A

Dissertation

On

MULTI-VIEW IMAGE SURVEILLANCE AND TRACKING

Submitted in Partial fulfillment of the requirement

for the award of Degree of

MASTER OF ENGINEERING

(Electronics & Communication Engineering)

Submitted By:

SANJAY RAKHECHA

College Roll No: 20/E&C/07

University Roll No: 12278

Under the Guidance of:

DR. RAJIV KAPOOR

Dept. of Electronics & Communication

Delhi College of Engineering, Delhi

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
DELHI COLLEGE OF ENGINEERING
DELHI UNIVERSITY

2007-2009

CERTIFICATE

This is to certify that the work contained in this dissertation entitled **"Multi View Image Surveillance And Tracking"** submitted by **Sanjay Rakhecha,** University Roll No- 12278 in the requirement for the partial fulfillment for the award of the degree of Master of Engineering in Electronics & Communication Engineering, Delhi College of Engineering is an account of his work carried out under our guidance and supervision in the academic year 2008-2009.

The work embodied in this minor project has not been submitted for the award of any other degree to the best of my knowledge

Dr. RAJIV KAPOOR
Dept. of Electronics and Communication
Delhi College of Engineering, Delhi-42

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extend my heartiest felt gratitude to everybody who helped me throughout the course of this project.

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor Dr. Rajiv Kapoor for their invaluable guidance, encouragement and patient reviews. I am very thankful to Prof. Asok Bhattacharyya, H.O.D Electronics & Communication Department who allows me to do project under the Guidance of Dr. Rajiv Kapoor on Image Processing. With their continuous inspiration only, it becomes possible to complete this dissertation and both of them kept on boosting me with time, to put an extra ounce of effort to realize this work.

I would also like to take this opportunity to present my sincere regards to all the faculty members of the Department for their support and encouragement.

I am grateful to my parents for their moral support all the time; they have been always around to cheer me up, in the odd times of this work. I am also thankful to my classmates for their unconditional support and motivation during this work.

SANJAY RAKHECHA

M.E. (Electronics & Communication Engineering)

College Roll No. 20/E&C/07

University Roll No. 12278

Department of Electronics & Communication Engineering

Delhi College of Engineering, Delhi-42

ABSTRACT

The work presented in this report provides a framework for object tracking using multiple camera views. The application uses several widely separated overlapping and non-overlapping camera views for object tracking in an outdoor environment. Multiple cameras are needed to cover large environments for monitoring activity. To track moving object successfully in multiple perspective images, one needs to establish correspondence between objects captured in multiple cameras. The system, require for tracking in multi-view environment, employs a centralised control strategy, where each intelligent camera unit transmits tracking data to a multi view-tracking server. The tracking data generated by each intelligent camera unit is stored in a central surveillance database during live operation. The data stored in a surveillance database is used to generate pseudo synthetic video sequences, which can be used for performance evaluation.

For overlapping camera views the homography constraint is used to match moving objects in each camera view. The homography is automatically learned by applying a robust search to a set of object trajectories in each overlapping camera view. The system uses symbolic scene information to reason about object handover between non-overlapping viewpoints that are separated by a small temporal distance, of the order of seconds. The major entry and exit regions between each non-overlapping view are used to improve the robustness of predicting where objects should re-appear having left the field of view of an adjacent camera.

Keyword: - Video surveillance, Multiple views, Camera calibration, Tracking in multiple cameras

CONTENTS

1 Introduction

1.1 Tracking: A Brief Review

In day to day life, there has been an increasing interest in image tracking and activity recognition systems. Due to the large amount of applications there those features can be used. Image tracking and activity recognition are receiving increasing attention among computer scientists due to the wide spectrum of applications where they can be used, ranging from athletic performance analysis to video surveillance. By image tracking we refer to the ability of a computer to recover the position and orientation of the object from a sequence of images. There have been several different approaches to allow computers to derive automatically the kinematics pose and activity from image sequences.

Video tracking is the process of locating a moving object in time using a camera. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. The main difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate. Here, video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track.

1.2 Surveillance and Monitoring

Video surveillance is a difficult task. Based on the field of computer vision, the automatic processing of video feeds often requires specialized encoding and decoding hardware, fast digital signal processors, and large amounts of storage media.

Image surveillance and monitoring is an area being actively investigated by the machine vision research community. Machine vision based surveillance systems can be applied to a number of application domains, for example retail outlets, traffic monitoring, banks, city centres, airports and building security, with each domain having its own specific requirements.

The best example of machine vision based surveillance system is Closed Circuit Television (CCTV) technology. However, even with these technological advances, there is still the problem of how information in such a surveillance network can be effectively managed. CCTV networks are normally monitored by a number of human operators located in a control room containing a bank of screens streaming live video from each camera.

Human operators are presented with a number of issues when monitoring a bank of video terminals. One problem is how to reliably navigate through the environment using each camera in the CCTV network. Each camera has a limited Field Of View (FOV) of the region, and hence it is necessary to switch between camera views appropriately to track suspicious individuals as they walk through the scene. In manually operated environments it has been observed that human operators have difficulty in performing this task if they are not completely familiar with the scene and placement of each camera. Second issue is that human operators are normally only interested in identifying certain events that can occur in the scene, for example crowd congestion on a railway platform or other atypical behaviour. Ideally, a system that could automate this task would be of great benefit, particularly in the CCTV network comprised of a large number of cameras. Another issue is the ability of humans to concentrate on multiple videos simultaneously is limited. Therefore, there has been an interest in developing computer vision systems that can analyse information from multiple cameras simultaneously and possibly present it in a compact symbolic fashion to the user.

A machine vision based solution for a visual surveillance application would comprise of many components to address the operations ranging from low-level video acquisition and pre-processing to high-level object tracking and visual

interpretation. Live video data can be acquired by using frame-grabbing hardware, which is available at a relatively low cost. The reduced hardware costs have made it economically feasible to deploy networks of cameras to perform visual surveillance tasks. The frame-grabbers also have an application-programming interface (API) in order to allow software to be developed for integration with the hardware.

Once live video data can be captured and stored, the next step of the surveillance application would be to identify any object activity within the camera field of view. This task is normally referred to as motion segmentation and requires that the surveillance application utilise vision algorithms to automatically detect possible moving objects of interest. This presents many challenges, since the motion detection must be robust with respect to illumination changes, and irrelevant motion. Illumination changes typically occur in outdoor environments due to varying weather conditions. For example, the appearance and disappearance of the sun on a partially cloudy day causes significant changes in lighting conditions and cast shadows.

In order to reduce the affect of each of these sources of error it has become common to employ adaptive background modeling techniques to provide a robust solution. A background subtraction process is then applied to identify possible moving objects of interest. Once moving objects have been identified in the camera view the next task of the visual surveillance system is perform feature extraction and tracking. Features extracted from detected objects can comprise of location, shape and colour. Feature extraction is important, since it provides a mechanism to represent each moving object using a compact model.

An important requirement for a machine vision based surveillance system is to be able to preserve the identity of an object as it moves through the field of view of the camera. This presents an additional challenge to the motion segmentation problem, since it is necessary to establish inter frame object correspondence between each captured image frame. This task can be resolved by employing an object tracking algorithm, which takes as input a set of detected object features and

attempts the maintain the correct tracked state of each object. Once a tracked object has been matched to a detected foreground object the measurement is used to update the state of the tracked object. New tracked objects can be created for any foreground objects that have not been assigned to an existing tracked object during the data association process.

For real world applications a visual surveillance system would comprise of many cameras, so a method is required to integrate all the tracking information from the multiple camera sources. This presents a number of additional issues than faced for single view tracking. Firstly, it is a necessary requirement to assign a unique identity to moving objects even if they are visible in more than one camera view simultaneously. Secondly, the identity of an object should be preserved when it moves between non overlapping camera views. Using multiple camera views for object tracking offers some advantages over single view tracking. With sensible camera placement the system would have an increased field of coverage, since the field of views of all the cameras could be combined by the system.

One common cause of single view tracking failure is due to Dynamic and Static Object Occlusions. A Dynamic Occlusion occurs when two objects interact or cross each other's path within the camera view. A Static Occlusion typically occurs when an object temporarily disappears from the camera field of view due to an occlusion plane, for example a tree located near a pedestrian path. Since multiple camera views provide a larger field of coverage it is expected that a multi view camera surveillance system should be capable of resolving dynamic and static occlusions better than single view tracking, since the start and end of an occlusion should occur at different times in each camera view increasing the possibility that the system should be able to correctly track the occluded objects.

1.3 Motivating Example

As a motivating example, consider the overhead view of a surveilled area as seen in Figure 1.1. Cameras A and B are disjoint – they look at different areas of the world and do not overlap. However, cameras A and C partially overlap, as do

cameras B and C. An object in either of the darker overlapping areas will be visible to two cameras simultaneously.

Now examine the output of the three cameras. There are two people in the world. However, between the three cameras they have been given four different labels: A-8, B-2, C-4, and C-5. Given these object labels, the most important piece of information that we could find is which labels refer to the same real-world objects. This is the consistent labeling problem.

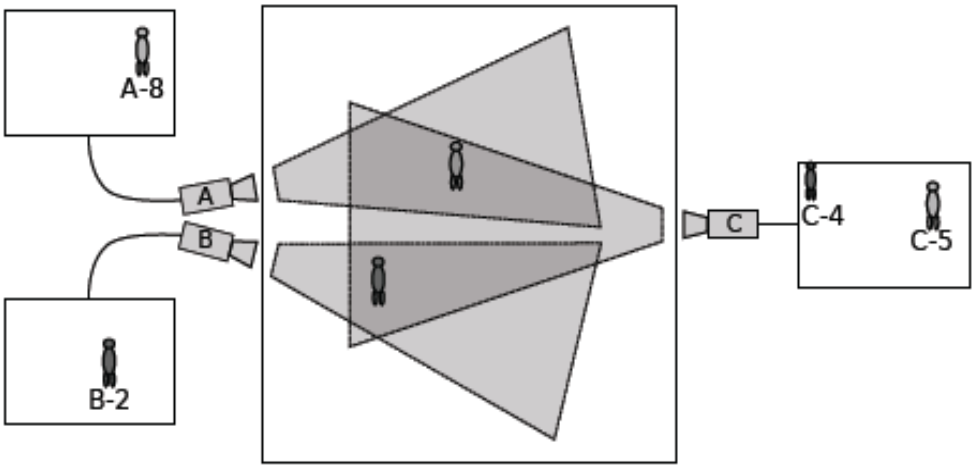

Figure 1.1: Three cameras look at the same general area in this overhead view. Across the three cameras, two targets are given four tracking labels.

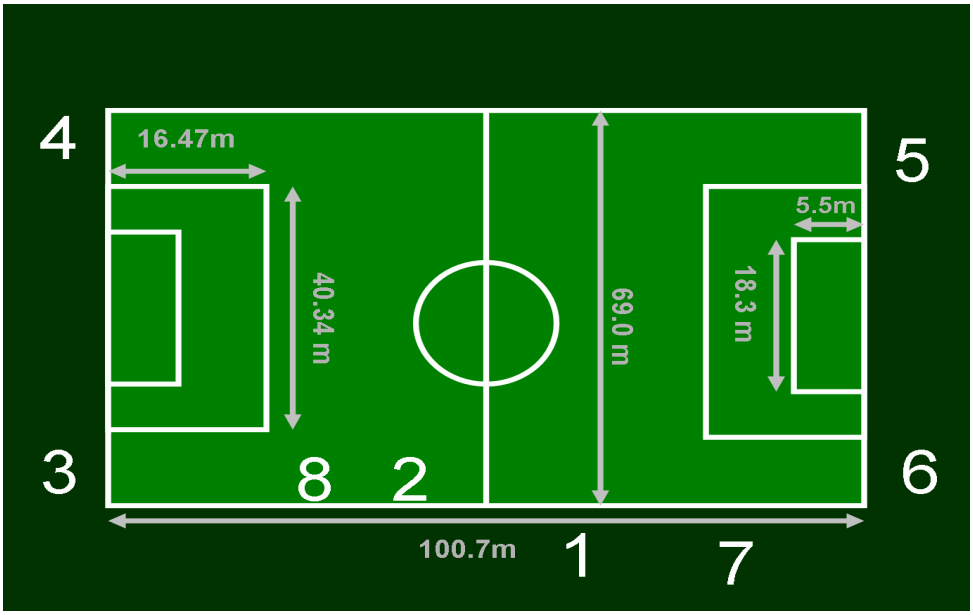

Figure 1.2: The eight cameras are placed in a configuration that covers the entire football pitch

1.4 Research Aims and Objectives

The main focus of this research was to create a framework for tracking objects using multiple camera views. Object tracking using multiple views has recently received much attention. The obvious benefits of tracking using multiple cameras are increased coverage of the scene, since each of the combined field of views of all the cameras should be greater than that of any individual camera. Using multiple camera views for object tracking, increases the possibility of preserving object identity across the region.

Initially, moving objects of interest must be identified in each camera. This represents a challenging problem, particularly in outdoor environments where lighting conditions cannot be controlled and image intensities are subject to large changes in illumination variation. It is assumed that each camera view is fixed and calibrated in a world coordinate system. The multi view object-tracking framework should be able to integrate tracking information from each camera and reliably track objects between views. In addition the multi view tracker should be able to resolve both dynamic occlusions that occur due to object interaction, and static occlusions that can occur due to the scene constraints, for example trees that form occlusion regions.

Tracking multiple people accurately in cluttered and crowded scenes is a challenging task primarily due to occlusion between people. If a person is visually isolated (i.e., neither occluded nor occluding another person in the scene), it is much simpler to perform the tasks of detection and tracking. This is because the physical attributes of the person's foreground blob, like color distribution, shape, and orientation, remain largely unchanged as he/she moves. Increasing the density of objects in the scene increases interobject occlusions. A foreground blob is no longer guaranteed to belong to a single person and may belong to several people in the scene. Even worse, a person might be completely occluded by other people. Under such conditions of limited visibility and clutter, it might be impossible to detect and track multiple people using only a single view. The next logical step is

to use multiple views of the same scene in an effort to recover information that might be missing in a particular view. In this paper, we propose a multiview approach to detect and track multiple people in crowded and cluttered scenes. We are interested in situations where the scene is sufficiently dense that partial or total occlusions are common and it cannot be guaranteed that any person will be visually isolated.

My method of detection and occlusion resolution is based on geometrical constructs and requires only the distinction of foreground from background, which is obtained using standard background modeling techniques. At the core of our method is a planar homographic occupancy constraint [1,2] that combines foreground likelihood information (probability of a pixel in the image belonging to the foreground) from different views to resolve occlusions and determine regions on scene planes that are occupied by people. The homographic occupancy constraint interprets foreground as scene occupancy by non-background objects (in effect using cameras as occupancy sensors) and states that pixels corresponding to occupancies on a reference plane will consistently warp (under homographies of the reference plane) to foreground regions in every view.

In order to achieve the requirements of online tracking, and video tracking performance evaluation framework, system architecture will be designed and implemented, which will support the real time capture and storage of object tracking data from multiple cameras. The captured data will comprise of object track information such as location, appearance features, bounding box dimensions, and pixel image data of each detected object. Central to the operation of the system will be a multiview tracking server (MTS), which will integrate all the tracking data observed by each camera in the surveillance network. Another design consideration is how the video data will be stored and managed for retrieval, particularly if the system runs continuously for many days, which would result in large quantities of tracking data. One requirement of any surveillance system is that it should be possible to access video data for specific times and dates. This functionality has been implemented in a surveillance database, which is appropriately indexed to support fast retrieval of data.

1.5 Scope – Goals

This thesis covers the development, implementation, and testing of a multiple camera surveillance algorithm. The algorithm shall have the following characteristics:

I. Independent of camera extrinsic parameters, i.e. location and orientation. The algorithm should smoothly handle widely disparate views of the world.

II. Independent of camera intrinsic parameters, i.e. focal length, pixel skew, and location of principal point. Different cameras are available on the market – the algorithm should be able to handle multiple focal lengths, differences in resolution, and the like.

III. Independent of camera modality. The algorithm should be able to handle the output of any single-camera tracker. The algorithm should not depend on whether the underlying camera hardware is RGB, near-infrared, thermal infrared, or some other image-forming technology.

IV. Solve the consistent labeling problem. One real-world target should be linked to one object label in each of the cameras in which that target is visible.

V. Robust to target occlusions and in-scene entrances. If a target enters the surveilled area in the middle of a scene, say, through a door, then the algorithm should correctly solve the consistent labeling problem. Similarly, if one target splits into two, as when two close people take different paths, the algorithm should identify and correctly label the two targets.

VI. Simple to set up. No camera calibration should be required. Training, if needed, should take as little time as possible and should be done with

normal in-scene traffic. Training should be automatic and should not require operator intervention.

VII. Capable of camera cueing. The algorithm should be able to determine which cameras should be able to see a given target.

1.6 Scope – limitations

The scope of the algorithm shall be limited as follows:

I. Frame rates will be sufficient to allow the single-camera tracking algorithm to work properly.

II. The algorithm shall be used for tracking walking people. Vehicles, animals, and other classes of moving objects are not included in the scope of this thesis.

III. Pairs of cameras to be processed will have at least partially overlapping fields of view. This requires the operator to make an initial judgment when installing the hardware and initializing the algorithm: to decide which cameras see the same parts of the world.

IV. The cameras shall be static. Once installed, both the intrinsic and extrinsic parameters of the camera shall be fixed. This means that a camera cannot be mounted on a pan-tilt turret, or if it is, the turret must not move.

V. The output images of the video cameras will be of a practical size. The algorithm will not include single-pixel detectors (e.g. infrared motion detectors, beam-breaking light detectors). This limitation is necessary to ensure that single-camera tracking is possible without major changes to the chosen algorithm.

VI. The cameras shall approximate regular central-projection cameras with basic pinhole optics. Cameras with extremely wide fields of view – fisheye lenses – or significant un-corrected distortions will not be used.

VII. Most importantly, the targets shall be walking on a ground plane. The overlapping area between any two cameras shall not have significant deviations from a planar surface. Code to deal with hilly areas or steps shall not be included in this algorithm.

VIII. The cameras shall not be located on the ground plane. This prevents a degenerate condition in the scene geometry, as shall be shown later.

2 Previous Work

2.1 Background

The purpose of this chapter is to provide a survey of the research that has already been published in relation to multi view object tracking, and video tracking performance evaluation. Some of the work discussed is outside the scope of this thesis including: adaptive background modelling, motion segmentation, and single view tracking but are included for completeness. In the previous chapter we discussed some the general issues that would be encountered when developing a surveillance application. We discuss some solutions to some of the surveillance tasks identified. This survey of existing multiple view tracking systems and methods of performance evaluation enabled us to identify the key requirements that needed to be considered by this research.

2.2 Single View Tracking

There are a number of techniques available for single view tracking. The tracking problem is primarily decomposed into a number of stages, which include: motion detection, object segmentation, and object tracking. The tracking system developed at MIT Media laboratory [3, 4] used a used a real-time tracking algorithm that uses contextual information. The system could track and analyse the actions and interactions of people and objects. The contextual information included knowledge about the objects being tracked and their current relationships between one another. The contextual information was used to weight the image features used for inter frame data association. Each object was detected using background subtraction allowing the blob's dimensions, location, and colour appearance attributes to be computed.

The W4 system [5] employed a set of techniques for implementing a real-time surveillance system using low cost hardware. The key components of W4 were: adaptive background modelling to statistically detect foreground regions, object classification to distinguishing between different object classes using shape and motion cues, and tracking multiple objects simultaneously in groups. The blob representation of objects particularly has problems when objects interact and form a dynamic occlusion. Using a blob representation it is not possible to distinguish between the foreground regions of each object. The W4 system uses an alternative appearance model for each tracked object that takes the form of silhouette description that includes the location of the head, hands, feet and torso.

Pfinder (Person-finder) was a real-time system for tracking and interpretation of human motion developed at the Massachusetts Institute of Technology (MIT) [6]. Motion detection is performed using background subtraction, where the statistics of background pixels are recursively updated using a simple adaptive filter. The human body is modelled as a connected set of blob regions using a combination of spatial and colour cues. Features of the human body are found by analysis of the foreground object's contour. The system can only track one human object, in future work they plan to extend Pfinder to use multiple cameras. Pfinder has been applied for a variety of applications including: video games, distributed virtual reality, providing interfaces to information spaces, and recognising sign language.

Blob tracking is a popular low-cost approach for tracking objects [7]. It entails extracting blobs in each frame, and tracking is performed by associating blobs from one frame to the next. The Bramble system [8] for example, is a multi blob tracker that generates a blob-likelihood based on a known background model and appearance models of the tracked people. Its performance degrades when multiple objects merge into one blob due to proximity or occlusions. Alternate approaches maintain explicit object states with position, appearance, and shape. Zhao and Nevatia [9] present interesting results when tracking multiple people with a single camera. They use articulated ellipsoids to model human shape, color

histograms to model different people's appearance, and an augmented Gaussian distribution to model the background for segmentation. Once moving head pixels are detected in the scene, a principled MCMC approach is used to maximize the posterior probability of a multiperson configuration. This concept of global trajectory optimization was previously explored in [10]. It also forms the basis of our tracking formulation; however, there is an important difference. Our approach utilizes fusion of multiple views at multiple scene planes and trajectory optimization on scene occupancy probabilistic data that combines the task of detection and tracking seamlessly.

2.3 Multi View Tracking Systems

The use of multiple cameras soon becomes necessary when one wishes to accurately detect and track multiple occluding people and compute their precise locations in a complex environment. Multi view tracking techniques intend to decrease the hidden regions and provide 3D information about the objects and the scene by making use of redundant information from different viewpoints.

In order to integrate the track data from multiple cameras, it is useful to consider the visibility of targets within the entire environment, and not just each camera view separately. Four region visibility criteria can be identified to define the different fields-of-view (FOV) available from the network of cameras.

- **Visible FOV** - this defines the regions that an individual camera will image. In cases where the camera view extends to the horizon, a practical limit on the view range is imposed by the finite spatial resolution of the camera or a practical limit on the minimum size of reliably detectable objects.
- **Camera FOV** - encompasses all the regions within the camera view, including occluded regions

- **Network FOV** - encompasses the visible FOV's of all the cameras in the network. Where a region is occluded in one camera's visible FOV, it may be observable within another FOV (i.e. overlap).
- **Virtual FOV** - covers the network FOV and all spaces in between the camera FOV's within which the target must exist. The "boundaries" of the system represent locations from which previously unseen targets can enter the network.

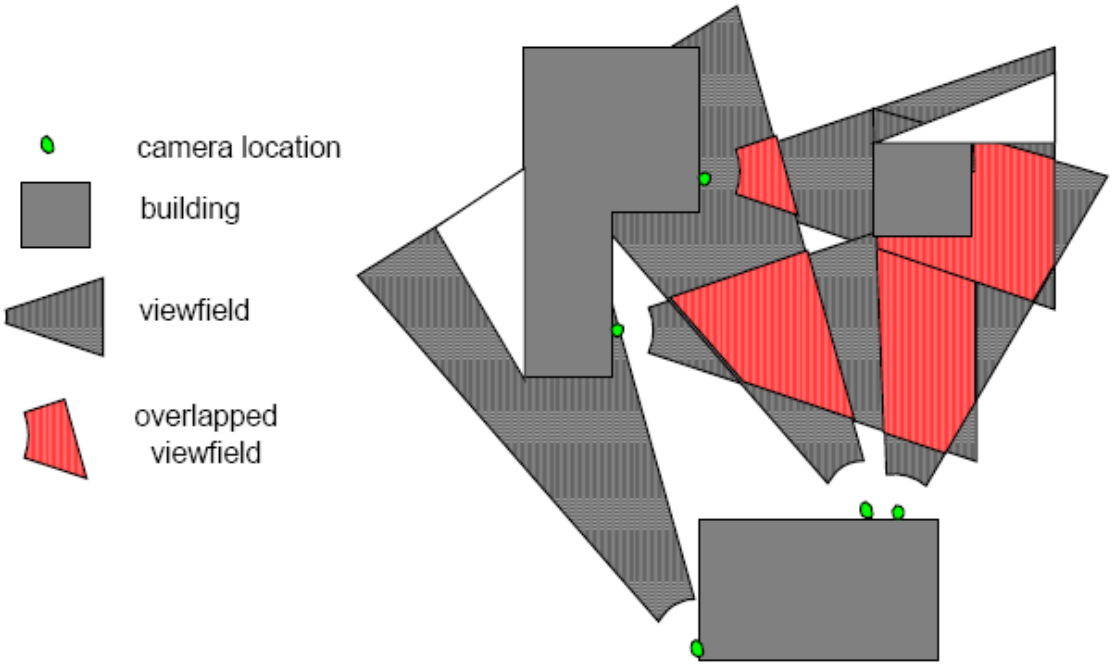

Figure 2.1 Visibility criteria of camera network

Figure 2.1 illustrates the camera network visibility regions for a simple environment projected onto the ground plane. Occluded regions are shown in white (if within the expected view field of a camera). The main requirement of the multi view tracking system is that a unique identity should be assigned for objects tracked within regions of overlap, and the identity should be preserved when objects move between adjacent non-overlapping views.

Cai and Aggarwal [11] extend a single-camera tracking system by starting with tracking in a single camera view and switching to another camera when the

system predicts that the current camera will no longer have a good view of the subject. Spatial matching was based on the euclidean distance of a point to its corresponding epipolar line. In [12] individuals are tracked both in image planes and top view using a combination of appearance and motion models. Bayesian networks were used in several papers as well. In [13] Chang and Gong used Bayesian networks to combine geometry (epipolar geometry, homographies, and landmarks) and recognition (height and appearance) based modalities to match objects across multiple sequences. Bayesian networks were also used by Dockstader and Tekalp in [14] to track objects and resolve occlusions across multiple calibrated cameras. Integration of stereo pairs is another popular approach, adopted by [15] among others. Krumm et al. use stereo cameras and combine information from multiple stereo cameras in 3D space. They perform background subtraction and then detect human-shaped blobs in 3D space. Color histograms are created for each person and are used to identify and track people.

The homography-based tracking by Kalman and particle filtering were presented in [16] respectively. The authors in [16] extracted the principal axes of upright humans tracked in each view and then combined multiple views using planar homography. Homography-based 2D segmentation and tracking of objects has also been studied in the intelligent transportation domain, for instance, the recent work by Park and Trivedi [17]. They propose to combine multiple view data, In [1, 2] which is then augmented with contextual domain knowledge for the analysis and query of person-vehicle interactions for situational awareness and pedestrian safety. Khan and Shah proposed an approach that avoided explicit calibration of cameras and instead utilized constraints on the field of view (FOV) lines between cameras, learned during a training phase, to track objects across the cameras.

The Video Surveillance and Monitoring (VSAM) project at Carnegie Mellon University (CMU) developed a system for multi view surveillance using a distributed network of active sensors. Their system operated in an outdoor environment that presents more challenges than indoor environments, where

lighting conditions can more easily be controlled, which makes motion segmentation an easier task. In outdoor environments lighting conditions can vary due to changing weather conditions or cast shadows. In order to overcome these problems they chose to use an adaptive background model to reflect slow changes in illumination. Each pixel in the background is modelled as a mixture of Gaussians, allowing slow varying changes in illumination, and bi-modal backgrounds (for instance leaves blowing in the wind) to be correctly represented. Foreground objects can be detected by applying a background subtraction technique. One problem with adaptive background modelling is that transient objects, for example a car stopping for a few seconds, can be absorbed into the background model after a period of time. They addressed this problem by employing a layered approach to adaptive background subtraction.

2.4 Summary

The systems and methods illustrate the considerable progress that has been made for video surveillance and performance evaluation. The research presented in this thesis is specifically concerned with multi view object tracking using widely separated views, and developing an automated framework that can be applied for quantitative video tracking performance evaluation. Implementing a system that can be applied for continuous twenty-four hour tracking requires detailed planning and must account for a number of design considerations. Hear we choose to cooperatively track objects between overlapping views to increase the likelihood of resolving both static and dynamic occlusions. Assuming the cameras are widely separated it is likely that dynamic occlusions will start and end at different times in each view increasing the likelihood of success. This issue was not considered where failure of single view tracking would cause failure of the multi view tracking. If the training data contains no object moving between the field of view of two overlapping cameras then the model will not have a handover policy in this region. One solution to this problem is to initialise the FOV boundaries by automatically recovering the homography relations between each pair of

overlapping views. The homography relations can be recovered from a set of sparse 2D object trajectories.

From the review of previously published work a set of general requirements was derived for the system that will be designed and implemented for multi-view surveillance and tracking:

- The system must support robust object tracking for single camera views.
- The system must support robust object tracking between multiple camera views.
- The system must be able to resolve both static and dynamic object occlusions, which are a common reason for tracking failure.
- Tracking between overlapping camera views must be coordinated appropriately to preserve object identity when objects move between the different fields of view of each camera.
- The system must support a temporal synchronisation strategy between multiple views that are located in different locations.
- The system must store the surveillance data in a compact format (preferably a database) that can be easily accessed to support the playback of video captured during a specific time interval.
- The system architecture must support the insertion or removal of an intelligent camera from the surveillance network. In a typical camera network it is likely that devices can fail, or the camera network can be increased in size. This maintenance of the camera network should be performed in a seamless manner.
- It should be possible to calibrate cameras in the surveillance network with limited supervision. This is particularly important in order to match objects between overlapping views, and tracking objects between non-overlapping views.

3 Feature Matching and 3D Measurements

3.1 Background

The objective of this chapter is to examine the methods and techniques available to perform camera calibration, match features between overlapping views, and extract 3D measurements from the scene. Camera calibration is important, since it provides a mechanism to translate 2D image features to a 3D world coordinate system, which can facilitate the integration of tracking information from multiple camera views. The camera calibration also provides a means of making accurate 3D measurements in terms of the world coordinate system, particularly if an object has been matched between several camera views.

This chapter is organised as follows: we first describe the method used to calibrate each camera. We then discuss the approach employed to extract 3D landmark points from the scene being monitored. A homography transformation is employed to correspond 2D object features between overlapping camera views. The homography transformation is utilised by the multi-view tracking framework in order to augment the tracking process. We then describe the techniques used to extract 3D measurements from the scene. A least squares estimation is used to perform 3D line intersection to estimate an object's 3D location using overlapping camera views. A 3D measurement is not of much practical use unless we have some idea of its accuracy, since this has an impact on the reliability of an object's location and consequently on how well the object will be tracked. The measurement uncertainty can be determined by propagating the 2D measurement uncertainty to the world coordinate system by using the calibrated camera parameters. We then discuss the results of homography calibration, along with 3D measurement and uncertainty, for various video sequences to test the validity of each approach.

3.2 Camera Calibration

In order to extract 3D measurements from the scene it is necessary to calibrate each camera within the surveillance system. The calibration model provides a mechanism to translate 2D image coordinates to 3D world coordinates. In general it is most common to derive the calibration information by using a set of known 3D landmark points that are visible within the camera field of view. The calibration model is defined in terms of intrinsic and extrinsic parameters. The intrinsic parameters characterize the internal parameters of the cameras such as: principal point, pixel dimensions, focal length, and radial lens distortion. While the extrinsic parameters describe the cameras position and orientation with respect to the world coordinate system.

It is possible to extract a number of 3D landmark points for a surveillance region. In general, most surveillance regions have a dominant ground plane. The accuracy of the calibration is sufficient for extracting 3D measurements and tracking objects as long as the survey points are sensibly distributed on the ground plane. An example of some of the survey points used in the calibration of cameras connected to the surveillance network is shown in figure 3.1.

The calibrations of each camera are required before start tracking. The parameters (Tx, Ty, Tz) define the translation vector between the world and camera coordinate space. The parameters (Rx, Ry, Rz) define the rotation angles for the transformation between the world and camera coordinate space. K defines the 1^{st} order radial lens distortion coefficient, F is the focal length of the camera, and (Cx, Cy) defines the centre of the radial lens distortion on the image plane.

Figure 3.1 Example of landmark points gathered by a survey of the surveillance region

The calibration errors are dependent on a number of factors including: the number of landmark points, the distribution of the features along the ground plane, the distance of the features from the camera, and the accuracy of the features selected on the image and on the scene ground plane. The accuracy of the camera calibration is to within a few centimetres in the world coordinate system. The accuracy should be sufficient to reliably extract measurements from the scene and track objects in 3D.

3.3 Homographic Occupancy Constraint

Consider a scene containing a reference plane being viewed by a set of wide-baseline stationary cameras. The background models in each view are available and, when an object appears in the scene, it can be detected as foreground in each view using background difference. Any scene point lying inside the foreground object in the scene will be projected to a foreground pixel in every view. This also applies for scene points inside the object that lie on the reference plane except, however, that the projected image locations in each view will be related by homography induced by the reference plane. We can state the following:

Proposition 1: If $\exists P \in R^3$ such that it lies on scene plane π and is inside the volume of a foreground object, then the image projections of the scene point *P* given by $p_1 , p_2 , \ldots . , p_n$ in any n views satisfy both of the following:

- $\forall_i$, if Ψ_i is the foreground region in view i , then $p_i \in \Psi_i$.
- $\forall_{i,j} p_i = \left[H_{i_\pi j}\right] p_j$, where $H_{i_\pi j}$ is the homography induced by plane π from view j to view i.

Warping a pixel from one image to another using the homography induced by a reference scene plane amounts to projecting a ray through the pixel onto the piercing point (point where the ray intersects the reference plane) and then projecting it to the second camera center. If the pixel's piercing point is inside (occupied by) a foreground object in the scene, it follows from Proposition 1 that the pixel will warp to foreground regions in all views. This can be formally stated as follows:

Proposition 2: Let Φ be the set of all pixels in a reference view r and let $H_{i_\pi r}$ be the homography of plane π in the scene from the reference view to view i. If $\exists p \in \Phi$ such that the piercing point of p with respect to π lies inside the volume of a foreground object in the scene, then $\forall_i p_i^{'} \in \Psi_i$, where $p_i^{'} = \left[H_{i_\pi r}\right] p$ and Ψ_i is the foreground region in view i.

We call Proposition 2 the homographic occupancy constraint [1]. Notice that this does not distinguish between foregrounds in different views that may correspond to different objects. It is essentially using camera sensors as scene occupancy detectors with foreground interpreted as occupancy in the line of sight of the image sensor. Although the foreground regions associated across views may correspond to different scene objects (specifically the nearest foreground object in the line of sight of the particular image sensor), the homographic occupancy constraint insures that they all correspond to the same scene occupancy.

This has the dual action of localizing people in the scene as well as resolving occlusion, which is described in Fig. 3.2. Fig. 3.2(a) shows a scene containing a person viewed by a set of cameras. The foreground regions in each view are shown as white on a black background. A pixel which is the image of the feet of the person will have a piercing point on the ground plane (the reference plane for this example) that is inside the volume of the person. According to the homographic occupancy constraint, such a pixel will be warped to foreground regions in all views. This is demonstrated by the pixel in view 1 of Fig. 3.2(a) that has a blue ray projected through it. Foreground pixels that do not satisfy the homographic occupancy constraint are images of points off the ground plane. Due to plane parallax, they are warped to background regions in other views. This is demonstrated by the pixel with a red ray projected through it.

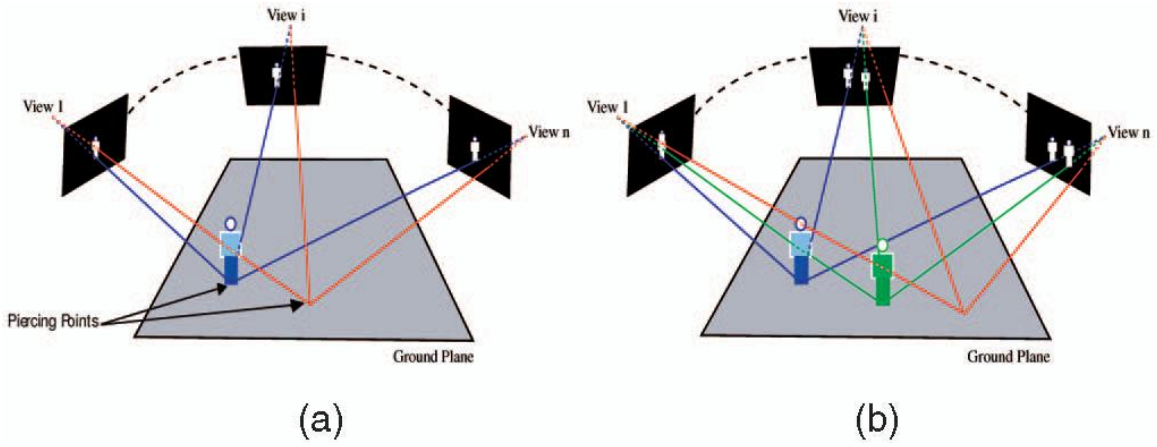

Figure 3.2 The figure shows people viewed by a set of cameras.

Fig. 3.2(b) shows how the homographic occupancy constraint would resolve occlusions. The blue person is occluding the green person in view 1. This is apparent by the merging of their foreground blobs. In such a case, there will be two sets of pixels in view 1 that satisfy the homography constraint. The first set will contain pixels that are image locations of blue person's feet [same as in Fig. 3.2(a)]. The other set of pixels is those that correspond to the blue person's torso region, but are occluding the feet of the green person. Even though these pixels are image locations of points off the ground plane, they have piercing points inside a foreground object, which in this case is the green person. This process creates a seemingly translucent effect detecting feet regions even if they are completely occluded by other people. Clearly, having more people between the blue and the green person will not affect the localization of the green person on the ground plane.

It should be noted that the homographic occupancy constraint is not limited to the ground plane and, depending on the application; any plane in the scene could be used. In the context of localizing people in a surveillance scenario, the ground plane is typically a good choice if it is clearly visible. In other scenarios, a building wall or any planar landmark can be used as the reference plane. In the next section, we develop an operator that uses this approach to localize people on a reference plane.

3.4 Two View Relations

3.4.1 Homography Transformation

A homography mapping defines a planar mapping between two camera views that have a degree of overlap.

$$x' = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+h_{33}} \tag{3.1}$$

$$y' = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+h_{33}} \tag{3.2}$$

Where (x, y) and (x', y') are image coordinates for the first and second camera views respectively. Hence, each correspondence point between two camera views results in two equations in terms of the coefficients of the homography. Given at least four correspondence points allows the homography to be evaluated. It is most common to use Singular Value Decomposition (SVD) for computing the homography [18]. The homography matrix can be written in vector form:

$$H = [h_{11}\ h_{12}\ h_{13}\ h_{21}\ h_{22}\ h_{23}\ h_{31}\ h_{32}\ h_{33}]^T \tag{3.3}$$

Each pair of correspondence points $\left((x, y), (x', y')\right)$ results in two equations in terms of the coefficients of the homography matrix. The following equations can be determined by rearranging equations 3.1 and 3.2.

$$\left[x_i\ y_i\ 1\ 0\ 0\ 0\ -x_i x_i'\ -y x_i'\ -x_i'\right]H = 0 \tag{3.4}$$

$$\left[0\ 0\ 0\ x_i\ y_i\ 1\ -x y_i'\ -y y_i'\ -y_i'\right]H = 0 \tag{3.5}$$

Given N correspondence points a (2N by 9) matrix M that can be constructed and then used to minimise $\|MH\|$ subject to the constraint $\|H\| = 1$ The value of the homography matrix can then be estimated by using Singular Value Decomposition (SVD). Methods such as Gaussian elimination or pseudo inverse could not be applied for the homography estimation, since these techniques cannot handle sets of equations that are singular or numerically very close to singular.

The result of the homography estimation is dependent on the coordinate system of the correspondence points, and their distribution across the image plane. In order to compensate for these differences we also apply an isotropic scaling function to the set of correspondence points, in order to normalise the data. The normalization is performed prior to estimating the homography transform, and reduces the effect of the coordinate system and scale on the estimation. The normalisation function defines a translation and scaling that maps each correspondence point such that centroid of the points is the coordinate origin (0, 0) and the average distance of each point from the origin is $\sqrt{2}$. The additional steps are added to the homography estimation to incorporate the isotropic scaling.

3.4.2 Epipoler Geometry

The epipoler geometry is another method that can be employed for feature correspondence between two overlapping camera views [18]. The key distinction between from the homography method is that feature matching involves a 2D line search, whilst the homography is a point based transform. This has benefits where the ground plane constraint is not valid but this method has increased ambiguity if several objects lie on the epipole line. The epipole geometry is graphically depicted in figure 3.3.

The epipoler geometry exists between any two camera systems. Consider the case of two cameras as shown in Fig. 3.3. Let C and C´ be the optical canters of the first and second cameras, respectively. Given a point **m** in the first image, its corresponding point in the second image is constrained to lie on a line called the *epipoler line* of **m**, denoted by $I_m^{'}$. The line $I_m^{'}$ is the intersection of the planeΠ, defined by **m**, C and C´ (known as the epipoler plane), with the second image plane $I^{'}$. This is because image point **m** may correspond to an arbitrary point on the semi line *C*M (**M** may be at infinity) and that the projection of *C*M on $I^{'}$ is the line $I_m^{'}$. This is called the epipoler constraint.

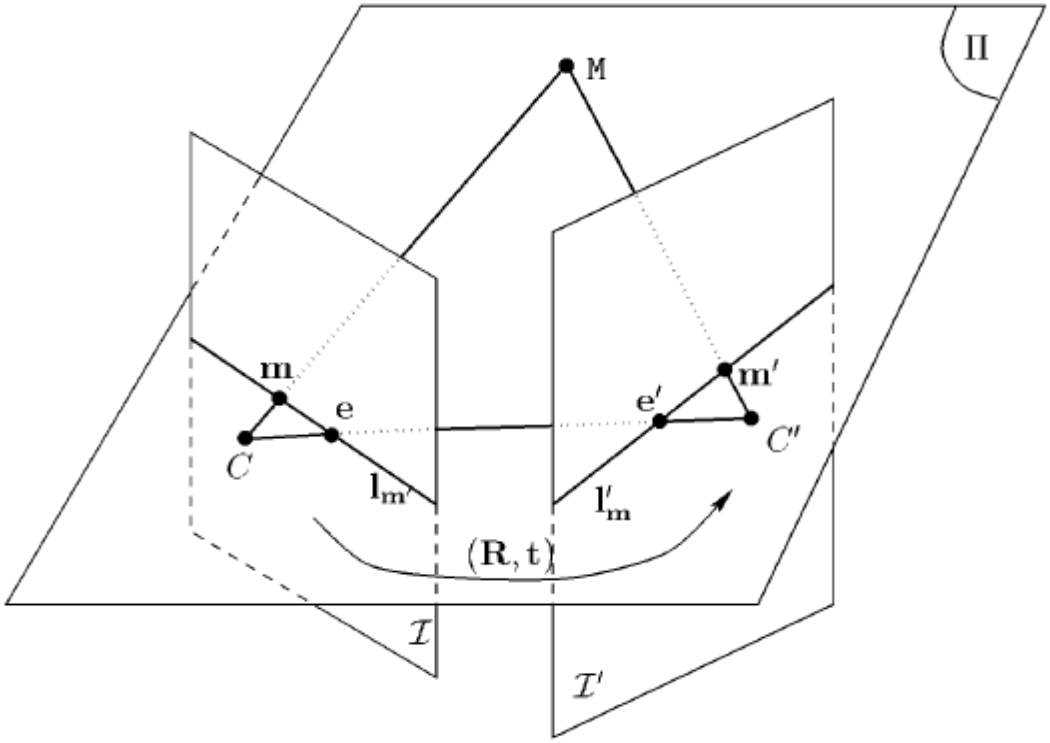

Figure 3.3 The epipole geometry between a pair of camera views.

3.5 Robust Homography Estimation

3.5.1 Feature Detection

Each single view tracker employs a background subtraction algorithm for motion detection [97] and a partial observation method for 2D tracking [98]. The background subtraction algorithm can handle small changes in illumination, which is particularly important in outdoor environments where lighting conditions can vary considerably. The object features required to perform the homography estimation takes the form of centroid measurements of objects detected by each single view tracker. The sequence of centroids for the same object represents the tracked path as it moves through the camera field of view.

3.5.2 Least Quantile of Squares

Given a set of correspondence points the homography can be estimated using the method described in section 3.4.1. The next step is to define a process that would allow a set of correspondence points to be determined automatically from a set of input data. Given a set of sparse object trajectories they can be used to provide training data for estimating the homography transformation between the two overlapping camera views. The object trajectories are taken during periods of low activity, in order to reduce the likelihood of finding false correspondence points. The trajectories take the form of a set of tracked object centroid that are found using the feature detection method described in section 3.5.1. A Least Quantile of Squares (LQS) approach is used to automatically recover a set of correspondence points between each pair of overlapping camera views, which can be used to compute the homography mapping. The LQS method performs an iterative search of a solution space by randomly selecting a minimal set of correspondence points to compute a homography mapping. The solution found to be the most consistent with the set of object trajectories is taken as the final solution. The LQS method was used since it is a robust alignment algorithm and can cope with data that contains a large number of outliers. The following steps are used to estimate the homography transformation using this approach:

I. Synchronise the tracking data using the timestamp information associated with each object. The internal clocks of each camera are synchronised via a LAN using the method that will be discussed in chapter five.

II. Create a list of the M possible object correspondence pairs between the two cameras for each synchronised image frame.

III. From the M possible pairs select four unique correspondence pairs randomly and use these to compute a homography from camera one to camera two.

IV. Compute the transfer error for each correspondence pair in the list created in step 2. The LQS score for this test is chosen as the worst of the top 20%. A value of 20% is chosen since it is expected that the list of correspondence pairs will contain more than 50% of outliers, particularly if there are several objects moving simultaneously between the cameras.

V. Repeat steps 3-4, N times saving the random choice that gives the smallest LQS score.

VI. After N tests we assume that the choice giving the smallest LQS score corresponds to object pairings that contain the smallest number of outliers. The top 20% of these object pairings are used to compute the final homography.

The steps used to perform the LQS search are illustrated with a simple example used to estimate a homography between two cameras, where a set of six features appear in each of the camera views as shown in Figure 3.4.

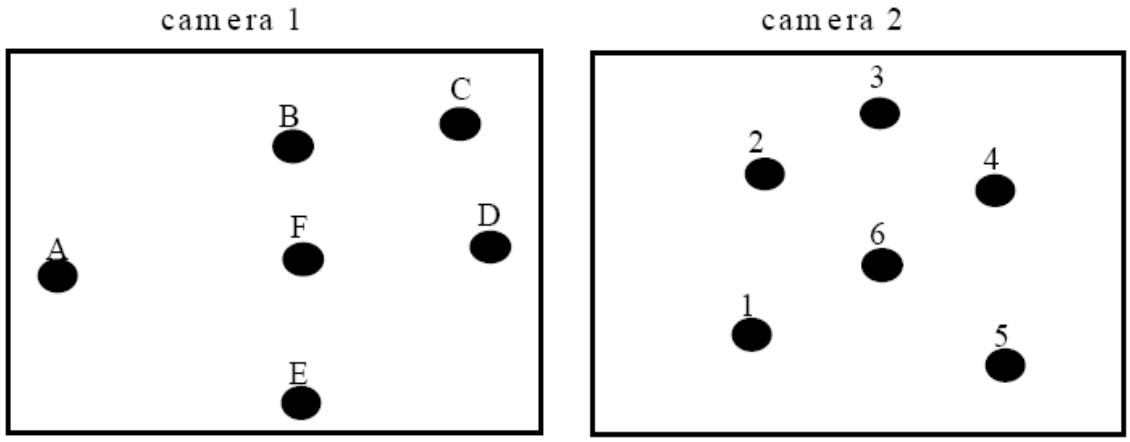

Figure 3.4 Features used to estimate the homography transformation between two camera views using LQS method.

- Step 2: The list of all possible combinations of correspondence points is created:

  (A, 1), (A, 2), (A, 3), (A, 4), (A, 5), (A, 6)

  (B, 1), (B, 2), (B, 3), (B, 4), (B, 5), (B, 6)

  …….

  (E, 1), (E, 2), (E, 3), (E, 4), (E, 5), (E, 6)

- Step 3: Select four correspondence points at random, for example: ((A, 1), (B, 2), (C, 3), (D, 4)), and use the points to estimate a homography transformation. In this instance the four points selected are a set of inlier correspondence points.

- Step 4: Let $r_i^2, i = 1, \ldots\ldots, M$ are the list of transfer errors associated with the correspondence points defined in step 2 and the homography transformation estimated in step 4. The list of transfer errors is then sorted in ascending order, resulting in the list $r_{k,M}^2$. The subscript k refers to the k^{th} largest transfer error in the ascending sorted list. The LQS score for this test is taken as the worst transfer error of the top 20% of the list $r_{k,M}^2$. We choose to take the top 20%, since we expect the list of correspondence points defined in step 2 to contain a large percentage of outliers. The correspondence points consistent with the estimated homography will appear at the top of the list $r_{k,M}^2$, while outlier correspondence points will appear at the bottom of the list of transfer errors.

- Step 5: The steps 3-4 are repeated N times. The value of N is chosen such that there is a 99% chance that one of the random samples of correspondence points selected in Step 3 is free from outliers. $(1 - w^s)^N = 1 - p$, where w is the probability that a correspondence point is an inlier, N is the number of selections (each of s correspondence points), and p is the probability that at least one of the N selections will be free from outliers.

$$N = \frac{\log(1-p)}{\log(1-w^s)} \tag{3.6}$$

where p=0.99, s=4, w=0.2 resulting in N=2875

The LQS test that has the lowest score is taken as the result with the best set of inlier correspondence points. We use the top 20% of correspondence points defined by the list $r_{k,M}^2$ to estimate the final homography transformation.

3.5.2.1 Transfer Error

The homography relations between each overlapping camera are used to match detected moving objects in each overlapping camera view. The transfer error is the summation of the projection error in each camera view for a pair of correspondence points. It indicates the size of the error between corresponded features and their expected projections according to some translating function, the object centroid homography in our case. The epipole line based approach can still function even if the two views do not share a common dominant ground plane but requires that the camera geometry between the two views is known in advance and fairly accurate. The homography-based method assumes that each camera view shares a dominant ground plane.

The biggest advantage of the homographic method over the epipole based method is that the homography maps points to points, while the epipole approach maps points to lines, so a one dimensional search still needs to be performed to establish an object correspondence. The transfer error is used by the homography alignment and viewpoint correspondence methods for assessing the quality of a corresponded pair of centroids in two different camera views. The transfer error associated with a correspondence pair is defined as:

$$(x' - H^{-1}x'')^2 + (x'' - Hx')^2 \tag{3.7}$$

Where x' and x'' are projective coordinates in view 1 and view 2, respectively and *H* is the homography transformation from view 1 to view 2.

3.6 3D Measurements

Given a set of corresponded object features and camera calibration information it is possible to extract 3D measurements from the scene. Using multiple viewpoints improves the estimation of the 3D measurement. In each camera view a 3D ray is projected through the centroid of the object in order to estimate its location. Using the camera calibration model it is possible to map the 2D object centroid to a 3D line in world coordinates.

3.6.1 Least Squares Estimation

Given a set of N 3D lines

$$\boldsymbol{r_i} = \boldsymbol{a_i} + \lambda_i \boldsymbol{b_i}$$

A point $\boldsymbol{p} = (x, y, z)^T$ must be evaluated which minimises the error measure:

$$\xi^2 = \sum_{i=1}^{N} d_i^2 \tag{3.8}$$

Here d_i is the perpendicular distance from the point **p** to the line $\boldsymbol{r_i}$, assuming that the direction vector $\boldsymbol{b_i}$ is a unit vector then we have:

$$d_i^2 = |\boldsymbol{p} - \boldsymbol{a_i}|^2 - \left((\boldsymbol{p} - \boldsymbol{a_i}) \cdot \boldsymbol{b_i}\right)^2 \tag{3.9}$$

Figure 3.5 provides an explanation of the error measure from a geometric viewpoint. The point $\boldsymbol{a_i}$ is a general point located on the line, and $\boldsymbol{b_i}$ is the unit direction vector of the line. The distance d_i^2 is the perpendicular distance between an arbitrary point **p** and the line $\boldsymbol{r_i}$. The origin of the world coordinate system is defined by **O**. Evaluating the partial derivatives of the summation of all d_i^2 with respect to x, y and z results in the equation for computing the least squares estimate of p.

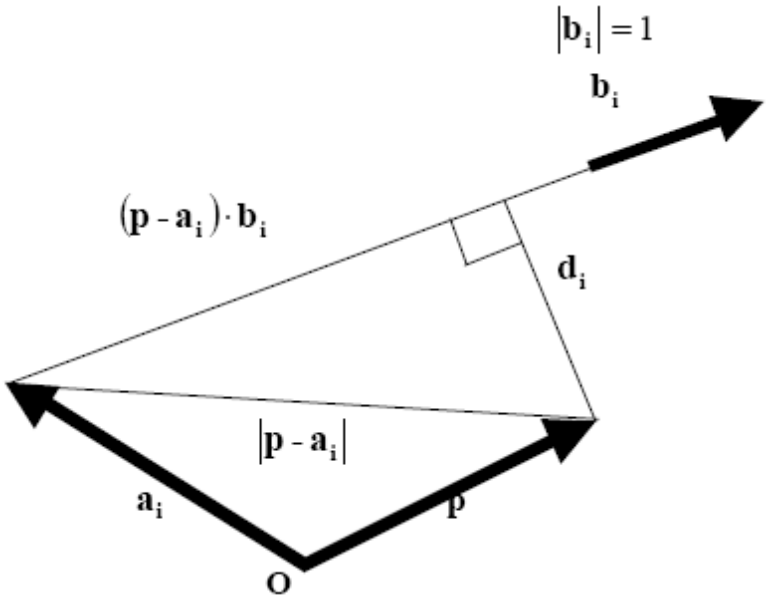

Figure 3.5 Geometric view of the minimum discrepancy

$$\xi^2 = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} \left\{ |\boldsymbol{p} - \boldsymbol{a}_i|^2 - \left((\boldsymbol{p} - \boldsymbol{a}_i) \cdot \boldsymbol{b}_i \right)^2 \right\} \tag{3.10}$$

Rearrangement of (3.10) leads to:

$$\frac{\partial \xi^2}{\partial x} = \sum_{i=1}^{N} \{2(x - a_{ix}) - 2(\boldsymbol{p} - \boldsymbol{a}_i) \cdot \boldsymbol{b}_i b_{ix}\} \tag{3.11}$$

$$\frac{\partial \xi^2}{\partial y} = \sum_{i=1}^{N} \{2(y - a_{iy}) - 2(\boldsymbol{p} - \boldsymbol{a}_i) \cdot \boldsymbol{b}_i b_{iy}\} \tag{3.12}$$

$$\frac{\partial \xi^2}{\partial z} = \sum_{i=1}^{N} \{2(z - a_{iz}) - 2(\boldsymbol{p} - \boldsymbol{a}_i) \cdot \boldsymbol{b}_i b_{iz}\} \tag{3.13}$$

$$\frac{\partial \xi^2}{\partial x} + \frac{\partial \xi^2}{\partial y} + \frac{\partial \xi^2}{\partial z} = 0 \tag{3.14}$$

Using matrix notation an equation can be derived to minimise the error function (3.14) for all N lines.

$$\begin{bmatrix} \sum_{i=1}^{N} 1 - b_{ix}^2 & \sum_{i=1}^{N} -b_{ix}b_{iy} & \sum_{i=1}^{N} -b_{ix}b_{iz} \\ \sum_{i=1}^{N} -b_{ix}b_{iy} & \sum_{i=1}^{N} 1 - b_{iy}^2 & \sum_{i=1}^{N} -b_{iy}b_{iz} \\ \sum_{i=1}^{N} -b_{ix}b_{iz} & \sum_{i=1}^{N} -b_{iy}b_{iz} & \sum_{i=1}^{N} 1 - b_{iz}^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} a_{ix} - b_{ix}\boldsymbol{a_i} \cdot \boldsymbol{b_i} \\ \sum_{i=1}^{N} a_{iy} - b_{iy}\boldsymbol{a_i} \cdot \boldsymbol{b_i} \\ \sum_{i=1}^{N} a_{iz} - b_{iz}\boldsymbol{a_i} \cdot \boldsymbol{b_i} \end{bmatrix} \quad (3.15)$$

$$\mathbf{KP = C}$$

$$\Rightarrow \mathbf{P = K^{-1}C}$$

The point **P** can now be calculated by solving the summation of each partial derivative for the N 3D lines. A 3D line intersection algorithm was used to find the optimal centroid point in the least squares sense.

3.6.2 Singular Value Decomposition

An alternative strategy for making 3D measurements is to use a Singular Value Decomposition (SVD) based approach. This approach is a full least squares approach and is numerically stable for N camera views. Although the matrix dimensions increase with the number of views the matrices are sparse, reducing the computational complexity. Using the Cartesian representation of a line:

$$\frac{x-a_{ix}}{b_{ix}} = \frac{x-a_{iy}}{b_{iy}} = \frac{x-a_{iz}}{b_{iz}} = \lambda_i \quad (3.16)$$

After rearranging (3.15):

$$x - a_{ix} = \lambda_i b_{ix} \quad (3.17)$$

$$y - a_{iy} = \lambda_i b_{iy} \quad (3.18)$$

$$z - a_{iz} = \lambda_i b_{iz} \quad (3.19)$$

The constraints described by the equations (3.16-18) can be transformed into matrix notation:

$$Bx = A$$

$$\begin{bmatrix} 1 & 0 & 0 & -b_{1x} & \cdots & 0 \\ 0 & 1 & 0 & -b_{1y} & & 0 \\ 0 & 0 & 1 & -b_{1z} & & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ 1 & 0 & 0 & 0 & & -b_{Nx} \\ 0 & 1 & 0 & 0 & & -b_{Ny} \\ 0 & 0 & 1 & 0 & \cdots & -b_{Nz} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} a_{1x} \\ a_{1y} \\ a_{1z} \\ \vdots \\ \vdots \\ \vdots \\ a_{Nx} \\ a_{Ny} \\ a_{Nz} \end{bmatrix} \quad (3.20)$$

Using SVD the least squares estimate of the 3D intersection of the N lines can be determined. This approach represents a complete least squares solution. Using SVD simplifies the solution of the value **x** from equation (3.19), which is (3N) by (3+N), where N is the number of cameras used to make the measurement. However, this approach is computationally more expensive that the least squares estimate approach discussed in section 3.5.1.1. In addition, our surveillance network typically does not contain more than three overlapping camera views, so the approach is numerically stable.

4 Object Tracking and Trajectory Prediction

4.1 Background

In the previous chapter we discussed a set of techniques that could be used to calibrate each camera in the surveillance network, match 2D object features between overlapping camera views, and extract a set of 3D measurements given a set of matched 2D features.

To track a moving object in multiple cameras environment, an algorithm is desired that receives video camera feeds from all cameras. The cameras have at least partially overlapping fields of view. By watching people move through the scene the algorithm should eventually be able to predict the location of a person in one camera based solely on their position in others cameras, and vice versa. This will implemented by three functional objectives:

- Single-camera tracking
- Learning the relationship between two cameras i.e. Homography
- Using the relationship to predict target locations

First, the two parts of single-camera tracking – background subtraction and object tracking will be discussed. This will be followed by the method used to learn the relationship between one camera view to reference view (i.e. top view of ground), which consists of two main parts: finding field of view lines and then accumulating enough corresponding points to calculate a plane-induced homography. Finally, once that homography is known, the novel algorithm that creates, updates, and refines relationships between targets and locates them to the reference plane.

4.2 Feature Detection and 2D Tracking

Each intelligent camera employs an adaptive background model for motion detection, which provides a robust framework for motion detection in outdoor environments that are subject to varying changes in illumination. Object tracking is performed using a partial observation-tracking algorithm for robust occlusion reasoning in 2D. The features tracked for each object include: bounding box dimensions, centroid location, and the mean colour components of the foreground object in the (R,G,B) colour space. The features of the tracked objects detected by each intelligent camera are used as input to the multi view tracking algorithm.

4.3 Feature Matching Between Overlapping Views

4.3.1 Viewpoint Correspondence (Two Views)

The LQS algorithm was used to determine a set of correspondence points, which were then used to compute the object centroid homography. This homography can be used to correspond object tracks in the testing video sequences. Once the calibration data and homography alignment model are available we can use the relationship between both camera views to correspond detected objects. From observing the results of motion detection it is apparent that the object centroid is a more stable feature to track in 3D, since it is more reliably detected than the top or bottom of the object, particularly in outdoor scenes where the object may be a far distance from the camera.

To summarize the following steps are used for matching 2D object tracks, taken from different views, for a given image frame:

1. Create a list of all possible correspondence pairs of objects for each camera view.
2. Compute the transfer error for each object pair.
3. Sort the correspondence points list by increasing transfer error.

4. Select the most likely correspondence pairs according to the transfer error. Apply a threshold so that correspondence pairs where Transfer Error > ε_{max} are not considered as potential matches.
5. Create a corresponding points list for each matching object.
6. Map each entry in the correspondence points list using 3D line intersection of the bundle of image rays to locate the object in 3D.
7. For each object centroid which does not have a match in the correspondence pair list use the calibration information to estimate the location of the object in 3D.

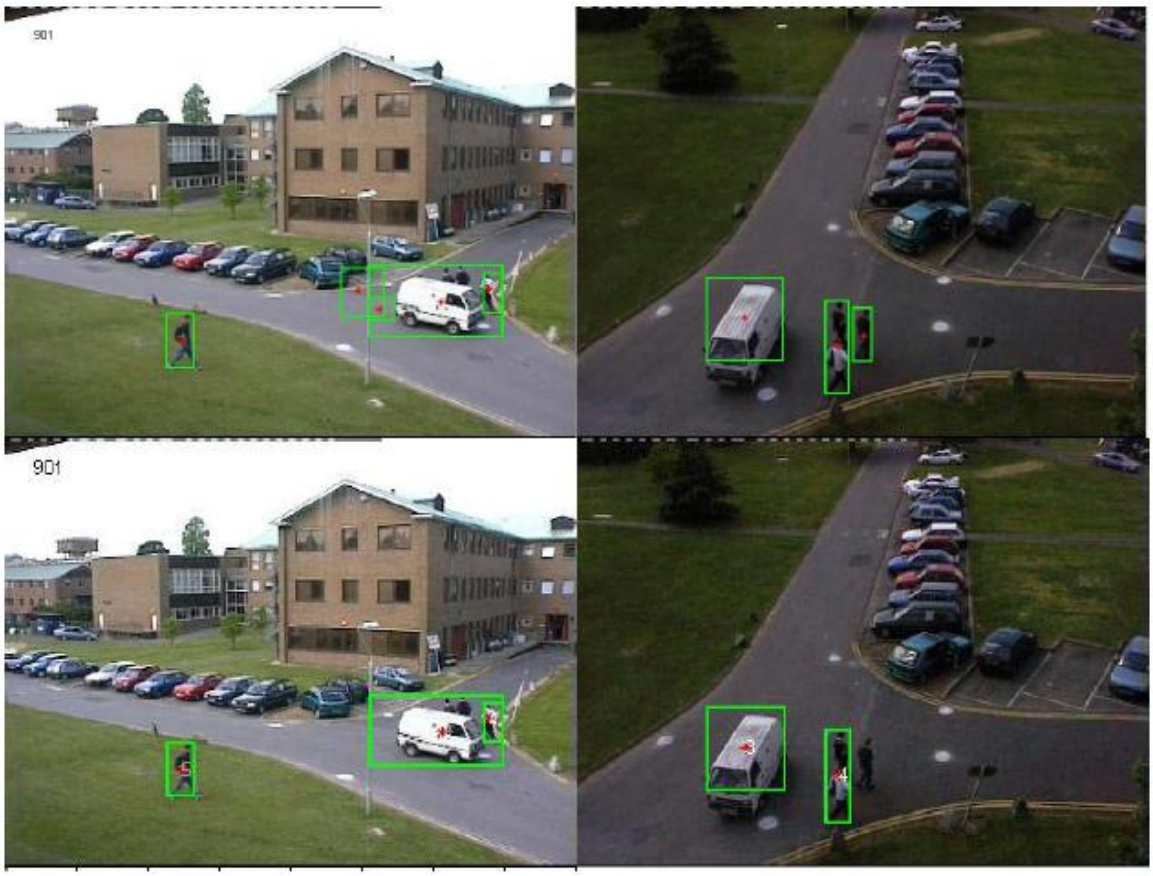

Figure 4.1 Example of feature matching in PETS2001 dataset one.

An example of viewpoint correspondence is shown in figure 4.1, the top image shows the original objects detected by the 2D object tracker, and the bottom

row shows the observations remaining once viewpoint correspondence has been applied. In addition, the three pedestrians are classified as a group because one to many matches are not allowed between the objects in each camera view. The viewpoint correspondence process has the affect of reducing the number of false objects that have been detected by the 2D object tracker.

4.3.2 Viewpoint Correspondence (Three Views)

The viewpoint correspondence algorithm can be extended to match objects between three camera views with a few modifications. It is assumed that the homography mappings between each pair of camera views have been determined by performing an LQS search. We first consider the triplets (*i*, *j*, *k*), where i, j and k are observations of moving objects in camera views one, two and three, respectively. We then evaluate the transfer errors between each pair of observations formed from the triplet (*i*, *j*, *k*), which results in three transfer errors: TE_{ij} , TE_{ik} and TE_{jk} , which represent the transfer errors between each camera pair. When the transfer error is below the threshold ε_{max} we can conclude that the observation pair forms a match between each of the corresponding pair of camera views. We can use the following transitive relationship to identify when a triplet of observations forms a match across all three camera views:

$$\left(TE_{ij} < \varepsilon_{max} \wedge TE_{ik} < \varepsilon_{max}\right) \vee \left(TE_{ij} < \varepsilon_{max} \wedge TE_{jk} < \varepsilon_{max}\right) \vee \left(TE_{ik} < \varepsilon_{max} \wedge TE_{jk} < \varepsilon_{max}\right)$$

The remaining correspondence pairs relationships are summarised below:

$TE_{ij} < \varepsilon_{max}$ The condition to form the correspondence pair (i, j), between camera views one and two

$TE_{ik} < \varepsilon_{max}$ The condition to form the correspondence pair (i, k), between camera views one and three

$TE_{jk} < \varepsilon_{max}$ The condition to form the correspondence pair (j, k), between camera views two and three

The algorithm then proceeds in the same manner as described in Section 4.3.1 for matching between two camera views. An example of homography transfer between three views is shown in figure 4.2. The black arrows indicate how the homography is used to match the three objects visible in each camera view.

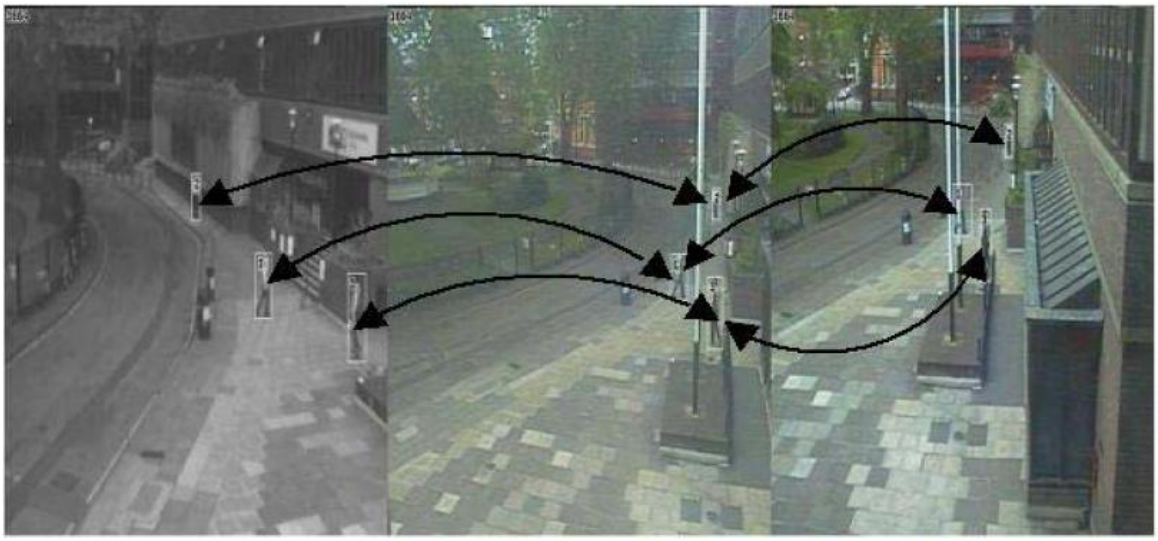

Figure 4.2 Example of viewpoint correspondence between three overlapping camera views

4.4 Some Algorithms

4.4.1 Background subtraction

The first step in processing each single-camera input frame is to determine which parts of the image are background, and hence un-interesting, and which are foreground. In this work, foreground objects are moving people. Although the system can track any moving object, so it also works on traffic scenes, where the targets are cars or any vehicles.

In general, the background subtraction is performed by subtraction of N^{th} frame to $(N-1)^{th}$ frame and get the foreground object. These foreground objects are divided into blobs. The blob should have an area larger than a threshold value.

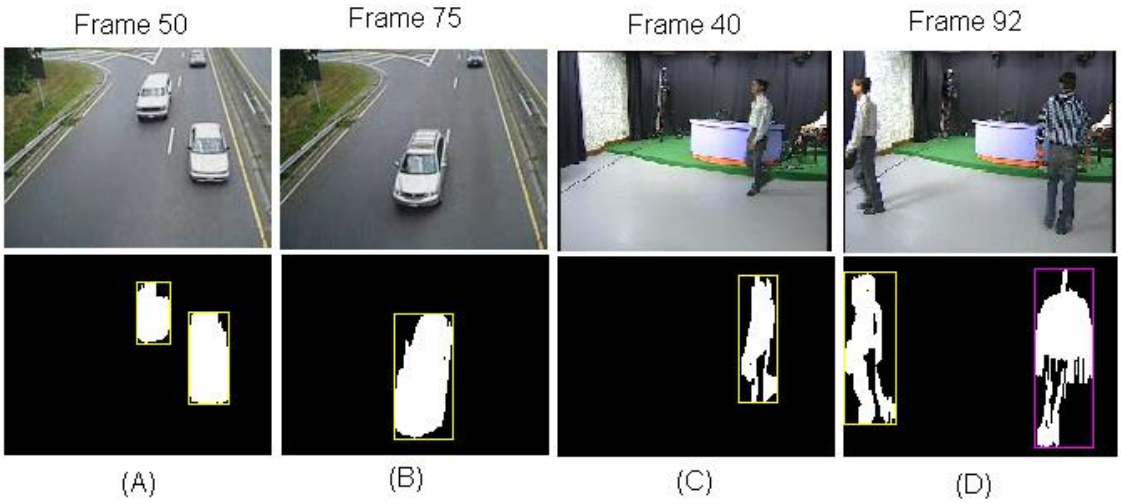

Figure 4.3 The first row shows a seen capture from road (a, b) or inside a room (c, d) at different time. Second row shoes the detected moving object inside the bounding box.

4.4.2 Single-camera tracking

In order to associate targets using multiple cameras, we first need to track moving objects in a single camera. The input to the single camera tracker is the output of the background subtraction algorithm, namely, a series of blob masks that is non-zero on foreground pixels, and zero elsewhere. The output of the single-camera tracker can be represented many ways. In the present work the output is a frame-sized array, with each pixel's value an integer corresponding to a blob label. As targets move around the scene the shape and position of their corresponding blob change with the output of the background subtraction algorithm as shows in figure 4.3. After single-camera tracking, the value of the pixels in each target's blob should be constant.

We add the additional requirement that target labels should be historically unique. This means that once a target disappears from the scene, whether by leaving the field of view of the camera or by being fully occluded, its tracking label is not used again.

4.4.3 Determining feet locations

Finding the feet of a moving object is essentially a feature detection problem. In the present case, the input to the feet locator function will be a mask of a moving object – a standing or walking person. The challenge is to find a single point that represents the "feet" of that person. In general, a person will have two feet touching or nearly touching the ground. The single point then can be thought of as representing the location of the person's centre of gravity, projected onto the ground plane.

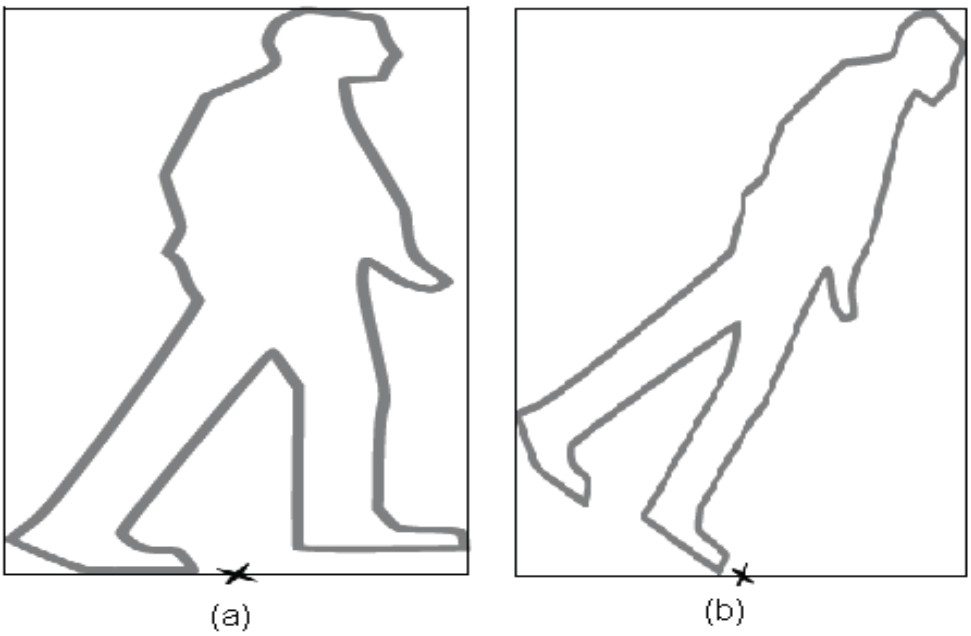

Figure 4.4 The method of finding the feet feature location, (a) The feet are said to be at the centre of the bottom edge of the bounding box. (b) If a target is tilted relative to the camera, then the method will fail to correctly identify the feet location.

As mentioned above, the requirement for a single point stems from our desire to project that point into another camera's coordinate system. The projection of a person's centre of gravity should be onto the same point on the world plane regardless of the location of the cameras.

In [1, 2] Khan and Shah drew a vertical rectangular bounding box around a moving object. The target's feet were then deemed to be positioned at the centre of the bottom edge of the bounding box. This can be seen in Figure 4.4(a). This method is widely used in the computer vision community.

The bounding-box method of finding feet described in has a significant potential failure mode. It effectively requires the person to have their height axis aligned with the vertical axis of the camera. If the camera is tilted relative to the person, which could occur if the camera is installed on a tilted platform or if the person has a consistent tilt in their gait, then the bounding box will enclose a large amount of non-target area. As a result, the bottom centre of the bounding box will not necessarily be close to the actual feet of the target. Figure 4.4(b) shows this error mode.

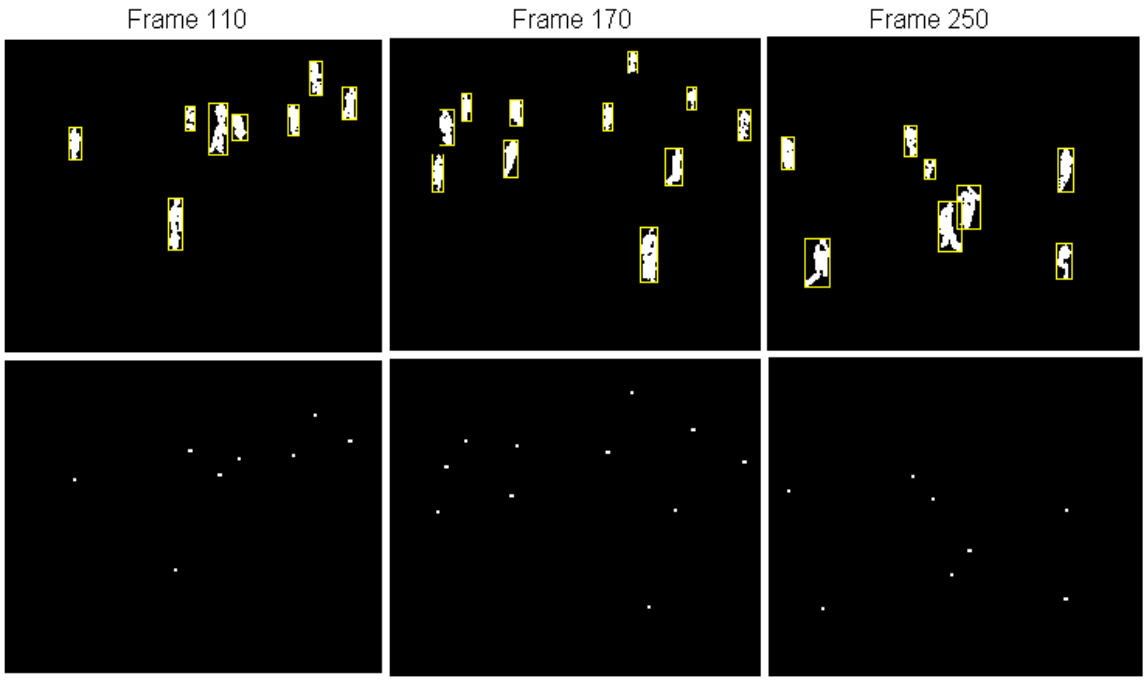

Figure 4.5 First row shows the output of tracked objects and second row shows their foot location of detected object.

The second failure mode of the bounding-box feet-finding method is that the feet will always be outside of the convex hull formed by the target mask. This means that the feet will always appear below the actual feet location. When considered in three dimensions for targets on a ground plane, this means that the feet will always be marked closer to the camera than they should be. Given a wide disparity in camera views, the projected locations of the feet in each view will not

be at the same point the feet will be marked somewhere between the best feet location and the cameras' locations.

4.4.4 Determination homography and field of view (FOV) line

Given a point that lies on a world plane π, $X = [X_1\ X_2\ 1]^T$ we wish to find a predictive relation between the images of X in different view of cameras. That is, given image points $x = [x_1\ x_2\ 1]^T$ we wish to find H_π such that

$$X = H_\pi x$$

H_π is said to be the homography induced by the world plane π. It can be thought of as two projectivities chained together: one that takes a two dimensional point from the image plane of the camera view, x to a two dimensional point on plane π. Note that a point that lies on a plane in the three dimensional world only has two degrees of freedom, those that are required to move on the two dimensional plane.

Finding the relationship between the objects in camera A and objects in camera B to reference view can be broken into two parts. The first part consists of using the meta-target information to create a list of corresponding points, and then using those points to find the plane-induced homography. The second part consists of finding Field of View (FOV) lines and creating target associations using a modified implementation. In short, we want to find the relationship between moving people as seen by any cameras to reference view. Whereas this thesis's approach wishes to find the correspondence between targets at any point in the frame.

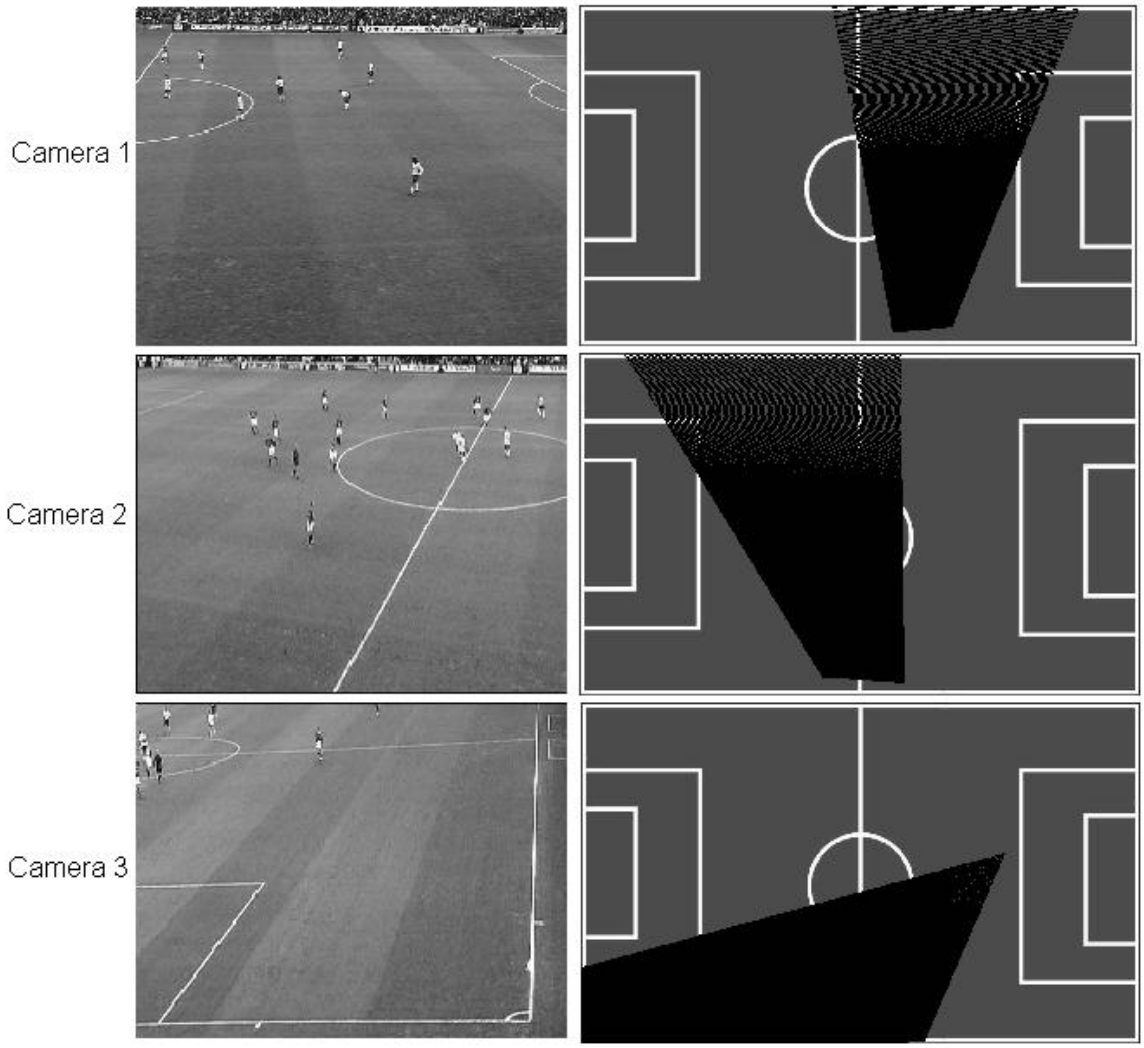

Figure 4.6 In first column, there is views of football ground captured by different camera & second column shows their Field of View (FOV).

4.5 Localization Algorithm

Our algorithm for localizing people is rather simple. The following are the steps of localize people to reference plane:

1. Obtain the foreground likelihood maps $\Psi_1, \Psi_2, \ldots\ldots, \Psi_n$
   - ❖ Model Background using a Mixture of Gaussians.
   - ❖ Perform Background Subtraction to obtain foreground likelihood information.

2. Obtain reference plane homography and vanishing point of reference direction.
3. Find the feet feature location for each active target in each view.
4. $for\ i = 1\ to\ N$
   - ❖ Warp foreground likelihood maps to a reference view using homographies of the reference plane.
   - ❖ Warped Foreground Likelihood maps: $\Psi_1', \Psi_2', \ldots\ldots, \Psi_n'$
   - ❖ Fuse $\Psi_1', \Psi_2', \ldots\ldots, \Psi_n'$ at each pixel location of the reference view
   - ❖ $end\ for$
5. Go to step 1 for next frames.

Figure 4.7 shows the algorithm applied to one of our test scenes. The first row of Figure 4.7 shows the foreground likelihood information in the available views. Football ground's top view was chosen as the reference view and the other views were warped to the reference view with the homography of the reference scene plane (the ground plane). In the last row it clearly highlights the feet regions of the people. Notice how occlusions are resolved and the ground locations of people are detected.

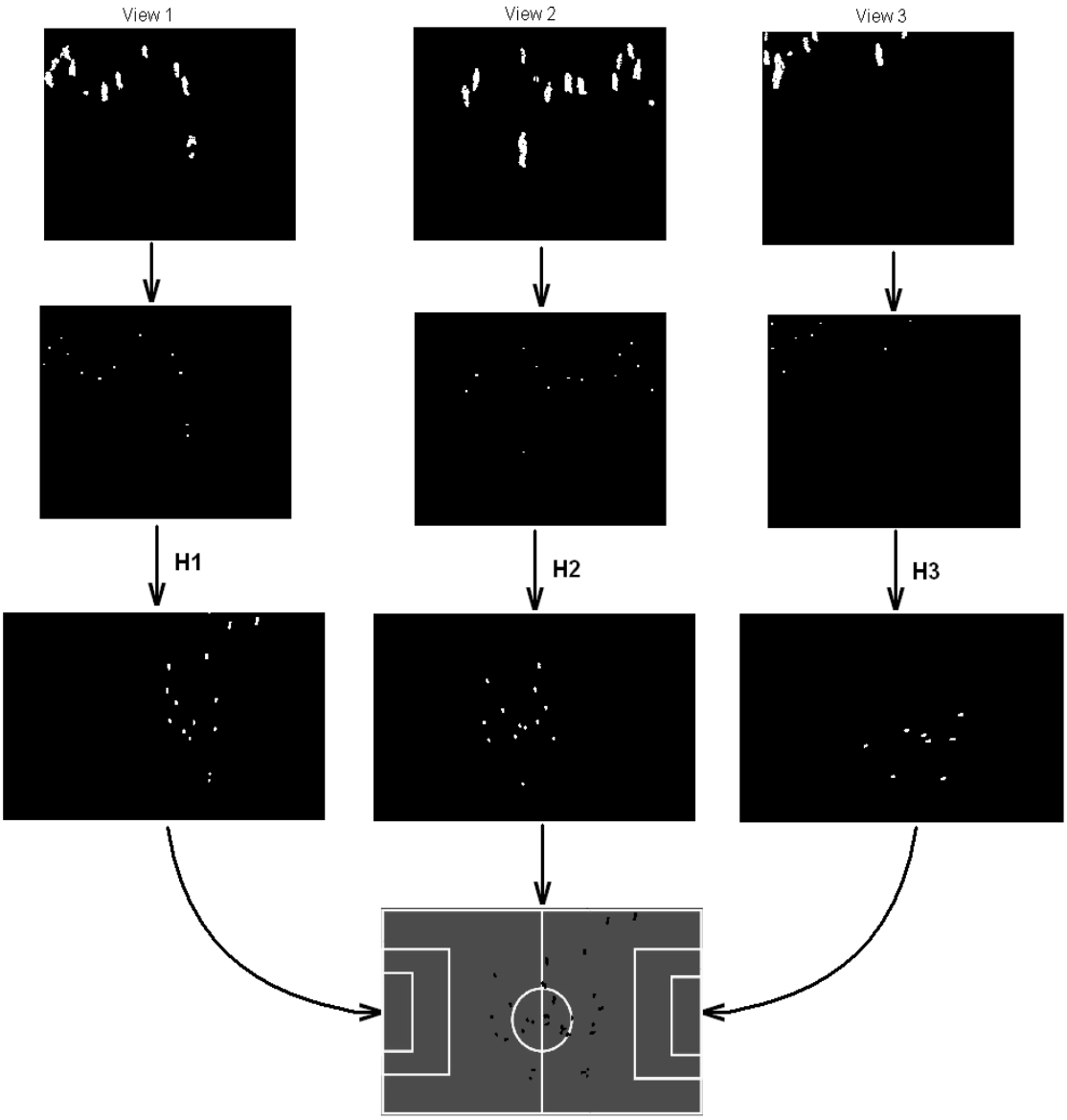

Figure 4.7 The first two rows show the foreground likelihood maps obtained from the background model on the available views. The second row shows their foot location. The third row shows warped foreground maps to a reference view using homography of the reference plane. The last row shows the ground locations of the players on the ground.

4.6 Non-Overlapping Views

In a typical image surveillance network the cameras are usually organised so as to maximise the total field of coverage. As a consequence there can be several cameras in the surveillance network that are separated by a short temporal and spatial distance, or have minimal overlap. In these situations the system needs to track an object when it leaves the field of view of one camera and then re-enters

the field of view of another after a short temporal delay. For short time durations of less than two seconds the trajectory prediction of the kalman filter can be used to predict where the object should become visible again to the system.

However, if the object changes direction significantly or disappears for a longer time period this approach is unreliable. In order to handle these cases the system uses an object handover policy between the pair of non-overlapping cameras. The object handover policy attempts to resolve the handover of objects that move between non-overlapping camera views. The system waits for a new object to be created in the adjacent camera view. A data association method is applied to check the temporal constraints of the objects exit and re-entry into the network field of view.

4.6.1 Entry and Exit Regions

In order to facilitate the object handover reasoning process a model of the major exit and entry regions is constructed for each pair of adjacent non-overlapping camera views. These models can be hand crafted or automatically learned by analysis of trajectory data stored in the surveillance database. The surveillance data can be accessed to retrieve the start and end of object trajectories. An algorithm can then be applied to construct a list of clustered regions, each modeled using a Gaussian distribution, to represent the major entry and exit regions of each camera view. Since each object trajectory has an associated timestamp it is possible to identify the spatial links between exit regions in one camera view and an entry region in the adjacent camera view. The spatial links can be found by identifying a model that is most consistent with respect to spatial and temporal constraints of the object trajectory data. These models of the entry and exit regions are used to improve the performance of the object handover reasoning. When an object is terminated within an exit region the system uses the exit and entry regions models to determine the regions where the object is most likely to reappear. The main benefits of using the model to facilitate the handover reasoning is that the method reduces the computational complexity, since the model is only

used to focus attention on the major entry and exit regions where object handover is most likely to occur, and even if the two cameras are calibrated in different world coordinates the system can still track objects since the model uses temporal properties to perform data association.

4.6.2 Object Handover Regions

The object handover region models consist of a linked entry and exit region along with the temporal delay between each region. The temporal delay can be determined manually by observation, or by generating statistics from the data stored in the database. The temporal delay gives an indication of the transit time for the handover region for a specific object class, so the temporal delay for a pedestrian object class and a vehicle object class would be different based on the set of observations used to generate the statistics. Each entry or exit region is modelled as a Gaussian:

$$\langle (x, y), \sum \rangle$$

Where (x, y) is the centre of the distribution in 2D image coordinates, and $\sum$ is the spatial covariance of the distribution. The following convention is used to describe the major entry and exit regions in each camera view:

X_i^k is the k^{th} exit region in the i^{th} camera view

E_j^l is the l^{th} entry region in the j^{th} camera view.

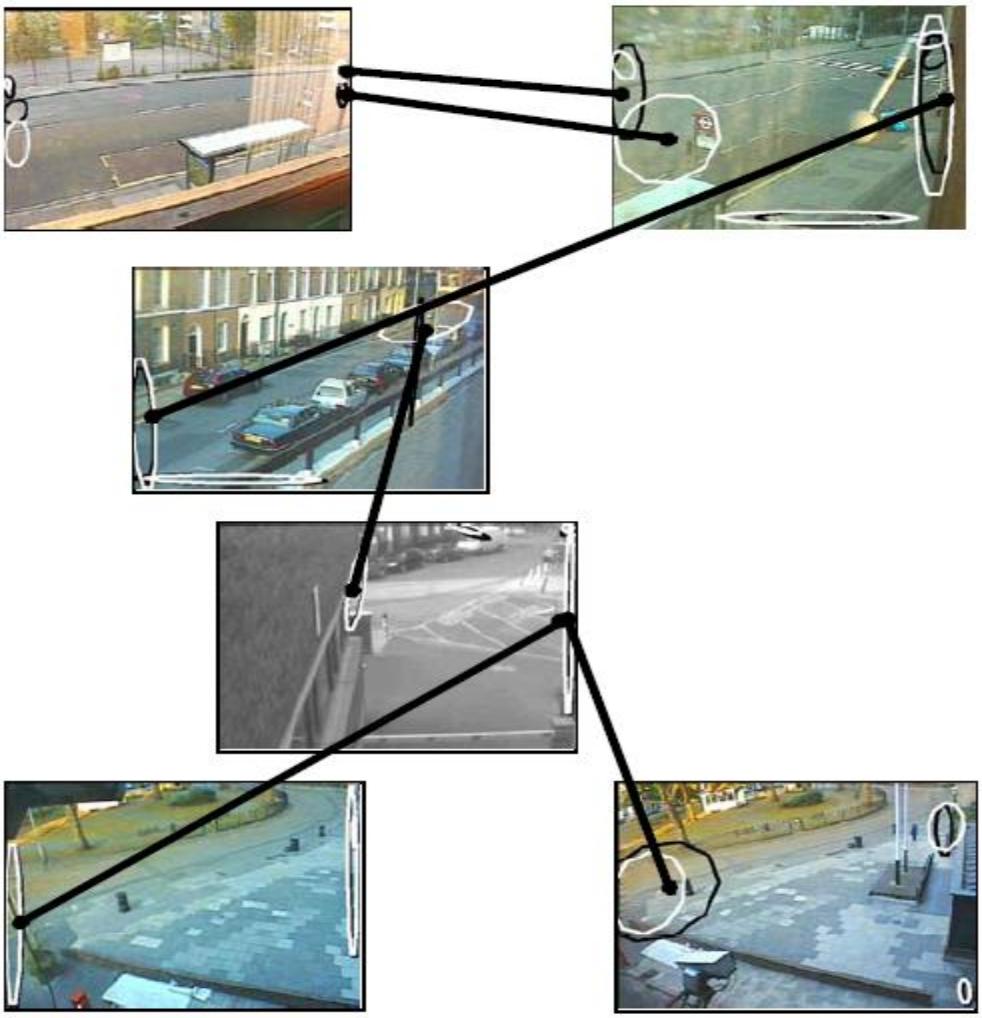

Figure 4.8 Handover regions for six cameras in the surveillance system.

Given the set of major exit and entry regions in each camera the following convention is used to define the handover regions between the non-overlapping camera views:

$H_{ij}^{p} = \langle X_i^k, E_j^l, t, \sigma \rangle$ is the p[th] handover region between camera i[th] and j[th] camera views.

As previously discussed each handover region H_{ij}^{p} consists of a spatially connected exit and entry region pair $\left(X_i^k, E_j^l\right)$, along with the temporal delay and the variance of the temporal delay (t, σ) between the exit and entry region. An example of object handover regions is visually depicted in figure 4.8. The black

and white ellipses in each camera view represent the major entry and exit regions in each camera. The links represent the handover regions between each camera.

4.6.3 Object Handover Agents

The object handover mechanism only needs to be activated when an object is terminated within an exit region that is linked to an entry region in the adjacent camera view. Once the object leaves the network field of view and is in transit between the non-overlapping views the system cannot reliably track the object. The handover agent is activated when an object is terminated within an exit region that is included in the handover region list. The handover agent records the geometric location and time when the object left the field of view of the i^{th} camera. The handover agent achieves completion when an object is created within the entry region that forms a handover region with the exit region, where the object was terminated in the i^{th} camera view.

4.7 Summary

In this chapter a method has been presented for tracking objects between multiple camera views, which have been calibrated using a set of known 3D landmark features. For overlapping camera views the homography is used to match features visible in several cameras. The homography computed using the LQS approach described in chapter 3 is preferred over feature correspondence using the 3D calibration information. The justification for this is that the homography can automatically be recovered from a set of training data, which is dependent upon the accuracy of the 2D tracker and not the camera calibration information that may not be accurate in all regions of the camera view.

Once image features are matched between overlapping views it is possible to generate 3D observations using the least squares estimate technique , along with the associated measurement uncertainty. The algorithm was shown to be robust in

resolving both dynamic and static object occlusions for a variety of video sequences, and coordinating the tracking of objects between a network of cameras in an outdoor environment. One problem with using the homography transformation for feature matching is that it is possible that the system will classify individual objects in close proximity (less than one metre) as a group. As a consequence it would not be possible to track the activity of individuals within the group, or scenes containing a large density of objects under these conditions. In addition, for complex dynamic occlusions where objects interact for long periods, or significantly change speed and direction during the occlusion the Kalman filter tracking becomes less reliable. This is due to limitation of the Kalman filter in that it cannot handle multiple hypotheses. It would be possible to resolve this deficiency by employing a multiple hypothesis tracker framework, which could robustly handle multiple object states during the occlusion, or using color cues to match objects based upon appearance once the occlusion has ended. However, for the environment where the system has been applied the Kalman filter was fit for purpose for robust real-time tracking.

5 System Architecture

5.1 Background

The objective of this chapter is to describe the system architecture of the surveillance system. The basic surveillance system is shown in figure 5.1. Where each synchronized camera input moving object are segmented using the foreground segmentation module. The moving objects are tracked using the single-view tacking module. Following this, the tracks are matched across views using the multi-view correspondence. Feedback from this module is used to improve the accuracy of the single-view tracking. The output from the system is a trajectory of each detected and tracked object.

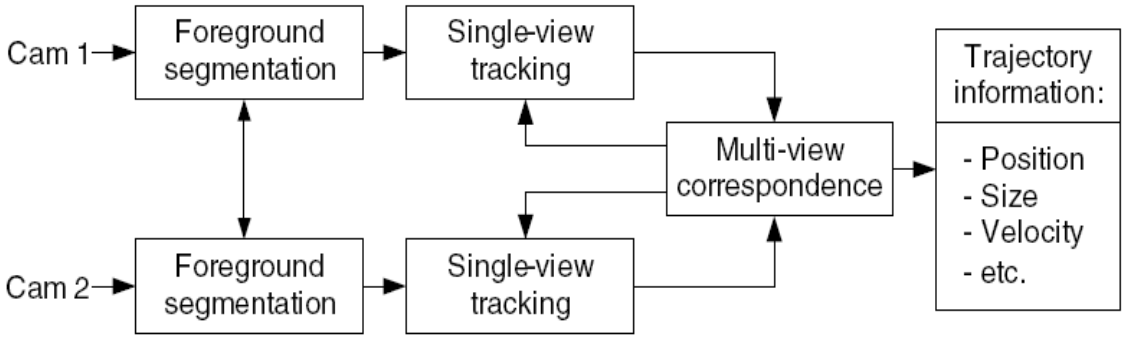

Figure 5.1 System overview of Multi-view tracking

Since, the system would need to run continuously over extended periods of time in order to capture surveillance data. It was not feasible to store raw video, because the surveillance system contained multi cameras and this would generate terabytes of storage each for twenty-four hour period. The surveillance system comprises of a set of intelligent camera units (ICU) that utilise vision algorithms for detecting and robustly tracking moving objects in 2D image coordinates. It is assumed that the viewpoint of each ICU is fixed and has been calibrated using a set of known 3D landmark points. Each ICU communicates to a central multi-

view-tracking server (MTS), which can integrate all the information received in order to generate global 3D tracking information. Each individual object, along with its associated tracking details is stored in a central surveillance database. The surveillance system employs a centralised control strategy as shown in figure 5.1. The multi view-tracking server (MTS) creates separate receiver threads (RT) to process the data transmitted by each intelligent camera unit (ICU) connected to the surveillance network. Each ICU transmits tracking data to each RT in the form of symbolic packets. The system uses TCP/IP sockets to exchange data between each ICU and RT. Once the object tracking information has been received it is loaded into the surveillance database that can be accessed for subsequent online or offline processing.

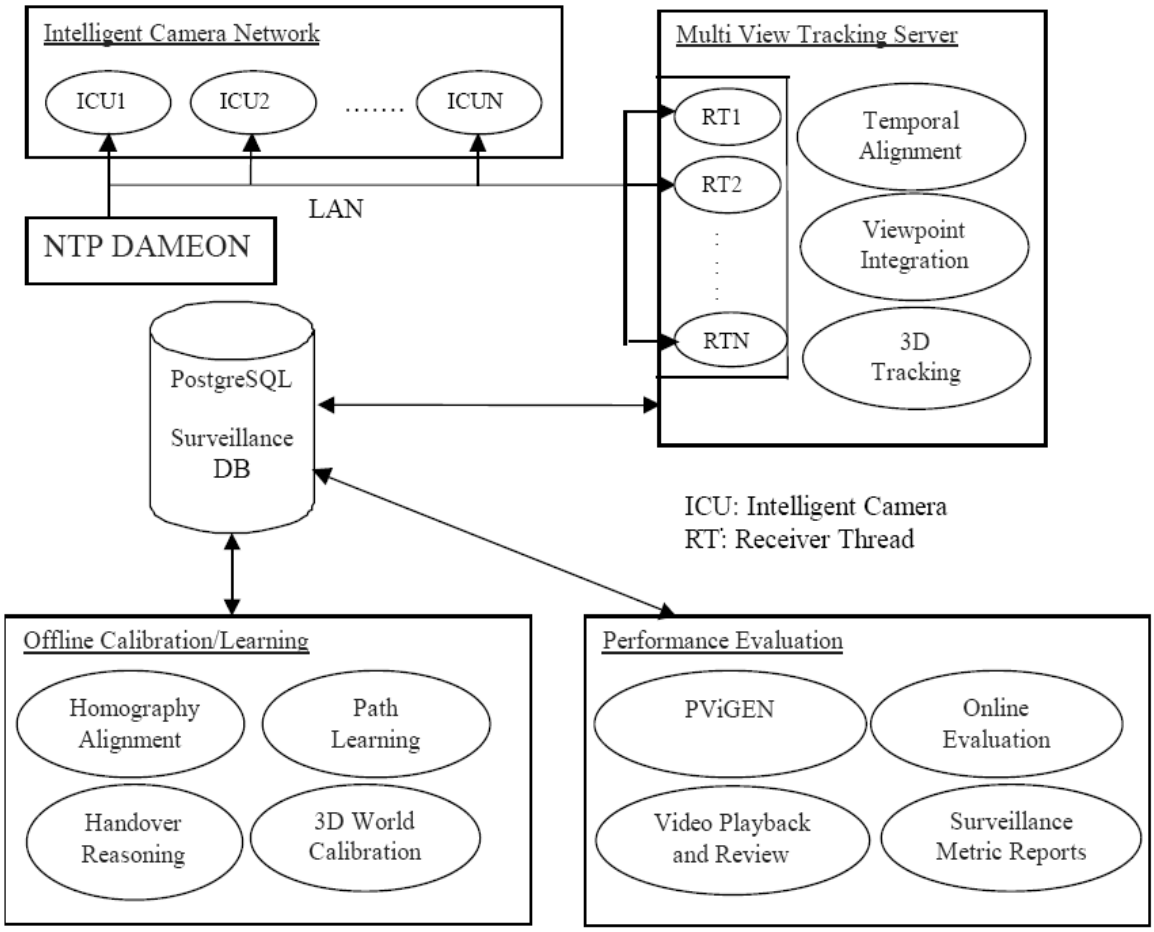

Figure 5.2 System Architecture of the Image Surveillance Network of Cameras.

5.2 Intelligent Camera Network

The surveillance system uses a network of ICUs that are located at various positions within the surveillance region. Each ICU uses an adaptive background-modelling algorithm to detect possible moving objects. A partial observation algorithm is used to track each detected object's location in 2D image coordinates. During live operation the ICUs can robustly track objects in scenes that undergo changes in illumination. The operating speed of each camera is typically between 5-10Hz, depending on the level of activity within the field of view of the camera.

5.3 Multi View Tracking Server (MTS)

The MTS forms one of the core components of the work discussed in the remainder of this thesis. The MTS receives object-tracking data from each ICU and integrates the information for tracking in 3D. Temporal alignment is performed online to synchronise the object tracks received from each ICU. For overlapping camera views the homography constraint is used to correspond objects visible in each viewpoint. A 3D Kalman filter is then employed for object tracking and trajectory prediction.

5.3.1 Temporal Alignment

Temporal calibration could be performed using a robust homography estimation for various time offsets between the set of input data. This approach is valid for pre-recorded video but cannot reliably be applied in a real application, since the temporal calibration would have to be performed continuously to account for the slight variations of each camera internal clock during continuous operation. In addition, some of the cameras are non-overlapping, so it would not be possible to use the homography estimation method.

5.3.2 Viewpoint Integration

For overlapping camera views the homography mapping is used to match 2D objects between each viewpoint. The transfer error, which represents the re-projection error of the homography transformation, is used to derive a list of corresponded 2D object tracks. The transfer error is evaluated for each combination of pairs of objects in each camera view. An appropriate threshold is applied to identify matched objects between each viewpoint. A transitive relationship is used to derive object matches between three or more overlapping camera views.

5.4 Offline Calibration/Learning

The camera calibration allows the 2D observations to be mapped to 3D world coordinates that can be used by the MTS for object tracking. The system also performs offline processing to recover the homography relations between each pair of overlapping camera views, in order to facilitate the multi view tracking. Using a Least Quantile of Squares (LQS) search it is possible to recover a set of correspondence points that can be used to compute the homography mapping between each pair of overlapping views. The LQS method performs an iterative search of a solution space by selecting a minimal set of correspondence points to compute the homography mapping. The solution found to be the most consistent with the selected object tracks is taken as the final solution.

6 Conclusion

6.1 Summary

In short, I have presented an algorithm that can reliably track multiple people in a complex environment. This is achieved by resolving occlusions and localizing people on multiple scene planes using a planar homographic occupancy constraint. By combining foreground likelihood information from multiple views and obtaining the global optimum of space-time scene occupancies over a window of frames, we segment out the individual trajectories of the people

In chapter two a set of requirements were identified and used to define the scope and research goals of this report. This work has primarily focused on the following problems associated with visual surveillance: multi view object tracking, surveillance database management, and performance evaluation of video tracking systems. In chapter three we described a technique that could be employed for automatically recovering the homography transformations between pairs of overlapping camera views. The approach was shown to provide robust estimation for real and synthetic video sequences in the experiments performed. In chapter three the methods used to extract 3D measurements from the scene were also discussed. It is assumed that each camera in the surveillance system is calibrated with respect to the same ground plane.

In chapter four we discussed how the system uses the estimated homography transformations to correspond features between overlapping camera views, and track objects in 3D. One of the key benefits of tracking objects in 3D is that it is possible to resolve both dynamic and static object occlusions. In addition, each camera is calibrated in the same world coordinate system. This enables the system to preserve an object's identity when it moves between non-overlapping cameras that are separated by a short temporal distance of less than two seconds. When objects change direction or the transition time is much longer. To handle

these types of tracking scenarios an object handover policy is defined between each pair of non-overlapping camera views. An object handover region is represented by a linked exit and entry region between adjacent camera views.

The system should also provide a suite of tools to allow the fast indexing and retrieval of the surveillance data. In chapter five we discussed the System Architecture of the Image Surveillance Network of Cameras. The database stores several different representations of the surveillance data, which support spatio-temporal queries at the highest level, to playback of video data at the lowest level. Each intelligent camera in the surveillance system streams 2D tracking data to the multi view-tracking server, where the tracking data is integrated and then stored in the surveillance database.

6.2 Limitations

- The multi view tracking algorithm uses a homography relation to correspond features between overlapping camera views. This presents a problem when objects are in close proximity. it is not possible for the system to track individual members within the group, or correctly track objects in scenes. For example crowds of pedestrians in a shopping center.
- When tracking objects between non-overlapping camera views the system relies on 3D trajectory prediction of the Kalman filter, and object handover policies between linked exit and entry zones. Currently the method does not make any provision to handle cases of ambiguity when several objects move between overlapping cameras.
- Trajectory prediction is used maintain an objects identity during a dynamic and static occlusion once the object interaction has completed. But there some instances where the tracking would fail. If the objects do not maintain the same trajectory during the occlusion, or there are more than three interacting objects it is possible the tracking will fail and the identities of the object be assigned incorrectly.

- It assumed that the surveillance region conforms to the ground plane constraint and 3D camera calibration is available for each camera. This could present a problem in applying the methods discussed in chapter three to scenes where the ground plane assumption is invalid, for example for tracking people between several floors of a building.

- The surveillance system presented in chapter five demonstrated a robust framework for continuous monitoring of a region over extended periods of several hours or days. The current system has been operating for several months using a network of six cameras connected to a standard 100MB/sec Ethernet local area network. One problem with the architecture is that a centralized control strategy is employed to integrate all the tracking data received from each camera connected to the network of intelligent cameras. This would represent a bottleneck if the system had to be scaled to cope with several hundred cameras, which is common in many surveillance environments.

6.3 Future Work

There are many possible extensions of this work. One direction is to incorporate color models in the detection and tracking of individual people. The color models can be used to disambiguate tracks in cases when two or more people come too close to be segmented as separate entities. Using articulated human shape models can be another addition that can act as a prior to prune out false detections and increase robustness of localization. Similarly, a human motion model that takes into account the consistency of speed and direction as well as modeling collision avoidance strategies between people could be an interesting addition. These models may be useful in situations where crowd densities increase and camera views are limited.

Spatial temporal cues are used to coordinate object tracking between multiple camera views. The system relies on 3D trajectory prediction to resolve dynamic occlusions as stated in the list of limitations. The Kalman filter is not effective in instances where objects change direction significantly, or the scene contains clutter. This limitation also presents problems when tracking objects between non-overlapping views, where the ambiguity for matching is increased if several objects move between cameras concurrently. An enhancement can be made to the system to use appearance information to improve the robustness of occlusion reasoning, and object handover between cameras.

When a surveillance application is installed in a new environment it should be possible to automatically configure the system with limited operator intervention. The current system can calibrate between overlapping views without supervision, however 3D camera calibration data is still required to coordinate tracking between multiple camera views. In future work it should be possible to perform self-calibration of the ground plane without the need for performing a landmark survey.

The current system has been running continuously over a period of several months allowing a large volume of object tracking data to be accumulated. All the information is stored in a surveillance database, which allows various types of activity queries to be executed to recognize single object behaviours. In future work new methods will be explored to recognize different types of interactions between multiple objects. This concept of data mining is an emerging research area for surveillance systems, since it would be possible for human operators to perform forensic analysis of data streams without having to manually review large volumes of video.

7 Bibliography

1. S.M. Khan and M. Shah, "A Multi-View Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint," Proc. Ninth European Conf. Computer Vision, 2006.

2. S.M. Khan and M. Shah, "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes", Ieee Transactions On Pattern Analysis And Machine Intelligence, March 2009

3. Intille S.S., Bobick A.F. Closed-world tracking. International Conference on Computer Vision (ICCV'95), Cambridge, Massachusetts ,June 1995, pp 672-678

4. Intille S.S., Davis J.W., Bobick A.F. Real-time closed-world tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97), Puerto Rico, June 1997, pp 697-703

5. Haritaoglu I., Harwood D., Davis L.S. W4: Real-Time Surveillance of People and Their Activities. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), August 2000, Vol. 22, No. 8, pp 809-830

6. Wren C.R., Azarbayejani A., Darrell T., Pentland A.P. Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), July 1997, Vol. 19, No. 7, pp 780-785

7. M. Han, W. Xu, H. Tao, and Y. Gong, "An Algorithm for Multiple Object Trajectory Tracking," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2004.

8. M. Isard and J. MacCormick, “Bramble: A Bayesian Multiple-Blob Tracker,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2001.

9. T. Zhao and R. Nevatia, “Tracking Multiple Humans in Crowded Environment,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2004.

10. P. Kornprobst and G. Medioni, “Tracking Segmented Objects Using Tensor Voting,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2000.

11. Q. Cai and J. Aggarwal, “Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams,” Proc. Sixth IEEE Int’l Conf. Computer Vision, 1998.

12. J. Kang, I. Cohen, and G. Medioni, “Continuous Tracking within and across Camera Streams,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2003.

13. T. Chang and S. Gong, “Tracking Multiple People with a Multi-Camera System,” Proc. IEEE Workshop Multi-Object Tracking, 2001.

14. S. Dockstader and A. Tekalp, “Multiple Camera Fusion for Multi-Object Tracking,” Proc. IEEE Workshop Multi-Object Tracking, 2001.

15. Mittal and S. Larry, “M2tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene,” Int’l J. Computer Vision, 2002.

16. W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, “Principal Axis-Based Correspondence between Multiple Cameras for People Tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, 2006.

17. S. Park and M.M. Trivedi, "Multi-Perspective Video Analysis of Persons and Vehicles for Enhanced Situational Awareness," Proc. IEEE Int'l Conf. Intelligence and Security Informatics, 2006.

18. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2002.

19. Black J., Ellis T.J., Rosin P. Multi View Image Surveillance and Tracking, IEEE Workshop on Motion and Video Computing, Orlando, December 2002, pp 169-174.

20. Ellis T.J., Black J. A Multi-view surveillance system. IEE Intelligent Distributed Surveillance Systems, February 2003.

21. Stein G. Tracking from Multiple View Points:Self Calibration of Space and Time. DARPA Image Understanding Workshop 1998, pp 1037-1042