view.



(a)



(b)

Figure 3.12: Block diagram of the proposed homography estimation with lens distortion correction. (a) Each module B_i describes the image-to-image homography; (b) complete general block diagram considering all the cameras.

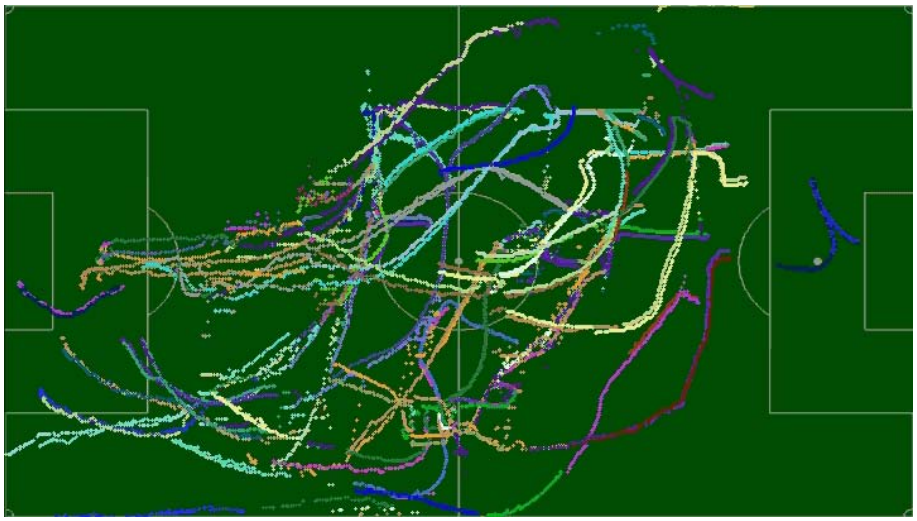


Figure 3.13: Fragments of trajectories after transformation of local tracks generated by individual cameras.

3.3 Trajectory integration on a common view

After object detection and the homography-based transformation onto a common view as shown in Fig. 3.13, the trajectories of objects are still represented by the fragments generated by the partial information from each individual camera. Additionally, there are gaps in objects' trajectories due to interruptions in tracking and occlusions. In areas of overlap, multiple traces of object's trajectories exist due to the simultaneous observations. Therefore, there is need to reconstruct complete objects' trajectories as they move across the multiple views. To this effect, given several trajectories generated in one camera, we perform object association, fusion of tracks in overlapping areas and linking the trajectory segments to obtain complete tracks.

Let $\mathcal{C} = \{C_1, ..., C_N\}$ be a set of N cameras that observe K moving objects. Let $O_m^i(x, y, t)$ and $O_n^i(x, y, t)$ be the trajectories of the i^{th} object, O^i, imaged in C_m and C_n, respectively with (x, y, t) indicating the two spatial coordinates and time. Let $T_m^i(x', y', t)$ and $T_n^i(x', y', t)$ be the trajectories on the ground plane $\mathcal{G}$ after the image-to-ground plane homographic transformations H_m and H_n, respectively. We aim at reconstructing the global trajectories $T^i(x', y', t)$ for object O^i on $\mathcal{G}$.

We express the problem of trajectory integration as the generation of the global trajectory from its fragments and break it into the following subproblems:

- to *estimate* object trajectories, O_m^i, on each camera plane (local trajectories); $i = \{1, \ldots, K\}$ and $m = \{1, \ldots, N\}$
- to *transform* the estimated trajectories onto a common frame coordinate, the ground plane $\mathcal{G}$, using the homographies H_k $(k = 1, 2, \ldots, N)$.
- to *associate* concurrent observations $T_m^i(x', y', t)$ and $T_n^j(x', y', t)$ on $\mathcal{G}$, generated by the same object O^i in overlapping areas. The trajectories are modeled as polygonal line in $2D + t$.
- to *fuse* the associated trajectory fragments $T_m^i(x', y', t)$ and $T_n^i(x', y', t)$ into a single one:

$$T_{m,n}^i(x', y', t) = \mathcal{F}\left(T_m^j, T_n^k\right), \tag{3.23}$$

  where $\mathcal{F}$ the fusion function.
- to *link* the fused fragments by connecting them across $\mathcal{G}$.

The first two subproblems have been treated in the preceding sections and the next will focus on the association, fusion and segment linkage across the scene top view.

3.3.1 Trajectory association and fusion

After having the transformed trajectories ($T_m^i(x', y', t)$), the next step is to compute their relative *pair-wise similarities* for association and fusion in order to have a single trajectory corresponding to an object across the entire field. We make the following assumption for association and fusion: i) two trajectories that are close in space and time and ii) having similar shape are considered to be generated from the same object observed by two cameras. It is noteworthy that two cameras mounted at different positions and having different orientations may force an affine transformation onto the object trajectory.

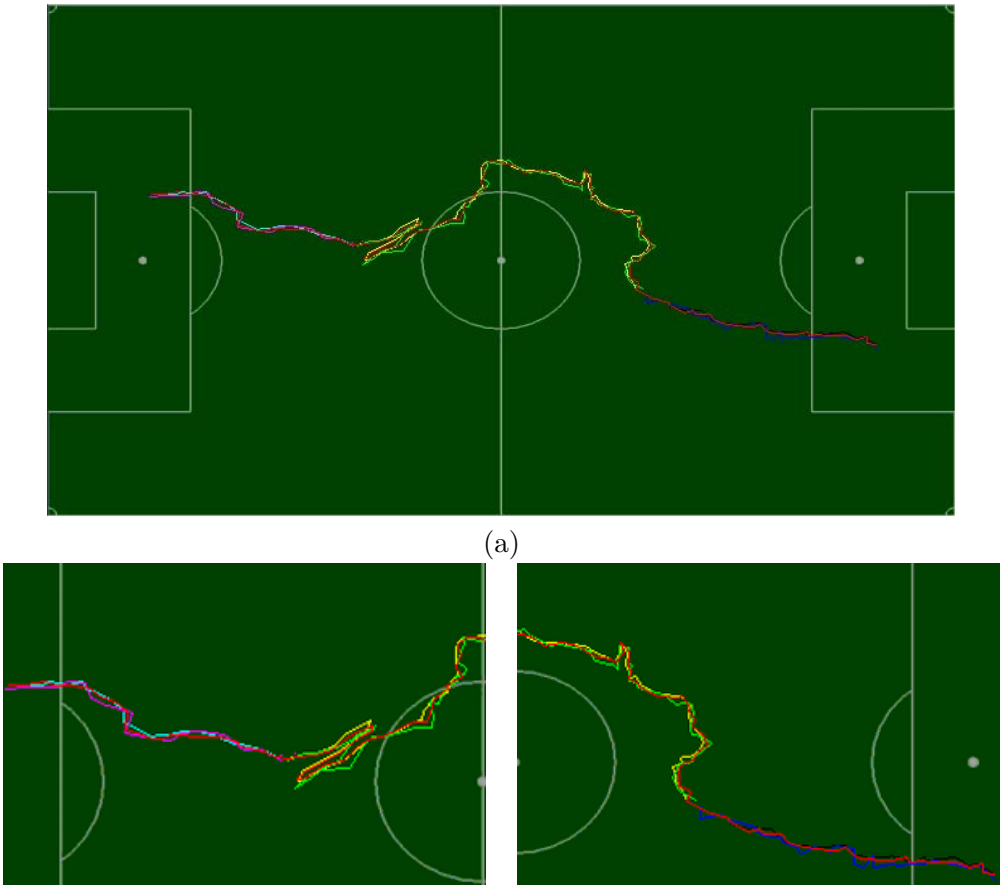


Figure 3.14: Example of trajectory association and fusion on the ground plane. (a) The red trace shows the fused segments of trajectories. (b)-(c) Close-ups on segments.

In order to diminish this effect the input trajectories are first transformed by subtracting the first moments, i.e., for T_m^j the transformation is done as

$$\begin{cases} c_{x'} = \sum_{i=1}^{L_j} (x'_i/L_j) \\ c_{y'} = \sum_{i=1}^{L_j} (y'_i/L_j) \end{cases} \tag{3.24}$$

$$T_m^j(x', y', t) = \left\{ \left(x'_1 - c_{x'}, y'_1 - c_{y'}, t\right), \ldots \ldots, \left(x'_{L_j} - c_{x'}, y'_{L_j} - c_{y'}, t\right) \right\} \tag{3.25}$$

where $m = 1, \ldots, N$; L_j is the total number of trajectory points in T_m^j. Next, both model-based and spatial features are computed. For model-based features, models of the trajectory parameters are learned using polynomial regressions. The matrix notation for

the model estimation $\tilde{T}_m^j(y)$, is written as:

$$\tilde{T}_m^j(y) = \begin{pmatrix} 1 \; T_m^j(x) \; T_m^j(x)^2 \; \ldots \; T_m^j(x)^P \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \ldots \\ \beta_P \end{pmatrix} + \psi, \tag{3.26}$$

where, the first term in the product on the right is a L_jxP matrix and the second is Px1 vector and last a L_jx1 vector. The resulting vector is also Lx1 dimension. The goal here is to find the optimal values of β_i that minimises $\psi = |T_m^{j'}(y) - T_m^j(y)|$. The process requires an inherent trade-off between accuracy and efficiency. As the degree of the polynomial increases, the fit grows in accuracy (up to a point), but the time and space needed increases as well. We find the appropriate degree by starting with a first degree (linear) polynomial and continuously monitoring the fit to see whether the degree needs to be increased. If so, the regression is restarted with the degree incremented by one. We fixed $P = 2$ as in our experiments, increasing P does not effect the overall accuracy. We use the initial and final positions as spatial information. The length of each trajectory thus obtained by taking absolute difference between the spatial coordinates of starting and final positions of the object trajectory:

$$\alpha_{m,n}^j = |T_m^j(x_1, y_1) - T_m^j(x_{L_j}, y_{L_j})|, \tag{3.27}$$

where the first and the second term indicates the initial and final coordinates, respectively, in the overlapping area $\Omega_{m,n}$. The final feature vector form is:

$$V_{m,n}^j = [\beta_0, \beta_1, \beta_2, \alpha_{m,n}^j, t]^T, \tag{3.28}$$

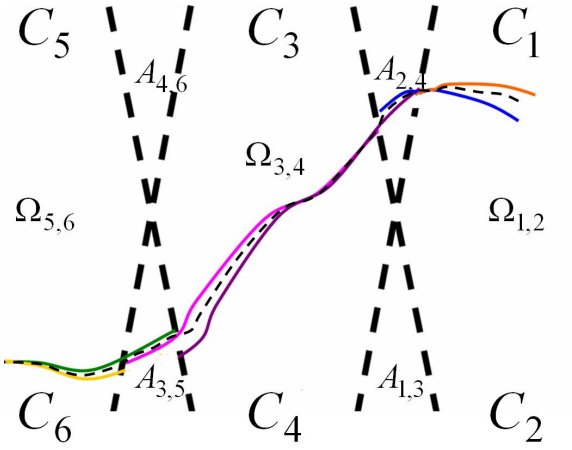


Figure 3.15: Linkage of segments across the ground plane. C_1 : *Orange*, C_2 : *Blue*, C_3 : *Pink*, C_4 : *Violet*, C_5 : *Green*, C_6 : *Yellow*. The dashed line is the complete trajectory.

where the upscript T denotes the transpose vector. The time dimension, t, is dropped for simplicity. Because of its robustness to the scale variation, we use cross correlation as similarity measure. For T_m^j and T_n^k in $\Omega_{m,n}$ the correlation matrix is calculated as:

$$\zeta_{m,n}^{j,k} = C(V_{m,n}^j, V_{m,n}^k) \quad k = 1, ..., K; \tag{3.29}$$

where, C is the correlation function. The final trajectory $T_{m,n}^i$ is estimated by fusing the trajectories (T_m^j, T_n^k), when $\zeta_{m,n}^{j,k} > \zeta_{m,n}^{j,i} \quad \forall k \neq i$. The fusion is computed as the average of the associated pair T_m^j and T_n^k coordinates (Fig. 3.14).

3.3.2 Segment linkage

After fusing the trajectories generated by an object O^i on the ground plane, the challenge we face here is to bridge gaps between fragments and link the pieces of trajectories into long spatio-temporal trajectories. This is trivial when gaps are brief and the tracks are spaced from each other. Figure 3.15 shows an example of trajectory linkage where a chain process that uses objects' attributes is carried out to reconstruct a complete trajectory on the entire field. The object trajectory starts from $\Omega_{5,6}$. The

connection between $\Omega_{5,6}$ and $A_{3,5}$ can be established by utilizing the object's attribute related to T_5^i $(Green)$. Further, the connection to $\Omega_{3,4}$ is established by using $T_3^i(Pink)$ segment. Next, connection of fragments between $\Omega_{3,4}$ and $A_{2,4}$ is made by using T_4^i $(Violet)$. Finally, $T_2^i(Blue)$ is used to connect $A_{2,4}$ to $\Omega_{1,2}$. The dashed line is the final trajectory constructed after association, fusion and linking.

The main steps of the method we propose in this work for trajectory integration is summarized in the following steps:

1. **Control points extraction**
   - We use the trajectory of an object moving across the multiple views and extract the corresponding segments that belong to the region of overlap.
   - We subsample points from the segments of the trajectories above to avoid collinearity in the set of control points as this might lead to a degenerate case during the homography estimation.
2. **Lens distortion correction**
   - The control points are put in a homographic correspondence
   - The parameter that corrects the lens distortion is computed
3. **Homography estimation**
   - The control points are corrected
   - The homography is estimated from the set of control points
4. **Trajectory transformation**

   All the trajectories are transformed and mapped into a common plane using the estimated homography.
5. **Trajectory association**

   The multiple traces generated by the same object in the overlapping areas are associated using their polinomial representation and fused to generate one segment.

6. **Segment linkage**

   Objects' fused traces in the overlapping areas are linked to segments generated by the same object in other regions across the views and we obtain the global trajectory.

3.4 Summary

We presented an algorithm for trajectory transformation for wide-baseline multi-camera scene analysis with embedded lens distortion correction. Using the transfer errors, we have demonstrated an improvement of trajectory transformation in terms of accuracy and a reduced error in the trajectory transformation compared to the traditional linear (SVD) and non linear (LSM) techniques. Moreover, we have demonstrated that this approach is more robust to errors in the estimation of the control points and that the perturbation in the trajectory transformation is smaller than that of traditional approaches using linear (SVD) or non–linear (LMS) homography estimation. We have also shown the benefits of the embedded lens distortion correction in the proposed algorithm by comparing the undistorted and distorted mosaics and transformed trajectories. Additionally, we have presented an algorithm for trajectory reconstruction in multi-view ensemble and applied it to the complex case of sport sequences. The proposed approach uses a trajectory generated by one single object visible in the cameras' overlapping area to estimate the homography. We use time constraint to extract the set of control points from the trajectory. Then, the trajectories generated by each camera are transformed and mapped onto the ground plane. We perform trajectory association and fusion using a similarity metric that identifies, within overlapping regions, fragments of transformed trajectories generated by a same object. These fragments are fused and connected across the field of view using temporal consistency and object identity.

Chapter 4

Results

4.1 Introdution

In order to evaluate the proposed approach, we conduct different sets of experiments using available data in five different multi-view scenarios that are described in the sections that follow. These scenarios vary from each other in terms of:

- the multiple view topology: the number of cameras ranges from two to six-views, their relative configuration from random to symmetrical, with different degrees of overlap between the different fields of view.
- the illumination and stability of the background in both indoor and outdoor settings: some scenarios have a stable background while others present changing daylight illumination, swaying leaves or a particularly noisy environment.
- the complexity of the scene: this ranges from scenes with one person only to scenarios with crowds.

The experiments cover the different aspects investigated in the present work. The cameras are synchronized and the objects are assumed to be moving on a ground plane and their relative positions on the image expressed as the mid-point of the bottom segment line of the bounding box (Fig. 4.1). Objects' trajectories are transformed with

Figure 4.1: Left: example of a frame with moving object; right: the detected object. The object's location is assumed to be the middle point of the bounding box bottom segment.

statistical estimation of homography and a comparison is made against existing methods. The experiments on estimating homography while correcting the radial lens distortion are presented, a comparison is performed against uncorrected images and a discussion on the impact of lens distortion is made. Trajectories extracted from objects in a crowd are transformed onto the scene top-view and reconstructed to recover the global tracks across the views. Additionally, we present a section that analyses the performance of the proposed algorithm in two ways: i) by comparison against existing state-of-the-art methods; ii) by measuring the discrepancies between the results from real data and the ground truth process. We conduct an extensive series of experiments to quantify tracking and trajectory mapping errors over a large amount of data. We highlight the most significant results and discuss them in the light of the underlying processes and methods provided by the proposed approach. We use the Euclidean distance as a reliable measure for trajectory distances against the ground truth, as suggested by Fu and Zhang in [129, 130].

4.2 Datasets

The following data were used in the evaluation of the proposed approach:

- ***ISSIA dataset***[1] This is a six-view football scene (Fig. 4.2) acquired by six Full-

[1]Raw videos courtesy of Institute on Intelligent Systems for Automation - C.N.R., Bari, Italy. *http://www.issia.cnr.it*

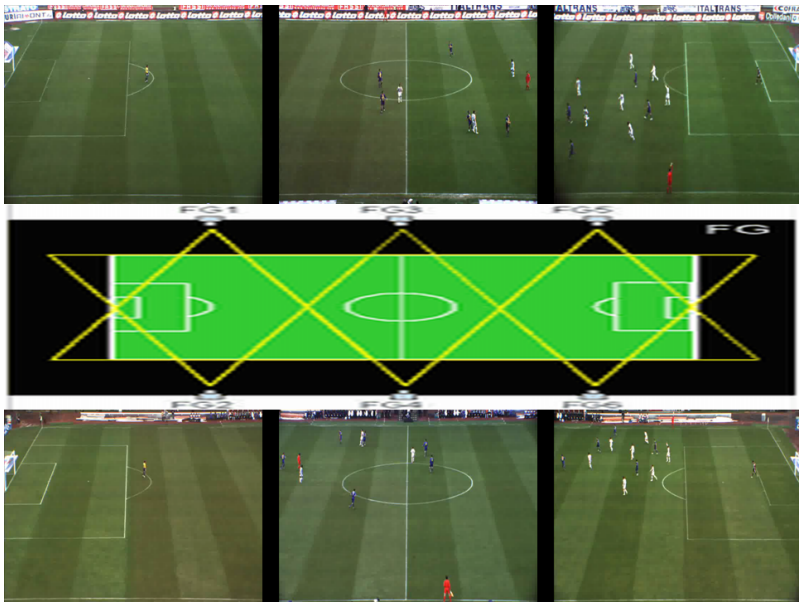


Figure 4.2: Example from the ISSIA dataset. This provides a six-view of a footbal scene. The middle illustration shows the locations of the six cameras.

Figure 4.3: Example from the Pets01 dataset (a two-view scene of a campus).

HD DALSA 25-2M30 cameras, three for each major side of the playing-field, at 25 fps. The six real footages are made of 18000 frames (1920 x 1088) describing a football game captured from 6 viewpoints symmetrically arranged in two rows of three cameras facing each other. The dataset is quite challenging with scenes presenting an increasing level of complexity. This includes various situations ranging from idle players moving at a slow pace to very dynamic scenes with building-up of groups of players with entangled trajectories. Additionally, the targets' motion is erratic, highly non-linear due to the nature of the game.

Figure 4.4: Example from the ETISEO dataset (a multiple view scene of an airport runway).

- ***Pets01***[2] This standard surveillance dataset describes different scenarios featuring isolated pedestrians or small groups moving along regular motion paths. PETS2001 represents a two-view monitoring of a campus site (Fig. 4.3). The level of illumination between the two views is different in the original images.

- ***Synthetic dataset*** We also use a simulated environment that generates synthetic trajectories originating from a multiple camera setup. The setup consists of four cameras whose fields of view present overlapping areas. We test the trajectory association and linkage algorithms on synthetic data generated by cameras placed on a perfect top view perspective. Therefore, the main axis of the camera is perpendicular

[2]http://peipa.essex.ac.uk/ipa/pix/pets/PETS2001/

Figure 4.5: Example from the CREDS dataset (a two-view scene of an underground).

to the plane on which objects are moving and therefore no perspective distortion is involved.

- ***Etiseo***[3] The ETISEO dataset depicts an airport scene taken from cameras whose images are affected by radial lens distortion. It depicts scenes on an airport tarmac with vehicles and pedestrians (Fig. 4.4). These images contain accentuated distortions and we apply the proposed approach to remove distortion while computing homographies.

- ***CREDS*** The Challenge for Real-time Events Detection Solutions CREDS dataset describes scenes in a metropolitan station (Fig. 4.5). It includes several scenarios such as people crossing rails, walking on the platform, passengers wedged in the train door and such like. It is a three-camera set-up but for the sake of simplicity we will not be using the third one. This dataset presents relevant disturbances due to noise and poor illumination conditions as well as low contrast between the objects and the background.

- ***Pets06***[4] describes a multiple-camera system to monitor a train station (Fig. 1.1). The resolution of these videos (PAL standard) is of 768 x 576 pixels and 25 frames

[3] http://www.silogic.fr/etiseo/index.html
[4] Copyright ISCAPS consortium. Permission of the PETS 2006 workshop

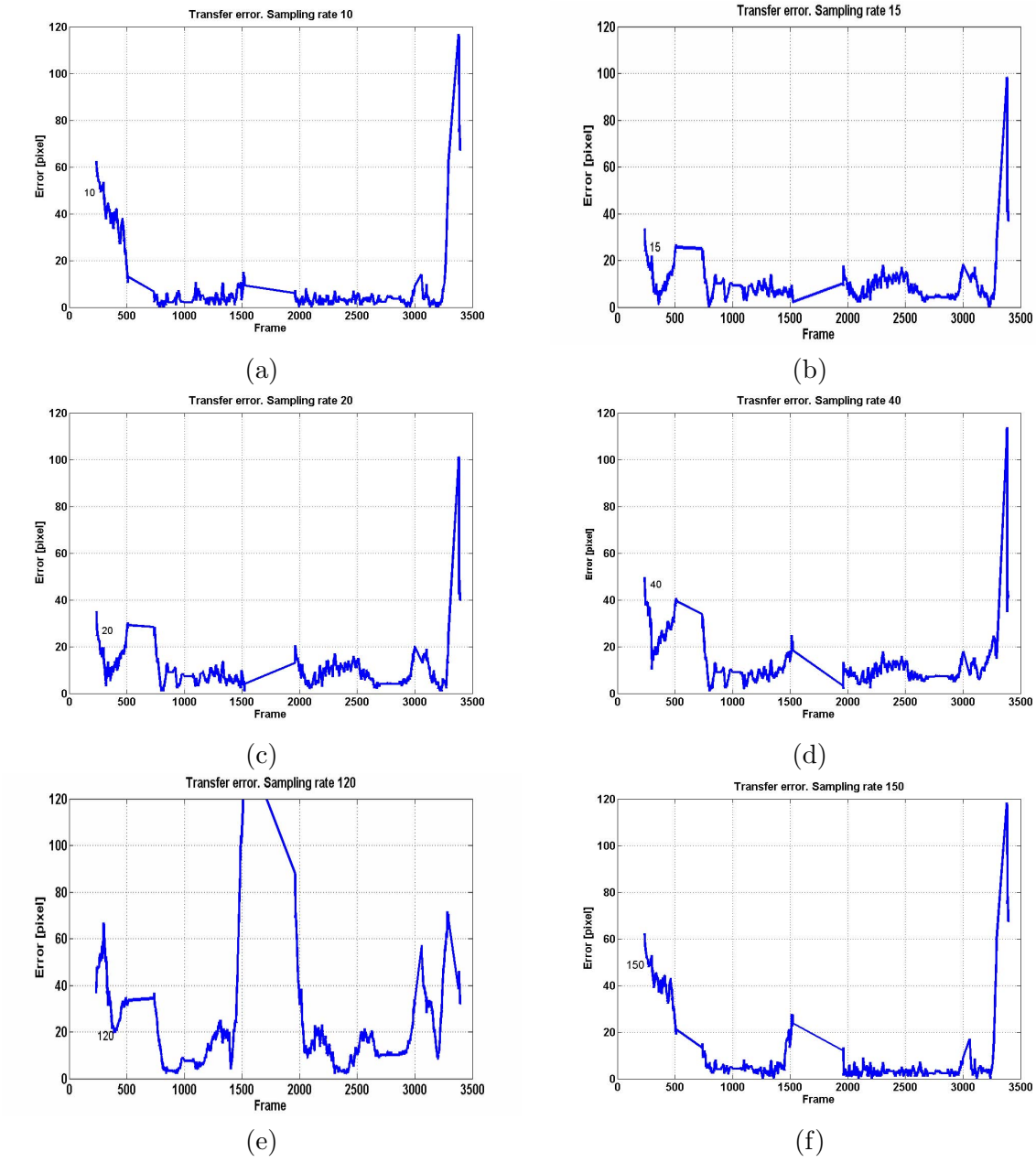


Figure 4.6: Transfer errors of transformation of *target*1's trajectory when estimating homography from *target*2's trajectory. The results we show of sampling trajectory points at different intervals (10, 15, 20, 40, 120 and 150 frames) to form sets of control points. A low interval means a high number of extracted features and vice-versa.

per second. Images are compressed as JPEG with approximately 90 % quality.

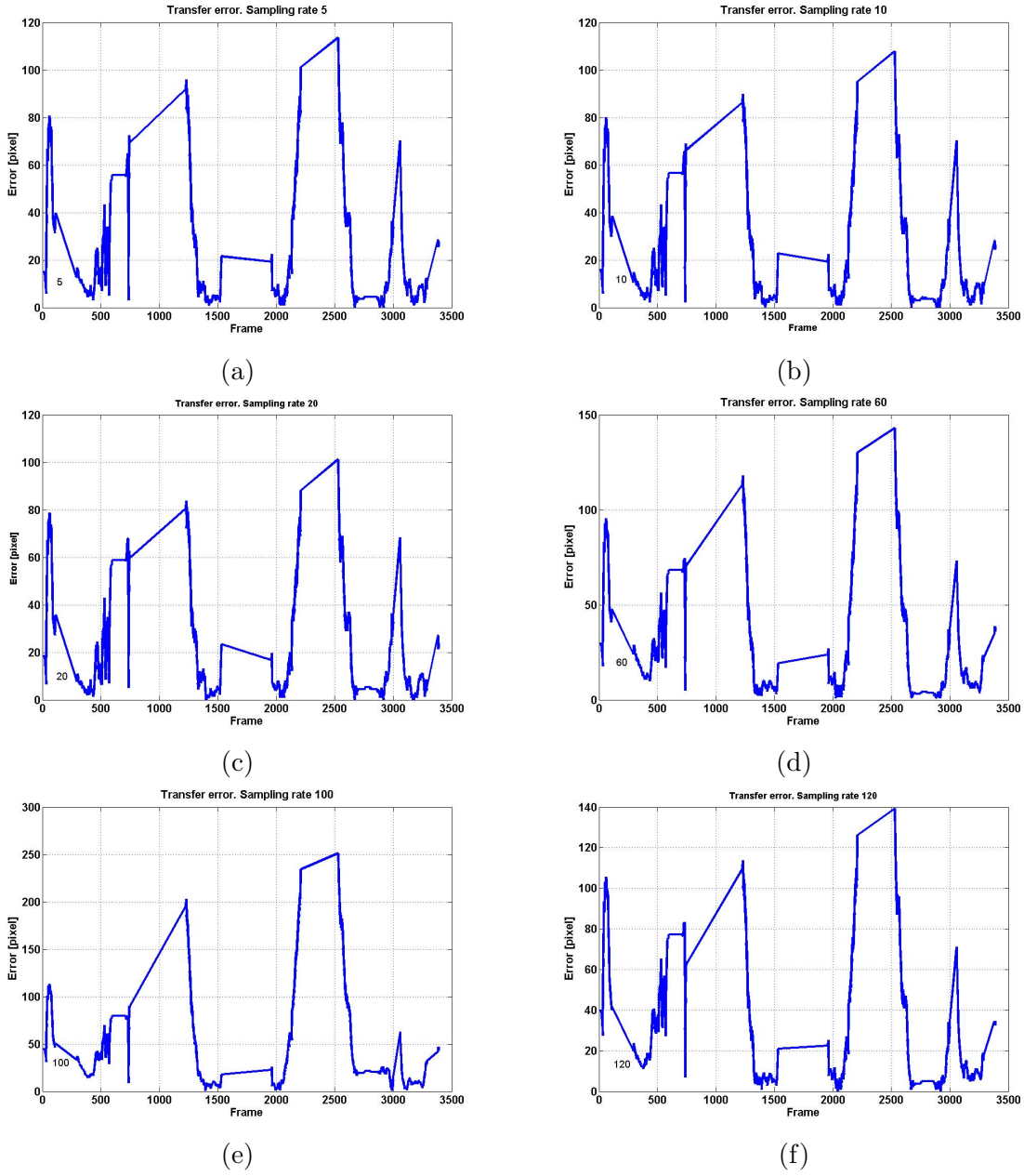


Figure 4.7: Transfer errors of transformation of *target*1's trajectory when estimating homography from *target*3's trajectory. The results we show of sampling trajectory points at different intervals (10, 15, 20, 40, 120 and 150 frames) to form sets of control points. A low interval means a high number of extracted features and vice-versa.

4.3 Trajectory mapping to one common view

We apply the control points extraction method and estimate homographies from objects' trajectories. In the following example, we use the trajectory of an object moving

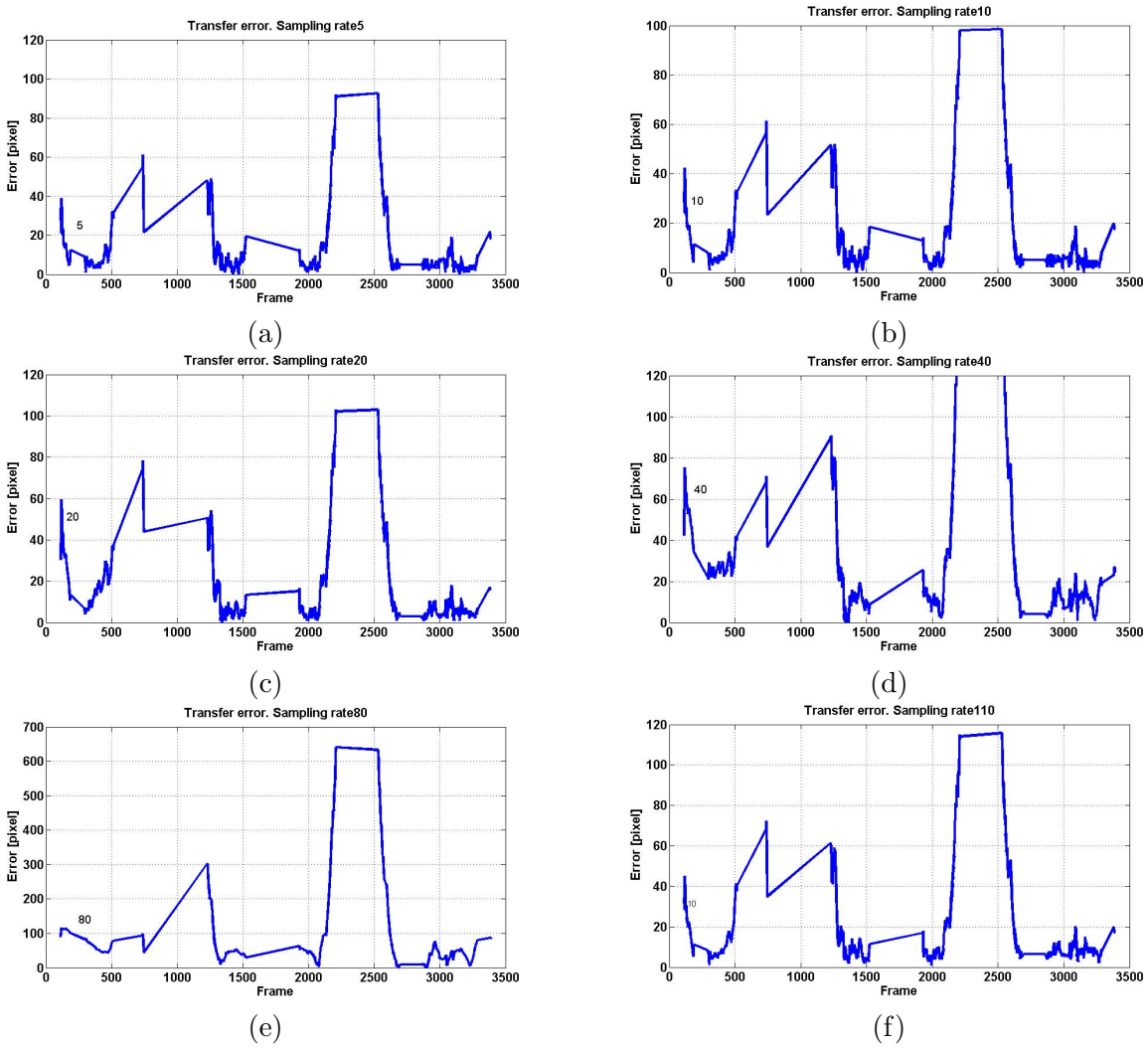


Figure 4.8: Transfer errors of transformation of *target*2's trajectory when estimating homography from *target*3's trajectory. The results we show of sampling trajectory points at different intervals (10, 15, 20, 40, 120 and 150 frames) to form sets of control points. A low interval means a high number of extracted features and vice-versa.

across the image in the two-view scene (Fig. 3.7). The homography obtained with this method is then used to transform other objects' trajectories. We study the accuracy of the transformation while varying the number of features (control points) and their distribution across the image. Figures 4.6 and 4.7 and 4.8 are an example from ETISEO dataset and illustrate the impact of the increasing density of control points on the transformation accuracy. This accuracy is measured as transfer error on trajectories other than the one used to estimate the homography. Higher sampling rates indicate lower number of

features and vice-versa. We see that the accuracy decreases while the number of control points decreases too. Large numbers of control points provide better estimation of the homography. However, very large numbers of control points seem to lead to a drop of accuracy. This is because when more features are added, more noise is consequently added to the homographic geometric constraints. When on the other hand there is only a small number of points, there are insufficient constraints to enable the homography fitting over the entire image. The main observation remains that the accuracy of the transformation is less sensitive to the number of points than their distribution across the image. This set of experiments consists of extracting objects' trajectories in multiple views, transforming and mapping them onto one integrating view. This view is either represented by a mosaic or a scene top view. We use scenarios featuring pedestrians from a standard surveillance dataset, PETS2001, and for clarity we show examples for three targets P_1, P_2 and P_3. The experiments are organised as follow: first, we extract a set of control points from each image. These corresponding points are used to estimate the plane-to-plane homography. One image is warped onto the second (reference image), using the estimated homographic transformation and then both are composed to generate a mosaic of the scene. The performance assessment is conducted by a visual and a quantitative evaluation. The first visualises the estimated trajectories against their expected spatial location (ground truth). The second measures the transfer errors and computes the robustness of the transformation against noise. The evaluation is conducted by comparing the proposed approach with state-of-the-art methods that use linear (SVD) and non linear (LMS) homography estimation [6]. For the quantitative evaluation we use the transfer error, e_{tr}, that computes the displacement of a point transferred by a homography with respect to its ground truth position as follows:

$$e_{tr} = d\left(x_i^{'}, Hx_i\right), \tag{4.1}$$

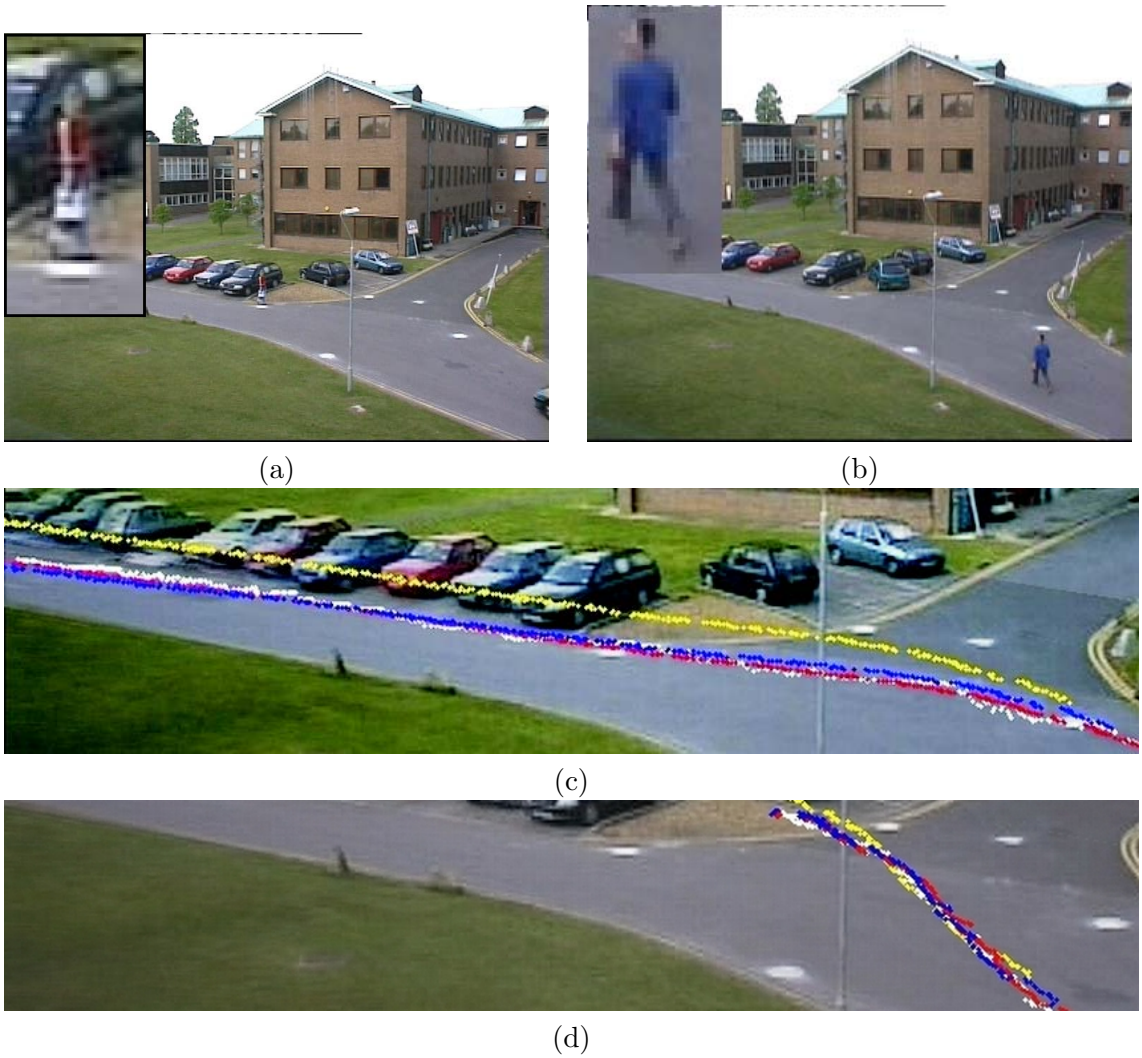


Figure 4.9: Comparison of trajectory transformations for two targets, (a) P_1 and (b) P_2, on the image mosaic of Fig. 3.8; (c)-(d) zoom on the transformed trajectories: ground truth (white), proposed approach (red), SVD (yellow) and LMS (blue).

where $x_i^{'}$ is the ground truth and Hx_i the estimated value. $d(.)$ is the Euclidean distance. Figure 4.9 compares the trajectories of two targets, superimposed on the mosaic generated with 3 different methods: the proposed algorithm, the SVD-based and the LMS-based algorithm [6]. Figures 4.9(c)-(d) show a close-up on the targets' transformed trajectories, illustrating the displacement between the expected location (ground-truth) of the target and their actual measurements after transformation. The linear SVD method presents higher errors than both LMS and the proposed approach. The linear estimation performs well as long as the targets are moving close to control points as in the case of

Table 4.1: Trajectory transfer error with and without renormalization. μ and σ indicate the mean and the standard deviation of the resulting transfer errors

	Without renormalization				With renormalization	
	SVD		LMS		Proposed approach	
$Target$	μ	σ	μ	σ	μ	σ
P_1	5.97	3.11	3.86	2.49	3.05	2.12
P_2	4.51	1.99	2.87	1.86	2.29	1.01

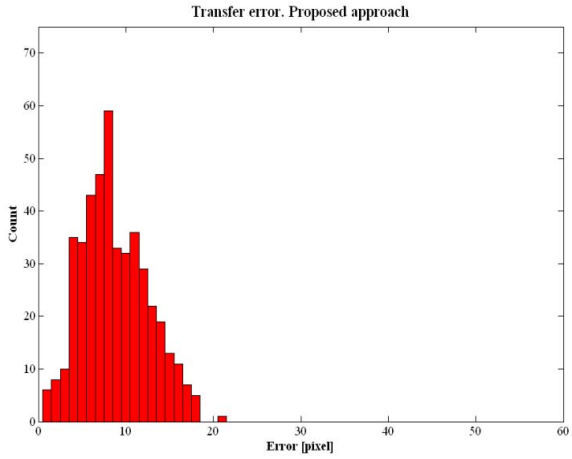
target P_2 (Fig. 4.9(d)). While getting further from the control points, the drift between the measured and the expected location becomes more and more important (target P_1 in Fig. 4.9(c)). The main reason behind the errors in the mapping using linear transformation resides in the way SVD estimates the homography transformation. Indeed, its linear estimation consists of a pure algebraic solution to the geometric problem of fitting noisy corresponding points in a homography relationship. The absence of additional geometric constraints to relate to the corresponding points is likely to degrade the homography in those remote areas. Table 4.1 reports the transfer errors for the aforementioned cases and shows the improvements in the trajectory mapping accuracy when using the renormalization technique as opposed to SVD and LMS, which do not use renormalization. The reported values are expressed as mean values, with their corresponding variances.

We compare the robustness of the three approaches against noise and report the mean values of the variation of the transfer errors for the targets P_1 and P_2 in Table 4.2. The robustness test is performed by corrupting the selected control points with varying magnitudes of Gaussian noise N and by then estimating the subsequent transfer error on the transformed trajectories. Figure 4.11 shows examples of computed transfer error for target P_1 and target P_2 and confirms this performance. The results show a smaller increase in the transfer error for the proposed approach than for the SVD-based and in LMS-based methods. This is due to the fact that the renormalization accounts for the geometric noise in the point-to-point correspondence. We use the paired student's t-test to verify the statistical significance of the differences in performance observed in the results presented above. Results present p values less than $p < 10^{-3}$ for targets P_1 and P_2 and

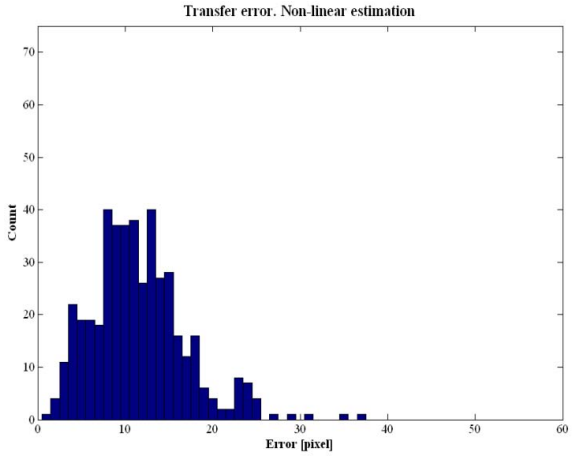
Table 4.2: Influence of noise on the trajectory transfer error. N indicates the noise amplitude (Gaussian) used to corrupt the control points; μ and σ denote the mean and the standard deviation of the resulting transfer errors for target P_1 and target P_2.

		SVD		LMS		Proposed approach	
	N	μ	σ	μ	σ	μ	σ
P_1	1	6.85	4.24	4.16	2.59	3.65	2.19
	2	7.69	5.32	5.35	2.98	4.08	2.31
	4	8.94	6.37	5.96	3.81	4.57	2.75
	5	9.35	5.92	6.66	4.41	5.38	2.93
	7	13.38	7.03	7.49	4.57	6.01	3.18
	8	14.84	7.19	9.01	5.05	6.72	4.01
	10	17.27	8.25	12.11	5.45	8.12	4.37
	12	19.97	9.33	13.89	5.88	9.25	4.49
	15	21.91	10.39	16.41	7.09	11.36	6.28
P_2	1	5.98	3.97	3.49	2.20	2.73	1.46
	2	6.71	4.56	3.89	2.25	2.97	1.80
	4	8.85	5.90	4.97	2.91	4.13	2.34
	5	11.34	6.90	6.92	3.53	5.37	2.93
	7	15.27	8.94	7.72	3.79	6.35	3.45
	8	19.97	10.66	10.17	4.66	7.64	3.95
	10	20.03	11.50	11.84	5.42	8.39	4.41
	12	20.89	12.18	13.91	5.95	9.44	5.06
	15	23.29	14.15	15.31	7.10	12.26	6.62

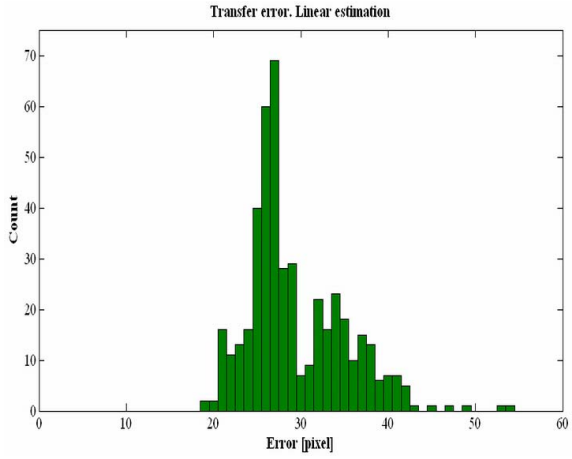
$p = 10^{-3}$ for target P_3. The p-value associated with these measurements is low ($p < 0.05$), thus we have evidence that there is a difference in means across the paired observations LMS versus proposed approach and SVD versus the proposed approach. Figure 4.10 illustrates an example of histograms of transfer errors (target P_3). SVD has a noticeable wider spread (larger errors) than LMS and the proposed approach (lower magnitude of errors). When comparing the proposed approach to the LMS method, the difference in performance is reduced. LMS presents a better fitting than SVD due to the introduction of a geometric cost function that minimizes the transfer error. However, one disadvantage of this model is that it requires an additional phase in the homography estimation that consists of an initialisation with the linearly estimated homography matrix. Besides, LMS assumes that the entire data can be interpreted by only one parameter vector of a given model and even when the data contains only one bad datum, LMS estimates may be completely perturbed [4]. Figure 4.12 shows an example of the trajectory transformation



(a)



(b)



(c)

Figure 4.10: Comparison of transfer errors distribution for the trajectory transformation of target P_3: (a) Proposed approach; (b) LMS; (c) SVD.

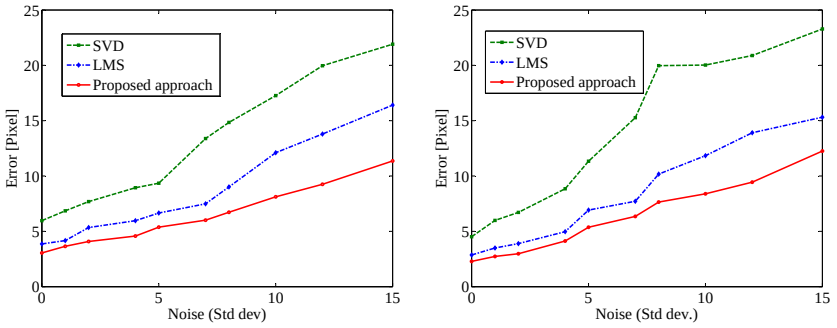
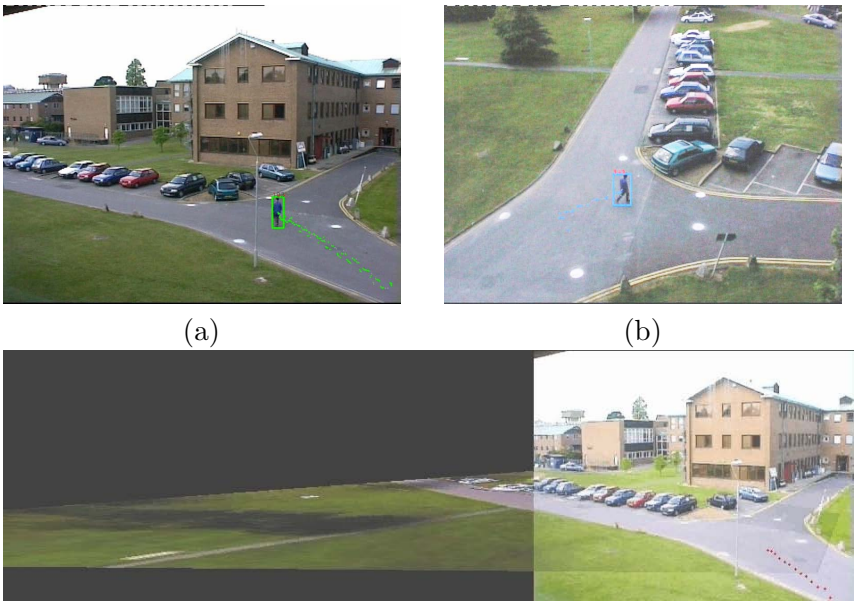


Figure 4.11: Comparison of computed positions of control points altered with Gaussian noise. Mean transfer error measured on the transformed trajectories ((a)P_1; (b)P_2) while increasing the noise magnitude.

(target P_2) superimposed on the generated mosaic. The landmarks (see white patches) in Fig. 4.12(a)-(b)) on the ground plane show the image location of the control points used to estimate the homography. The target enclosed in the bounding box is moving across the two cameras' fields of view. The final decision on the target's location (red dots on the mosaic on Fig. 4.12(c)) is based on camera switching that selects the closest location in case of concurrent observations. Since there are concurrent object observations in areas corresponding to overlapping field of views, a decision is taken that results in a single observation on the mosaic [121]. We assume that the most reliable measurement of an object spatial location is given by the observation from the closest camera (*c*amera switching). At a time t, given a detected object located at spatial co-ordinates $\left(x_1^t, y_1^t\right)$ in camera 1 and $\left(x_2^t, y_2^t\right)$ in camera 2, the closest camera is the one whose object's y co-ordinate is closer to the bottom of the image plane. Although the proposed approach improves the trajectory transformation, there are some misalignment residuals due to the image segmentation process that equally affects all transformation methods. A faulty segmentation in one camera generates a truncated blob that in turn undermines the computation of an object's ground location. Since the image-to-image homography requires point coplanarity, one consequence of a wrong ground location computation is a displacement of the transformed trajectory point. Besides, a wrong ground location computation in one view



(a) (b)

(c)

Figure 4.12: Final object trajectory on the mosaic; (a)-(b) a pedestrian is tracked; (c) the red dots indicate the resulting final on the ground plane.

leads to a wrong point-to-point correspondence between views. A solution to this problem could consist of improving the estimation of blobs' ground locations.

4.4 Lens distortion correction in multiple view

We demonstrate the proposed approach for trajectory transformation with lens distortion correction on the ETISEO dataset and compare the results with those of SVD and LMS. We analyze examples of resulting object detection and tracking across multiple views and of trajectory mapping on mosaics whose distorted images have been corrected. Four sequences of 110, 300, 100 and 170 frames (the image size is 720×576 pixels) with moving pedestrians have been used. For fairness of comparison, the same distortion correction is applied to all methods. Figure 4.13 shows examples of object segmentation and tracking of targets E_1, E_2, E_3 and E_4.

Figure 4.14 visualizes the benefits of the correction of radial lens distortion on image mosaics. A mosaic from two views with overlapping areas is shown with and with-

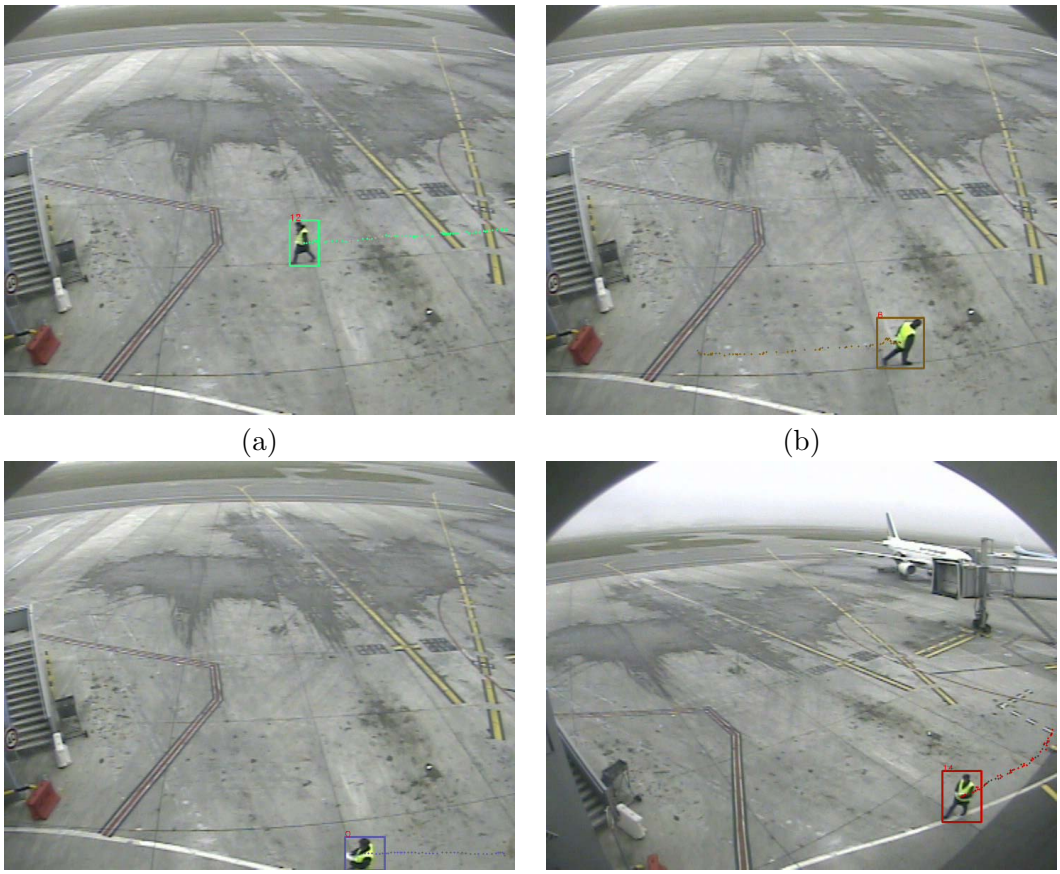
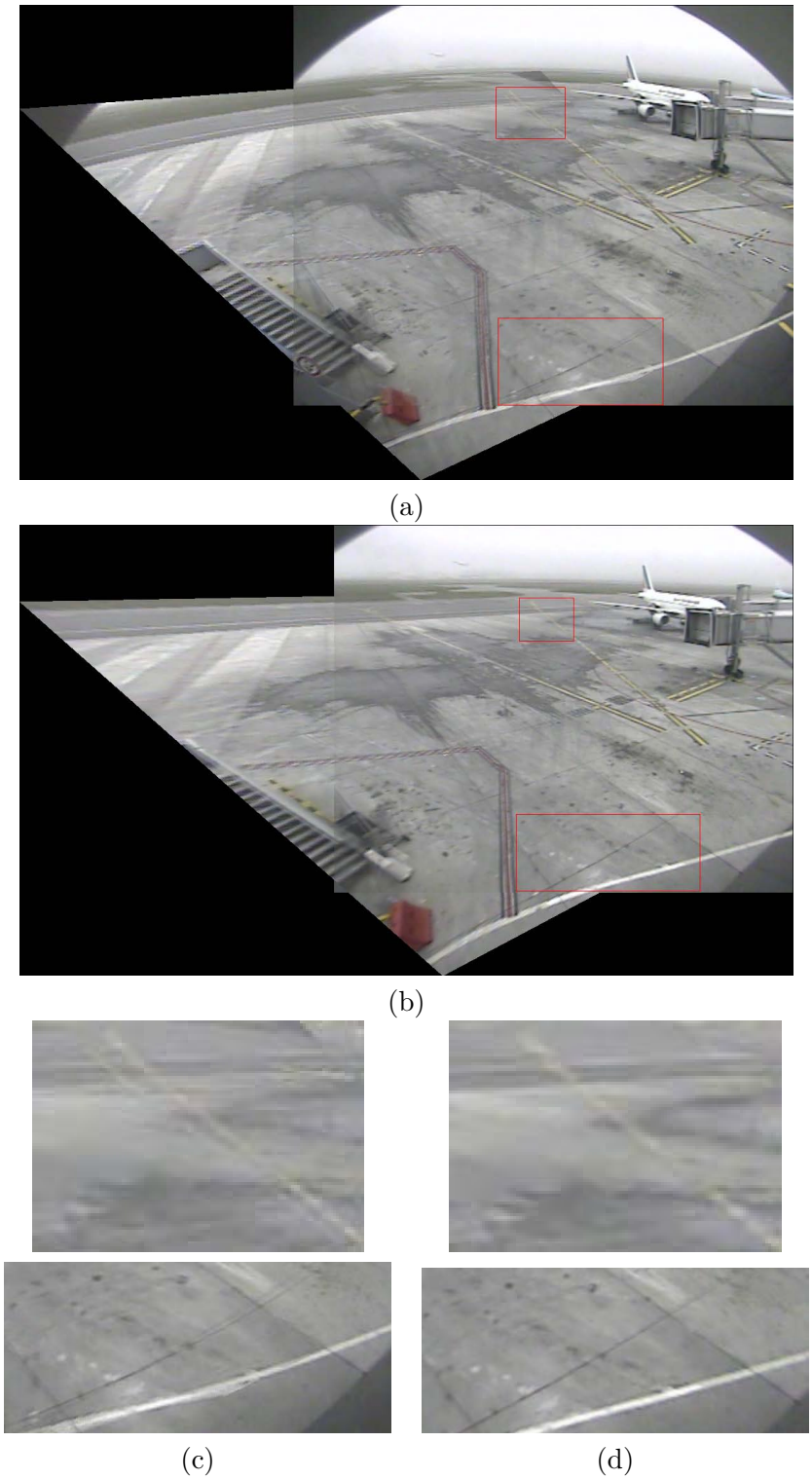


Figure 4.13: Sample targets from the ETISEO dataset. (a) Target E_1; (b) Target E_2; (c) Target E_3; (d) Target E_4.

out lens distortion correction. Because of the radial distortion, residual misalignments are visible on Fig. 4.14(a) (before lens distortion correction), particularly with the white and yellow lines located near the borders of the image. A significant improvement is obtained after correction as illustrated by the alignment in Fig. 4.14(b). Similarly, Fig. 4.17 shows the distortion correction on a two-view scene of a chessboard with both significant perspective and radial lens distortion. Aligning the two original images (left column) results in residual errors, particularly visible at the bottom of the image. These errors are reduced on the mosaic generated with the proposed approach (bottom right). Figure 4.15 shows the correction of two objects' trajectories (target E_1 and target E_2). Note the difference between the distorted (red) and corrected (blue) trajectory when the target moves closer

to the image periphery. With lens distortion correction, we rectify the trajectory points location with distances that reach 10 pixels for E_1, 45 pixels for E_2, 62 pixels for E_3, 88 pixels for E_4. These quantities measure the differences between the trajectory points before and after lens distortion correction. Figure 4.16 illustrates an example of the variation of the transfer errors over time for two of the ETISEO targets.

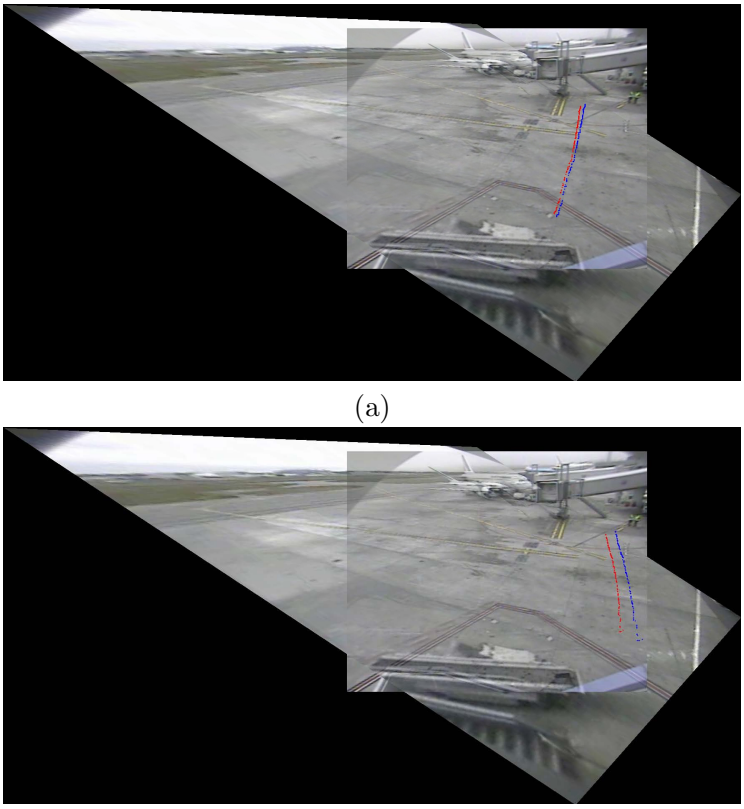


(a)

(b)

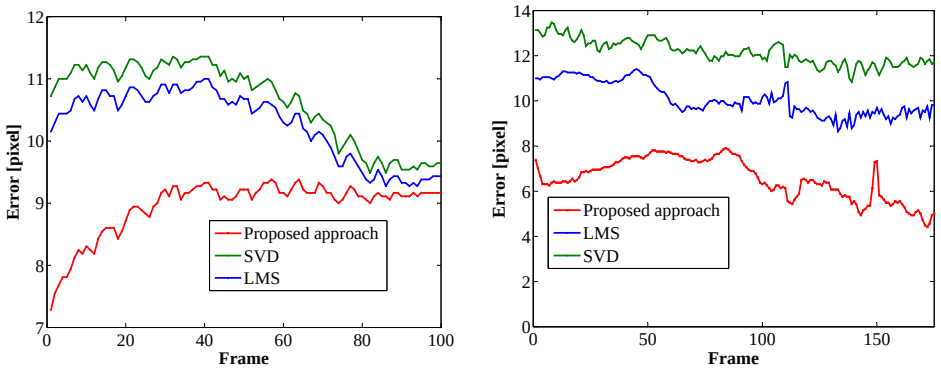
(c) (d)

Figure 4.14: Distortion correction on mosaics. (a) Mosaic without distortion correction; (b) image mosaic after lens distortion correction; (c) and (d) refer to areas enclosed by the red rectangles in (a) and (b). Note the different in residual misalignments between (c) before and (d) after lens distortion correction.



(a) (b)

Figure 4.15: Example of distorted trajectory (in red) and corrected (in blue) on the mosaic. (a) target E_1; (b) target E_2.



(a) (b)

Figure 4.16: Comparison of transfer errors over time for (a) target E_3 and (b) target E_4. The distortion correction is applied to all 3 methods.

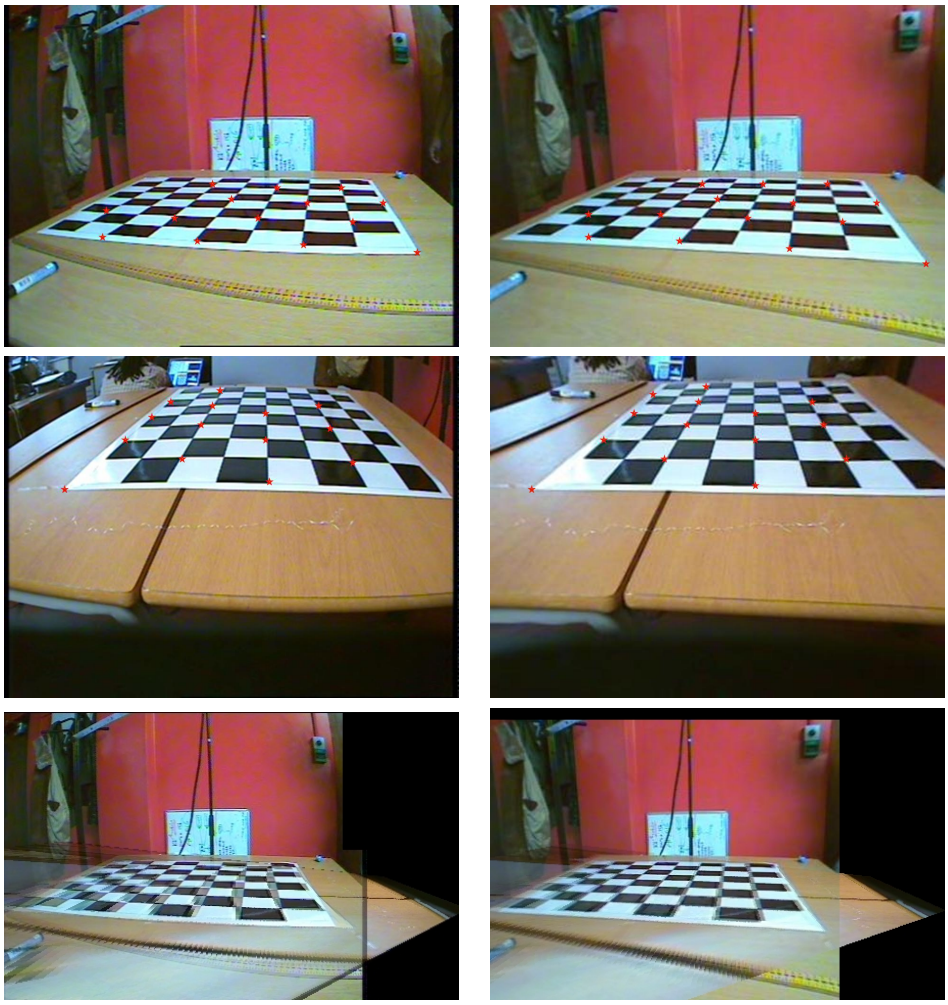


Figure 4.17: Two-view of a chessboard (first and second row). The red dots indicate control points. Radial lens distortion affecting the images (left) and distortion correction (right). Mosaics of images with (left) and without lens distortion correction (right). Note the residual misalignments on the mosaic from the uncorrected images.

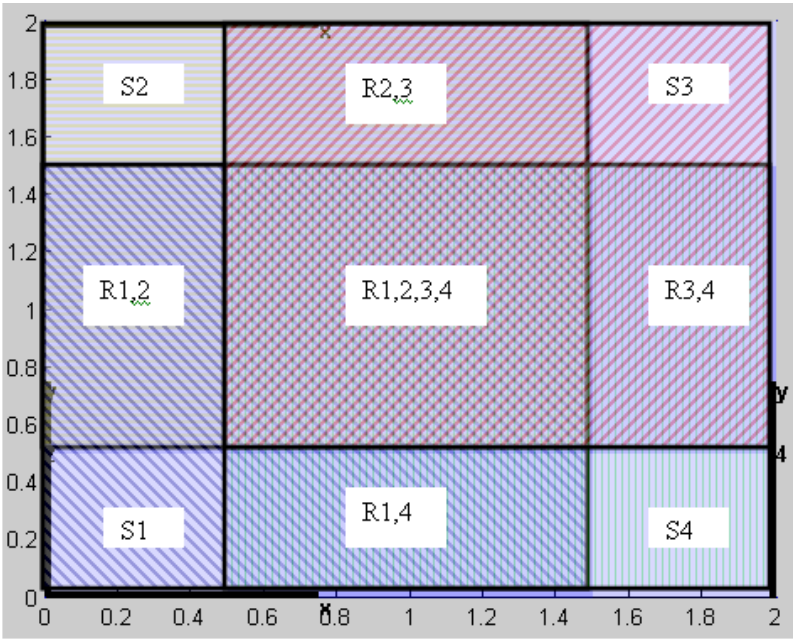


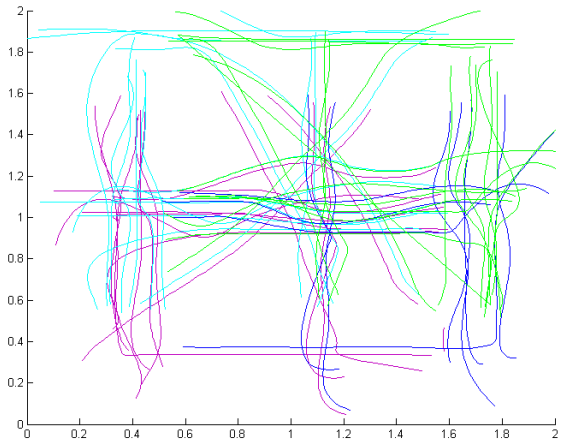
Figure 4.18: Top-view of camera fields of view for the synthetically generated trajectories. S_i is the non-overlapping region imaged in *camera i* only whereas R_{ij} show overlapping areas viewed by cameras indicated in the subscript.

4.5 Trajectory reconstruction from multiple views

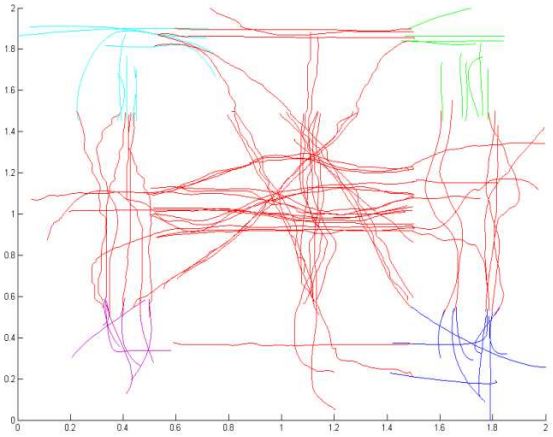
Tests on the algorithm presented have been performed on synthetic data and real data from football footage. In the first series of experiments, we simulate trajectories originating from a multiple camera setup whose configuration is illustrated in Fig. 4.18. The setup consists of four cameras whose fields of view present overlapping areas. In the figure, S_i represents a non-overlapping area and R_{ij} the region commonly monitored by camera C_i and C_j. We test the trajectory association and linkage algorithms on synthetic data generated by cameras placed on a perfect top view perspective. This means that

Table 4.3: Comparison of trajectory association *Precision* (P) and *Recall* (R) for the methods under analysis on synthetic data.

	S12		S13		S14		S24		S34	
Algorithm	R	P	R	P	R	P	R	P	R	P
Proposed approach	0.94	1.00	0.78	1.00	0.86	1.00	0.70	1.00	0.86	1.00
KNN-interpolated	0.89	0.94	0.71	0.68	0.84	0.78	0.76	0.82	0.78	0.80
KNN-LCSS	0.90	0.94	0.72	0.69	0.85	0.80	0.70	0.80	0.79	0.80



(a)



(b)

Figure 4.19: Global trajectory with overlapping cameras on synthetic data: (a) input trajectories; (b) corresponding fused trajectories.

the main axis of the camera is perpendicular to the plane on which objects are moving and therefore no perspective distortion is involved. Fig. 4.19(a) shows the initial and Fig. 4.19(b) the final (fused) trajectories. To evaluate the proposed approach, we compute two measures, the *precision* P and the *recall* R defined as

$$\begin{cases} R = \frac{|G \bigcap E|}{|G|} \\ P = \frac{|G \bigcap E|}{|E|} \end{cases}, \tag{4.2}$$

where G and E indicate the set of manually constructed pairs of concurrent trajectories and the matching results from the proposed approach, respectively. We compare the proposed approach with *interpolated* and *LCSS* KNN algorithms. Table 4.3 reports higher scores with the proposed approach in terms of precision and recall measures. The second series of experiments were conducted on the ISSIA dataset that consists of footage of 3000 frames, describing a football scene simultaneously recorded by 6 cameras located at different viewpoints. There are 22 players and 1 referee on the pitch and 2 linesmen. When acquiring the sequences, no constraints were imposed on players' trajectories. Unlike the first series of experiments, the oblique camera principal axis induces a perspective distortion. The homography is computed to obtain the top view observation of the scene. The process of mapping trajectories onto the ground plane carries errors reflected by the discrepancy between observations in an overlapping area. We expect the reconstruction of trajectories to be affected by the proximity of moving objects' tracks, the gaps caused by discontinuity in field of view borders and faulty object extraction. Figure 4.21(a) presents results obtained after reconstruction of global trajectories on $\mathcal{G}$. The proposed approach is able to reconstruct global trajectories across $\mathcal{G}$. Nevertheless, the reconstruction of global trajectories is still difficult, particularly in areas of high density (regions of the image in the centre of the pitch). In the crowded scene, blobs splitting undermines the reconstruction as it causes the generation of several tracks of one object, all moving close to each other and with similar motion. This spatio-temporal proximity with tracks generated by the corresponding object on another view causes ambiguity in track associations. This in turn affects the segment linkage in the global reconstruction phase.

Table 4.4 report results on trajectory association for C_3 and C_4. We have achieved the highest *Recall* and *Precision* score on segment 1 whereas segment 4 presents the lowest score. The matching performance is related to the segment length which in turn is related to tracking performance and ground plane transformation as it can affect the accuracy of objects' attributes (Fig. 4.20). Segment 1 contained longer trajectories compared to

Table 4.4: Performance evaluation of the trajectory association using *Precision* (P) and *Recall* (R) on real data.

Segment no.	Fragment time interval (number of frames)	P	R
1	230	0.80	0.90
2	230	0.60	0.80
3	139	0.80	0.80
4	249	0.50	0.60
5	290	0.67	0.70
6	360	0.64	0.65
7	130	0.71	0.70
8	400	0.62	0.80

segments 4 and 6 and they were close in the spatio-temporal domain. Conversely, segments 4 and 6 showed isolated and short trajectories that hampered the association process. These singularities are mainly due to tracking failures and transformation errors. The results from the association process can be further improved by enhancing these two components of the proposed approach.

4.6 Performance Evaluation

We conduct a series of experiments over a large amount of data and under various scenarios to bring to light critical situations, crucial to the understanding of the behaviour of the proposed approach under critical tracking situations and test its limits. These situations, illustrated in Fig 4.22, include objects' interactions such as multiple objects crossing paths and causing occlusions, crowds and abrupt motion variations. We compare the performance of tracking and trajectory transformation modules of the proposed approach against that of another algorithm, based on linear estimation of the homography (hereinafter named LE). The two algorithms (Fig. 4.23 and 3.12) are tested on ISSIA dataset [43] and the output measured in terms of the drift of trajectory points with respect to the ideal position of the ground truth data. The errors are computed at each instant for each object in the image plane as the Euclidean distance between the object's

trajectory point in the automatic detection and the ground truth. Additionally, we compute a stability score that estimates the object's identity switches along their trajectories and compare the outcome for the two algorithms. The ground truth has been generated manually with Viper tool [131].

The main differences in the performance of the two algorithms are due to the approach used in object tracking and the estimation of the homographic transformation. Because of its linear prediction and blob split/merge mechanisms, LE handles the crucial aspect of blob assignment to objects in a more efficient way than the proposed approach. On the other hand, because its estimation of the object's localization is solely based on detection, the proposed approach is more responsive than LE to abrupt changes in object's motion. Therefore, the proposed approach tends to be more accurate than LE, in absence of partial or total occlusion. Interactions between different objects (partial or full occlusion) usually give rise to a sudden increase in errors. This happens because when coming close to each other, blobs can temporarily merge, causing errors in blobs' assignment hence object's localization. Likewise, faulty detections can cause blob split. Figure 4.24 show examples of typical errors related to those cases. Sharp rises are reported at instants of sudden change in either the speed or the direction of motion. Interactions between more objects trigger sharp rises in the tracking errors for both algorithms. These errors occur at instants of blob merging/splitting for which it is difficult for the algorithms to assign to each object an accurate estimation of its location.

Figure 4.25 illustrates examples related to the occurrence of a split. There is no split detection mechanism in the tracking algorithm (GM) of the proposed approach. A small piece created by the splitting of a blob initially associated to one object is identified as a new object and this triggers the initialization of a new track. This means the association of the various pieces of blobs to one tracked object is made difficult by the one-to-one assignment policy in the GM algorithm. GM associates one of the new tracks to the old ones depending on the position and velocity of the new detected object. Conversely,

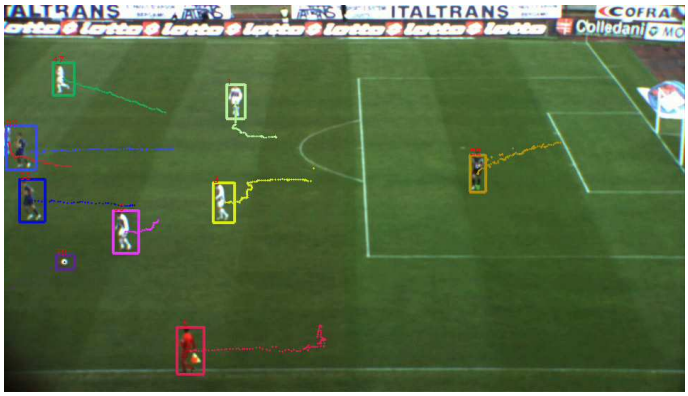
because LE uses a HMM whose hidden state includes information about the location, velocity, acceleration and importantly the single bounding box, this strategy allows a detection of the split occurrence. The various parts of a blob resulting from a split are assigned to one object.

Figure 4.26(a)-(c) illustrates the case of object's trajectory crossing and generating a merging blobs event. The corresponding evaluation of the LE and the proposed algorithms against ground-truth shows increasing errors. Figure 4.26(d) shows a peak of errors around $frame200$ at the occurrence of blob merge with tracking errors lower in LE than in the proposed approach. By estimating the multi-person configuration dynamics and the blob observation likelihood, LE procedures are capable of describing the temporal behaviour of the target configuration. This includes the targets' position changes, entrance and exit from scene, occlusion and blob merging. The probability of having blobs merging between two consecutive frames is estimated based on the blob distances, velocities and dimensions. The maintenance of the state vectors for solved blobs in merge blobs allows the recognition of splitting situations.

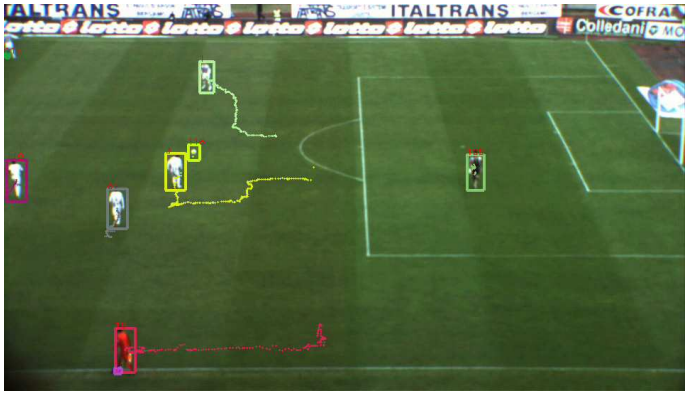
Figure 4.27 shows an example of final trajectories on the ground plane computed with the two algorithms. The resulting tracks from the proposed approach (red) is closer to the ground truth (black) than the LE (green). This is due to the lower registration errors in trajectory mapping with the proposed approach and likewise to the accuracy in object location by the GE tracking algorithm. Both approaches are using simple averaging to compute the fused location of the object.

To evaluate the stability of LE and GM compute a stability score as the ratio between the occurrences of a successful tracking for given object and its actual occurrences in the scene (GT). Values range between 0 and 1 and those close to 1 show better tracking stability and conversely those close to 0 a poor performance. Table 4.28 reports the stability measurements for a sample of 50 objects. The mean and variance for LE and GM algorithm's tracking stability are respectively $(0.7019, 0.218)$ and $(0.5861, 0.252)$.The

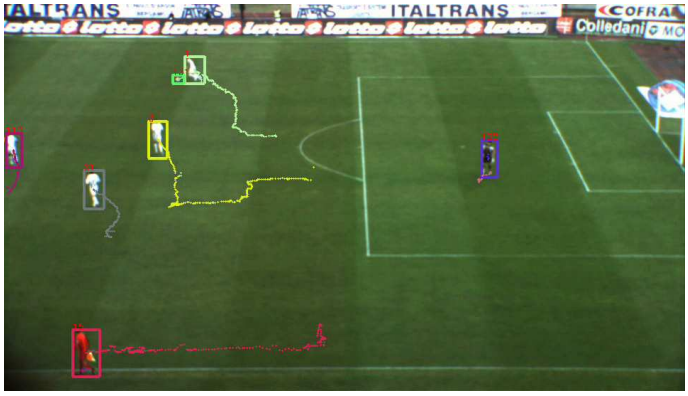
tracking stability for target P_{18} in sequence *camera*1 is 0.096, which is below the average, due to GM difficulty in handling sudden and speed and direction changes [16]. P_{115} and P_{119} both LE and GM present noticeably lower values than the average due to identity switches in cases where several objects are interacting. Other noticeable values are reported in bold in Table 4.28 to show either particularly very low, high or considerable differences of performances between the two algorithms. These results show that the absence in the proposed approach of a mechanism that handles the merging/split of blobs is critical and undermines its performances. However, under all other situations, the tracking procedure in the proposed approach tends to achieves higher stability than LE algorithm.



frame 130

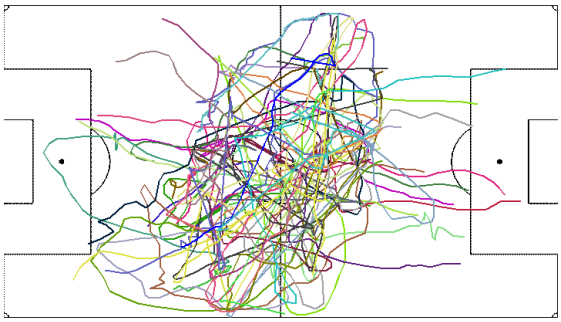


frame 244

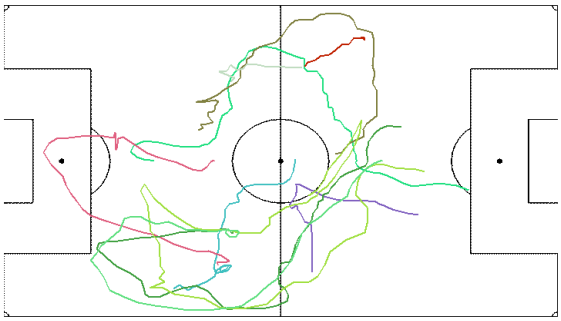


frame 284

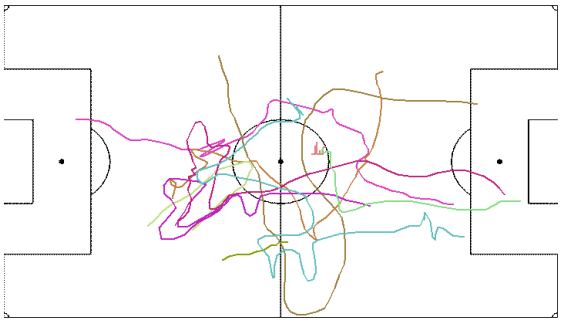
Figure 4.20: Sample tracking results on the image plane, with an example of tracking error. The goalkeeper's identity switches between 22 in frame 130, 100 in frame 244 and 132 in frame 284 (the identity switch is highlighted by the different colour of the corresponding bounding box).



(a)



(b)



(c)

Figure 4.21: Reconstructed global trajectories on the ground plane. (a) Final trajectories. (b)-(c) Sample reconstructed trajectories.

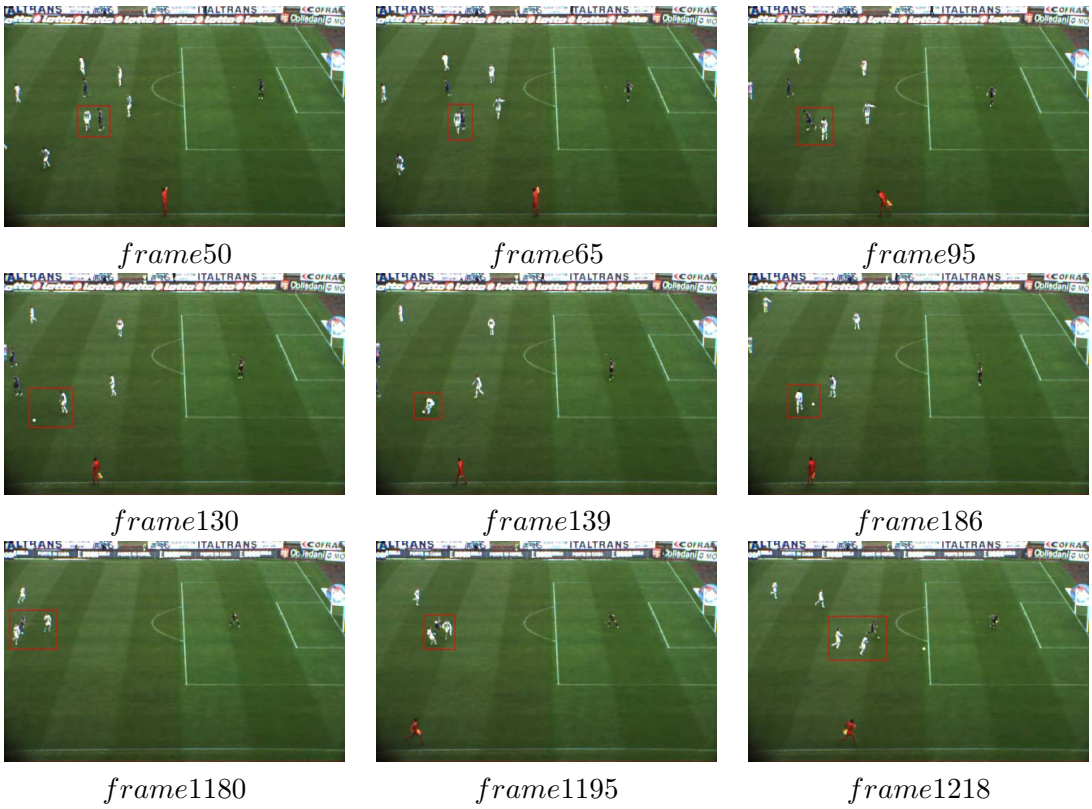


Figure 4.22: Examples of different interactions between objects in the examined scenes. Two players (first row); object-ball (second row); several objects (third row).

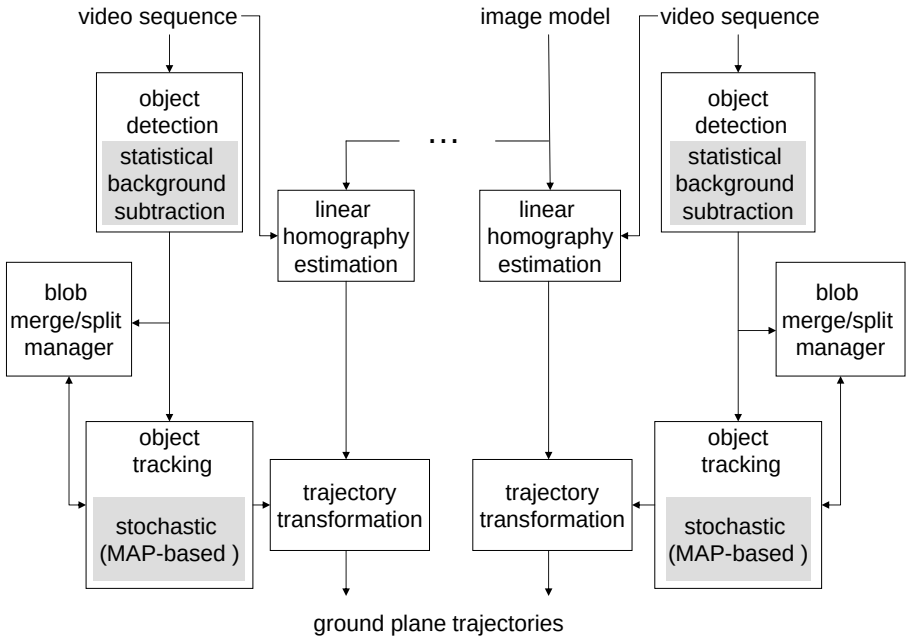


Figure 4.23: block diagrams of the LE algorithm.

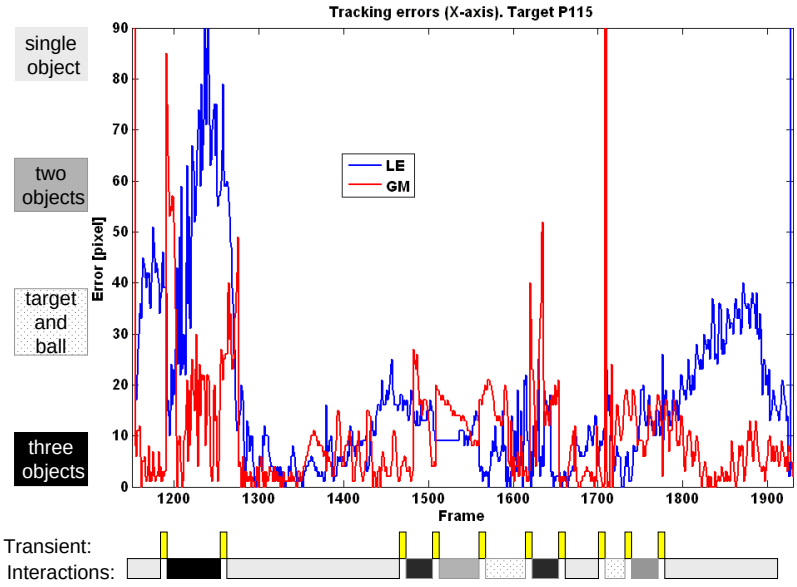
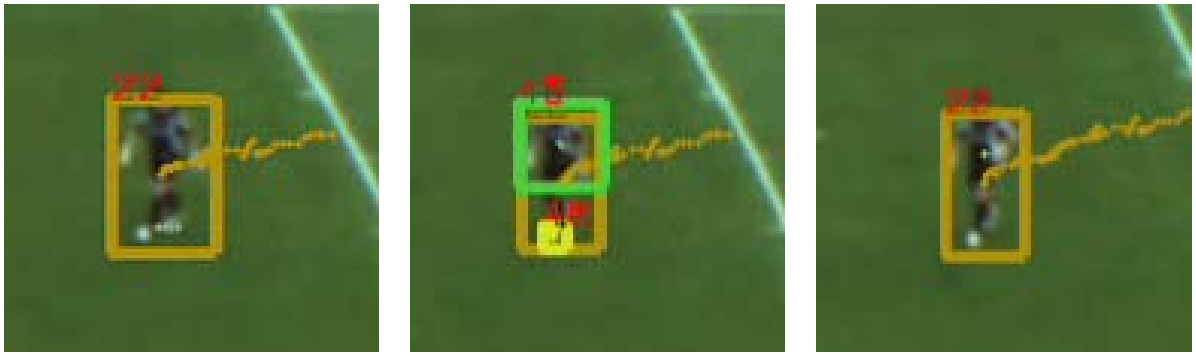
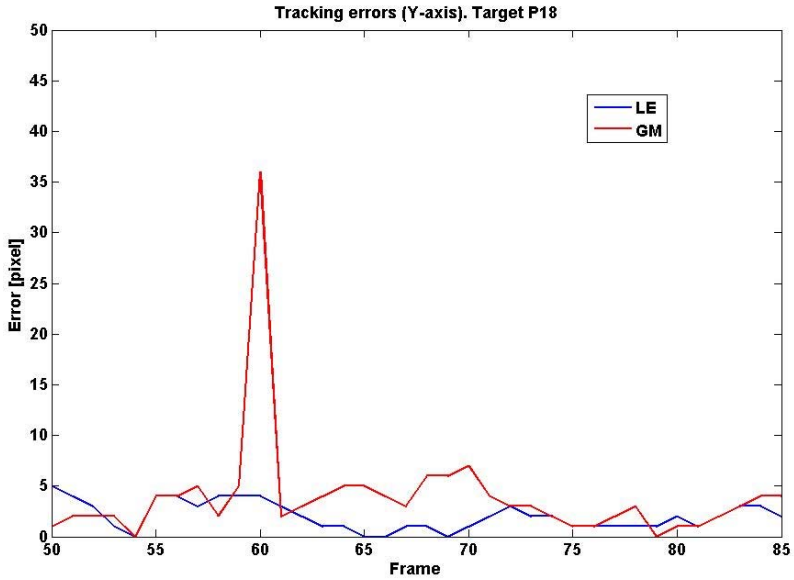


Figure 4.24: Example of typical tracking errors when objects enter each other's vicinity. The bar with different patterns highlights these interactions and show rises at moments of sudden change in speed or direction of motion.

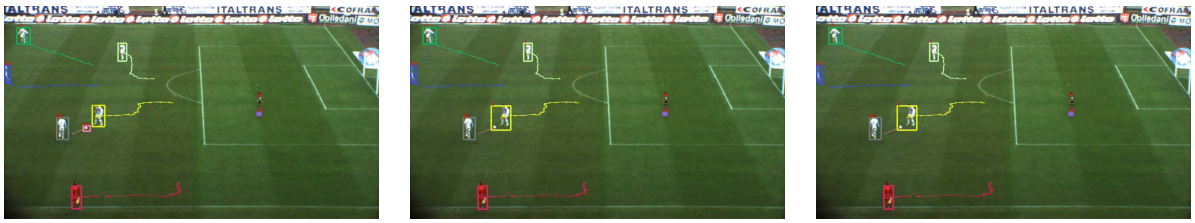


(a) (b) (c)

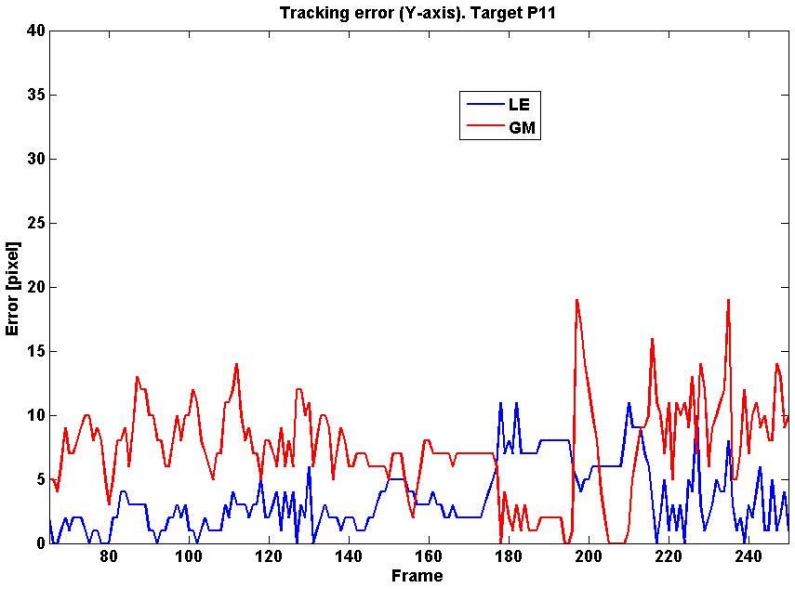


(e)

Figure 4.25: Example of blob split (target P_{18}, frame $55-70$) and resulting tracking errors. (a), (b) and (c) blob split; (d) errors.



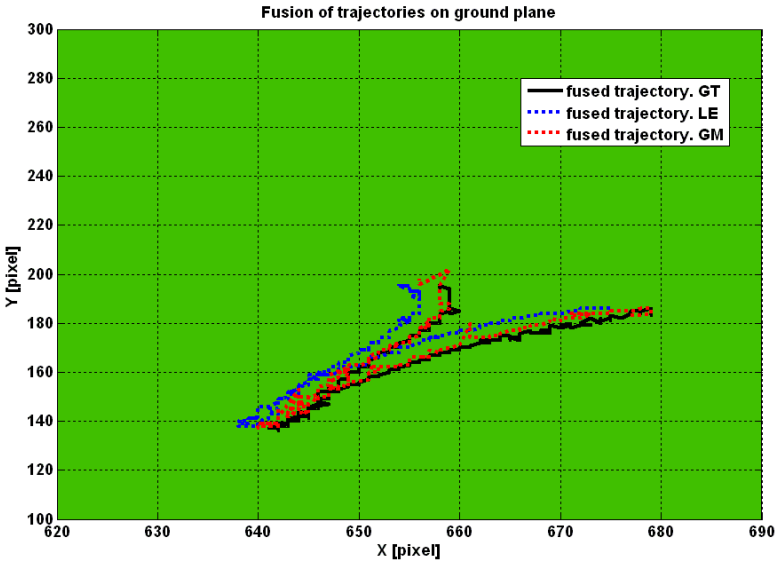
(a) (b) (c)



(e)

Figure 4.26: Example of blob merge (target P_{11}, frame $40 - 240$) and resulting tracking errors. (a), (b) and (c) blob merge; (d) errors.

(a) (b)



(c)

Figure 4.27: Example of trajectory fusion on the ground plane. (a) Example of frame from View 1 and View 2; (c) Fused trajectories on the ground plane.

		LE			Proposed algorithm		
Object ID	Occurrences	detection	ID switches	stability	detection	ID switches	stability
P_{18}^{C1}	2882	2867	7	**0.952**	2857	172	**0.096**
P_{11}^{C1}	448	436	3	0.975	444	2	0.993
P_{12}^{C1}	145	135	7	0.486	132	9	0.336
P_{13}^{C1}	358	351	2	0.981	357	1	**1**
P_{14}^{C1}	148	148	2	0.940	147	8	0.631
P_{15}^{C1}	253	246	3	0.969	187	7	0.654
P_{16}^{C1}	90	77	2	0.879	89	2	0.945
P_{17}^{C1}	369	358	5	0.919	368	4	0.627
P_{19}^{C1}	53	44	3	0.815	52	1	**1**
P_{110}^{C1}	210	192	5	0.673	198	3	0.607
P_{111}^{C1}	19	12	3	0.600	16	3	0.550
P_{112}^{C1}	28	18	2	0.621	26	4	0.690
P_{113}^{C1}	347	328	2	0.966	344	20	0.724
P_{114}^{C1}	78	63	2	0.797	78	1	**1**
P_{115}^{C1}	781	719	19	**0.327**	744	9	**0.285**
P_{116}^{C1}	83	67	3	0.798	82	2	0.976
P_{117}^{C1}	25	2	2	0.885	24	3	0.808
P_{118}^{C1}	766	728	11	0.494	495	10	0.412
P_{119}^{C1}	604	515	22	**0.207**	500	43	**0.210**
P_{120}^{C1}	700	673	8	0.524	695	6	0.522
P_{121}^{C1}	584	359	4	0.793	579	15	0.346
P_{122}^{C1}	652	631	6	0.580	588	8	0.588
P_{123}^{C1}	736	721	14	0.259	723	36	0.374
P_{124}^{C1}	560	542	8	0.533	522	5	0.446
P_{125}^{C1}	477	422	10	0.588	469	32	0.289
P_{126}^{C1}	513	505	4	0.529	511	7	0.508
P_{127}^{C1}	540	531	6	0.640	533	4	0.636
P_{128}^{C1}	463	445	10	0.448	315	10	0.386
P_{129}^{C1}	306	301	5	0.893	303	4	0.687
P_{130}^{C1}	228	215	4	0.755	220	24	0.624
P_{131}^{C1}	260	247	4	0.686	255	7	0.563
P_{132}^{C1}	149	137	3	0.927	148	3	0.967
P_{133}^{C1}	101	98	2	**0.931**	95	7	**0.441**
P_{134}^{C1}	87	87	2	0.898	79	2	0.977
P_{135}^{C1}	109	108	2	0.909	100	2	0.982
P_{11}^{C3}	477	465	3	**0.749**	340	39	**0.377**
P_{12}^{C3}	434	397	8	0.554	274	5	0.543
P_{13}^{C3}	411	299	7	0.497	256	17	0.398
P_{14}^{C3}	374	356	10	**0.448**	361	29	**0.184**
P_{15}^{C3}	423	416	3	0.892	418	3	0.877
P_{16}^{C3}	424	414	6	**0.951**	418	9	**0.468**
P_{17}^{C3}	391	376	9	0.367	339	9	0.474
P_{18}^{C3}	386	374	10	0.517	383	5	0.555
P_{19}^{C3}	372	363	11	0.**609**	293	28	**0.252**
P_{110}^{C3}	141	137	4	0.560	139	3	0.500
P_{111}^{C3}	141	137	2	**0.972**	139	11	**0.451**

Figure 4.28: Tracking stability of the *LE* and the proposed algorithm. Here, the stability in object tracking represents the number of times this object's ID has changed.

4.7 Summary

Results for trajectory transformation show a reduction in errors - the difference between the expected and measured positions of transformed trajectory points - in the object tracking when using statistical technique in homography estimation [132] as opposed to linear methods [133]. The proposed approach allows the recovery of a complete trajectory across views, in sport scenes, where objects' paths often happen to cross and are being interrupted. The integration of statistical homography estimation and the correction of radial distortion in camera lenses enhances accuracy in objects' correspondence across views whilst achieving lower residuals in image alignment [134]. The absence of mechanisms that detect blobs' merging/split generates erratic positions in object's localization and undermines accuracy of the proposed approach in situations of occlusion. The analysis suggests that respective blocks in the two algorithms can either be combined, dropped or augmented to obviate to problems encountered. The proposed approach presents a noticeable advantage in terms of accuracy in object location because it is detection-based. However, in LE, the prediction in object's localization can improve smoothing the trajectories. There is improvement when detections are combined with predictions. The blob split and merge block is an important block that is present in LE and needs to be integrated into the framework of a future tracking algorithm. This will allow the improvement of tracking stability and reduce the object identity switches.

Chapter 5

Conclusions

5.1 Summary of achievements

We have presented an algorithm for the integration of trajectories of objects that are simultaneously viewed in multiple cameras. The proposed approach performs a homography-based trajectory transformation onto a common view whilst reducing registration errors. We operate in the context of a wide baseline configuration and static cameras. Using the transfer errors, we have demonstrated an improvement of trajectory transformation in terms of accuracy as compared to the traditional linear (SVD) and non linear (LSM) techniques. Further reductions of errors were achieved by embeddeding lens distortion correction in the algorithm for trajectory transformation. The proposed approach estimates homography from multiple overlapping uncalibrated cameras and then blends them to generate mosaics on which object trajectories are registered. Alternatively, the estimated homographic transformation is used to map trajectories on a scene top view. To obtain objects' complete trajectories across views, from the segments generated in each camera, we have proposed a trajectory association and fusion that operates on a scene ground plane. The association is based on a similarity metric that, within overlapping regions, identifies fragments of transformed trajectories generated by each object. These

fragments are fused and connected across the fields of view using temporal consistency and object identity. We have presented an estimation of tracking errors in terms of the Euclidean distance between the expected results from ground truth and those measured with real data. From the analysis of these errors, we have identified the limitations of the proposed approach and proposed enhancing mechanisms to overcome them.

The proposed algorithm is suitable for use as either a stand-alone application for multiple view analysis or a geometric cue to be integrated into an object tracking algorithm for multiple cameras.

5.2 Future work

In the light of discussions on the experiments, we propose three main directions for further works:

- **Use of a feedback loop**

  To insert a two-level feedback process that allows the detection of a drifting process in object tracking and simultaneously, the collaboration between the cameras in a distributed system. At the local level, the output of the tracking module will be fed-back to that of the motion segmentation to ascertain the coherence of objects' tracks. At the global level, the final position of an object will be back-projected onto the corresponding image planes sources. This latter feed-back loop aims at exploiting the multiple camera set by reinforcing the detection/tracking output between corresponding areas across views.

- **Use of spatial uncertainty in 2D location**

  The proposed approach in this thesis relies on noisy feature points extracted from object tracking and on the trajectory transformation that is subject to strong collinearity constraint. Therefore, it is important in future to consider an estimate over the uncertainty with which the trajectory data is known. The nature of the fluctuation

of measurements giving object's position is noted in Douxchamps's work [52]. We suggest, for future works, an initial training phase that collects data and estimates tracking errors on the image plane. Additionally, a map that estimates the magnitude of perspective distortion in the common view (scene top view or image mosaic) will be stored in order to estimate the expected homographic errors in all the area of the common view. These two pieces of information model the relative reliability of each source when computing the objects' final position.

- **Distributed sensing**

  Another relevant direction to explore is the protocols of transmission of information between the cameras themselves, and between the cameras and a central server of camera networks [135, 136]. This research will simulate the transmission of metadata in a network, study the impact of delays and losses at destination with regards to the information on objects' spatio-temporal localization. The goal is to quantify the results of the differences between centralised and distributed configurations in multiple view object tracking.

Bibliography

[1] O. Javed and M. Shah. *Automated Multi-Camera Surveillance: Algorithms and Practice*. Springer Publishing Company, Incorporated. Springer-Verlag US, 2008.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004.

[4] O. Faugeras, Q.-T. Luong, and T. Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, Cambridge, MA, USA, 2001.

[5] R. A. Romano. *Projective minimal analysis of camera geometry*. Ph.D. thesis, 2002. Supervisor-Grimson, W. Eric and Supervisor-Faugeras, Olivier D.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second ed. Cambridge University Press (U.K), 2004.

[7] K. Kanatani, N. Ohta, and Y. Kanazawa. Optimal homography computation with a reliability measure. *Trans. on Information and Systems. Special Issue on Machine Applications*, 7:1369–1374, 2000.

[8] J. Black and T. Ellis. Multi camera image tracking. *Image and Vision Computing*, 24(11):1256–1267, 2006.

[9] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Pittsburgh, PA, 2000.

[10] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004.

[11] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Survey*, 38(4), 2006.

[12] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 744–750. 2006.

[13] P. Kornprobst and G. Medioni. Tracking segmented objects using tensor voting. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 118–125. 2000.

[14] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.

[15] F. Tang and H. Tao. Object tracking with dynamic feature graph. In *Proc. of IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 25–32. 2005.

[16] M. Taj, E. Maggio, and A. Cavallaro. Multi-feature graph-based object tracking. In *CLEAR, Springer LNCS 4122*, pages 190–199. Southampton, UK, 2006.

[17] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion

in crowds. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 594–601. 2006.

[18] Z. Khan, 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(11):1805–1918, 2005.

[19] S. Mckenna. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.

[20] R. Rosales and S. Sclaroff. 3d trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 117–123. 1999.

[21] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proc. of European Conf. on Computer Vision*, pages 702–718. 2000.

[22] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 789–795. 2003.

[23] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24(11):1233–1243, 2006.

[24] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 199–206. 2001.

[25] H. Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):75–89, 2002.

[26] K. Okuma, A. Taleghani, N. D. Freitas, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. of European Conf. on Computer Vision*, pages 28–39. 2004.

[27] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 666–673. 2006.

[28] Y. Ricquebourg and P. Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):797–808, 2000.

[29] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1531–1536, 2004.

[30] J. Maccormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *Int. Jrnl. of Computer Vision*, 39(1):57–71, 2000.

[31] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, page 1182. 2005.

[32] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1208–1221, 2004.

[33] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 406–413. 2004.

[34] M. Isard and J. Maccormick. Bramble: a bayesian multiple-blob tracker. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 34–41. 2001.

[35] 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.

[36] O. Javed. *Scene monitoring with a forest of cooperative sensors*. Ph.D. thesis, Orlando, FL, USA, 2005. Major Professor-Shah, Mubarak.

[37] J. Ren, M. Xu, James, and G. A. Jones. Real-time modeling of 3-d soccer ball

trajectories from multiple fixed cameras. *IEEE Trans. on Circuits Systems and Video Technology*, 18(3):350–362, 2008.

[38] Y. Sheikh and M. Shah. Trajectory association across multiple airborne cameras. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):361–367, 2008.

[39] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *In Proc. of the Int. Conf. on Multimedia*, pages 58–67. 2007.

[40] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and J.-L. Van Gool. Color-based object tracking in multi-camera environments. In *DAGM-Symposium*, pages 591–599. 2003.

[41] W. Du and J. Piater. Multi-view tracking using sequential belief propagation. In *Proc. of the Asian Conference on Computer Vision*. 2006.

[42] J. Kang, I. Cohen, and G. Medioni. Tracking people in crowded scenes across multiple cameras. In *Proc. of the Asian Conference on Computer Vision*. 2004.

[43] T. D'Orazio, M. Leo, P. Spagnolo, P. Mazzeo, N. Mosca, and M. Nitti. A visual tracking algorithm for real time people detection. In *Int. Workshop on Image Analysis for Multimedia Interactive Services, 2007*. 2007.

[44] S. Calderara, R. Cucchiara, and A. Prati. Bayesian-competitive consistent labeling for people surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):354–360, 2008.

[45] A. Goshtasby. *2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications*. Wiley, 2005.

[46] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: establishing a common coordinate frame. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):758–767, 2000.

[47] R. Patil, P. E. Rybski, T. Kanade, and M. M. Veloso. People detection and tracking in high resolution panoramic video mosaic. In *Int. Conf. on Intelligent Robots and Systems*, pages 1323–1328. 2004.

[48] J. Kang, I. Cohen, and G. Medioni. Continuous multi-views tracking using tensor voting. In *Proc. of the Workshop on Motion and Video Computing*. 2002.

[49] J. B. Gao and C. J. Harris. Some remarks on kalman filters for the multisensor fusion. *Information Fusion*, 3(3):191–201, 2002.

[50] S. Dockstader and A. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proc. of the IEEE*, 89(10):1441–1455, 2001.

[51] G. Wang, H.-T. Tsui, Z. Hu, and F. Wu. Camera calibration and 3d reconstruction from a single view based on scene constraints. *Image and Vision Computing*, 23(3):311–323, 2005.

[52] D. Douxchamps and K. Chihara. High-accuracy and robust localization of large control markers for geometric camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):376–383, 2009.

[53] J.-S. Kim and I. S. Kweon. Camera calibration based on arbitrary parallelograms. *Computer Vision and Image Understanding*, 113(1):1–10, 2009.

[54] Q. Cai and K. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:1241–1247, 1999.

[55] H. Jiang, S. Fels, and J. Little. Optimizing multiple object tracking and best view video synthesis. *IEEE Trans. on Multimedia*, 10(6):997–1012, 2008.

[56] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31(2):133 – 147, 2007.

[57] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, 2008.

[58] 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*, pages 187–194. 2004.

[59] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –259 Vol. 2. 1999.

[60] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl. Autonomous multicamera tracking on embedded smart cameras. *EURASIP Journal on Embedded Systems*, 2007(1):35–35, 2007.

[61] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proc. of IEEE Int Workshop on Visual Surveillance*, pages 3–10. 2000.

[62] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[63] 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.

[64] H. Kim, Y. Seo, S. Choi, and K. Hong. Where are the ball and players? soccer game analysis with color-based tracking and image mosaic. In *Lecture Notes in Computer Science Volume 1311*, pages 976–983. 1997.

[65] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. Jones. A multi-agent frame-

work for visual surveillance. In *Proc. of Int Conf. on Image Analysis and Processing*. 1999.

[66] I. Junejo and H. Foroosh. Trajectory rectification and path modeling for video surveillance. In *Proc. of IEEE Int. Conf. on Computer Vision*. 2007.

[67] I. Sachiko and S. Hideo. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Proc. of the IEEE Int. Conf. on Pattern Recognition*, pages 751–754. Washington, DC, USA, 2004.

[68] Y. L. de Meneses, P. Roduit, F. Luisier, and J. Jacot. Trajectory analysis for sport and video surveillance. *Electronic Letters on Computer Vision and Image Analysis*, 5(3):148–156, 2005.

[69] J. S. Franco and E. Boyer. Fusion of multi-view silhouette cues using a space occupancy grid. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 1747–1753. 2005.

[70] D. Yang, H. Gonzalez-Banos, and L. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 122–129. 2003.

[71] S. Thrun. Learning occupancy grids with forward sensor models. *Autonomous Robots*, 15:111–127, 2002.

[72] S. M. 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.

[73] P. H. Kelly, A. Katkere, D. Y. Kuramura, S. Moezzi, and S. Chatterjee. An architecture for multiple perspective interactive video. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 201–212. 1995.

[74] R. Jain and K. Wakimoto. Multiple perspective interactive video. *Proc. of Int. Conf. on Mathematical Sciences*, 1995.

[75] K. Sato, T. Maeda, H. Kato, and S. Inokuchi. Cad-based object tracking with distributed monocular camera for security monitoring. In *Proc. of the Workshop on CAD-Based Vision Workshop*, pages 291–297. 1994.

[76] P. J. Figueroa, N. J. Leite, and R. M. Barros. Tracking soccer players aiming their kinematical motion analysis. *Trans. on Computer Vision and Image Understanding*, 101(2):122–135, 2006.

[77] T. Misu, S. Gohshi, Y. Izumi, Y. Fujita, and M. Naemura. Robust tracking of athletes using multiple features of multiple views. In *Proc. of the International Conf. of WSCG*, pages 285–292. 2004.

[78] W. Du, J. Hayet, J. Piater, and J. Verly. Collaborative multi-camera tracking of athletes in team sports. In *Workshop on Computer Vision Based Analysis in Sport Environments*, pages 2–13. 2006.

[79] Y. Busnel, L. Querzoni, R. Baldoni, M. Bertier, and A.-M. Kermarrec. On the Deterministic Tracking of Moving Objects with a Binary Sensor Network. In *Proc. of Int. Conf. on Distributed Computing in Sensor Systems*. 2008.

[80] Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *Proc. of European Conf. on Computer Vision*, pages 98–109. 2006.

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

[82] T. H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *Proc. of IEEE Conf. on Multi-Object Tracking*, pages 19–26. 2001.

[83] S. Park and M. M. Trivedi. Analysis and query of person-vehicle interactions in homography domain. In *Proc. of the ACM Int. workshop on Video Surveillance and Sensor Networks*, pages 101–110. 2006.

[84] S. Park and M. M. Trivedi. Multi-perspective video analysis of persons and vehicles for enhanced situational awareness. In *Proc. of Conf. on Intelligence and Security Informatics*, pages 440–451. 2006.

[85] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[86] G. Cross, A. W. Fitzgibbon, and A. Zisserman. Parallax geometry of smooth surfaces in multiple views. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 323–329. 1999.

[87] R. Szeliski and H. Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proc. of Conf. on Computer Graphics and Interactive Techniques*, pages 251–258. New York, NY, USA, 1997.

[88] I. Skog and P. Handel. In-car positioning and navigation technologiesa survey. *IEEE Trans. on Intelligent Transportation Systems*, 10(1):4–21, 2009.

[89] S. Benhimane, E. Malis, P. Rives, and J. Azinheira. Vision-based control for car platooning using homography decomposition. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2161–2166. 2005.

[90] M. Kumar, C. Jawahar, and P. Narayanan. Building blocks for autonomous navigation using contour correspondences. In *Proc. of IEEE Int. Conf. on Image Processing*, volume 2, pages 1381– 1384. 2005.

[91] S. Park and M. M. Trivedi. Analysis and query of person-vehicle interactions in homography domain. In *Proc. of the ACM Int. Workshop on Video Surveillance and Sensor Networks*, pages 101–110. 2006.

[92] A. Agarwal, C. V. Jawahar, and P. J. Narayanan. A survey of planar homography estimation techniques. Technical Report IIIT Technical Report IIIT/TR/2005/12, Centre for Visual Information Technology; International Institute of Information Technology, 2005.

[93] L. Lucchese. A hybrid frequency-space domain algorithm for estimating projective transformations of color images. In *Proc. of IEEE Int. Conf. on Image Processing*, volume 2, pages 913–916 vol.2. 2001.

[94] M. Kumar, S. Kuthirumunal, C. Jawahar, and P. Narayanan. Planar homography from fourier domain representation. In *Proc. of Int. Conf. on Signal Processing and Communications*, pages 560–564. 2004.

[95] L. Lucchese. A frequency domain technique based on energy radial projections for robust estimation of global 2d affine transformations. *Comput. Vis. Image Underst.*, 81(1):72–116, 2001.

[96] S. Kruger and A. Calway. Image registration using multiresolution frequency domain correlation. Technical report, Bristol, UK, UK, 1998.

[97] P. F. Sturm. General imaging - design, modelling and applications. In *Proc. of Int. Conf. on Computer Vision Theory and Applications*, pages 9–10. 2008.

[98] J. Kaminski and A. Shashua. Multiple view geometry of general algebraic curves. *Int. Jrnl. of Computer Vision*, 56(3):195–219, 2004.

[99] P. Chen and D. Suter. Error analysis in homography estimation by first order approximation tools: A general technique. 33(3):281–295, 2009.

[100] K. Kanatani. Unbiased estimation and statistical analysis of 3-d rigid motion from two views. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(1):37–50, 1993.

[101] K. Kanatani. Statistical bias of conic fitting and renormalization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(3):320–326, 1994.

[102] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. 1996.

[103] P. K. Saurabh, M. P. Kumar, S. Goyal, S. Kuthirummal, C. V. Jawahar, and P. J.

Narayanan. Discrete contours in multiple views: approximation and recognition. *Image and Vision Computing*, 22, 2004.

[104] P. K. Mudigonda, P. Kumar, M. C. V. Jawahar, and P. J. Narayanan. Geometric structure computation from conics. In *In ICVGIP*, pages 9–14. 2004.

[105] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 125–132. 2001.

[106] A. Sugimoto. A linear algorithm for computing the homography from conics in correspondence. *Jrnl. of Mathematical Imaging and Vision*, 13(2):115–130, 2000.

[107] D. P. Huttenlocher, G. A. Klanderman, G. A. Kl, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[108] C. Schmid and A. Zisserman. The geometry and matching of curves in multiple views. In *Proc. of European Conf. on Computer Vision*, pages 394+. 1998.

[109] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *Int. Jrnl. of Computer Vision*, 40(3):199–233, 2000.

[110] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Proc. of Workshop on Motion and Video Computing*, pages 169–174. 2002.

[111] Y. Abdel-Aziz and H. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proc. of American Society of Photogrammetry the Symposium on Close-Range Photogrammetry*, pages 1–18. 1971.

[112] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 246–252. 1999.

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

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

[115] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 510–517. IEEE Computer Society, Washington, DC, USA, 2005.

[116] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[117] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1409–1424, 2002.

[118] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *Int. J. Comput. Vision*, 68(1):53–64, 2006.

[119] K. Kanatani. Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis. *Int. Jrnl. of Comput. Vision*, 80(2):167–188, 2008.

[120] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.

[121] G. Kayumbi and A. Cavallaro. Robust homography-based trajectory transformation for multi-camera scene analysis. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*. 2007.

[122] C. Slama. *Manual of Photogrammetry, 4th Ed.* American Society of Photogrammetry, Falls Church, VA, USA, 1980.

[123] F. Devernay and O. Faugeras. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments. *Mach. Vision Appl.*, 13(1):14–24, 2001.

[124] R. Swaminathan, M. D. Grossberg, and S. K. Nayar. A perspective on distortions. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:594, 2003.

[125] J.-P. Tardif, P. Sturm, M. Trudeau, and S. Roy. Calibration of cameras with radially symmetric distortion. 31(9):1552–1566, 2009.

[126] R. Swarninathan and S. Nayar. Non-metric calibration of wide-angle lenses and poly-cameras. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2. 1999.

[127] G. P. Stein. Lens distortion calibration using point correspondences. In *In Proc. CVPR*, pages 602–608. 1996.

[128] Z. Zhang. On the epipolar geometry between two images with lens distortion. In *In Proc. ICPR*, pages 407–411. 1996.

[129] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Proc. of IEEE Int. Conf. on Image Processing*, pages 602–605. 2005.

[130] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. of the IEEE Int. Conf. on Pattern Recognition*, pages 1135–1138. 2006.

[131] Viper: The vide performance evaluation resource. In *http://viper-toolkit.sourceforge.net*. 2005.

[132] G. Kayumbi and A. Cavallaro. Robust homography-based trajectory transformation for multi-camera scene analysis. In *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*. Vienna, Austria, 2007.

[133] J. Black. A novel method for video tracking performance evaluation. In *In Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 125–132. 2003.

[134] G. Kayumbi and A. Cavallaro. Multi-view trajectory mapping using homography with lens distortion correction. *EURASIP Journal on Image and Video Processing*, 2008(September), 2008.

[135] H. Medeiros, J. Park, and A. Kak. Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *Selected Topics in Signal Processing, IEEE Journal of*, 2(4):448–463, 2008.

[136] D. Agrafiotis, T.-K. Chiew, D. R. Ferre, P.and Bull, A. Nix, A. R.and Doufexi, J. Chung-How, and D. Nicholson. Seamless wireless networking for video surveillance applications. In *Proc. of SPIE Conf. on Image and Video Communications and Processing*. San Jose, CA, USA, 2005.