UNIVERSITY OF LIVERPOOL

Robust Moving Object Detection by Information Fusion from Multiple Cameras

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of

Doctor in Philosophy

Jie REN

January 2014

Department of Electrical Engineering & Electronics

ABSTRACT

Moving object detection is an essential process before tracking and event recognition in video surveillance can take place. To monitor a wider field of view and avoid occlusions in pedestrian tracking, multiple cameras are usually used and homography can be employed to associate multiple camera views. Foreground regions detected from each of the multiple camera views are projected into a virtual top view according to the homography for a plane. The intersection regions of the foreground projections indicate the locations of moving objects on that plane. The homography mapping for a set of parallel planes at different heights can increase the robustness of the detection. However, homography mapping is very time consuming and the intersections of non-corresponding foreground regions can cause false-positive detections.

In this thesis, a real-time moving object detection algorithm using multiple cameras is proposed. Unlike the pixelwise homography mapping which projects binary foreground images, the approach used in the research described in this thesis was to approximate the contour of each foreground region with a polygon and only transmit and project the polygon vertices. The foreground projections are rebuilt from the projected polygons in the reference view. The experimental results have shown that this method can be run in real time and generate results similar to those using foreground images.

To identify the false-positive detections, both geometrical information and colour cues are utilized. The former is a height matching algorithm based on the geometry between the camera views. The latter is a colour matching algorithm based on the Mahalanobis distance of the colour distributions of two foreground regions. Since the height matching is uncertain in the scenarios with the adjacent pedestrian and colour matching cannot handle occluded pedestrians, the two algorithms are combined to improve the robustness of the foreground intersection classification. The robustness of the proposed algorithm is demonstrated in real-world image sequences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgements

I would like to thank my supervisors Prof. Jeremy S. Smith and Dr. Ming Xu. Without their patience and support in supervising my research and revising this thesis, I would have been unable to complete the work for my PhD.

I am very grateful to Xi'an Jiaotong-Liverpool University for providing a PhD studentship and a living allowance to me. I would like to thank the support of the National Natural Science Foundation of China (NSFC) under Grant 60975082 and the Natural Science Foundation of Jiangsu Province, China, under Grant BK2008180.

I am thankful to Dr. Shi Cheng, Dr. Jimin Xiao and Mr. Yungang Zhang for their advice on programming. I am also grateful to Mr. Chili Li and Mr. Yuyao Yan for their reviewing of my reports and thesis.

I want to thank the classmates and colleagues who helped and supported me in my study: Ms. Guifen Wang, Ms. Yanfei Qi, Mr. Dongyong Chen, Mr. Tianyuan Jia, Mr. Lei Lu, Mr. Buoyuan Sun, Mr. Daoman Hu and Mr. Jie Yang.

Finally, I would like to thank my parents and my cousin Dr. Jinchang Ren for their support and encouragement.

1 INTRODUCTION

1.1 Visual Surveillance and Moving Object Detection

Intelligent visual surveillance is an active research area in artificial intelligence and computer vision. The aim of an intelligent visual surveillance system is to detect, track, classify objects and recognize events automatically. Visual surveillance system can be applied in a variety of situations such as airport, subway/railway stations, sport events, shopping centres and parking lots. The applications of the intelligent visual surveillance system can be divided into two main categories: online and offline. The offline applications involve a forensic mode and statistical information collection. In the forensic mode, the videos captured by the intelligent visual surveillance system are scanned to find out what happened before and after the event once the event had been detected. In statistical information collection, the video is further analysed by the intelligent visual surveillance system to provide overall statistics such as the number of people or vehicles passing through a location in a certain time, the shopping habits of people in a store and the queue lengths in a store. The statistical information can be used to determine people's behaviour and to access the efficiency of operations.

Traditional online video surveillance systems need human operators working in a control room where a set of surveillance scenes are displayed on monitors. The human operators monitor these scenes and respond when special events occur. Although human operators can provide accurate detection and recognition of interesting objects and events, their performance is influenced by the operators' skill, response and fatigue. Since cameras in the video surveillance system are located at different places and each camera covers a limited field of view, the operator needs to switch between camera views and to concentrate on the camera which can provide the best view to monitor the important events. This means that only one view is monitored and information from other scenes is lost. Besides that, it requires that the operator should be familiar with the arrangement and placement of the cameras and their corresponding scenes respectively. When some important events occur at different locations simultaneously, the operator cannot monitor and response to them all at the same time. Furthermore, the response of the operator is influenced by the fatigue of the operator especially when the

operator is working during anti-social hours. However, the tasks for offline video surveillance system are more laborious because it requires the operators to review a significant archive of recoded video data.

Intelligent visual surveillance systems can now aid or replace the operators in traditional video surveillance systems. The online applications are mainly used for security, which can provide real-time detection of moving objects and respond to special events after tracking and classification. These special events include illegal car parking, left luggage and human loitering. In the offline applications, videos are stored in a different format for fast retrieval, searching and analysis. For example, a suitable detected object with index and other information.

Using a single camera, a typical intelligent visual surveillance system undertakes four main tasks: foreground detection, object tracking, classification and event detection. Figure 1.1 shows a schematic diagram of a typical intelligent visual surveillance system with a single camera. The first step involves using a foreground detection method to detect moving objects in each frame. The detected objects are tracked over time according to the spatial-temporal coherence. Then, the features of the individual tracked objects are classified. Given the event definitions, an alert is triggered when an important event is detected.

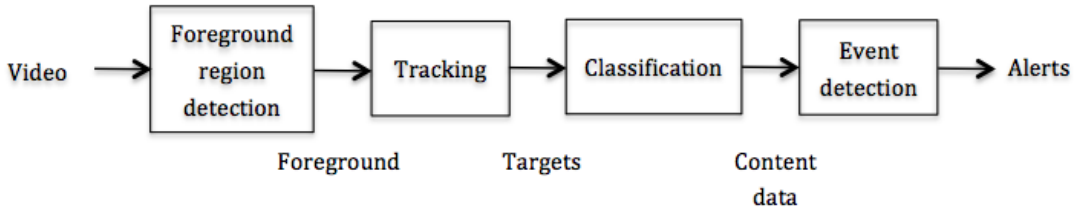


Figure 1.1 A schematic diagram of a typical intelligent visual surveillance system with a single camera.

If objects are isolated in the scene, it is easy to fulfil the tasks in a single-camera intelligent visual surveillance system. In reality, dynamic and static occlusions are the main problems that degrade the performance of intelligent visual surveillance systems. A dynamic occlusion occurs when an object is occluded by another moving object in the scene whilst a static occlusion occurs when an object is occluded by obstacles such as buildings or trees. Using multiple cameras is a reasonable method to solve occlusions, because when an object is occluded in one view, it may be visible in the other camera views. Many multi-camera visual surveillance systems have been developed [1, 2]. In a multi-camera visual surveillance system, cameras are usually arranged in different locations to monitor the same area, which means

these cameras have overlapping field of views (FOVs). Within the overlapping FOVs, the information lost in a particular view can be recovered from the other views. Furthermore, the multiple camera views can extend the overall field of view.

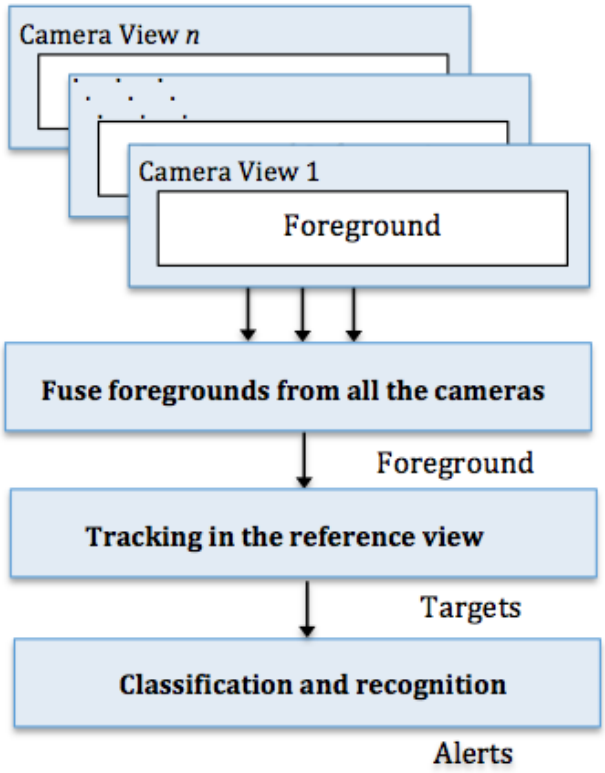


Figure 1.2 A schematic diagram of a multi-camera approach with high-degree information fusion.

Since the systems use many cameras, an active research topic is how to utilize the information from the multiple cameras? According to the degrees of information fusion, the multi-camera approaches can be divided into three categories. The first category is the low-degree information fusion which starts tracking with a single camera view and switches to another camera when the system predicts that the current camera will no longer have a good view. In the intermediate-degree of information fusion, measurements, extracted features or tracked targets are first detected in the individual camera views and then integrated to obtain the global estimation. The third category no longer extracts features or tracks targets but provides foreground bitmap information in the individual camera views. The foreground information from all camera views is fused in the fusion centre and then detection and tracking based on the fused information is undertaken. The third category has emerged in recent years and belongs to the category of high-degree information fusion. This approach can help moving object detection when the scene

is crowded. Figure 1.2 shows a schematic diagram of the multi-camera approach with high-degree information fusion.

To associate camera views and to fuse information from all the camera views, one useful assumption is that in all the camera views the objects of interest are on a common plane. This assumption is valid for most scenarios in intelligent visual surveillance systems. In this case, the ground plane is the common plane. Then, homography, a geometric transformation which shows a pixelwise mapping between two views according to a common plane, can be used as an efficient method to associate multiple camera views. Using a homography transformation, the foregrounds from one camera view can be projected to a reference view according to the homography for a specific plane.

1.2 Aims and Objectives

Collins, Lipton and Kanade [3] divided the active research areas in video surveillance into detection, tracking, human motion analysis and activity analysis. Since all intelligent visual surveillance systems require accurate detection, the main focus of this research is to detect objects using multiple camera views in real time. In this research, after the foreground detection process, the foregrounds in the individual camera views are warped to a virtual top view according to the homography for a certain plane. All camera views are associated with the top view and global detection is generated in the top view.

Homography mapping is a 2.5D method which embeds a two-dimensional plane in a three-dimensional space that limits the 2D to a 3D projection on a certain plane. In previous work, this method achieved good results in detection and is robust in coping with occlusion. In Khan and Shah's work [4], the foreground likelihood image, which is extracted from each of the multiple camera views, is warped to a reference camera view according to the ground-plane homography and overlaid with those from other camera views. A threshold is applied in the reference view to determine the locations of people on the ground-plane. Then, the homographies for a set of parallel planes at different heights are employed to increase the robustness of the detection. This work achieves good results in moderately crowded scenes, because regions at the locations of true objects reinforce each other whereas the false locations are scattered around. However, there are two problems with this method, low speed homography mapping and false-positive detections. The research described in this thesis is an extension of Khan and Shah's work, which

focuses on moving object detection with multiple cameras and with multi-layer homography mapping while solving these two problems.

1.2.1 Real-time Foreground Projection with Homography

When the homography matrix is defined, the homography transformation is a pixelwise projection, in which each pixel in the camera view is projected to the reference view according to the planar homography. The number of pixels which are projected is decided by the resolution of the camera views.

Since the projection from the camera view to the reference view is affected by the perspective geometry, foreground openings or holes are generated at the locations which are far away from the cameras. Simply projecting each foreground pixel in the camera view to the reference view is infeasible. Therefore, the homography mapping starts by scanning each pixel in the reference view and projecting each pixel in the reference view back to the camera view according to the inverse homography. The pixel is classified as a foreground pixel in the top view when the warped back pixel is a foreground pixel in the camera view. Since the top view usually covers a larger area and has a higher resolution, scanning and projecting all the pixels in the top view is very time consuming.

Since undertaking the pixelwise homographic transformations at image level is very time consuming, the homography approach is difficult to apply in real-time applications. It also brings a challenging requirement on the bandwidth of multi-camera networks, if the foreground detection and the multi-view foreground fusion are carried out by different computers. Furthermore, when the homographic transformations are applied to multiple cameras and multiple parallel planes, the time consuming problem becomes worse and thus prevents the homography approach from being used in any low cost real-time implementation.

A real-time homography mapping algorithm is proposed to solve this time consuming problem. Unlike the most recent algorithms which transmit and fuse foreground likelihood maps or binary foreground images, in this research an approximation of the contour of each foreground region by a polygon is generated and only the polygon vertices need to be projected. These polygons are rebuilt and fused in the reference image. This greatly reduces the requirement on the network bandwidth and avoids homographic transformations at image levels.

1.2.2 False Positives in the Detection

Another problem with the homography approach is that the intersections of non-corresponding foreground regions can cause false-positive detections known as phantoms. Figure 1.3 is a schematic diagram to illustrate how non-corresponding foreground regions intersect and give rise to false positives. The warped foreground region in the top view is observed as the intersection of the ground-plane and the cones swept out by the silhouette of the underlying object. When the foreground regions for the same object are warped from multiple views to the top view, they will intersect at a location where the object touches the ground. However, if the warped foreground regions from different objects intersect in the top view, the intersection region will lead to a phantom detection. When homography mapping is based on a plane parallel to but higher than the ground plane, the projected foreground regions will move towards the cameras and additional phantoms may be generated. In Figure 1.3, the projected foregrounds from the two cameras intersect in four regions on plane *p* which is parallel to and off the ground plane. The white intersection regions are the locations of the two objects, whilst the black region that is intersected by the warped foreground regions of different objects is most likely to be a phantom. The grey region is an addition phantom.

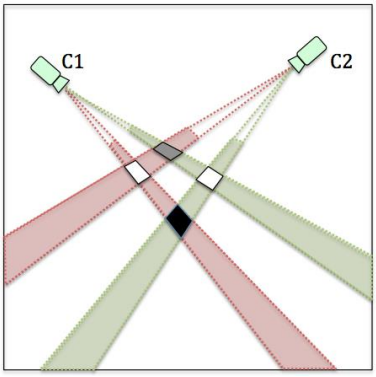


Figure 1.3 A schematic diagram of the phantom occurrence in multi-view detection via homography mapping of foregrounds.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows: In chapter 2, previously published work is discussed and reviewed. Chapter 3 describes the techniques used to estimate the homography for the ground plane and homographies for a set of planes parallel to the ground plane and at some height. In chapter 4, the developed real-time moving object detection algorithm using multiple cameras is explained,

where the contour of each foreground region is approximated with a polygon and only the polygon vertices need to be transmitted and projected. Chapter 5 describes a geometrical approach to identify objects and phantom detections. Chapter 6 presents a combination method in which height matching and colour matching are applied successively to identify whether a foreground intersection region in the top view is due to same object or not. In this way, phantom detections can be identified. Conclusions and the future work are presented in chapter 7.

1.4 Contributions

The main research contributions of this thesis are highlighted as follows:

1) To accelerate the transmission and projection of the foreground information to a reference image, it is reasonable to focus on foreground regions. To warp the foreground regions in a camera view to the reference image, the contour of each foreground region is approximated by a polygon and the vertices of the polygon are projected onto the reference image using homography mapping. Then the foreground region is rebuilt by filling the polygon projected in the reference image. This approach greatly saves network bandwidth and accelerates the processing by avoiding the image-level homographic transformation.

2) To identify false-positive detections caused by the foreground intersections of non-corresponding objects in the top view, geometrical information of the foreground regions is utilised. As such, a height matching algorithm is proposed to match the intersection regions in the top view with the foreground regions in individual camera views to identify whether an intersection region is due to the same object.

3) Based on the colours of each foreground region, an appearance model and a colour matching algorithm is proposed, with the Mahalanobis distance applied to calculate the colour similarity of two foreground regions. In addition, the height matching algorithm and the colour matching approach are combined to further improve the robustness in the classification of the foreground intersection regions.

2. Literature Review

Although some good surveys on visual surveillance with multiple cameras have already been published [5-9], in this chapter a survey of the research in relation to this thesis is presented. It begins with a review of multi-camera visual surveillance systems and divides the existing systems into three categories according to the degrees of the information fusion from the multiple camera views. Then, the geometrical methods which are used to associate multiple cameras in the information fusion are reviewed. Since homography mapping is the method used in this thesis to associate individual camera views with a reference view, the related research using homography in multi-camera visual surveillance is discussed and then grouped according to the features used in the homography projection. Finally, the existing algorithms for the removal of false-positive detections are introduced.

2.1 A Review of Multi-Camera Visual Surveillance

As using multiple cameras in a visual surveillance system can provide a larger overall field of views and can reduce occlusions in the overlapping field of view, one of the key issues for visual surveillance with multiple cameras is how to utilize the information from the multiple cameras for the purpose of detection and tracking. The more information from the multiple cameras which can be used simultaneously, the more robust and accurate the system becomes. Depending on the degree of information fusion, the existing multi-camera surveillance systems can be categorized as low-degree information fusion, intermediate-degree information fusion and high-degree information fusion.

2.1.1 Low-degree Information Fusion

The first category of the visual surveillance systems uses multiple cameras to extend the limited field of view for a single camera [10-13]. It is known as the camera handoff method which starts tracking objects with a single camera view and switches to the next camera when the tracked object goes beyond the field of view (FOV) of the current camera. For the camera handoff method, existing research differ on three points: when the handoff process is triggered, which camera will be

selected as the next optimal camera to track and monitor the object of interest and how to establish the correspondence of the objects between the cameras.

Cai and Aggarwal [10] proposed an algorithm, which starts tracking with a single camera view and switches to another camera when the system predicts that the current camera will no longer have a good view. The features extracted from the upper human bodies are used to build the object correspondence between cameras. The next camera should not only provide the best view but also have the least switching to continue tracking in that camera view.

In [11], the edges of the field of view of each camera, which can be seen in other cameras, are defined as field of view lines. The field of view lines are used to establish the correspondence of trajectories between cameras. The camera handoff is triggered when the object becomes too close to the edge of the camera's field of view (EFOV). However, the authors did not give quantitative values of what is considered as too close to the EFOV and which camera is the most qualified camera to track the handoff object.

In these approaches, the detection and tracking are applied in separate cameras, and only one camera is actively working at a particular time stamp. Therefore, it fails to detect and track objects during dynamic occlusions as this is one of the problems with single-camera visual surveillance systems. Since there is very limited information exchange between the cameras, this camera switch approach is classed as a low-degree information fusion method.

2.1.2 Intermediate-degree Information Fusion

If the correspondence of the objects between cameras is established, it is possible to not only track objects as they move from one camera view to the other, but also to robustly align the trajectories in multiple views and fuse them for a improved tracking result. Khan and Shah [14] extended their work proposed in [11] by creating associations across views for a better localization of the object. The trajectories from each camera are fused into the reference view if that camera view can be visible in the reference view. In [15], the authors align the multiple views with the viewpoint of the camera which can give a clear view of the scene and fuse the trajectories from multiple views in that view. In [16], the multiview process uses Kalman trackers to model the object position and velocity, to which the multiple measurements input from the single-view stage are associated.

In addition to the tracking data, extracted features in individual camera views can be integrated into a reference view to obtain a global estimation. The extracted features include bounding boxes, centroids, principal axes and classification results. In [17], a motion model and an appearance model of each detected moving object are built. Then the moving objects are tracked using a joint probability data association filter in a single camera view. The bounding boxes of the moving objects are projected to a reference view according to the ground-plane homography to correct falsely detected bounding boxes and handle occlusions in the reference view. Du and Piater [18] use particle filters to track targets in the individual camera views and then project the principal axes of the targets onto the ground plane. After tracking the intersections of the principal axes using the particle filter on the ground plane, the tracking results are warped back into each camera view to improve the tracking in the individual camera views. Hu et al. [19] also project the extracted centre principal axes of each foreground object from the individual camera views to a top view according to the homography mapping for the ground plane. The foot point of each object in the top view is determined by the intersection of the axes projections from two camera views. The tracking is based on the foot point locations in the top view. In [20], global tracking is based on the intersections of the 3D lines, in which the centroids of the tracking targets are mapped from multiple views to 3D lines in terms of the world coordinates.

These methods are grouped into the intermediate-degree information fusion category of the multiview methods. Although these methods attempt to resolve dynamic occlusions through the integration of information from additional cameras as occlusions might not occur simultaneously in all the cameras viewing an object, they are still vulnerable to occlusion. The reason is that features are extracted from the individual camera views before fusion, and problems that arise in the detection and tracking with a single camera will affect the final fusion result.

2.1.3 High-degree Information Fusion

In recent years a third category of multiview methods has emerged, in which the individual cameras no longer extract features but provide foreground bitmap information to the fusion centre. The objects are detected as the visual hull intersections of these foreground bitmaps from multiple views. In [21], homography mapping is used to combine foreground likelihood images from different views to resolve occlusions and determine regions on the ground plane. In [22] and [23], the midpoints of the matched foreground segments in each pair of

cameras are back-projected to yield points in the 3D world. These points are then projected onto the ground plane to generate the probability distribution map of the object locations. Berclaz, Fleuret and Fua [24, 25] divided the ground plane into grids to calculate the occupancy map in the ground plane. The probability that each sub-image corresponds to the average size of a person in each camera view is warped from each camera view to the top view for the ground-plane homographies independently.

The ground plane was later extended to a set of planes parallel to, but at some height off the ground plane to reduce false positives and missed detections [4]. In [26], a similar procedure was followed but the set of parallel planes are at the height of people's heads. This method is able to handle highly crowded scenes because the feet of a pedestrian are more likely to be occluded in a crowd than the head. Their work achieves good results in moderately crowded scenes. The third category fully utilizes the visual cues from multiple cameras and has high-degree information fusion. This thesis will focus on the approaches in this category.

2.2 A Review of Multi-View Association

In multiple camera visual surveillance systems, one essential step is to establish the correspondence of objects between the cameras. In low-degree information fusion, feature matching is normally used to determine the correspondence among cameras and label the same target constantly when the camera is switched from one to another. In intermediate-degree and high-degree information fusions, the geometric constraints are used to associate different camera. The geometric constraints can be grouped into three categories: central projection based methods, epipolar line based methods and homography based methods.

2.2.1 Central Projections

Central projection is a fundamental method to establish correspondence across multiple camera views [27, 28]. The information used in the fusion is projected from individual camera views into the world coordinate system to establish equivalence between projected objects at the same location [29-34]. Using a pinhole camera, in the forward projection, the three-dimensional world is mapped onto a two-dimensional image plane which means all points that lie on a line passing through the centre of the pinhole camera are mapped onto the same point in the image plane. When an image point is warped back from the image plane into the

world coordinate system, the mapping is an invertible one–to-many problem as that point corresponds to all points on that line. Since this leads to some of the classical problems in the establishment of correspondence across views, some features detected in the individual camera views are used to establish the correspondences. When feature points are transformed into the same world coordinate system, matching of the feature points is undertaken to indicate the corresponding feature points in the different views. The corresponding feature points in different views are projected at the same point in the world coordinates. In [35, 36], the centroids of the detected objects are used to establish the correspondences.

2.2.2 Epipolar Lines

The epipolar line is another constraint to associate objects across multiple camera views. When a point in one camera view is mapped to another camera view, all points lying on the epipolar line in the second view are potential candidates corresponding to the point in the first view. Since the epipolar constraint cannot build a unique correspondence between the two camera views, some feature matching approaches need to be added in the line search to help establish the correspondence across different camera views.

In [37], the epipolar geometry is improved by using detected faces, view volumes and colour cues. When the centroid of a face box in one view is projected to another view, the centroid of the face box in the new view, which has the least distance with the projected epipolar line, is considered as the corresponding centroid of the face box. Moreover, each view is divided into some non-overlapping view-volumes according to vertical features in the image. The matching of the view-volumes between the two cameras is embedded in the epipolar line method. The authors also used the hue and saturation values of each person as features to help establish the correspondence.

In [22] and [23], after a background subtraction process, Mittal and Davis match the colours of all the foreground regions along the epipolar lines, in pairs of cameras. Then, the midpoints of the matched regions in each pair of cameras are back-projected to yield 3D points. These points are then projected onto the ground plane to generate a probability distribution map to indicate object locations. Using the outlier-rejection technique, the probability distribution map is then used to compute the 2D positions of each object.

2.2.3 Planar Homographies

To solve the one–to-many problem in the central projection, a world plane is introduced [19, 38-40]. Using a common plane, when a point on that plane in the image is warped into the world coordinate system, its corresponding point in the world coordinate system lies on the intersection of that plane and the line passing through the centre of the pinhole camera. The common plane is a realistic assumption because most of the scenarios being monitored contain the ground plane, which make the imaging equation invertible.

In addition to using homography to link a camera view and the world coordinates on the ground plane, the projected world point can be projected into the same ground plane of a second image [41]. The correspondence between the points on the two image planes can be easily found. This property is referred to as the homography induced by that plane, which can be used to find correspondences across different camera views. A literature review on using homography to associate objects in multi-camera visual surveillance is provided in the next section.

2.3 A Review of Foreground Fusion with Homographies

Based on the features which are detected in individual camera views and are fused in the reference view to locate objects, homography mapping can be divided into three categories: pixel-based methods, line-based methods and region-based methods.

2.3.1 Foreground Pixel-based Methods

Since homography is a pixel-level projection, most of the foreground projections and fusions are based on pixels. In Khan and Shah's work [4], the foreground likelihood image, which is extracted from each of the multiple camera views, is warped to a reference camera view according to the ground-plane homography. The projected foreground likelihoods from multiple cameras are then overlaid in the reference view. A threshold is applied in the reference view to determine the locations of people on the ground-plane. Then, the homographies for a set of parallel planes at different heights are employed to reduce false positives and missed detections. This approach achieves good results in moderately crowded scenes, because regions at the locations of true objects reinforce each other whereas the false locations are scattered around. In [42], the number of foreground

pixels above each foreground pixel on the vertical direction is calculated as a support of that foreground pixel and is normalized by a factor related to the perspective cross-ratio. The supports in each camera view are projected onto a virtual ground plane to determine the locations of pedestrians.

2.3.2 Foreground Line-based Methods

Foreground line-based methods use the principal axes of people as the feature in the homography mapping, as foreground pixels of a person are often symmetrically distributed along the principle axis. This can reduce the influence of motion segmentation errors. In [18, 19, 43], the authors projected the central vertical axis of each foreground object from individual camera views to a top view according to the homography of the ground plane. Then, the foot point of each pedestrian is determined as the intersection of the projected axes in the top view.

2.3.3 Foreground Region-based Methods

Arsic et al. [44] warped the contours of the detected foreground regions from each camera view to a virtual top view according to homographies. In [45], the silhouettes of the foreground in each camera view are applied in the multi-view moving object detection. Berclaz, Fleuret and Fua [24] divided the ground plane into grids to calculate an occupancy map in the ground plane. Each sub-image delimited by the grids is a rectangle that corresponds to the average size of a person in each camera view. The probability the each sub-image has a person is warped from the camera view to the top view by using the ground-plane homographies.

2.4 A Review of Colour Matching

As colour is a strong cue to identify moving objects, colour matching has been successfully applied in intra-camera tracking and inter-camera tracking. For intra-camera tracking, objects from different frames within one camera view are matched. For inter-camera tracking, it focuses on the association of the moving objects observed in different camera views.

Orwell et al. [46] proposed two methods to build the colour histogram of each object. In the first method, mixture of Gaussians parameterization is combined with cross-entropy distance to determine a matched measure using the aggregate colour signal of the observed objects. Based on maximum entropy and a χ^2 distance

measure, an explicit representation of colour is employed in the second method. Cheng et al. [47] built an appearance model of pedestrians based on the major colour spectrum histogram representation (MCSHR), where integrated MCSHR within 3-5 frames was applied to measure the similarity of moving objects in coping with small pose changes.

In Bowden and KaewTraKulPong [48], intersection of colour histograms in three colour spaces, RGB, HSL, and consensus – colour conversion of Munsell colour space (CCCM), were used for colour based object matching. In Gilbert and Bowdens [49], an incremental learning method was applied to model both the colour variations and posterior probability distributions of spatio-temporal links between cameras. In Park et al [50], each detected pedestrian was divided into three parts from the top to bottom, and the colours of the lower two parts in the HSV space were combined to generate a histogram for object matching. Porikli [51] proposed a distance metric and a non-parametric function to model colour distortion for pair-wise cameras and evaluate the inter-camera radiometric properties. Javed et al. [52] introduced an appearance model for each pedestrian by using colour histograms, in which the correspondence of pedestrians was established based on learning the usual change in colour of pedestrian during their moving between cameras, with the Bhattacharyya distance used to measure whether two observations were from the same object.

2.5 A Review of Phantom Removal

There are a number of algorithms which aim to remove these phantoms in foreground projection intersections. One solution is to avoid the generation of phantoms. Adding more cameras can provide a wider field of view and reduce the probability that a region cannot be visible in all views. Although additional cameras can reduce the sizes and number of phantoms, it is limited by the cost of the additional cameras [26]. Stering et al. [53] applied the idea of generalized Hough voting in the homography projection. Hough voting relates all foreground probabilities to a position on the ground plane and restrains the shadow generation. However, the authors stated that they cannot handle the case when objects cannot be visible in all camera views.

Since phantoms are often gradually created and merge back into real objects, distinguishing and detecting them on the basis of position is difficult [54]. Therefore, another approach often removes phantoms in the tracking process. In [55], Yang et

al. pointed out that phantoms appear from nowhere and checked their temporal coherence to test if a foreground intersection region existed in the previous frame. Khan and Shah [4] also filtered out the phantoms according to the temporal coherence. In Liem and Gavrila's work, they assumed that phantoms are often unsteadily detected and checked the temporal coherence measured by a 'hidden' time rather than a single previous frame. If such a candidate cannot survive over the hidden time in tracking, then it is classified as a phantom. They also proposed that a new real object can only appear from the boundary of the overlapping field of views (FOVs); objects which are first detected in the middle of the overlapping FOVs are phantoms [54, 56].

The geometric approach is built on the comparison of features between phantoms and real objects. This approach can be further divided into two sub-classes: 3D space and 2D image methods, according to the types of geometric constraints that are used. The features applied in the geometric approach include heights and sizes. In the 3D space method, the comparison is in 3D space or in a virtual top view. Tong et al. [57] utilized foreground projection on multiple planes at different heights to removed phantoms. In [55], Yang et al. pointed out that the size of a phantom is often smaller than the minimum object size in the top view. However, this assumption is related to the height and viewing angle of the camera, and it does not work when a phantom region is covered by a real object in all camera views. Eshel and Moses [26] used the height information and assumed that the cameras are looking downwards. They found that if the viewing rays from two cameras intersect behind a true object, the phantoms are lower than the true object, taller phantoms occur when the rays intersect in front of true objects. By limiting the heights of real objects within an appropriate range, they could remove some phantoms.

Some methods use the 2D information to identify the phantoms. Arics and Hristov [44] warped the intersection regions from the top view back into each camera view and checked if they are totally covered by the foreground regions. If warped back regions are totally covered in all views, it is considered a phantom. Peng et al. [58] learned an occlusion relationship by using a Bayesian network in a single camera view and then removed phantoms according to a multi-view Bayesian network. In [42], a filtering algorithm is applied to remove covered pixels by checking whether a pixel on the virtual ground plane is occluded in all views. In Eshel and Moses's work [26], they applied the pixelwise intensity correlation between aligned frames in a reference view to remove phantoms. In [59], the

foreground masks from all camera views are projected to a centroid plane to generate a occupancy likelihood map. The occupancy likelihood map is transformed to occupancy likelihood rays in the polar coordinate representation in each camera view, in which the origin of the polar coordinate is at the camera centre. The distance between the intersection region and the origin and the angle that each intersection region covered illustrate the death information and the size of that intersection region. Then, the depth information and covered angles are used to identify the occlusion relationship and remove phantoms.

2.6 Summary

In this chapter, a literature review on the multi-camera visual surveillance has been presented. In the review of multi-camera visual surveillance systems, the existing systems are divided into three categories according to the degrees of the information fusion from the multiple camera views. To associate multiple cameras in the information fusion, the geometrical methods are reviewed. According to the features used in the homography projection, the related research using homography in multi-camera visual surveillance is reviewed. Finally, the existing algorithms related to the false-positive detections are discussed. In the next four chapters, the details of the proposed algorithms form moving object detection and phantom removals researched in this thesis are presented.

3 HOMOGRAPHY ESTIMATION

In the previous chapter, methods to associate multiple camera views, which include the homography approach, were discussed. The objective of this chapter is to introduce the concepts of homography such as the homography transformation and homography estimation. Initially an introduction to projective geometry, the basics of a perspective camera such as the camera model, projections in homogeneous coordinate and camera calibration are discussed.

3.1 Camera Models and Calibration

Camera calibration is an important aspect in measuring the three-dimensional world. It provides a mechanism to build the relationship between a point in the 3D world and a point in the 2D image. It aims to estimate both the intrinsic parameters (such as focal length, principal points, skew coefficients and distortion coefficients) and extrinsic parameters (such as position of the camera centre and the orientation of the camera in world coordinates).

In camera calibration, the first step is to select a camera model and then to estimate parameters in that model. For many computer vision applications, the pinhole camera model has been widely used. It imagines a tiny hole on a virtual wall and assumes that the tiny hole only accepts the light rays passing through the tiny aperture in the centre and blocks other light rays. The central projection in the pinhole camera is depicted in Figure 3.1.

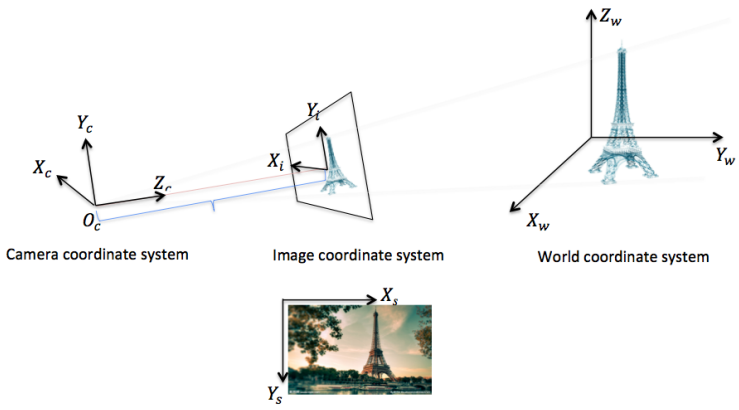


Figure 3.1 The geometry of a pinhole camera.

In Figure 3.1, there are four coordinate systems: the world coordinate system with subscript *w*, the camera coordinate system with subscript *c*, the image coordinate system with subscript *i*, and the pixel coordinate system with subscript *s*. A point in the world coordinate system needs three steps to be transformed into the pixel coordinate system. In the first step, the world coordinate system and the camera coordinate system are aligned by translating the origin of the world coordinate system to the origin of the camera coordinate system with a translation vector **t** and then by rotating the align axes with a 3×3 rotation matrix **R** which can be expressed by three elementary rotations. Since the translation vector **t** also contains three elements, the six parameters which define the orientations and 3D position of the camera are called extrinsic parameters in the camera calibration. Using the homogeneous coordinates, let $\mathbf{X_c} = (X_c, Y_c, Z_c, 1)$ be a point in the camera and $\mathbf{X_w} = (X_w, Y_w, Z_w, 1)$ be the corresponding point of $\mathbf{X_c}$ in the 3D world, the relationship that maps $\mathbf{X_w}$ to $\mathbf{X_c}$ can be denoted as:

$$\mathbf{X_c} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{Rt} \\ \mathbf{0^T} & \mathbf{1} \end{bmatrix} \mathbf{X_w} \tag{3.1}$$

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix} \tag{3.3}$$

In the second step, the point is projected from the camera coordinate system to the image plane. The similar triangle relationships between these two coordinate systems are used because they have collinear axes in the *Z* direction. Let $\mathbf{x_i} = (x_i, y_i, 1)$ be the corresponding point of $\mathbf{X_c}$ in the image coordinate system, the relationship that maps $\mathbf{X_c}$ to $\mathbf{x_i}$ can be denoted by two equations:

$$x_i = f\,\frac{X_c}{Z_c}, \quad y_i = f\,\frac{Y_c}{Z_c} \tag{3.4}$$

where *f* is the focal length of the lens, which is the distance between the principal point and the image plane.

In the third step, the ratio between the camera sensor and the image pixels defines another transformation. Let $\mathbf{x_s} = (x_s, y_s, 1)$ be the corresponding point of $\mathbf{x_i}$ in the pixel coordinate system, the relationship that maps $\mathbf{x_i}$ to $\mathbf{x_s}$ can be denoted as two equations:

$$x_s = \frac{1}{s_x}\,x_i + C_x, \quad y_s = \frac{1}{s_y} y_i + C_y \tag{3.5}$$

where s_x and s_y are the sampling frequencies in the X_s and Y_s axis, which are the number of pixels per unit length; $C_x$and C_y are the principal point.

These two transformations which are a function of the camera can be described by a 3×3 intrinsic calibration matrix **K:**

$$\mathbf{K} = \begin{bmatrix} f/s_x & s & C_x \\ 0 & f/s_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

where s in **K** is an effective scale factor which defines how far the axes are skewed in the direction of axis.

Then the full pinhole camera projection is generated. The relationship that maps $\mathbf{X_w}$ to $\mathbf{x_s}$ can be denoted as:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x & 0 \\ 0 & f/s_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.7}$$

The equation of the perspective projection is rewritten using a 3×4 projection matrix **M**:

$$\mathbf{x_s} = \mathbf{M}\mathbf{X_w} \tag{3.8}$$

Using a set of image points and the corresponding world coordinates, the extrinsic parameters and the intrinsic parameters in the perspective projection matrix **M** can be recovered and the camera calibrated. A simple approach uses a set of reference points involved the determination of transformation parameters to solve linear equations with known reference parameters [60]. Although it can provide an accurate and fast calibration result, the lens distortion cannot be handled and the reference points are hard to select in this approach.

The photogrammetric calibration method can calibrate not only intrinsic parameters and extrinsic parameters but also distortion factors. Tsai's algorithm can recover one distortion factor in the camera [61]. Since it deals with coplanar and non-coplanar points, Tsai's algorithm is suited to the wide area scenarios. The coplaner approach uses at least five pairs of corresponding points on the same plane of the 3D world. Without the assumption that all landmark points are on the same plane, the non-coplaner approach needs at least seven pairs of corresponding points. Zhang's method can calibrate five distortion factors [62]. It needs at least 2 frames of the same chessboard captured from different orientations.

Some method uses vanishing points to estimate the camera's intrinsic and extrinsic parameters on the basis of geometric relationships such as parallelism and orthogonally in the scene [63]. The vanishing points are the converging points of parallel lines in a perspective projected image, which can be estimated from static scene structures, such as image edges which are either parallel or perpendicular in the world and landmarks [64] [65]. When architectural structures do not exist in the scene, the vanishing points can be estimated from object motion. In [66] [67], the authors detect the head and feet position of people walking during their leg-crossing phases. The line segments between heads and feet are used to estimate the camera's intrinsic and extrinsic parameters. Zhang et al. [68] use the motion and appearance of moving objects and assumed camera height to estimate three vanishing points corresponding to three orthogonal directions in the 3D world coordinate system.

Since this research didn't need very accurate calibration results, the simpler Tsai's algorithm was used to calibrate the cameras and decide the relationship between a point in the 2D image and the same point in the 3D real world.

3.2 An Introduction to Homography

As discussed in chapter 2, planar homography is a special relationship, defined by a 3×3 transformation matrix $\mathbf{H}$ between a pair of captured images of the same plane with different cameras:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{3.9}$$

Let (x, y) and (x', y') be a pair of corresponding points on that plane in the two images. $\mathbf{x} = [x, y, 1]^T$ and $\mathbf{x}' = [x', y', 1]^T$ are the homogeneous coordinates of those two points. They are associated by the homography matrix $\mathbf{H}$:

$$\mathbf{x}' \cong \mathbf{H}\,\mathbf{x} \tag{3.10}$$

where $\cong$ denotes that the homography is given up to an unknown scalar.

3.3 Estimation of Planar Homography

Homographies are usually estimated between a pair of images by finding feature correspondence in these images. The most commonly used feature is corresponded

points in different images, though other features such as lines or conics in the individual images may be used [28] [69] [70]. These features are selected and matched manually or automatically from 2D images to compute the homography between two camera views or the homography between one camera view and the top view [71] [39]. Point based features are the most commonly used in estimating the homography, such as Harris corner points [72] and Scale-Invariant Feature Transform (SIFT) points [73]. In this thesis a set of manually selected corresponded points are used to estimate the homographies.

3.3.1 Point Correspondences

Given a set of corresponding points, the algorithms used to calculate the homography matrix can be divided into two classes: maximum likelihood estimation method and linear estimation method [74]. Given a pair of corresponding points (x_i, y_i) and (x_i', y_i'), equation (3.10) becomes two linear equations about **H**. The homography estimation is a process to calculate the solution of a set of linear equations about **H**. Since the homography matrix **H** is a homogeneous matrix, it only has 8 degrees of freedom or 8 unknowns which need to be solved. Then, if the number of correspondence point pairs N is 4 and no three points are collinear, the 8 unknowns of the homography matrix **H** can be solved uniquely with the 8 equations. If the number of correspondence point pairs N is more than 4, no exact **H** cannot be determined uniquely. Solving linear equations becomes the problem of optimal estimation of the parameters in **H**. Maximizing the likelihood and minimizing the algebraic distance are two methods to find the optimal parameters in **H**.

The Direct Linear Transform (DLT) algorithm [28] is the most widely used method to estimate the homography matrix **H**. Using the pair of correspondence points $\mathbf{x}_i = (x_i, y_i)$ and $\mathbf{x}_i' = (x_i', y_i')$, if the first row and the second row of equation (3.10) is divided by the third row respectively, equation (3.10) can be re-written as:

$$-h_{11}x_i - h_{12}y_i - h_{13} + (h_{31}x_i + h_{32}y_i + 1)x_i' = 0 \tag{3.11}$$

$$-h_{21}x_i - h_{22}y_i - h_{23} + (h_{31}x_i + h_{32}y_i + 1)y_i' = 0 \tag{3.12}$$

Equations (3.11) and (3.12) can be further rearranged in a matrix form:

$$A_i\mathbf{h} = \mathbf{0} \tag{3.13}$$

where $A_i = \begin{pmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i'x_i & x_i'y_i & x_i' \\ 0 & 0 & 0 & -x_i & -y_i & -1 & y_i'x_i & y_i'y_i & y_i' \end{pmatrix}$, $i \in [1, N]$

and $\mathbf{h} = (h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad 1)^T$.

A point matrix **A** is constructed by $\boldsymbol{A}_i, i \in [1, N]$ and then equation (3.13) can be rewritten as:

$$\mathbf{Ah} = \mathbf{0} \tag{3.14}$$

Equation (3.14) is over-determined and there is no solution in general. Choosing a suitable distance is considered and minimizing that distance can compute vector **h**. For the algebraic distance, the vector **h** can be computed by using Singular Value Decomposition (SVD) that minimizes the norm $\|\mathbf{Ah}\|$ subject to $\|\mathbf{h}\| = 1$ [75]. In this research, the algebraic distance was selected because it has the least computation cost. Beside the algebraic distance, the geometric distance such as the total transfer error, the symmetric transfer error or the re-projection error for the corresponding point pairs $\mathbf{x}_i$ and $\mathbf{x}_i', i \in [1, N]$, can also be used.

3.3.2 Robust Estimation

When the input point correspondences are affected by inaccurate point correspondences or corrupted by false correspondences, to maintain the robustness of the homography estimation, these outlier correspondences should be removed. Random Sample Consensus (RANSAC) [76, 77]is a commonly used optimization method to remove outliers. The application of RANSAC for homography estimation works as follows:

1. A random sample of four correspondences is selected and a homography matrix **H** is computed from those four correspondences.
2. Each other correspondence is then classified as an inlier or outlier according to its concurrence with **H**.
3. Step 1 and 2 are repeated for a number of iteration. When all of the iterations are completed, the iteration which contains the largest number of inliers is selected. These inliers correspondences are used to estimate the final homography **H**.

3.3.3 Camera Calibration

The homography transformation is a special variation of the projective transformation. The point $\mathbf{x} = (x_s, y_s, 1)$ in the image without distortion and the point $\mathbf{X} = (X_w, Y_w, Z_w, 1)$ in the 3D world are limited on the ground plane. Therefore Z_w is 0 and the projection transformation from **X** to **x** becomes:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x & 0 \\ 0 & f/s_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} \tag{3.15}$$

Then, equation (3.15) can be rewritten as:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x \\ 0 & f/s_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{3.17}$$

The parameters recovered in the camera calibration can be used to determine the homography matrix for the ground plane.

3.4 Estimation of Multi-Plane Homographies

Homography mapping is not limited to the homography for the ground plane, and can be extended to a set of planes parallel to the ground plane and at some height. If the camera is calibrated, the multi-plane homographies can be calculated through the parameters recovered in the calibration process directly. After the homography matrix for the ground plane is estimated, the homography matrix for the planes parallel to and off the ground plane can be recovered from the perspective properties such as the vanishing point in the vertical direction [78] [4] or the cross-ratio of four collinear points [26]. The homography matrix can also be approximated according to the interpreted points [79].

3.4.1 Calibration

For a plane *p* at some height *h*, by assuming that the points **X** and **x** are on that plane, the point $\mathbf{X} = (X_w, Y_w, Z_w, 1)$ on plane p in the 3D world can be denoted as

$\mathbf{X}_p = (X_w, Y_w, 1)$, where $Z_w = h$ is removed. To represent the projection transformation matrix simply, **M** is rewritten as:

$$\mathbf{M} = [\boldsymbol{m_1} \ \ \boldsymbol{m_2} \ \ \boldsymbol{m_3} \ \ \boldsymbol{m_4}] \tag{3.18}$$

According to equation (3.15), the ground-plane homography $\mathbf{H}_g$ can be denoted as:

$$\mathbf{H}_g = [\boldsymbol{m_1} \ \ \boldsymbol{m_2} \ \ \boldsymbol{m_4}] \tag{3.19}$$

Since the result, in which each element in the third column $\boldsymbol{m_3}$ multiples the value $Z_w = h$ in **X,** is a constant value and the last element in the homogrenous vector **X** is 1, according to the homography projection for plane p, the projection from point $\mathbf{X}_p$ in the 3D world to point $\mathbf{x}_p$ in the 2D image is:

$$\mathbf{x}_p = \mathbf{H}_p\mathbf{X}_p = [\boldsymbol{m_1} \ \ \boldsymbol{m_2} \ \ \boldsymbol{m_4} + h\,\boldsymbol{m_3}]\mathbf{X}_p \tag{3.20}$$

The homography of plane p can be represented as a combination of the homography for the ground plane and the third column of the projection matrix **M** multiplied by a given height h:

$$\mathbf{H}_p = \mathbf{H}_g + [\mathbf{0} \mid h\,\boldsymbol{m_3}] \tag{3.21}$$

where [**0**] is a 3 ×2 zero matrix [78].

3.4.2 Vanishing Point

Under perspective projection, parallel lines in the 3D world space intersect at a point in the image. Therefore, one way to determine the perspective projection is to use vanishing points. In equation (3.18), the third column $\boldsymbol{m_3}$ is defined as the vanishing point in the direction normal to the plane defined by $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$. When the camera is not calibrated, the homography for the plane at some height can be estimated by using the vanishing point with a scale factor [78]. Let $\mathbf{v}_{ref}$ be the vanishing point in the normal direction and $\mathbf{H}_g^{a,g}$ be the homography between camera view a and ground plane g, the homography between camera view a and plane p parallel to plane g and at some height h is given as:

$$\mathbf{H}_p^{a,p} = \mathbf{H}_g^{a,g} + [\mathbf{0} \mid \gamma\mathbf{v}_{ref}] \tag{3.22}$$

where γ is a scalar multiple proportional to h and [**0**] is a 3 ×2 zero matrix.

Let $\mathbf{H}_g^{a,b}$ be the homography between camera view *a* and camera view *b* induced by ground plane *g*, the homography between the two camera views induced by plane *p* at a given height *h* is given as by [4]:

$$\mathbf{H}_p^{a,b} = \left(\mathbf{H}_g^{a,b} + \left[\mathbf{0} \mid \gamma \mathbf{v}_{ref}\right]\right)\left(I_{3\times 3} - \frac{1}{1+\gamma}\left[\mathbf{0} \mid \gamma \mathbf{v}_{ref}\right]\right) \tag{3.23}$$

The main work focuses on the extraction of the vertical vanishing point. A static scene structures often contain many vertical parallel lines. These parallel lines in the scene are projected into the image. The projected lines in the image will ideally intersect at the vanishing point in the vertical direction. Therefore, the detection of the vertical vanishing point begins with the Canny edge detection [80] and the Hough transform [81]. Then a line detection algorithm is used to extract the dominant vertical lines. Finally, the intersection point of these vertical lines is calculated by minimizing the sum of its squared distances to all these lines.

3.5 Experiments and Analysis

Two video sequences were used to evaluate the performance of the algorithms proposed in this thesis. The first dataset is the PETS'2001 dataset which is standard image sequences for testing tracking and surveillance algorithms. The second dataset was captured at the thesis author's campus. The ground-plane homography and homographies for planes parallel to and off the ground plane were calculated for each dataset.

3.5.1 Experimental Results of the PETS'2001

For the homography estimation, the top view image was selected as the reference image. To compute the homography matrix between each camera view and the top view, at least 4 pairs of correspondence points are needed. Although the world coordinates of the five landmark points in the dataset are provided, they cluster on one side of the image, which leads to inaccuracies in the homography estimation. As an alternative, a Google satellite image (http://maps.google.com/) for the same site was used as the top view image.

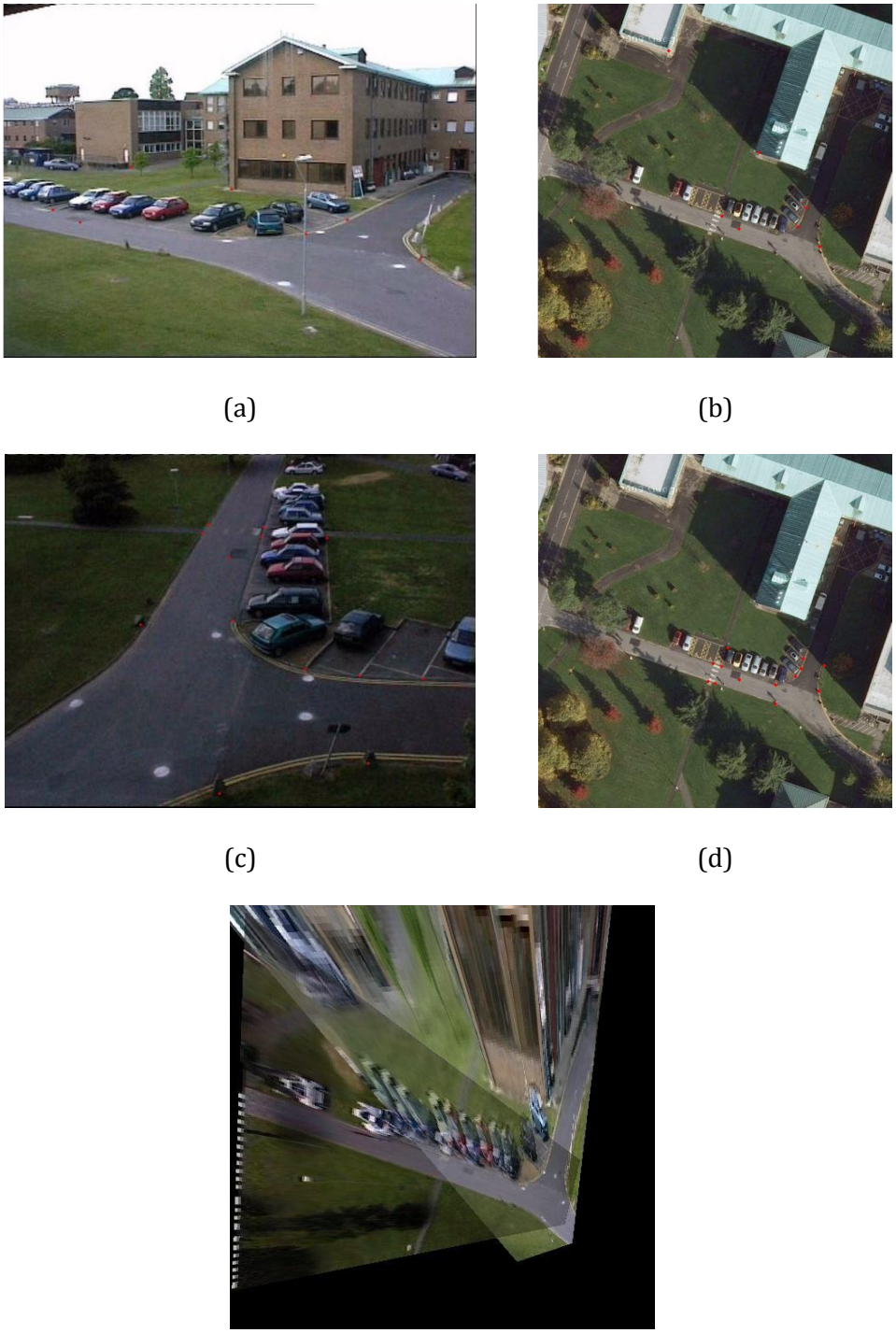
There has been some research which can calculate the homography matrix or calibrate multiple cameras automatically. The matching of the feature points such as Harris corners and Scale-Invariant Feature Transform points needs to be robust to the variations of viewpoints and lightings between camera views. The

correspondences between feature points which are obtained automatically may include a significant amount of false matches. In this thesis, to calculate the homography for the ground plane, a set of landmark points were manually selected in the two camera views and the top view. The homography matrix is estimated as an offline process before the online moving object detection, which is a common practice in video surveillance systems.

Figure 3.2 illustrates the process of the homography estimation by selecting a set of points in the top view. Figure 3.2 (a)- (d) shows the point set in two camera views and their corresponding top views. It should be noticed that (a) and (c) were not captured at the same time with the top view. Figure 3.2 (e) is a synthetic image generated by projecting and fusing the two camera views in the top view, in which the edges of the road are aligned.

3.5.2 Experimental Results of the Campus Dataset

Using a set of corresponding point pairs which were selected manually from each of the two camera views and a virtual top view, the camera can be calibrated by using Tsai's algorithm. Figure 3.3 shows the landmark points collected in two camera views and a virtual top view. In Figure 3.3 (a) and (b), some orange and pink landmark points were marked on the ground so that the orange marks and the pink marks are staggered. The distance between an orange mark and its adjacent pink mark is 0.6 m. The points in Figure 3.3 (c)-(e) show the landmarks selected in the two camera views and the virtual top view. Using these corresponding point pairs, the two cameras were calibrated using Tsai's algorithm. The intrinsic and extrinsic parameters of the two cameras are shown in Table 3.1 and Table 3.2.



(a) (b)

(c) (d)

(e)

Figure 3.2 Procedure for the homography estimation in the PETS'2001 dataset, (a) camera view *a* with selected feature points, (b) the top view with the corresponding feature points, (c) camera view *b* with selected feature points, (d) top view with the corresponding feature points, (e) the homography projection and fusion of the two camera views in the top view.

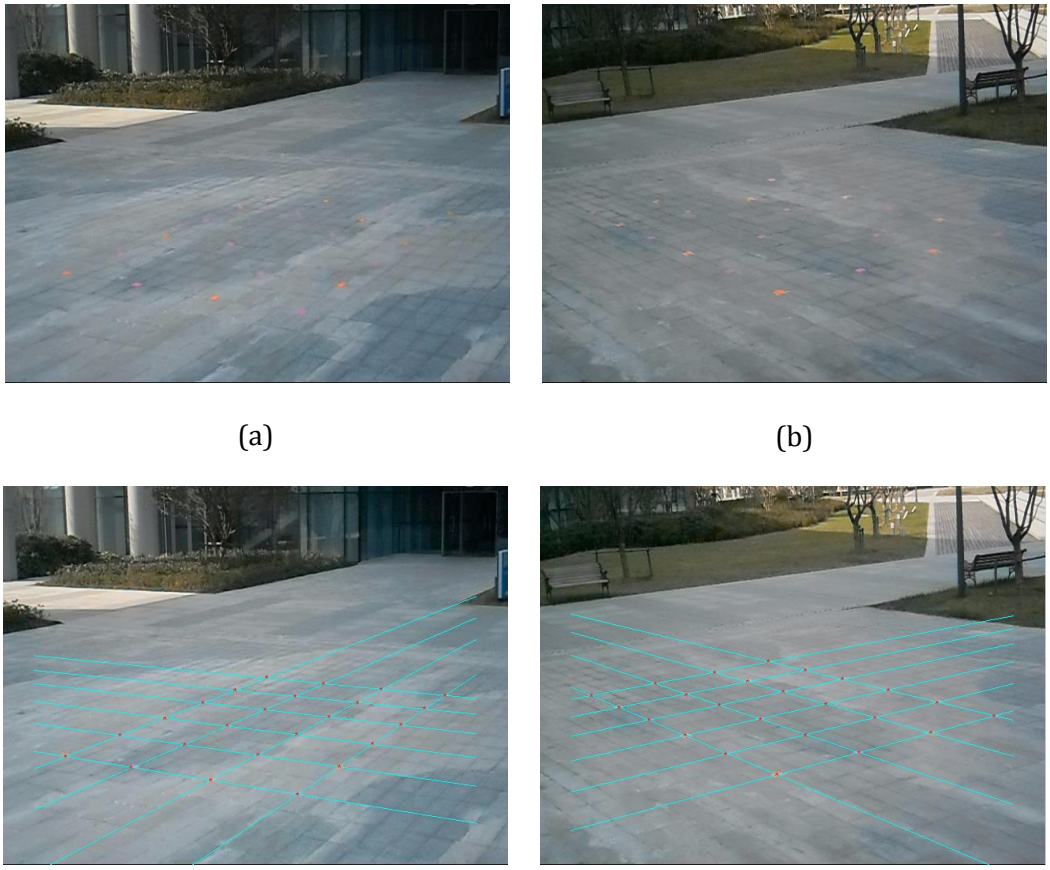


Figure 3.3 The landmark points collected in two camera views and a virtual top view, (a) original background image in camera view *a*, (b) original background image in camera view *b*, (c) landmark points in camera view *a*, (d) landmark points in camera view *b*, and (e) landmark points in the virtual top view.

Table 3.1 The intrinsic parameters of the two cameras in campus dataset.

	K (mm^{-2})	f (mm)	C_x (pixels)	C_y (pixels)
Camera *a*	-2.37×10^{-4}	22.76	320	240
Camera *b*	3.07×10^{-4}	26.59	320	240

Table 3.2 The extrinsic parameters of the two cameras in campus dataset.

	t_x	t_y	t_z
Camera *a*	-98.0	-68.4	1.47×10^3
Camera *b*	-6.43×10^2	32.28	1.13×10^3

(a) The translation vector **t**.

	r_{11}	r_{12}	r_{13}	r_{21}	r_{22}	r_{23}	r_{31}	r_{32}	r_{33}
Camera *a*	8.73 $\times 10^{-1}$	-4.86 $\times 10^{-1}$	1.80 $\times 10^{-2}$	7.54 $\times 10^{-2}$	1.72 $\times 10^{-1}$	9.82 $\times 10^{-1}$	-4.80 $\times 10^{-1}$	-8.56 $\times 10^{-1}$	1.87 $\times 10^{-1}$
Camera *b*	8.37 $\times 10^{-1}$	5.43 $\times 10^{-1}$	6.59 $\times 10^{-2}$	-1.64 $\times 10^{-1}$	1.35 $\times 10^{-1}$	9.77 $\times 10^{-1}$	5.21 $\times 10^{-1}$	-8.29 $\times 10^{-1}$	2.02 $\times 10^{-1}$

(b) The rotation matrix **R**.

When the intrinsic and extrinsic parameters have been recovered, using equation (3.16), the ground-plane homographies for the each of the two camera views are calculated. To show the accuracy of the estimated homography, the background images of the two camera views in Figure 3.3 are projected and fused in the top view (Figure 3.4).

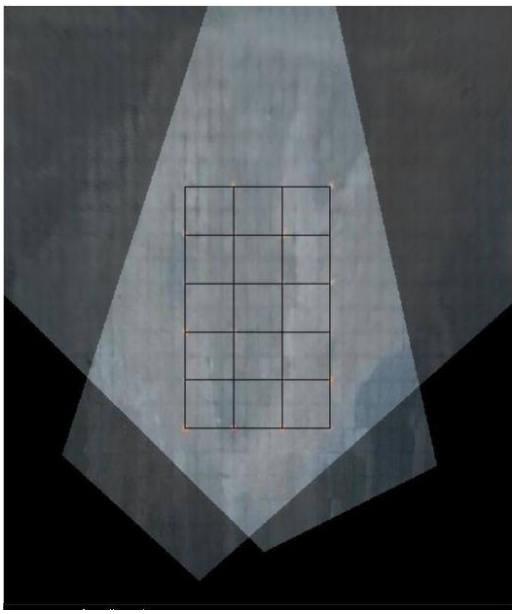


Figure 3.4 Fusion of the projected background images in the top view.

Then, using equation (3.21), the homography for a plane parallel to and at the height of h can be calculated. In Figure 3.5, the selected points in Figure 3.3(e) are warped back from the top view to camera view a according to the homography for a plane at the height of 1 meter. The warped back points are the green points in Figure 3.5, which are connected with their corresponding landmark points on the ground plane.

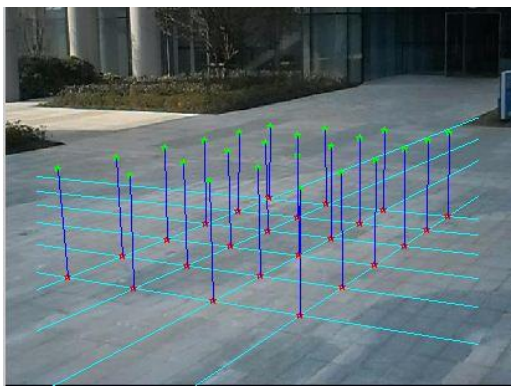


Figure 3.5 Warped back points corresponding to the selected points in Figure 3.3(e).

3.6 Summary

In this chapter, the projective geometry and the concept of homography were introduced. The method to estimate the ground plane homography and homographies for the plane parallel to the ground plane and at some heights were

discussed. If the distortion can be ignored, when the camera is calibrated, the homography for the ground plane and homographies for the plane parallel to the ground plane and at a given height can be calculated from the perspective transformation matrix directly. If the camera is not calibrated, the homography for the ground plane can be calculated according to a set of corresponding landmark points which are selected from the camera view and the reference view. The calculation of the homographies for the plane parallel to the ground plane is divided into two categories: the vanishing point based method and the approximation based method.

4 MOVING OBJECT DETECTION WITH REAL-TIME FUSION

In the previous chapter, the algorithms to calculate homographies for the ground plane and the multiple planes parallel to the ground plane at different heights were introduced. In high-degree information fusion, using the calculated homography matrices, the foreground image, which is extracted from each of the multiple camera views, can be mapped to a reference view.

In traditional homography mapping, each pixel in a camera view needs to be projected into the top view according to the homography for a plane. Since this mapping is a pixelwise projection, the number of pixels to project is determined by the resolution of the camera view. To avoid perspective openings or holes which are generated during the mapping from the camera view to the top view, each pixel in the top view is warped back to the camera view according to the inverse homography transformation. If a warped back pixel is located in a foreground region in the camera view, the original pixel in the reference view is labelled as a foreground pixel. The number of pixels in the homography mapping is decided by the resolution of the top view, which is usually higher than that of the camera view because the top view normally covers a larger area.

Due to the high resolution of the top view, the pixelwise homography mapping is very time consuming and needs high bandwidth, which makes it hard to apply the homography approach in real-time applications. This brings about a challenging requirement on the bandwidth of multi-camera networks. If the foreground detection and multiview foreground fusion are carried out by different computers, the pixelwise homographic transformations at image level, for multiple cameras and multiple parallel planes, are more time consuming and thus prevent any low cost real-time implementation.

In this chapter, a real-time homography projection algorithm for the fusion of the detected foregrounds from multiple cameras will be discussed. The moving object detection starts with a single-camera foreground detection, in which a Gaussian mixture model and background subtraction are used to detect the foreground pixels in the individual camera views. Then, the detected foreground

pixels in each frame are grouped into foreground regions by applying connected component analysis, morphological operations and a size filter. Once the foreground regions have been identified in a camera view, the foreground regions need to be projected to a reference view. As a pixelwise homographic transformation is time consuming, each foreground region is approximated by the polygon of the foreground region's contour. The Douglas-Peucker (DP) method [82] has been used for the polygon approximation. Instead of applying the inverse homography to each pixel in the reference view, the vertices of the polygon of each detected foreground region are projected to a reference view through homography mapping. To evaluate the performance, the proposed polygon approximation method has been compared with the contour based method and the bounding-box based method.

4.1 Foreground Segmentation in a Single View

As an essential process in visual surveillance systems, foreground segmentation aims to separate moving objects from a background image in each frame. For a fixed camera, foreground detection suffers mainly from the change of lighting conditions and noise as well as the motion of the moving objects. Relevant techniques for foreground segmentation are discussed in the following sections.

4.1.1 Introduction to Foreground Segmentation

The foreground segmentation methods can be divided into three categories: optical flow, temporal differencing, and background subtraction. The first category is the optical flow method which detects moving regions on the basis of the flow vectors of moving objects [83]. The distance that each pixel has moved between the previous frame and the current frame indicates the velocity of that pixel in an image. This method can be divided into two categories: dense optical flow method and sparse optical flow method. For the dense optical flow method, the velocity of every pixel in the image needs to be calculated, which leads to a high computational cost. In [84], the position of each pixel was computed by using the monotony operator in two successive frames. These positions are used to compute a displacement vector field which can be used to extract articulated objects during the tracking. The sparse optical flow method only tracks the feature points in the image. Although the computation cost of the sparse optical flow method is less than that of the dense optical flow method, the computation cost is higher than those for the temporal differencing and background subtraction methods.

Temporal differencing methods use the pixelwise differences between consecutive frames in an image sequence. Each pixel with a significant difference is classified as a foreground pixel. The image-differencing method can adapt to a dynamic environment quickly [85]. This approach can be improved by applying three-frame differencing instead of two-frame differencing [86]. However, if there is an object which moves slowly, the neighbours of a foreground pixel will be similar to the pixel itself, therefore, part of the foreground region may unexpectedly be detected as the background.

The background subtraction method involves calculating a background image, subtracting each new frame from the background image and thresholding the subtraction result. Since the foreground pixels are identified according to the pixelwise difference between the new frame and the background image, the method is highly dependent on a good background model, which should not be sensitive to illumination variations, shadows and waving vegetation. The existing algorithms have been proposed to use different temporal and spatial representations. In [87], Elgammal et al. represent a background pixel using a kernel estimator, which can calculate the probability density function of each pixel according to its values in previous N frames. This kernel-based approach can overcome the drawbacks that are caused by a faster or slower updating rate. Seki et al. [88] assumed that the neighbouring blocks of the background should have similar variations over time. After the average and an eigenvector transformation of each block are calculated, a block can be reconstructed by using the linear interpolation ratio obtained from the eigenspace. If the original block and its reconstructed block are similar to each other, it will be recognized as a foreground block. Oliver et al. [89] used eigenvalue decomposition and calculated the average and the eigenvector for each block in an image. By comparing the difference between the input image and the double projected image via the eigenspace, the foreground region image can be identified.

According to the assumption that a background pixel is more stable than a foreground pixel in pixel values, the value of a background pixel is modelled with a Gaussian which is updated by applying a running average algorithm to adapt for actual background variation [90]. Stauffer and Grimson [91] extended this adaptive method by using a mixture of Gaussian distributions to model switching, multiple backgrounds. The sum of the probability density functions weighted by the corresponding priors represents the probability that a pixel is observed at a particular intensity or colour. Pakorn and Richard [92] proposed an improved Mixture of Gaussians (MoG) model which reduces the learning time and can remove

moving shadows in the foreground regions. Although all the methods introduced can be used in foreground detection, the MoG model is the most widely used method to cope with switching background elements (e.g., waving trees) [7].

4.1.2 Gaussian Mixture Models

In this thesis, the colour value of each pixel is modelled by a mixture of K (K = 5) Gaussian distributions which is used to represent the variations of the background [92]. The computation cost increases when K increases. Let $\mathbf{p}_t = [R_t\ G_t\ B_t]^T$ be the value of a pixel at time t, the probability of that pixel taking this value is:

$$P(\mathbf{p}_t) = \sum_{j=1}^{K} \frac{w_j}{(2\pi)^{\frac{d}{2}}\left|\boldsymbol{\Sigma}_j\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{p}_t-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\mathbf{p}_t-\boldsymbol{\mu}_j)} \tag{4.1}$$

where d is the dimension of the colour value (currently d = 3), w_j is the weight, $\boldsymbol{\mu}_j$ is the temporal mean and $\sum_j$ is the covariance matrix for the j-th distribution. Let σ_j^2 be the trace of $\sum_j$. These K distributions are ordered according to w_j/σ_j^2, which means a distribution occurring frequently with low variation has a high rank. The first B ranked distributions, whose sum of weights is over a threshold T, are thought of as background models.

$$B = arg\min_b \left(\frac{\sum_{j=1}^{b} w_j}{\sum_{j=1}^{K} w_j} > Th_B \right) \tag{4.2}$$

After a new frame $\mathbf{I}_t$ arrives at time t, each pixel $\mathbf{I}_t(r,c)$ is compared with its background models. If it is more than Th_B times the standard deviation away from all the B distributions, it is regarded as a foreground pixel. Th_B can be set empirical. If $Th_B$is too small, some background pixels will be misclassified as foreground pixels. On the other hand, if the threshold is too large, some foreground pixels will be missed in the detection. In this thesis, Th_B was set 2.5.

$$F_t = \left\{(r,c): \left\|\mathbf{I}_t(r,c) - \boldsymbol{\mu}_{t-1,j}(r,c)\right\| > Th_B\sigma_{t-1,j}(r,c)\right\} \quad j \in [1,B] \tag{4.3}$$

If the pixel value is matched with one of the B background distributions, then the matched background distribution k is updated by incorporating the observed pixel value.

$$w_{t,k} = (1-\alpha)w_{t-1,k} + \alpha \tag{4.4}$$

$$\boldsymbol{\mu}_{t,k} = (1-\alpha)\boldsymbol{\mu}_{t-1,k} + \alpha\mathbf{I}_{t,k} \tag{4.5}$$

$$\boldsymbol{\sigma}_{t,k}^2 = (1-\alpha)\boldsymbol{\sigma}_{t-1,k}^2 + \alpha\left\|\mathbf{I}_t - \boldsymbol{\mu}_{t-1,k}\right\|^2 \tag{4.6}$$

where α is the updating rate and $\alpha \in (0,1)$. The weights, means and standard deviations of the other K-1 distributions remain the same. The weight of each distribution is normalized by the sum of the new K weights.

$$w_{t,j} = \frac{w_{t,j}}{\sum_{j=1}^{K} w_j} \tag{4.7}$$

If the pixel value fails to match any of the K background distributions, it will be used to build a new distribution and to replace the distribution which has the least weight in the K distributions. The mean of the new distribution is initialized by the pixel value while the weight and the variance of the new distribution are initialized with small values [92].

4.2 Foreground Regions

After the foreground pixels in each single camera view are detected, these pixels need to be grouped into foreground regions by applying connected component analysis, morphological operations and a size filter.

4.2.1 Connected Component Analysis

Flood fill is a useful method to merge isolated pixels into connected regions. When a seed point is selected, all neighbouring points having the same value with the seed point are used to generate a single contiguous region. As the 8-neighbour method also considers whether the nodes touching at the corners are connected or not, the results of the 8-neighbour method are more connective than ones obtained from the 4-neighbour method.

4.2.2 Morphological Processing

Morphological operations use the algebra of non-linear operators to bring a transformation and variation of the object shape in image processing such as pre-processing, segmentation, and object quantification. It provides a better and faster process than the linear algebraic system of convolution. Given a binary template image, zero-valued pixels are background pixels and the others are foreground pixels. Let the binary image I (p) be the set of all pixel locations in the foreground, a structuring element X_p, which is centred at foreground pixel p, is used to define

arbitrary neighbourhood structures. Although the structuring element is not limited to a normal small solid square and can be any shape or size, the 4-neighborhood is a widely used structuring element.

The dilation operation changes a background pixel to foreground if it has a foreground pixel as a 4-neighbour. The image I dilated by structure element X_p can be denoted as:

$$I \oplus X = \bigcup_{p \in I} X_p \tag{4.8}$$

If a foreground pixel has a background pixel as a 4-neighbour, it can be changed to background by the erosion operation.

$$I \ominus X = \{p | X_p \subseteq I\} \tag{4.9}$$

Figure 4.1 is a schematic diagram showing the results of the dilation and erosion operations.

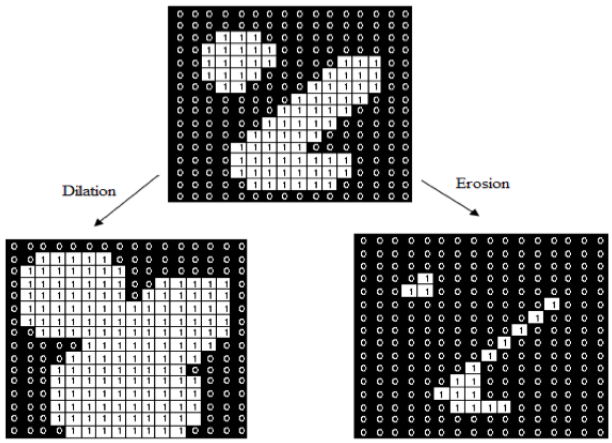


Figure 4.1 Schematic diagram of dilation and erosion.

The combinations of dilation and erosion can further introduce two other morphological operators: opening and closing. If erosion is followed by dilation, the combination is called opening. Opening can remove noise and isolated elements in the binary image, as the foreground structures which are smaller than the structuring element X are removed and the larger foreground structures are retained. The opening operation is denoted as:

$$I \circ X = (I \ominus X) \oplus X \tag{4.10}$$

Closing is defined as dilation followed by erosion. When the structuring element has a suitable size and shape, closing can connect objects that are close to each other, fill up small holes, and smooth the object outline by filling up narrow gulfs. The closing operation is denoted as:

$$I \cdot X = (I \oplus X) \ominus X \tag{4.11}$$

4.2.3 Post-processing

After the connected component analysis and the morphological operations, the foreground pixels are grouped into foreground regions, and the holes in the foreground regions are filled. To improve the performance of the foreground detection, small regions that are caused by noise are filtered out by a size filter.

4.3 Foreground Polygons

Once the foreground regions have been identified in a camera view, each foreground region need to be projected to a reference view according to the homography for a certain plane. Instead of applying a pixelwise homography mapping, the algorithm focuses on the vertices of each foreground polygon. Then, the image-level projection is replaced by the projection of a small number of the vertices of each foreground polygon. This section describes how to represent the foreground regions with vertex points.

4.3.1 Contour Extraction

Each foreground region in foreground image can be represented by the contour of that foreground region. Let F_i^a be the i-th foreground region detected in camera a. The contour of F_i^a is represented by an ordered set of N points $C_i^a = \{p_1, p_2, \dots, p_N\}$ on the contour curve. The algorithm proposed by Suzuki and Abe [93] is used to extract the contour of each foreground region in the camera view. Given a binary image, connect component analysis is used to group binary pixels into two kinds of connected components: the foreground connected components and the background connected components. According to the surround relations, the borders can be divided into two categories: the outer borders and the inner borders. The outer border, in which the foreground component is surrounded by the background component, indicates the contour of the foreground. If a pixel in

a foreground connected component has a background pixel in its 4-neighborhood, it is classified as an outer border pixel which is a border pixel between a foreground component and a background component. The detected pixel coordinates are used to represent the contour which is the border of a connected component of the foreground and a connected component of the background.

4.3.2 Polygon Approximation

To make the representation of the contour point set C_i^a more compact, the original contour is approximated by a polygon; that is, to find a subset of these contour points that can best represent the contour. The Douglas-Peucker (DP) algorithm [82] is used for the polygon approximation. The DP algorithm can be described as follows:

1. It starts with the original contour and picks up two extreme points which are the most distant from each other:

$$m, n = arg \max_{i,j \in [1,N]} dist(p_i, p_j) \tag{4.12}$$

2. These two points are connected with a line that divides the original contour into two segments. For each of these two segments, say segment $C' = \{p_m, p_{m+1}, \dots, p_n\}$, it is searched to find the point farthest from the line just drawn. That point is added to the approximation if its distance to the line is over a predetermined value ε that controls the accuracy of the approximation:

$$q = arg \max_{i \in [m,n]} dist(p_i, \overline{p_m p_n}), dist(p_q, \overline{p_m p_n}) > \varepsilon \tag{4.13}$$

3. Then segment C' is split at point p_q and the process is recursively applied to the two resultant smaller segments until all the contour points are within distance ε to the edges of the polygon.

This algorithm can be applied to either convex or concave contours. Moreover, it produces simplification within a hierarchical structure, in which the top layer represents the dominant shape properties, and the bottom layer describes the fine details. The DP algorithm is illustrated in Figure 4.2.

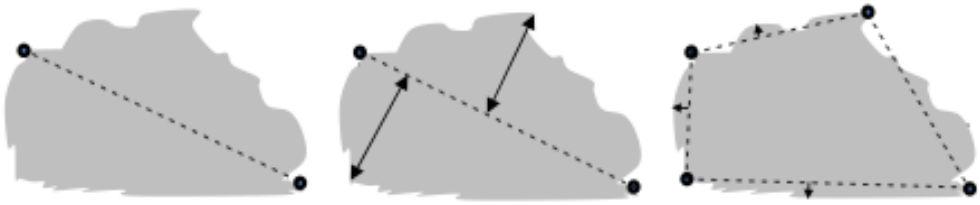


Figure 4.2 The polygon approximation for a foreground region.

Then, the contour of F_i^a is approximated by a set of vertices V_i^a. The most time consuming part of the Douglas-Peucker algorithm is the evaluation of the distances between contour points to line segments. Its worst-case running time is $O(N^2)$ where N is the number of the contour points. An improvement for speeding up the Douglas-Peucker algorithm, making it a $O(N \log N)$ time algorithm in the worst case, can be found in [94].

4.4 Foreground Projection

After the contour of F_i^a is approximated by vertices V_i^a, instead of applying pixelwise homographic transformations to the foreground images, only the vertices of the foreground polygons need to be projected onto the reference image. The foreground regions are then rebuilt by filling the internal area of each polygon with a fixed value.

4.4.1 Polygon Projection

To improve the computational efficiency of the homography projection, the vertices of the foreground polygons in each camera view are projected into a virtual top view according to the homography for a certain plane. The ground-plane homography $H_g^{a,t}$, which was estimated in Chapter 3, is used to project the vertices V_i^a of the i-th foreground polygon from camera view a to the top view t. Let $V_{i,g}^{a,t}$ be the set of projected vertices in the top view t, which can be described as:

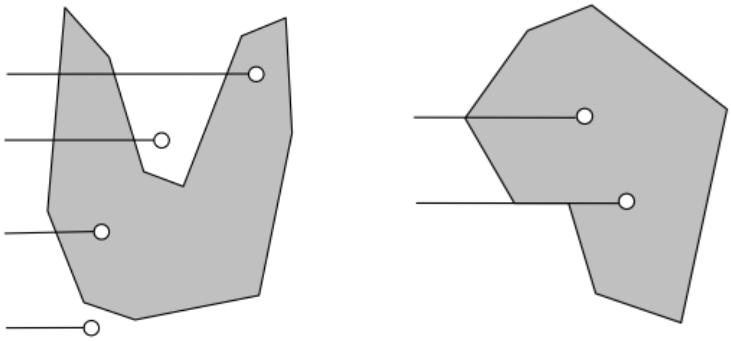
$$V_{i,g}^{a,t} = H_g^{a,t}(V_i^a) \tag{4.14}$$

4.4.2 Reconstruction of the Projected Foreground

Since the vertices in V_i^a are arranged in order, connecting each projected vertex with its neighbour sequentially can generated a new contour in the top view *t*. This new contour approximates the contour of the projected foreground region $F_{i,g}^{a,t}$ which is the projection of F_i^a from camera view *a* to the top view *t* according to the ground-plane homography. Then $F_{i,g}^{a,t}$ is rebuilt by filling the internal area of the projected foreground polygon with a fixed value. Thus, $F_{i,g}^{a,t}$ can be described as:

$$F_{i,g}^{a,t} = H_g^{a,t}(F_i^a) \tag{4.15}$$

In filling the projected polygons, a decision need to be made as to whether a given pixel in the top-view image lies inside, outside, or on the boundary of a polygon. This is the point-in-polygon problem in computational geometry. In this research, the ray-casting algorithm [95] has been used, in which the number of times that a ray (say in a horizontal direction) intersects the edges of the polygon is counted.



(a) (b)

Figure 4.3 The ray-casting algorithm to decide whether a given point is inside a polygon, (a) when the ray crosses the edges, and (b) when the ray crosses a vertex or lie on an edge.

If the point in question is not on the boundary of the polygon, it is outside if the number of intersections is an even number; it is inside if this number is odd. However, a vertex of the polygon may fall on the ray or one side of the polygon may lie entirely on the ray. To avoid duplicate counts of the edge crossing, if the intersection point is a vertex of a polygon side being tested, then the intersection is counted only if the second vertex of the side lies below the ray. Figure 4.3 shows a schematic diagram of the ray-casting algorithm to decide whether a given point is

inside a polygon. The time to test one point against a polygon with L sides or $L + 1$ vertices is $O(L)$. This algorithm can be applied to either convex or concave polygons.

The ray-casting algorithm is described as follows:

Algorithm 1: Ray-Casting

1:	count ← 0
2:	**for** each side in polygon **do**
3:	**for** each horizontal ray **do**
4:	**if** the ray intersects the polygon **then**
5:	count ← count + 1
6:	**end if**
7:	**end for**
8	**end for**
9	**if** *is_odd*(count) **then**
10	**return** inside
11	**else**
12	**return** outside
13	**end if**

When each projected foreground polygon is reconstructed, the projected foreground image is recognized as the bitmap projection of the foreground image F^a from camera view a to the top view t:

$$F_g^{a,t} = \sum_i F_{i,g}^{a,t} = H_g^{a,t} F^a \tag{4.16}$$

The warped foreground region of an object in the top view is observed as the intersection of the ground-plane and the cones swept out by the silhouette of that object. Figure 4.4 illustrates the homography projection based on a single camera view according to the ground plane g. If the camera is considered as a light source, the grey region which is the projected foreground region is like the shadow of the blue object on the plane g.

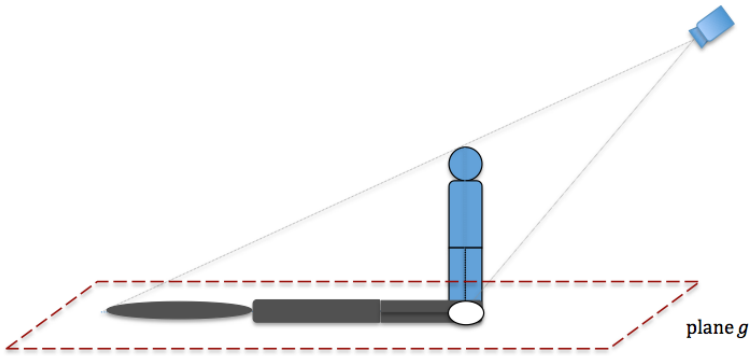


Figure 4.4 Schematic diagram of the homography projection according to the ground plane.

4.5 Foreground Fusion

4.5.1 Fusion with Multiple Views

The foreground projection is extended from a single camera to multiple cameras. Let lower-case c be the index of the camera in C cameras. The fusion of the projected foregrounds in the top view is carried out by overlaying the projected foreground images from the multiple camera views:

$$F_g^t = \sum_c F_g^{c,t} \tag{4.17}$$

The projected foreground regions from different camera views may intersect in the top view. The intersection regions correspond to enhanced regions in the overlaid foreground projection image F_g^t and indicate the locations of moving objects on the ground plane. The intersection regions are denoted by:

$$P_g^t = \bigcap_c F_g^{c,t} \tag{4.18}$$

When the foreground regions for the same object are warped from multiple views to the top view, they will intersect at a location where the object touches the ground. Figure 4.5 shows a schematic diagram of the overlaid foreground projections and the intersection region. Although the blue region which is the intersection of the two projected foregrounds illustrates the location of the object on the plane g, the size and the shape of that region is not exactly the cross section of the object.

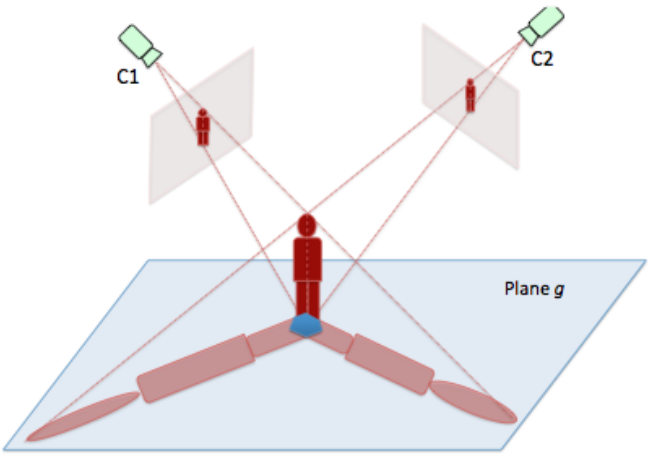


Figure 4.5 Schematic diagram of the overlaid foreground projections and the intersection region.

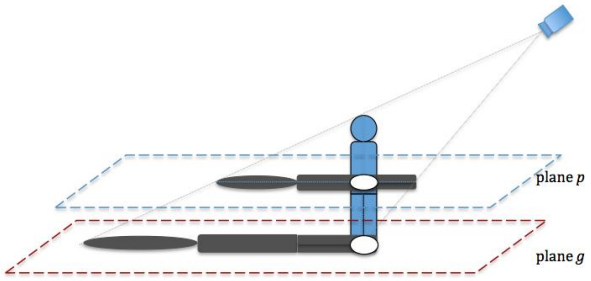
4.5.2 Fusion with Multi-Plane Homographies

For the ground plane, the intersection patches are at locations where the moving objects touch the ground. To improve the robustness of the moving object detection algorithm, the foreground projection and the fusion in the top view can be extended from the ground plane to a set of parallel planes. Figure 4.6 shows a schematic diagram of the homography projection according to the ground plane and a plane parallel to the ground plane. Plane p is an imaginary plane parallel to the ground plane g at the height of a person's waist. In Figure 4.6 (a), the projected foreground region in the plane p moves in the camera direction when the height of the plane p increases. In Figure 4.6 (b), the projected foregrounds from two camera views intersect at the waist of the person on plane p. If multi-camera and multi-plane are used, the result of the logically ANDed intersection patches at different heights is similar to the projection of the person's three-dimensional volume on the ground plane.

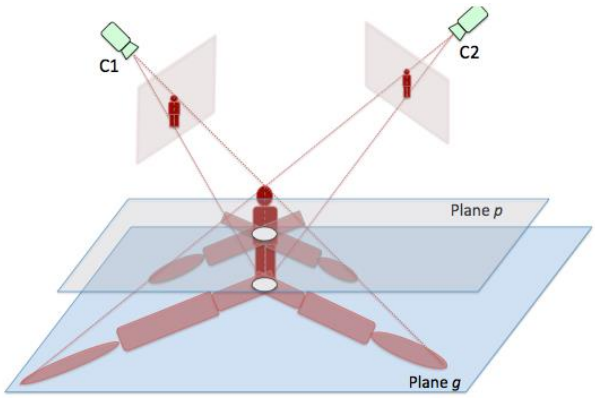
If the lower-case p denotes a plane index, the fusion of the projected foregrounds in the top view and the intersection of the overlaid foreground projection are:

$$F^t = \sum_p F_p^t \tag{4.19}$$

$$P^t = \bigcap_p P_p^t \tag{4.20}$$



(a)



(b)

Figure 4.6 Schematic diagram of the homography projection according to the ground plane and a plane parallel to the ground plane, (a) based on a single camera view, (b) based on two camera views.

An alternative way to identify the intersection regions is thresholding the overlaid multi-layer foreground projection image F^t. If the value of a pixel in the overlaid multi-layer foreground projection image is larger than a threshold Th_i, that pixel is recognized as being in the intersection regions.

$$P^t = \{(r, c): O^t(r, c) > Th_i\} \tag{4.21}$$

4.6 Experimental Results

The real-time moving algorithm using multiple cameras has been tested on a number of video sequences. To show the performance evaluation of this algorithm, two datasets were used. The first dataset was captured in the author's campus, where the cameras were placed close to pedestrians, and the second dataset is a standard dataset in which the cameras were located far away from walking people and vehicles.

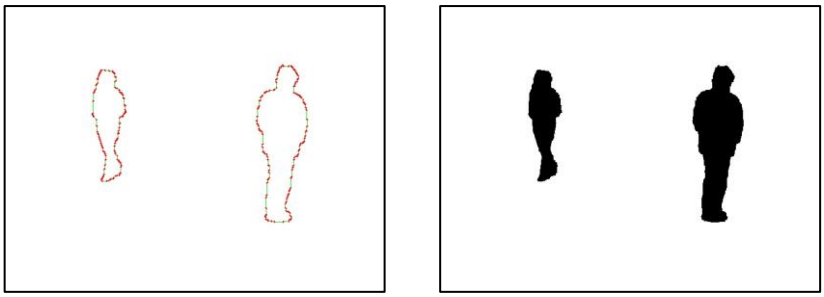
4.6.1 Experimental Results of the Campus Dataset

In the first testing sequences, two cameras were placed with small viewing angles and with significant overlapping field of views. People walked around within a $4.0\,m \times 2.4\,m$ region to ensure some degree of occlusion. There are 2790 frames captured in each camera view with a resolution of 640×480 pixels and a frame rate 15 fps. 2155 frames were used that contained two or three pedestrians in the tests (the first 660 fames contained no pedestrians or only one pedestrian). The test results of the polygon projection were compared with the bitmap projection results over 142 frames, each of which was sampled every 15 frames in the 2155 frames of the testing video. In this experiment, a virtual top view image was selected as the reference image with a resolution of 840×1000 pixels. In the testing of the processing speeds, the polygon projection was run on a single PC with an Intel Core i7 CPU running at 2.9 GHz.

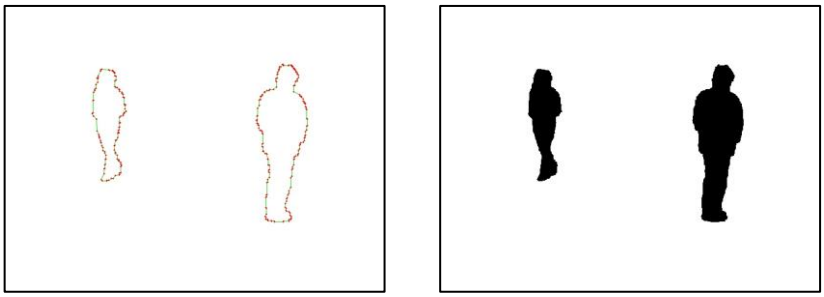
4.6.1.1 Foreground Polygons

Figure 4.7 shows the results of the foreground detection and foreground polygons with different ε at frame 1020 in camera view a. Figure 4.7 (a) and (b) are the original image and the foreground image. The times of morphologic operations (dilution and erosion) is 2, and the window size for post-processing is 200 pixels. Figure 4.7 (c), (e), (g) and (i) show the contour extraction results and the polygon approximation results with different distance ε, in which the green lines are the polygon edges and the red dots are the polygon vertices. The results of the foreground reconstruction with the ray-casting algorithm are shown in the Figure 4.7 (d), (f), (h) and (j).

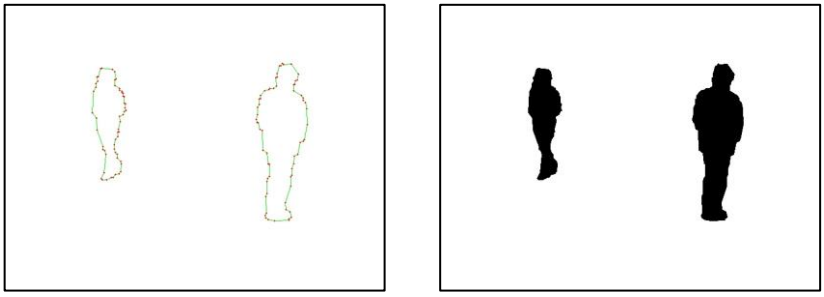
(a) (b)



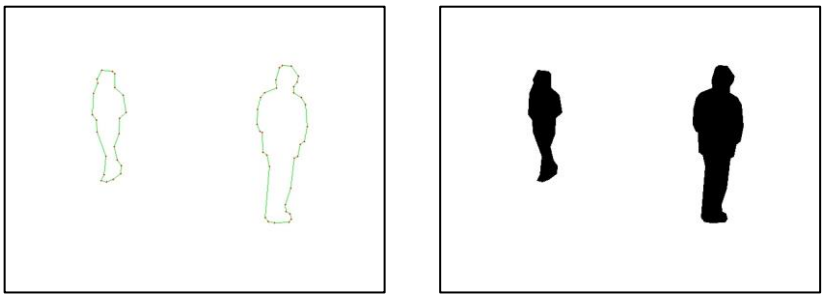
(c) (d)



(e) (f)



(g) (h)



(i) (j)

Figure 4.7 The foreground polygon approximation at frame 1020 in camera view *a*, (a) the original image, (b) the foreground image, (c) and (d) the foreground polygon image and the reconstructed foreground image ($\varepsilon = 0$), (e) and (f) the foreground polygon image and the reconstructed foreground image ($\varepsilon = 0.5$ pixel), (g) and (h) the foreground polygon image and the reconstructed foreground image ($\varepsilon = 1$ pixel), (i) and (j) the foreground polygon image and the reconstructed foreground image ($\varepsilon = 2$ pixels).

A comparison of the processing speeds for the polygon approximations using different ε, contour (no approximation) and bounding box (very rough approximation) are shown in Table 4.1. The comparison is based on the 142 sampled frames in the two camera views. The contour based method needed, on average, 217.84 vertices to represent each region and the bounding box based method needed 4 vertices. For the polygon approximation based method, when the distance ε is increased from 1 to 10 pixels, the vertices of the foreground polygons decrease dramatically while the cost of processing speed decreases slightly.

Table 4.1 The processing speeds for the contour, polygon approximations (with different distance ε) and the bounding box method.

	Contour	Polygon approximations Distance ε (pixel)				Bounding box
		1.0	2.0	5.0	10.0	
Total Number of Foreground Regions	667					
Total Number of Vertices	145297	41846	21601	10751	5919	2668
Average Number of Vertices per Region	217.84	62.73	32.39	16.12	8.87	4
Total Time(s)	1.07	0.69	0.61	0.58	0.55	0.35

The accuracy of the polygon approximation has been compared with different distances ε. To evaluate the performance, it also has been compared with the contour based method and bounding box based method. The comparison results of the accuracy for the polygon approximations are shown in Table 4.2. Suppose the original foreground image is $F_{temp,k}$ at frame k and the reconstructed foreground image from the polygon approximation is $F_{poly,k}$. The false negatives (missed detections) are the pixels that are 1 (representing foreground) in $F_{temp,k}$ but 0 (representing background) in $F_{poly,k}$. The false positives (false alarms) are the pixels that are 0 in $F_{temp,k}$ but 1 in $F_{poly,k}$. The false negative rate R_{FN} and the false positive rate R_{FP} are measured over all the frames as follows:

$$R_{FN} = \sum_{k} \#\left(F_{temp,k} \cap F^{c}_{poly,k}\right) / \sum_{k} \#\left(F_{temp,k}\right) \tag{4.22}$$

$$R_{FP} = \sum_{k} \#\left(F_{temp,k}^{c} \cap F_{poly,k}\right) / \sum_{k} \#\left(F_{temp,k}\right) \tag{4.23}$$

where #() is the function to count nonzero pixels in a set and the superscript c denotes the complement of a set.

Table 4.2 The accuracy of the contour, polygon approximations (with different distance ε) and the bounding box method.

	Contour	Polygon approximations Distance ε (pixel)				Bounding box
		1.0	2.0	5.0	10.0	
False Negative Rate R_{FN} (%)	0.04	0.53	1.33	4.03	7.28	0.03
False Positive Rate R_{FP} (%)	0.55	0.67	1.27	3.17	7.36	80.55

From Table 4.2, it can be concluded that the polygon approximation can accurately represent the foreground regions. The accuracy can be improved by using a smaller distance ε but at the cost of a slightly increased processing speed (see Table 4.1). The bounding-box based method leads to a very high false-positive rate (80.55%).

4.6.1.2 Polygon Projections

When each foreground region is approximated by polygon, instead of applying homographic transformations to the foreground images, it only needs to project the vertices of the foreground polygons to the reference image. Figure 4.9 shows the foreground projections using the bitmap projection method.

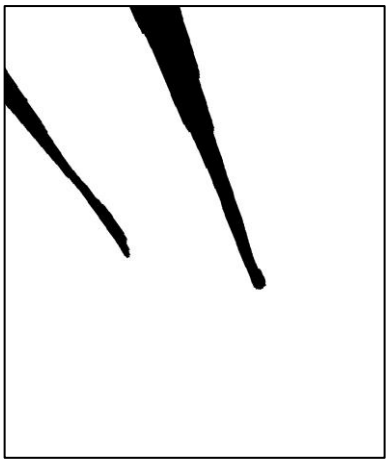
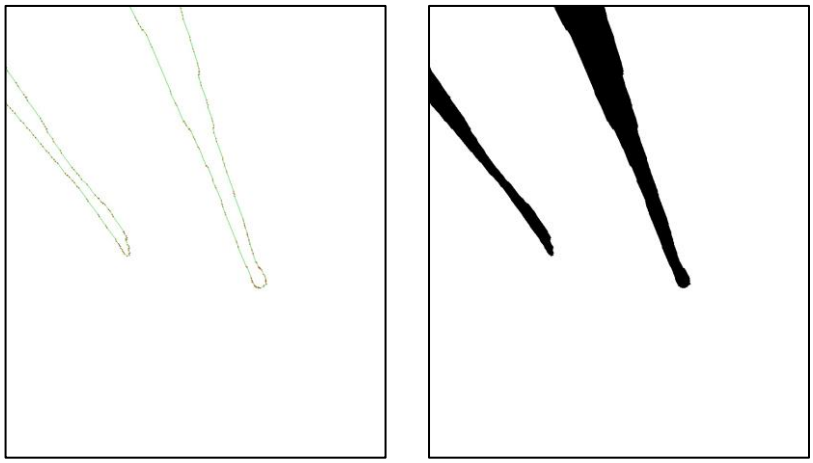


Figure 4.8 Foreground projection using the bitmap method at frame 1020 in camera view *a*.

Figure 4.9 shows the results of the foreground polygon projections at frame 1020 in camera view *a*. Figure 4.9 (a), (c), (e) and (g) show the polygon projection results in the top view, in which the vertices are projected from Figure 4.7 (b), (d), (f) and (h) respectively. The green lines are the projected polygon edges and the red dots are the projected vertices. The reconstruction results of the projected foreground regions by using the ray-casting algorithm are shown in Figure 4.9 (b), (d), (f) and (h) respectively.



(a) (b)

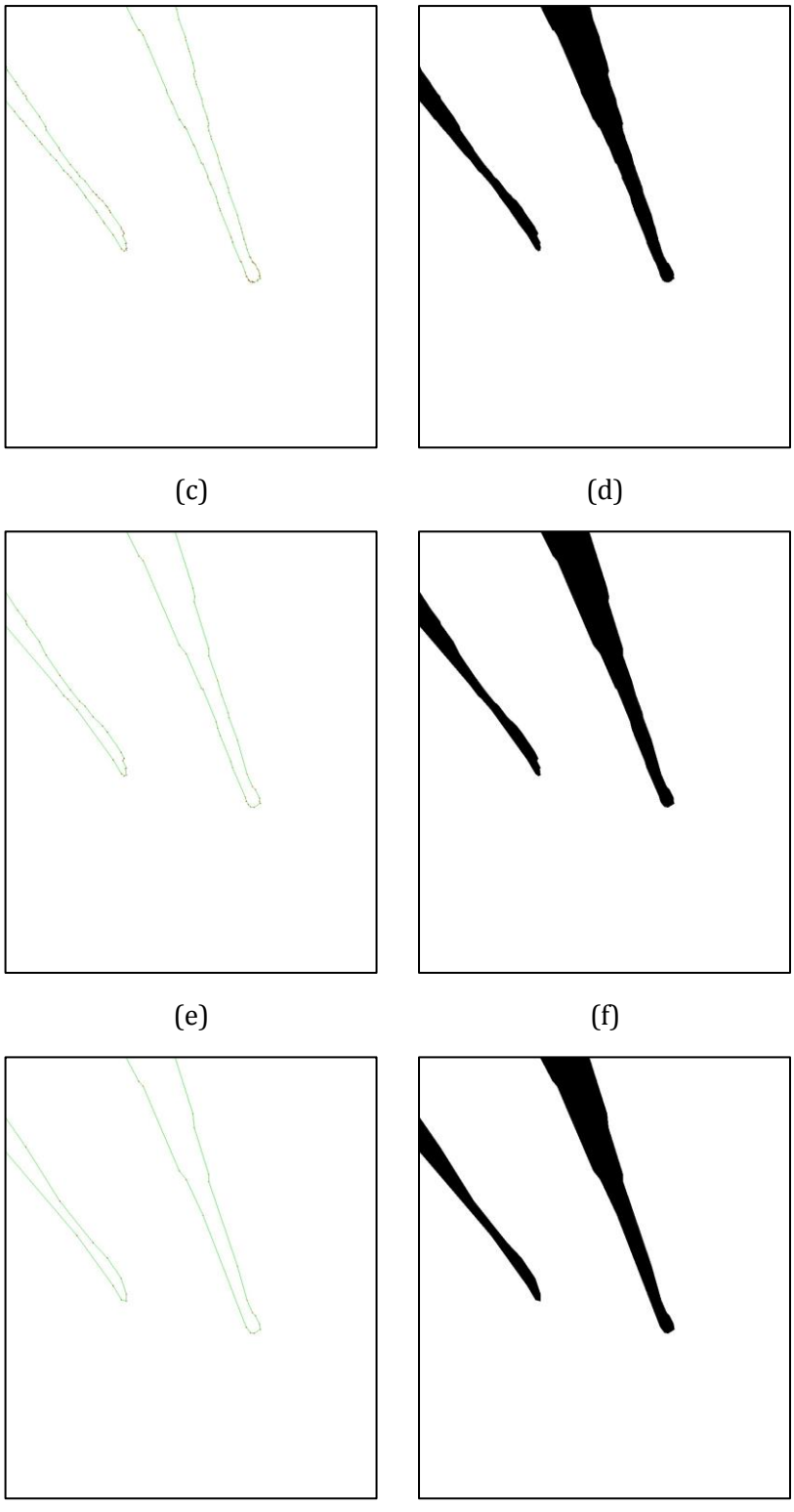


Figure 4.9 Results of the foreground polygon projection at frame 1020 in camera view *a*. (a)(b) the polygon projection and the reconstructed foreground regions ($\varepsilon = 0$), (c)(d) the polygon projection and the reconstructed foreground regions ($\varepsilon = 0.5$ pixel), (e)(f) the polygon projection and the reconstructed foreground regions ($\varepsilon = 1$ pixel), and (g)(h) the polygon projection and the reconstructed foreground regions ($\varepsilon = 2$ pixels).

The projected foregrounds between the polygon projection and the bitmap projection are then compared. Suppose the projected binarised foreground region map in the top view using the bitmap projection is the template $F_{temp,k}$ at frame k and that uses the polygon projection is $F_{poly,k}$. The results of the false negative rate R_{FN} and the false positive rate R_{FP} are measured over all the frames with different distance ε, as shown in Table 4.3. Compared with the contour based method and polygon approximation method, the bounding box method has the lowest false negative rate but has a very high false positive rate. Therefore, bounding boxes are not accurate enough to be used in this project. For the polygon approximation method, when ε is less than 2, its accuracy is similar to that of the contour based method. The accuracy decreases when ε increases. Therefore, the polygon approximation method is very flexible and can control the amount of data to transmit (the number of projected vertices) according to ε.

Table 4.3 The projection accuracy of the contour, polygon projection (with different ε) and the bounding box method.

	Contour	Polygon approximations Distance ε (pixel)				Bounding box
		1.0	2.0	5.0	10.0	
False Negative Rate R_{FN} (%)	3.09	3.02	3.26	4.96	6.16	0.01
False Positive Rate R_{FP} (%)	2.98	3.07	3.26	4.51	9.69	78.72

4.6.1.3 Projected Foreground Fusion based on a Single Plane

When the foregrounds of the individual cameras are projected into the top view according to the homography for the ground plane, the projected foregrounds from all the camera views are fused in the top view according to Equation (4.17). Then, the ground plane in the homography projection is extended to planes parallel to the ground plane and at some heights.

Figure 4.10 shows the results of fusion of the projected foregrounds from the two camera views according to the homography for one such plane at frame 1275. The distance ε for the polygon approximation was set to one pixel. Figure 4.10 (a) and (c) are the original images in the two camera views. Figure 4.10 (b) and (d) are

the results of the polygon approximation. Figure 4.10 (e) is the overlaid foreground projections in the top view using the ground-plane homography. Figure 4.10 (f)-(j) are those using the homographies for the planes at heights of 0.5 m, 0.75 m, 1.0 m, 1.25 m, and 1.5 m respectively.

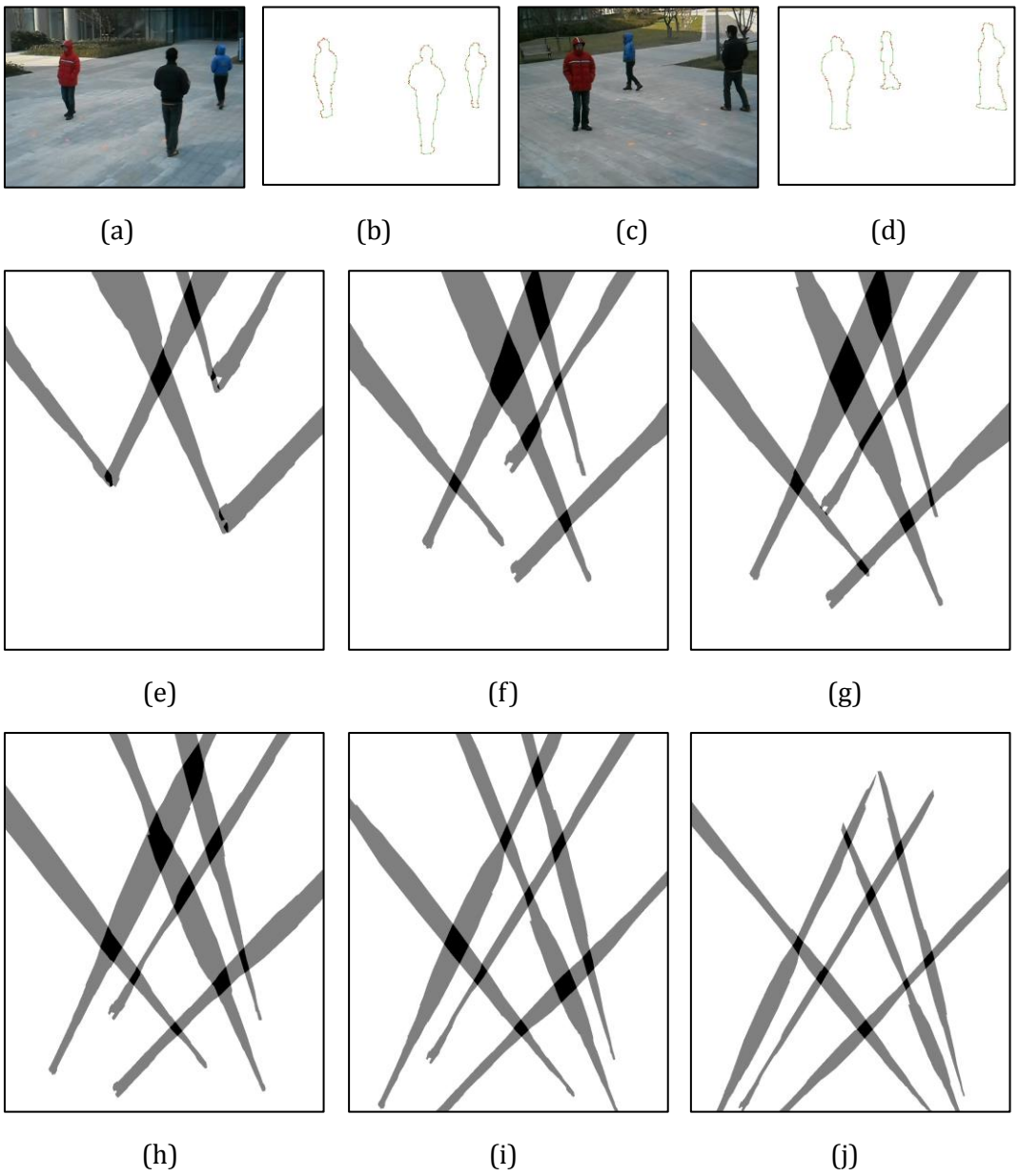


Figure 4.10 Fusion of the foreground projections according to the homographies for a set of parallel planes at different heights, (a) and (c) the original images in the two camera views; (b) and (d) the foreground polygons in the two camera views; (e) fusion of the projected foregrounds in the top view using the ground-plane homography; (f)-(j) fusion of the projected foregrounds in the top view using homographies for a set of parallel planes.

To evaluate the processing speed of the polygon projection method based on a single plane, the results were compared with those of the bitmap projection method.

In tests of the processing speeds over 142 frames in both the camera views, the plane used in the homography mapping was the ground plane and the distance ε was set to 1.

Table 4.4 Execution times for running the bitmap projection and the polygon projection, the total time for the foreground projections are in bold font.

	Bitmap Method	Polygon Method	
Foreground Detection (s)	GMM and Background Subtraction 31.22		
Foreground Projection (s)	**71.56**	Polygon Approximation	0.69
		Vertex Projection	0.03
		Polygon Filling	0.31
		Total Projection	**1.03**
Foreground Addition (s)	2.08		

The whole implementations include (1) the foreground detection in the two camera views, (2) the projection of foreground information from the two camera views and (3) fusion of the projected foregrounds in the top view. Then the time spent for processing each frame from both the camera views were obtained by taking the average. Usually in a video surveillance network, part (1) is executed by individual clients, and the other two parts are executed by a central server. Part (1) and Part (3) are not related to the improvement in the new algorithm. Part (2) was implemented using either the bitmap projection method or the polygon projection method. The polygon projection method is further subdivided into three stages: polygon approximation, vertex projection, and polygon filling in the top view. Since the implementations in [96], great efforts have been made to optimize the code and accelerate the bitmap projection method. The bitmap projection takes 71.559 s and the polygon projection takes 1.03 s to process 142 frames from the two cameras. Although 97% time of the polygon projection was used in the polygon approximation and polygon filling, the polygon projection is 69.47 times faster than the bitmap projection.

4.6.1.4 Projected Foreground Fusion based on Multiple Planes

The computational burden in fusing foreground images lies in the homography mapping for multiple cameras and multiple parallel planes. The more cameras and more planes, the more accurate and more robust the object localization is. As an example, four camera views and 10 parallel planes were used in [4]. Using Equation (4.19), the projected foregrounds from the individual camera views using the homographies for multiple parallel planes are overlaid in the top view. Figure 4.11 shows the results of the overlaid foreground projections at frames 810, 1270, and 2385. The top row is the results of the polygon projection and the bottom row is those of the bitmap projection. In Figure 4.11, two camera views and six parallel planes were used in the homography mapping. The bitmap projection takes 3.02 s, and the polygon projection takes 0.043 s to process one frame. Therefore, it provides a great boost in computational speeds.

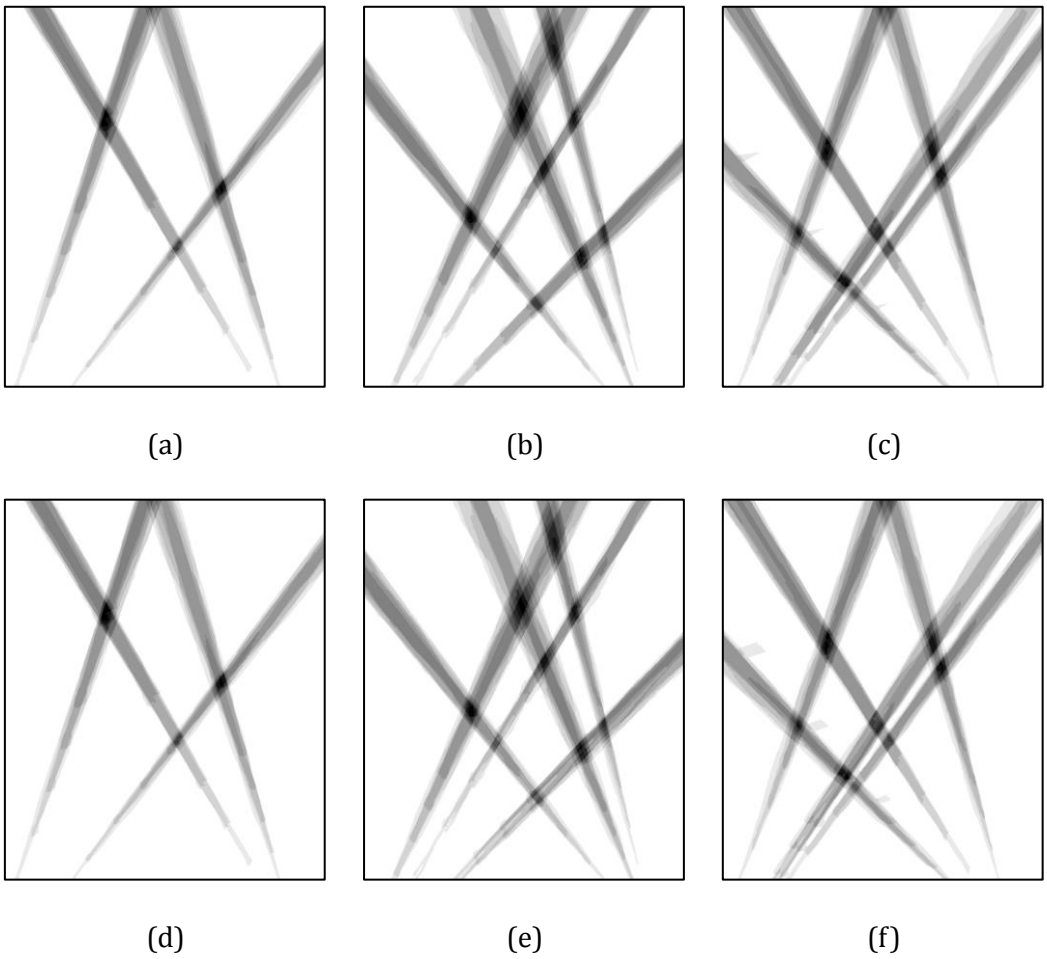


Figure 4.11 Overlaid foreground projections from two camera views and with multi-plane homographies, (a)-(c) the results using the polygon projection at frames 810, 1275, and 2385, and (d)-(f) the results using the bitmap projection at frames 810, 1275, and 2385.

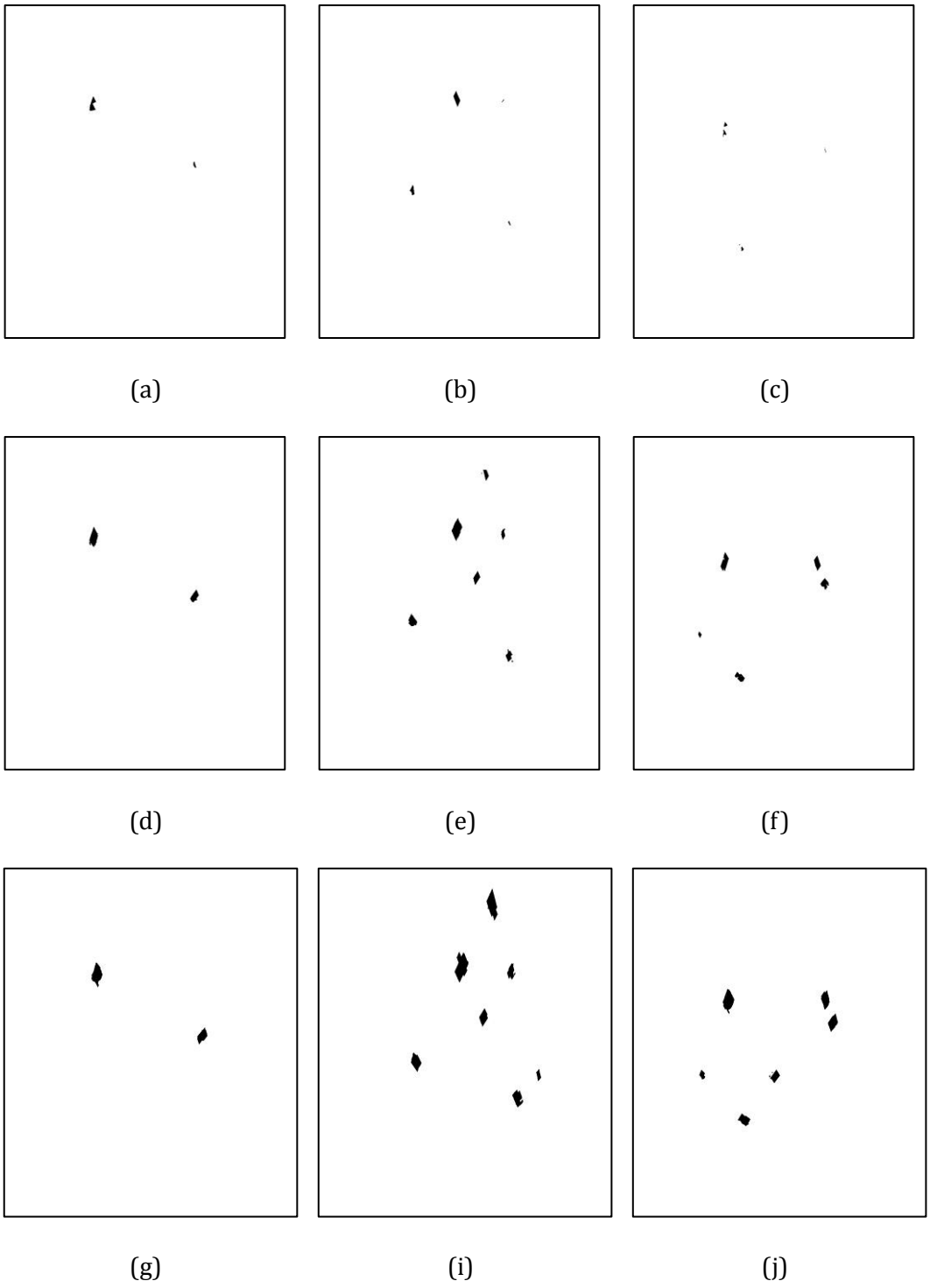


Figure 4.12 Intersection regions identified with differnet thresholds Th_i at frames 810, 1270, and 2385, (a)-(c) $Th_i = 255$, (d)-(f) $Th_i = 230$, and (g)-(i) $Th_i = 210$.

Figure 4.12 shows the foreground intersection regions of the multi-plane and multi-camera homography mapping, which are identified with different thresholds on Figure 4.11 (a)-(c). The number and the size of the intersection regions vary with the threshold Th_i. When the threshold is high, the sizes of the intersection regions are small and the number of the intersection regions is low. A Lower threshold

leads to a larger size of each intersection region and additional intersection regions which are phantoms (this will be discussed in the next chapter).

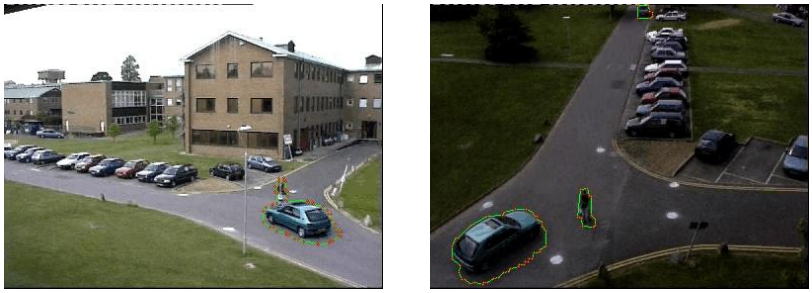
The threshold can be decided empirically. To provide a satisfactory performance with respect to most applications, the number of parallel planes can be used to set the threshold. Let D be the number of planes used in the homography fusion. Since the image of the overlaid foreground projections from two camera views and with multi-plane homographies is a grey level image, the value of each pixel in the overlaid image is from 0 to 255. The threshold Th_i can be denoted as:

$$Th_i = \frac{D-1}{D} \times 255 \tag{4.24}$$

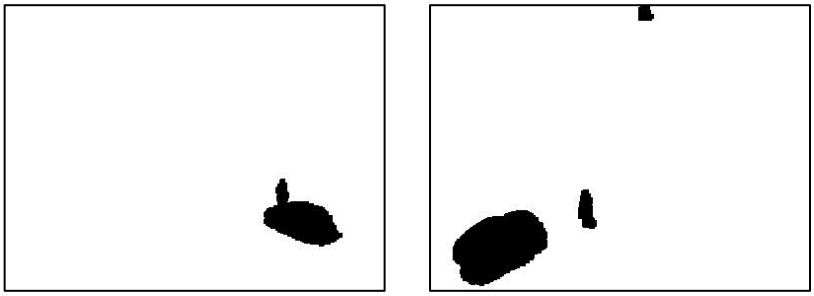
4.6.2 Experimental Results of the PETS'2001 Dataset

To further demonstrate the performance evaluation results, this polygon projection algorithm has also been tested on the PETS'2001 dataset which contains significant dynamic occlusion and scene activity. The PETS'2001 dataset contains standard image sequences for testing tracking and surveillance algorithms. In these experiments, the top view image was selected as the reference image. The top view image for the PETS'2001 is of 500×500pixels and the original sequences were spatially subsampled to half-PAL (384×288 pixels). Figure 4.13 and Figure 4.14 show the process of moving object detection using multiple plane homography fusions. The result of foreground fusion from two camera views to the top view with multi-plane homographies can minimize occlusion.

Then, the projected foregrounds between the polygon projection and the bitmap projection were compared for accuracy and processing speeds. Suppose the projected binarized foreground region map in the top view using the bitmap projection is the template $F_{temp,k}$ at frame k and that using the polygon projection is $F_{poly,k}$. The results of the false negative rate R_{FN} and the false positive rate R_{FP}, which are measured over all the frames using equations (4.22) and (4.23). Table 4.5 is the comparison results of the accuracy for the polygon projection with different predetermined distance ε. From Table 4.5, it can be concluded that the polygon approximation is accurate for the half-sized video sequences. The accuracy can be improved by using a smaller distance ε or full-sized video sequences. Comparing Table 4.5 with Table 4.3, it can be found that the sequences which have larger foreground objects have a lower false negative rate and a lower false positive rate.

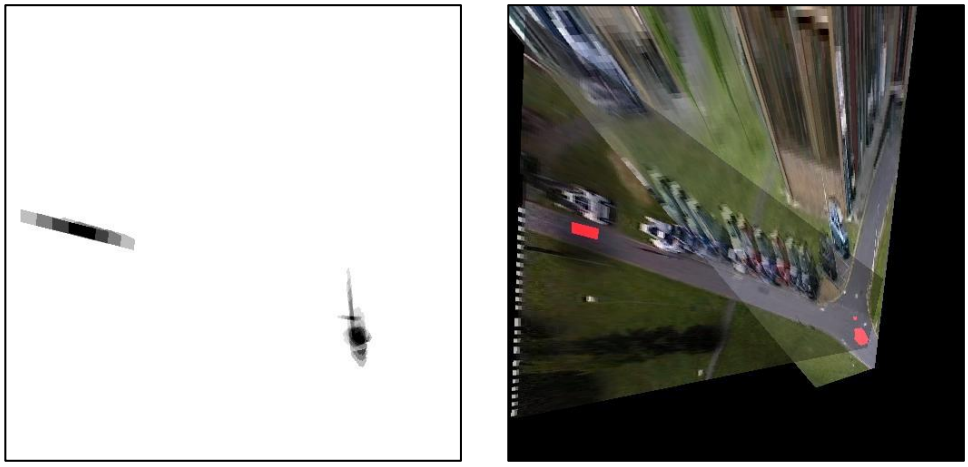


(a) (b)



(c) (d)

Figure 4.13 Foreground detections and the foreground polygons in two camera views, (a) and (b) foreground polygons, (c) and (d) foreground regions.



(a) (b)

Figure 4.14 Moving object detection by the foreground fusion for multi-planes homographies, (a) overlaid foreground projections, and (b) detection results from (b).

Table 4.5 The accuracy of the polygon projection method.

Polygon Distance ε (pixel)	False Negative Rate $R_{FN}(\%)$	False Positive Rate $R_{FP}(\%)$
0.5	2.9	4.8
1.0	3.4	5.0
2.0	4.4	5.4

Table 4.6 Execution times for running different algorithms on one camera, the total time for the foreground projections and fusions are in bold font.

	Bitmap Method	Polygon Method
Foreground Detection (ms)	65	
Foreground Projection and Fusion (ms)	**108.5**	Polygon Approximation 4.5
		Vertex Projection 0.1
		Polygon Filling 2.3
		Foreground Addition 1.6
		Subtotal **8.5**

In testing the processing speeds, the polygon projection and the bitmap projection were run on a single PC with an Intel Core 2 Duo CPU running at 2.66 GHz. To evaluate the performance of the multiple camera homography mapping of the polygon projection method for a single plane, the speeds of the moving object detection with the bitmap projection method were compared. Then the time spent for processing each frame from one camera view was obtained by taking the average. In the testing of the processing speeds, the plane used in the homography mapping is the ground plane and the predetermined distance ε is chose as 1. Table 4.6 shows execution times for running different algorithms on one camera. Both the implementations include (1) the foreground detection in two camera views and (2) the projection and fusion of foreground information from the two camera views. The Gaussian mixture model takes 65.0 *ms* to process one frame for one camera. Part (2) was implemented using either the bitmap projection method or the

polygon projection method. The polygon projection method is further subdivided into four stages: polygon approximation, vertex projection, polygon filling, and foreground addition to the top-view image. The bitmap projection takes 108.5 *ms*, and the polygon projection takes 8.5 *ms* to process one frame for one camera. Therefore, the latter is 12.8 times faster than the former.

4.7 Summary

In this chapter, a method has been presented for real-time moving object detection with multiple cameras. The proposed method is based on multi-plane homography mapping of the foregrounds from multiple cameras. Instead of applying a bitmap homography mapping, each foreground region is approximated by a polygon and only the vertices of the polygon are projected to the reference view through homography mapping. The experimental results have shown that the proposed algorithm can run in real time and produce competitive results comparing with the method using foreground bitmaps those by mapping. The execution time for the polygon approximation method changes linearly with the number of foreground regions. The polygon approximation method has been compared with the contour based method (no approximation) and the bounding box based method which has a very rough approximation. The polygon approximation method has an accuracy similar to that of the contour based method when the distance ε is low. The accuracy reduces when ε increases. The amount of data to transmit (the number of projected vertices) can be controlled by ε, which means the polygon approximation method is more flexible than the contour based method.

5 PHANTOM REMOVAL WITH GEOMETRICAL INFOMATION

The objective of the research described in this chapter is to identify the false-positive detections, which occur due to the foreground intersections of non-corresponding objects, in the top view using geometrical information. A height matching algorithm is proposed to match the intersection regions in the top view with the foreground regions from the individual camera views to identify whether an intersection region is due to the same object. Since the matching is carried out in each camera view, the intersection regions in the top view are warped back to the individual camera views to generate warped back patches. A correct correspondence is identified if the foot location of a foreground region is matched to the position of a warped back patch in an individual camera view. Based on the analysis of unmatched patches and the matched patch, the patches for each foreground region in a camera view are divided into three classes: the object patch, upper patches and lower patches. Then, the classes from different cameras are grouped together to identify object regions, phantoms, covered regions and occluded regions in the top view. Since foreground detection in a single camera is not the main focus of this thesis, the foreground segmentation is assumed to be in good quality. When the foreground segmentation error rate is high, the classes from different cameras could be grouped with a different classification approach.

5.1 Introduction to Phantoms

When the foreground images in the individual camera views are projected into the top view according to the homography for the ground plane or a plane parallel to the ground plane and at some height, the foreground regions from the different camera views may intersect in the top view, in which the intersections indicate the regions which may contain objects. If the intersecting foreground regions from the different camera views correspond to the same object, the intersection region reports the location where the object touches the plane used in the homography projection. If the intersection regions are caused by non-corresponding foreground regions from different camera views, they are false positive detections or phantoms.

This is an important problem in multi-camera moving object detection using foreground homography mapping.

Figure 5.1 is a schematic diagram which illustrates how non-corresponding foreground regions intersect and give rise to a false-positive detection in ground plane based homography mapping. The warped foreground region of an object in the top view is observed as the intersection of the ground plane and the cones swept out by the silhouette of that object. When the foreground regions for the same object are warped from multiple views to the top view, they will intersect at the location where the object touches the ground. However, if the warped foreground regions from different objects intersect in the top view, the intersection region will lead to a phantom detection. In Figure 5.1, the foreground regions of two objects are projected from two camera views into the top view. The foreground projections intersect in three regions on the ground plane. The white intersection regions are the locations of the two objects, whilst the black region may be a phantom.

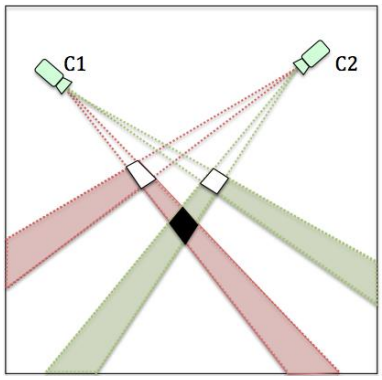
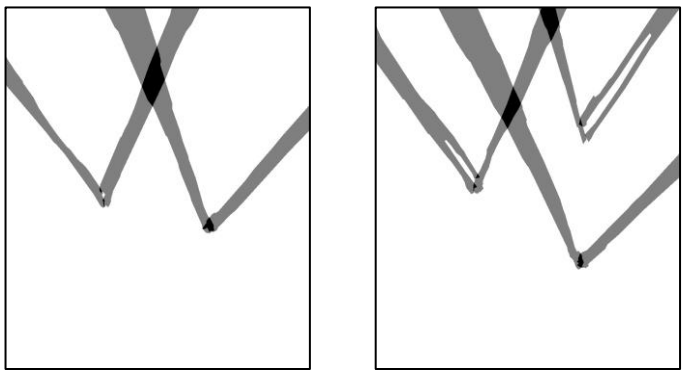


Figure 5.1 A schematic diagram of phantom occurrence using ground-plane homography.

In the previous chapter, Figure 4.10 (e) showed the overlaid foreground projections from the two camera views to the top view according to the homography for the ground plane. Although the ground plane is the most commonly used plane in homography mapping, the foreground projections of the same object, each from one of multiple camera views, may have missed intersections in the reference view. This may happen in at least three scenarios. Firstly, pedestrians' feet are quite small objects and are frequently missed in detection, when a pedestrian is striding and hence has their two legs separated. Furthermore, their feet are not necessarily touching the ground while they are walking. Finally, homography estimation errors are another reason for missed intersections. These

are illustrated in Figure 5.2. Figure 5.2 (a) shows an example of missed intersections due to inaccurate foreground detection when the homography mapping based on the ground plane is applied. Figure 5.2 (b) is another example for missed intersections when one foot of a pedestrian is not touching the ground.



(a) (b)

Figure 5.2 Examples of missed intersections by using ground-plane homography mapping.

When the foreground projections from individual camera views to the top view are based on the homography for a plane off the ground, the intersections of the projected foreground regions are more robust. This was clearly demonstrated in Figure 4.10. Each intersection region in Figure 4.10 (f)-(j) is bigger than the corresponding intersection region in Figure 4.10 (e). However, utilizing homography mapping for a plane higher than the ground plane can cause additional phantoms. The reason for this is that the projected foreground regions are moving to the camera. A schematic diagram of the foreground projection according to the homographies for the ground plane and a plane parallel to and off the ground plane is shown in Figure 4.6 (a). Compared with the foreground projection on plane *g*, the projected foreground region on plane *p* moves towards the camera. When such projected foreground regions on the plane off the ground intersect those from other camera views in the top view, additional phantoms may be generated. A schematic diagram of the homography mapping according to plane *p* is shown in Figure 5.3, in which the grey region is an additional phantom.

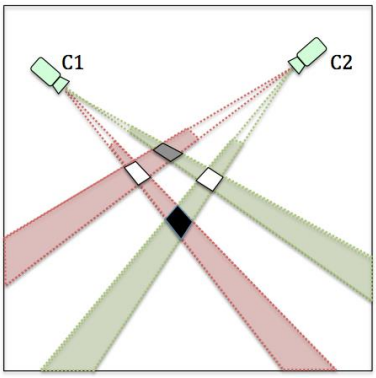


Figure 5.3 A schematic diagram of the homography mapping according to plane *p*.

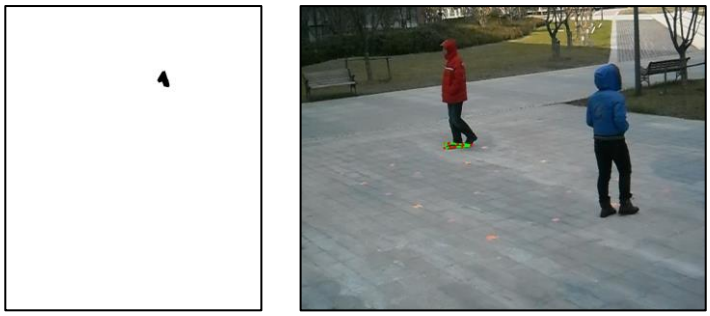
A related work proposed by Peng et al. [58] is an extension of Berclaz, Fleuret and Fuas' work [24, 25]. After the top view occupancy map according to the ground plane homography is generated, the geometrical information is used to build a single Bayesian network model to learn an occlusion relationship in each camera view. Then the results from each of the multiple camera views are integrated by using a multi-view Bayesian network to identify any phantoms. Therefore, this method is very slow and typically needs 3 s to process 1 frame. The height-matching algorithm is much faster.

5.2 Region Based Foreground Fusion

Given the foreground region F_i^a from camera view a and F_j^b from camera view b, $F_{i,p}^{a,t}$ and $F_{j,p}^{b,t}$ are the projected foreground regions from the two camera views to the top view according to the homographies $H_p^{a,t}$ and $H_p^{b,t}$ for the waist plane. Then the foreground projections are overlaid in the top view. If the two projected foreground regions from each of the two camera views intersect in the top view, these two projected foreground regions in the top view and their original foreground regions in each camera view are defined as a pair of projected foreground regions and a foreground region pair respectively. The intersection region of the projected foreground regions $F_{i,p}^{a,t}$ and $F_{j,p}^{b,t}$ is denoted as:

$$P_{i,j,p}^t = F_{i,p}^{a,t} \cap F_{j,p}^{b,t} = \left(H_p^{a,t}(F_i^a)\right) \cap \left(H_p^{b,t}(F_j^b)\right) \tag{5.1}$$

If the intersection region $P_{i,j,p}^t$ is formed by an object, it indicates the location of where the object is intersected by plane p. When plane p is at different heights and parallel to the ground plane, the intersection region $P_{i,j,p}^t$ varies in its size and shape, which approximates the widths of the corresponding body parts at different heights.



(a) (b)

(c) (d)

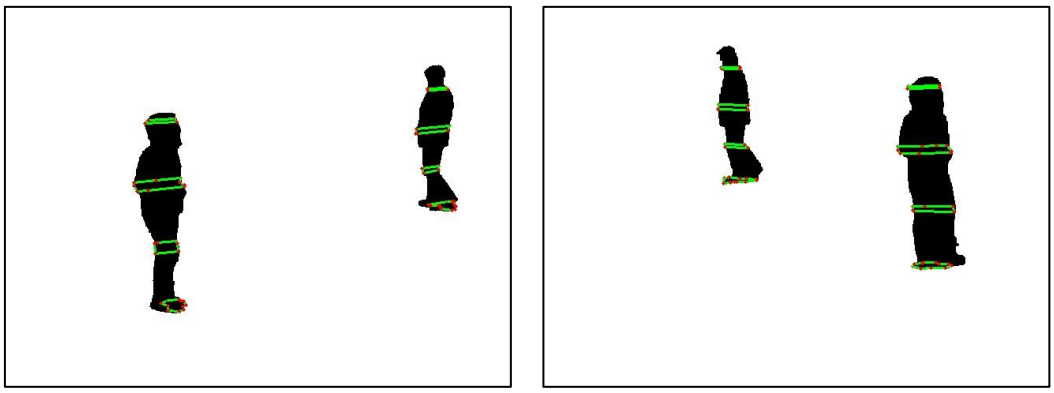
(e) (f)

(g) (h)

Figure 5.4 An example of the projected foreground intersections due to the same object by using the homographies for a set of planes at different heights. (a) and (b) are the intersection region in the top view and the warped back region in camera view *b* for the ground plane, (c) and (d) for the plane at a height of 0.5 m, (e) and (f) for the plane at a height of 1.0 m, (g) and (h) for the plane at a height of 1.5 m.

Figure 5.4 shows an example of the projected foreground intersections generated by the same pedestrian according to the homographies for planes at different heights. There are two pedestrians in each of two camera views. When the foreground polygons of the same pedestrian in both camera views are projected into the top view according to the homography for a plane, the intersection of the projected foreground polygons shows the location of that object intersected by that plane. Figure 5.4 (a) shows the intersection region according to the homography for the ground plane. Such an intersection region is then warped back into a single camera view (Figure 5.4 (b)), where the green lines and red dots represent the polygon edges and vertices of the back-warped intersection region in camera view *b*. Figure 5.4 (c)-(h) show the results at heights of 0.5 m, 1.0 m and 1.5 m.

Assuming that the pedestrians are standing upright, the ground plane and D virtual planes at different heights are considered. Let h be the height of plane p with a height range $[0, 2]$. $\{P_{i,j,p}^t\}_{p\in[0,D]}$ represent a set of foreground intersection regions at different heights but at the same location in the top view. When $P_{i,j,p}^t$ with different h value are projected onto the ground plane, they are at the same position in the ground plane. Therefore, $P_{i,j,p}^t$ can be observed at the location where the object touches the ground. Then, for the intersection region $P_{i,j,g}^t$, the index p of the plane can be removed. Figure 5.5 shows an example of the warped back foreground intersections in the two camera views, in which the intersection regions are represented with green lines and red dots. For each foreground region, although the warped back intersection regions are located at different heights, they projected to the same position on the ground plane.



(a) (b)

Figure 5.5 An example of the warped back foreground intersections in two camera views.

5.3 Warped Back Patches in a Single View

The height matching algorithm proposed in this chapter is derived from the geometry between the top view and the individual camera views. Since it is based on each camera view, each intersection region in the top view needs to be warped back to the individual camera views first. Given an intersection region $P_{i,j}^{t}$ in the top view, the image patch in camera view a, which is warped back from the top view using the ground-plane homography, is as follows:

$$P_{i,j}^{a} = \left(H_{g}^{a,t}\right)^{-1}\left(P_{i,j}^{t}\right) \tag{5.2}$$

For each foreground region in camera view a, the image patches which are warped back on that foreground region are grouped into a patch set of that foreground region. For example, if the i-th foreground region in camera view a is F_{i}^{a} and the J foreground regions in camera view b are $\left\{F_{j}^{b}\right\}_{j\in[1,J]}$, there will be up to J intersection regions $\left\{P_{i,j}^{t}\right\}_{j\in[1,J]}$ in the top view, which are due to F_{i}^{a}. When these intersections $\left\{P_{i,j}^{t}\right\}_{j\in[1,J]}$ are warped back into camera view a, the image patches $\left\{P_{i,j}^{a}\right\}_{j\in[1,J]}$ is defined as the patch set corresponding to the foreground region F_{i}^{a} in camera view a.

5.4 Height Matching in a Single View

In the height matching algorithm, geometrical relationships are utilized to identify the top-view intersection regions that are due to corresponding foreground region pairs in the individual camera views. The foreground correspondence is determined by comparing the feet of a foreground region and the warped back patches corresponding to that foreground region in an individual camera view. Here, the foreground segmentation error is assumed to be relatively low.

5.4.1 Normalized Distances

The normalized distance is the distance between the centroid of a warped back patch and the foot point of that patch's corresponding foreground region in a camera view. Given a foreground region F_{i}^{a} and a warped back patch $P_{i,j}^{a}$ whose corresponding foreground region in camera view a is F_{i}^{a}, the distance between the

centroid of $P_{i,j}^a$ and the bottom of F_i^a is denoted as $h_{i,j}^a$. To remove the perspective effects, $h_{i,j}^a$ is normalized by h_i^a, which is the height of F_i^a:

$$d_{i,j}^a = \frac{h_{i,j}^a}{h_i^a} \tag{5.3}$$

The normalized height $d_{i,j}^a$ indicates the likelihood that $P_{i,j}^a$ is located around the foot area of F_i^a and that $P_{i,j}^a$ contains an object. Figure 5.6 shows a schematic diagram of how to calculate the normalized height in camera view a. $h_{i,j}^a$ can be either positive or negative. When $P_{i,j}^a$ is located below the bottom of F_i^a, $h_{i,j}^a$ has a negative value, otherwise it has a positive value. Therefore, the range of the normalized height $d_{i,j}^a$ is from a negative value to 1.

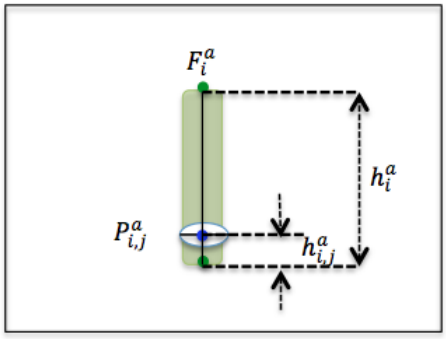


Figure 5.6 A schematic diagram of height matching in a camera view.

5.4.2 Height Matching of a Patch Set

Given a patch set $\{P_{i,j}^a\}_{j\in[1,J]}$ for the i-th foreground region in camera view a, the normalized distance of each patch in $\{P_{i,j}^a\}_{j\in[1,J]}$ is calculated and the normalized distance set is denoted as $\{d_{i,j}^a\}_{j\in[1,J]}$. Ideally, only one patch in $\{P_{i,j}^a\}_{j\in[1,J]}$ should be located around the foot area of F_i^a and be recognized as the correct match of F_i^a. Therefore, the patch which has the minimal normalized distance in a patch set is identified as the match, if such a distance is less than a threshold.

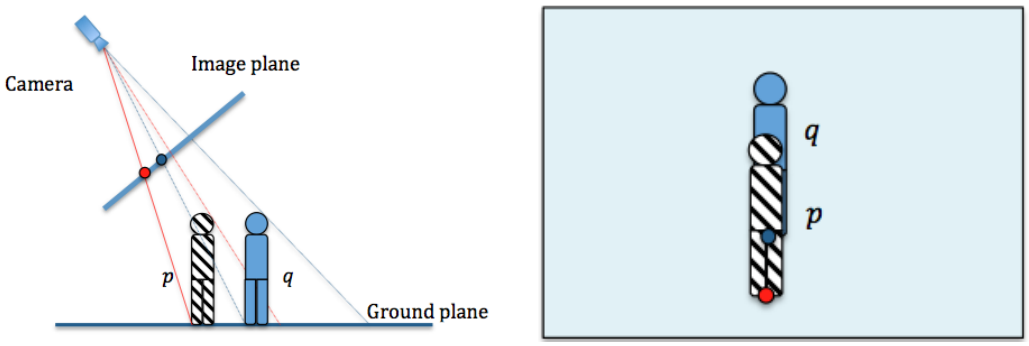
$$J_i^a = arg\min_{j\in[1,J]}\{d_{i,j}^a:\ |d_{i,j}^a| \le Th_d\} \tag{5.4}$$

5.5 Patch Classification in a Single View

Except for the patch matched with F_i^a, other patches in a patch set which corresponds to F_i^a are classified through position analysis.

5.5.1 Position Analysis

In position analysis, the camera is assumed to be viewing downward. Therefore the vanishing point is in the direction of positive infinity in the image coordinates. This assumption is satisfied in most visual surveillance systems. According to projective geometry, if an object moves closer to the camera in the top view, that object will move downward in that camera view. Suppose p and q are two objects on the same ray passing through a camera centre. Figure 5.7 shows a schematic diagram of the position analysis in a camera view. There are two objects p and q on the ground plane in Figure 5.7 (a), in which object p is closer to the camera than object q. Therefore, object p is located under object q and may partly occluded by q in the camera view (Figure 5.7 (b)).



(a) (b)

Figure 5.7 A schematic diagram of position analysis in a camera view.

5.5.2 Patch Classification in a Single View

Position analysis can be applied in patch classification for a single view. It is based on the normalized distances of the patches. For two warped back patches in a patch set, the patch which has a larger normalized distance locates above the other patch in the camera view.

During the height matching in the camera view, the warped back patch which matches the foreground region for each patch set is identified. The normalized distances of the other patches are compared with that of the matched patch in the

same patch set to decide whether the other patches are above or below the matched patch. The patches in that patch set can be divided into three categories: the object patch, upper patches and lower patches. These categories are labelled with 'Op', 'Up' and 'Lp' respectively, and further explained as follows. The object patch 'Op' corresponds to the foot location of an object which is visible in that camera view. The lower patches 'Lp' indicate the locations in front of the object location, thus no objects can be located in front of the object patch in this category. The upper patches 'Up' show the locations behind the object location. It is impossible to identify whether there is an object located at the upper patch or not in a single view.

5.6 Height Matching in the Top View

After the warped back regions are classified in a single camera view, the classification results from both views are incorporated to classify the intersection regions in the top view. The intersection regions in the top view are classified into four categories: object regions, occluded regions, covered regions and phantoms, which are labelled with 'Ob', 'Oc', 'Cv' and 'Ph', respectively. Table 5.1 summarizes the classification of the intersection regions from the two camera views.

Table 5.1 Classification of the intersection regions from two camera views.

Camera View *a* / Camera View *b*	Op	Up	Lp
Op	Ob	Oc	Ph
Up	Oc	Cv	Ph
Lp	Ph	Ph	Ph

In Table 5.1, if the warped back patches of an intersection region are identified as object patches in both camera views, that intersection region contains an object and is visible in the two camera views. If an intersection region is identified as an object patch in one camera view and an upper patch in another, the corresponding object is visible in the first camera view and occluded in the second camera view. As a result, that intersection region is classified as an occluded region. When an intersection region is identified as an upper patch in both camera views, it is labelled as a covered region, because it may be a phantom or contain a real object. If

the warped back patch for an intersection region is identified as a lower patch in either camera view, it is determined as a phantom.

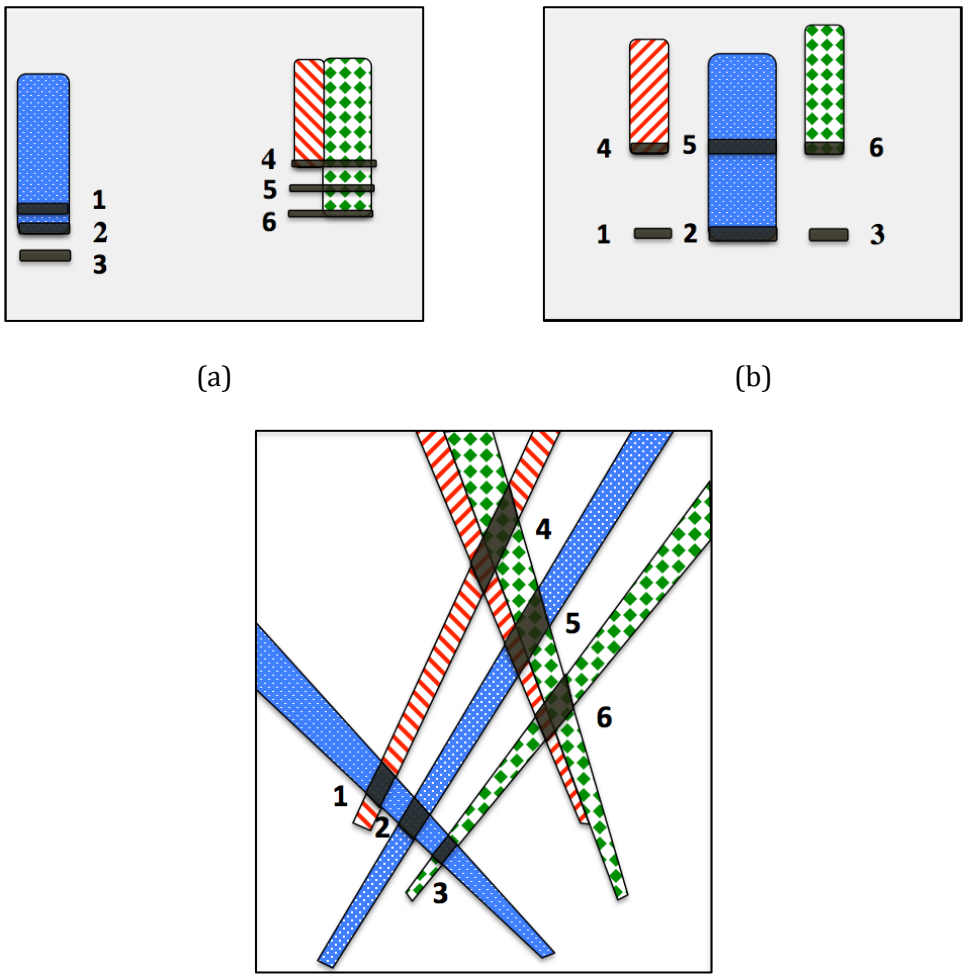


Figure 5.8 A schematic diagram of the position analysis in two camera views, (a) overlaid warped back patches in camera view *a*, (b) warped back patches in camera view *b*, and (c) overlaid foreground projections in the top view.

Figure 5.8 shows an example of the position analysis in the top view. In Figure 5.8 (a) and (b), there are three objects which are illustrated in red stripes, green squares and blue dots in each camera view. The objects with the same colours and patterns in the two camera views are the same objects. Since the green square object occludes the red striped object in Figure 5.8 (a), they are grouped into a single foreground region. The foreground regions in each camera view are projected from the two camera views to the top view according to the homography for a plane parallel to the ground plane and at some height. The two foreground regions from camera view *a* and three foreground regions from camera view *b* intersect in 6 regions in the top view. Figure 5.8 (c) shows the overlaid foreground projections

and darker intersection regions which are labelled with 1 to 6 in the top view. The ground-truth object locations are intersection regions 2, 4 and 6. The warped back patches of these intersection regions from the top view to the individual camera views according to the ground-plane homography are the black patches in Figure 5.8 (a) and (b). Each warped back patch is given the same label as the corresponding intersection region in the top view.

Intersection region 1, which corresponds to an upper patch in Figure 5.8 (a) and a lower patch in Figure 5.8 (b), is a phantom region. Intersection region 3 is a phantom, as its warped back patches are lower patches in both camera views. The warped back patches of intersection regions 2 and 6 are located at the foot area of the corresponding foreground objects in the two camera views. Those intersection regions are the object regions indicating the locations of the blue dot object and the green square object in the top view. Intersection region 4 is an occluded region because its warped back patch is an object patch in camera view *b* but is an upper patch in camera view *a*, indicating that intersection region contains an object but is occluded by another object in one camera view. Warped back patch 5 is an upper patch in both camera views and corresponds to a covered region. It is occluded by the green squared object at intersection region 6 in camera view *a* and the blue dot object at intersection region 2 in camera view *b*.

The details of the phantom pruning algorithm using height matching are described as **Algorithm 2**.

Algorithm 2: Phantom pruning using height matching

1:	**for** all camera views **do**
2:	each intersection region $P_{i,j}^{t}$ of foreground projections in the top view are warped back to the camera view by using the homography for the ground plane;
3:	**for** all foreground regions in the camera view (using a as the index of the camera) **do**
4:	the patch set $\{P_{i,j}^{a}\}_{j\in[1,J]}$ of F_i^a is generated;
5:	**for** all the warped back patches in $\{P_{i,j}^{a}\}_{j\in[1,J]}$ **do**
6:	calculate the normalized distance $d_{i,j}^{a}$;
7:	$J_i^a = arg\min_{j\in[1,J]}\{d_{i,j}^{a}: \lvert d_{i,j}^{a}\rvert \le Th_d\}$

8:	$d_i^a = d_{i,j}^a$
9:	**end for**
10:	**for** all the warped back patches in $\{P_{i,j}^a\}_{j \in [1,J]}$ **do**
11:	**If** $d_{i,j}^a == d_i^a$, **then** $P_{i,j}^a$ is labeled as Object Patch (Op);
12:	**else if** $d_{i,j}^a > d_i^a$, **then** $P_{i,j}^a$ is labeled as Upper Patch (Up);
13:	**else** $P_{i,j}^a$ is labeled as Lower Patch (Lp);
14:	**end if**
15:	**end for**
16:	Classify the foreground intersection regions based on the integration of the results from all the camera views.
17:	**end for**
18:	**end for**

5.7 Experimental Results

In this chapter, the focus has been on the identification of false-positive pedestrian detections. The campus dataset was used in the experiments. Figure 5.9 illustrates the results of foreground detection in the two camera views. The first curve shows the ground-truth number of objects in the FOVs in each sampled frame. The second and third curves show the number of detected foreground regions in the two camera views. The valleys in these two curves correspond to the frames which have an occlusion. It is noted that this dataset contains frequent occlusions.

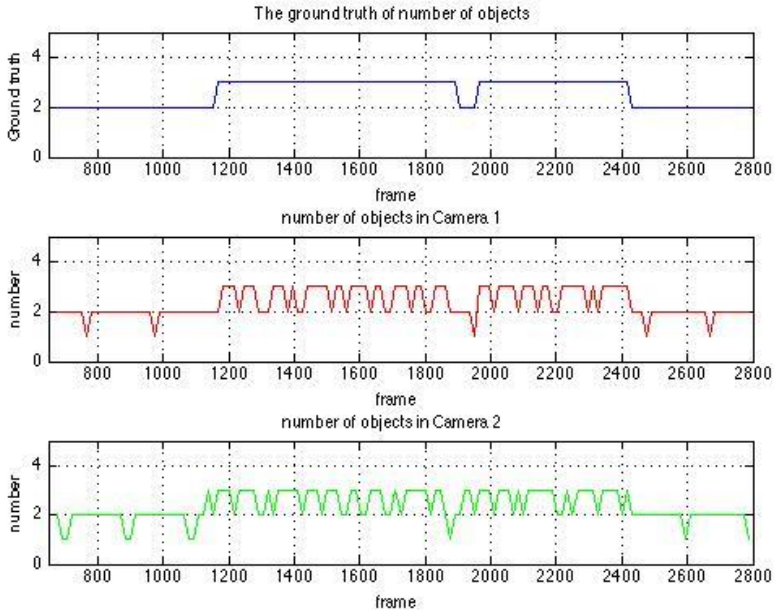


Figure 5.9 The ground-truth number of objects (the first curve) and the numbers of the detected foreground regions in two camera views (the second and third curve).

5.7.1 Intersection Region Analysis

In these experiments, the homography mapping is based on a plane parallel to and one meter above the ground plane, which is at the waist level of an average pedestrian. Each foreground polygon in a camera view was warped to the top view according to the homography for this plane. A threshold was applied to the overlaid foreground projections in the top view. The number of the intersected regions, which is determined by the numbers of the detected foreground regions in both camera views, is slightly affected by the separated legs of pedestrians. Figure 5.10 shows a comparison of the number of the detected intersection regions and the number of the expected intersection regions in the FOVs. The first curve shows the number of the detected intersection regions in the FOVs whereas the expected number of intersection regions is depicted in the second curve. The third curve is the difference between the first two curves and shows the frames in which there is an additional intersection region caused by separating legs.

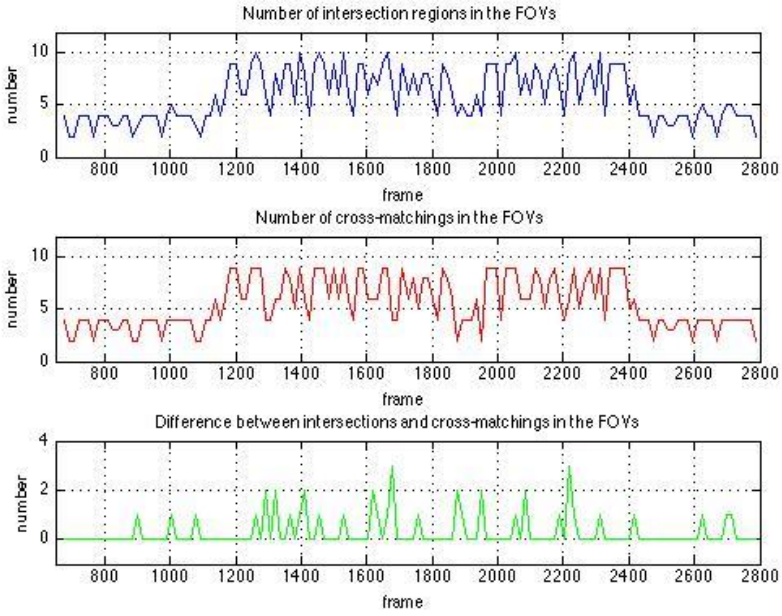


Figure 5.10 The number of the detected intersection regions (the first curve), the number of the expected intersection regions (the second curve), and the difference between these two curves (the third curve).

5.7.2 Phantom Pruning with Height Matching

The approximated polygon of each intersection region in the top view was warped back into the individual camera views according to the ground-plane homography. The distance between the warped back patch and the foot area of its corresponding foreground region is calculated in each camera view. The location of each warped back patch in a single camera view is represented by its centroid. The location of the foot area of a foreground region is represented by the bottom of the bounding box. If the ratio between the distance and the height of the bounding box is less than 0.1, the warped back patch was thought of as being located near the foot area of its corresponding foreground region.

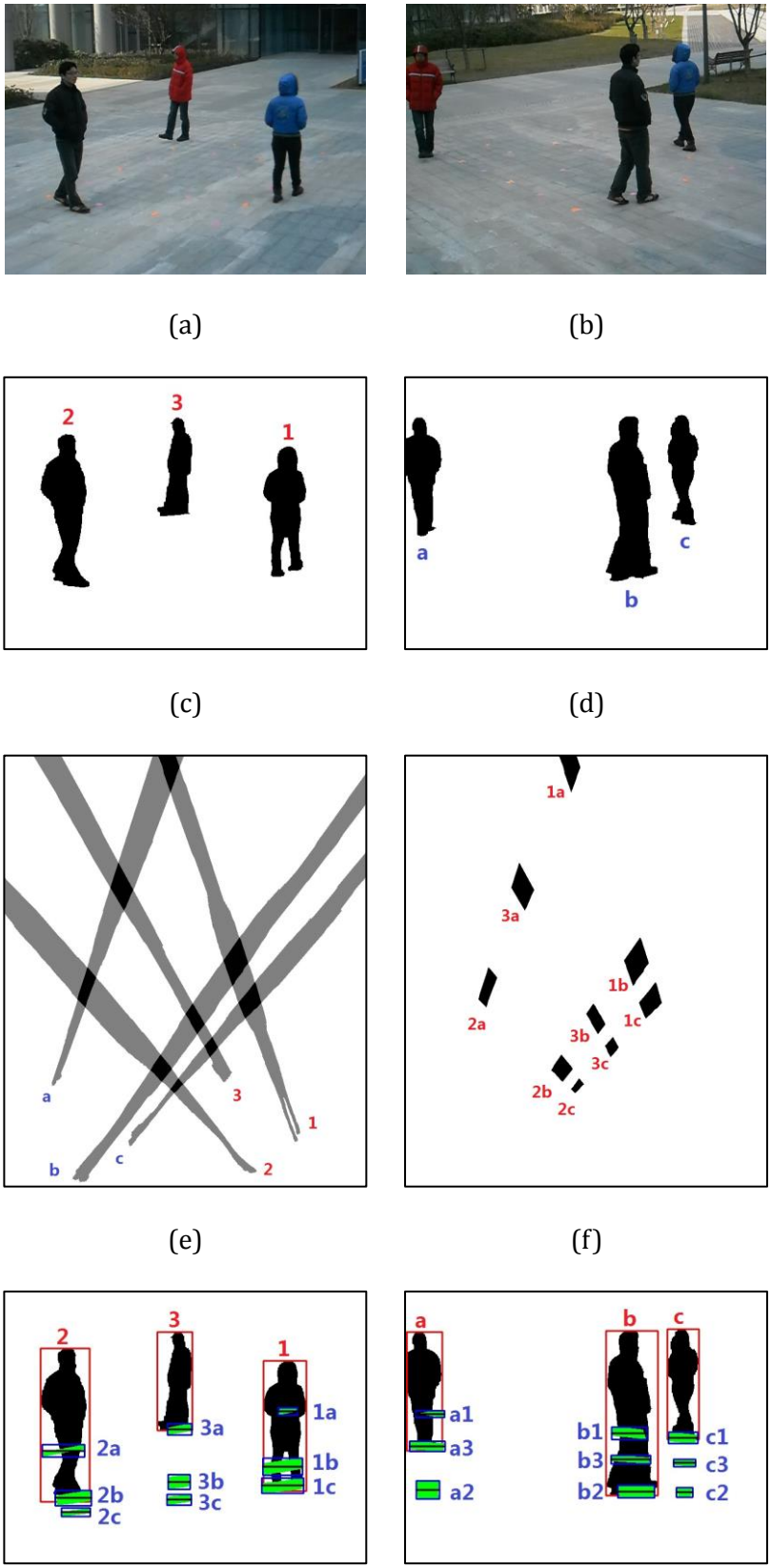


Figure 5.11 The process of phantom removal using height matching at frame 1200: (a)(b) the original images, (c)(d) the foreground regions in the two camera views, (e) the overlaid foreground projections in the top view, (f) the intersection regions in the top view, (g)(h) the warped back patches in the two camera views.

Figure 5.11 shows the procedure for the phantom removal algorithm using geometric information at frame 1200. Figure 5.11 (a)-(d) are the original images and the results of foreground detection in the two camera views. In each camera view, there are three pedestrians which are labelled with 1 to 3 in camera view *a* and labelled with a to c in camera view *b*. Figure 5.11 (e) shows the overlaid foreground projections from the two camera views to the top view with the homography for a plane at a height of one meter. Each foreground projection in the top view is given the same label of the corresponding foreground region in individual camera views. The foreground projections intersect in 9 regions. Each intersection region is given a label to indicate the corresponding foreground regions in the two camera views. For example, region 1a is the intersection of foreground region 1 in camera view *a* and foreground region a in camera *b*. Figure 5.11 (f) shows the intersection regions in the top view. The intersection regions are warped back to each camera view according to the ground-plane homography. Figure 5.11 (g) and (h) are the warped back patches overlaid on the original camera views. Green regions are the warped back patches with blue boxes as their bounding boxes. The blue line in the middle of the bounding box for a warped back patch represents the centroid of the warped back patch. Each warped back patch is given the same label of the corresponding intersection region in the top view. The red bounding box and the label of each foreground region are also marked in the two camera views. The bottom line of the red bounding box illustrates the foot point of each foreground region.

The normalized distance of each warped back patch in a single camera view is calculated according to equation (5.3), using the height of its corresponding foreground bounding boxes and the distance between the centroid line of the patch and the bottom line of the corresponding foreground bounding box. Table 5.2 and Table 5.3 show the results of the height matching for the warped back patches in camera view *a* and camera view *b* respectively. For foreground region 1 in camera view *a*, it is related to three warped back patches labelled with 1a, 1b and 1c. Patch 1c, which has the minimal normalized distance less than the threshold 0.1, is identified as an object patch. Patches 1a and 1b which have normalized distances larger than that for patch 1c are recognized as upper patches. For foreground region 3 in camera view *a*, patch 3a is identified as an object patch. Patches 3b and 3c are identified as lower patches because their normalize distances are less than that for patch 3a. Since foreground region b and c in camera view *b* are close to each other, the warped back patches related to b and c are closely located in camera view *a*.

Table 5.2 Height matching at frame 1200 in camera view *a*, the data in bold is the smallest normalized distance in each patch set.

Foreground Region in Camera View *a*	Foreground Region in Camera View *b*	Normalized Distance
1	a	0.613
	b	0.187
	c	**0.043**
2	a	0.332
	b	**0.026**
	c	-0.067
3	a	**0.012**
	b	-0.523
	c	-0.701

Table 5.3 Height matching at frame 1200 in camera view *b*, the data in bold is the smallest normalized distance in each patch set.

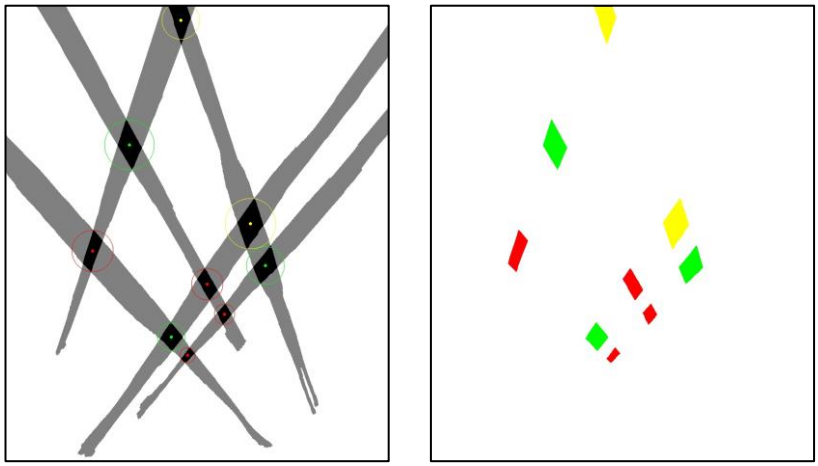
Foreground Region in Camera View *b*	Foreground Region in Camera View *a*	Normalized Distance
a	1	0.310
	2	-0.329
	3	**0.038**
b	1	0.378
	2	**0.024**
	3	0.220
c	1	**0.015**
	2	-0.477
	3	-0.210

The classification results in the two camera views are combined to make a final decision according to Table 5.1. Table 5.4 shows the classification results of the intersection regions. To visualize the results, in Figure 5.12, each intersection region

in the top view is filled with a different colour, in which red indicates phantom regions, green indicates object regions and yellow is for covered regions that cannot be visible in either view. The blue colour is used for the regions that are occluded in one view and are visible in the other view. There is no such region in frame 1200. However, it can be found in frame 1335.

Table 5.4 The results of regions classification using height matching.

Region	1a	1b	1c	2a	2b	2c	3a	3b	3c
Camera *a*	Up	Up	Op	Up	Op	Lp	Op	Lp	Lp
Camera *b*	Up	Up	Op	Lp	Op	Lp	Op	Up	Lp
Label	Cv	Cv	Ob	Ph	Ob	Ph	Ob	Ph	Ph



(a) (b)

Figure 5.12 Classification results of the intersection regions at frame 1200, (a) in the overlaid foreground projection image, and (b) in the foot location map.

Figure 5.13 illustrates the procedure for phantom removal algorithm using geometric information at frame 1335. Figure 5.13 (a)-(d) are the original images and the results of foreground detection in the two camera views. Although there are three pedestrians in each camera view, the black pedestrian is occluded by the red pedestrian in Figure 5.13 (b). In Figure 5.13 (d), those two pedestrians are detected as a single foreground region which is labelled with b and the other foreground region is labelled with a. The foreground regions in Figure 5.13 (c) are labelled with 1 to 3. When the foreground regions are projected from the two camera views to the top view with the homography for a plane at a height of one meter, they intersect in 6 regions.

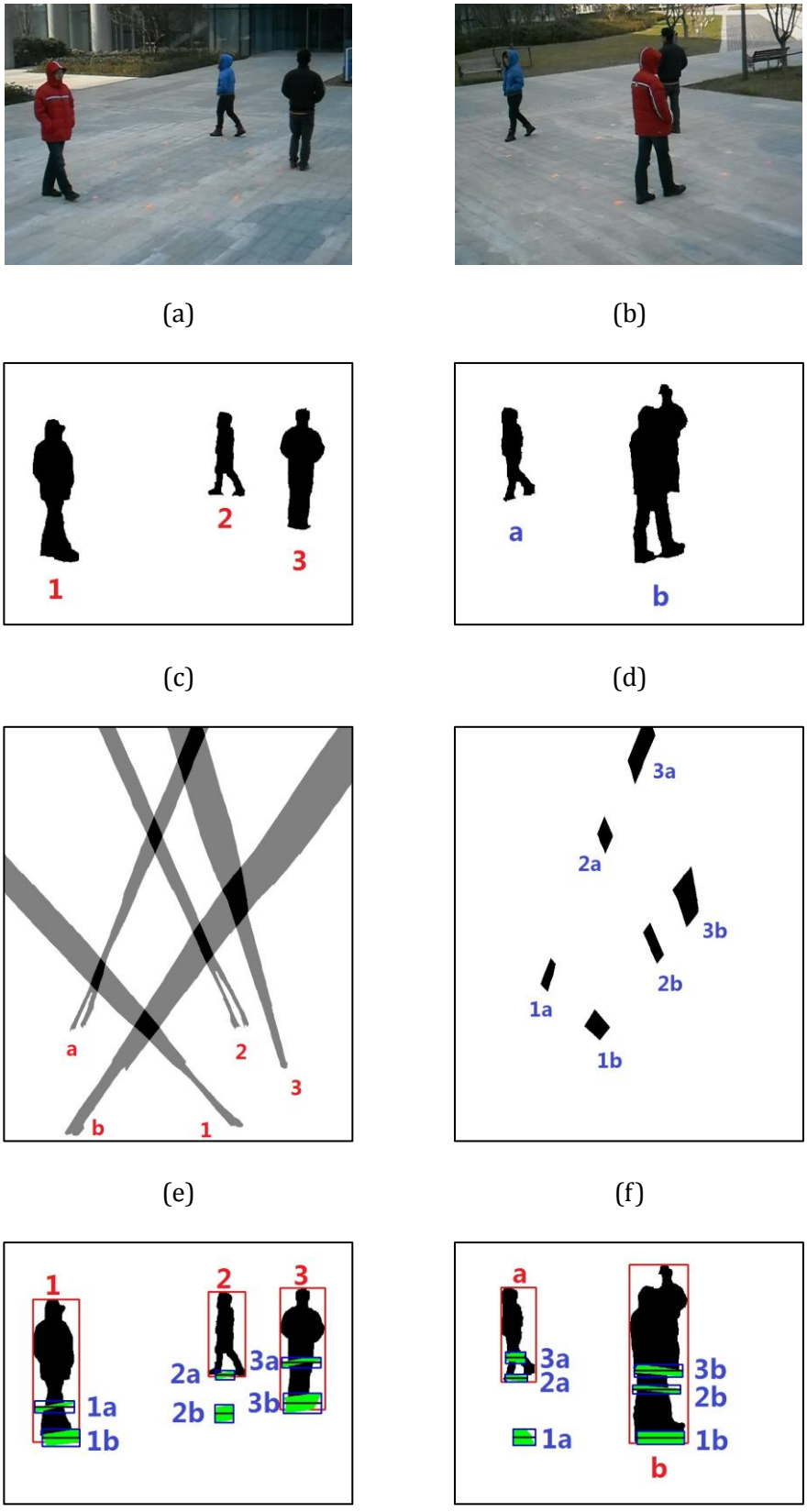


Figure 5.13 The process of phantom removal using height matching at frame 1335: (a)(b) the original images, (c)(d) the foreground regions in two camera views, (e) the overlaid foreground projections in the top view, (f) the intersection regions in the top view, and (g)(h) the warped back patches in the two camera views.

Figure 5.13 (e) and (f) show the overlaid foreground projections and intersection regions. The labels of the intersection regions are given according to those corresponding foreground regions in the two camera views. The intersection regions are warped back to each camera view according to the ground-plane homography. Figure 5.13 (g) and (h) are the warped back patches overlaid in the original camera views.

The normalized distance between the centroid of each warped back patch and its corresponding foreground foot location in the two camera views are shown in Table 5.5 and Table 5.6. In Table 5.5, for foreground regions 1 to 3 in camera view *a*, the warped back patches which have the minimal normalized distances less than the threshold 0.1 are 1b, 2a and 3b respectively. Those are identified as matched patches for their corresponding foreground regions and are labelled with 'Op'. As the normalized distances of patches 1a and 3a are larger than those corresponding matched patches 1b and 3b, patches 1a and 3a are classified as upper patches with the label 'Up'. Patch 2b is identified as lower patches because its normalize distance is less than that for patch 2a. The results of the patch classification in camera view *a* are shown in the second row in Table 5.7.

Table 5.5 The height matching at frame 1335 in camera view *a*, the data in bold is the smallest normalized distance in each patch set.

Foreground Region in Camera View *a*	Foreground Region in Camera View *b*	Normalized Distance
1	a	0.247
	b	**0.030**
2	a	**0.019**
	b	-0.436
3	a	0.388
	b	**0.054**

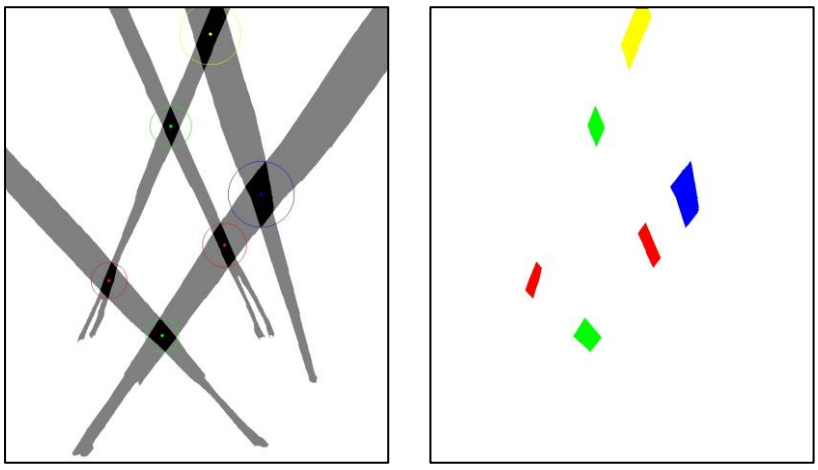
In Table 5.6, the matched patches for foreground region a and b are patches 2a and 1b. Patch 1a is identified as a lower patch whereas patches 3a, 2b and 3b are classified as upper patches. The third row in Table 5.7 shows the results of patch classification in camera view *b*. In the fourth row, the classification results in the two camera views are combined to make a final decision according to Table 5.1. At the end, the classification result at frame 1335 is visually illustrated in Figure 5.14.

Table 5.6 The height matching at frame 1335 in camera view *b*, the data in bold is the smallest normalized distance in each patch set.

Foreground Region in Camera View *b*	Foreground Region in Camera View *a*	Normalized Distance
a	1	-0.586
	2	**0.040**
	3	0.259
b	1	**0.030**
	2	0.301
	3	0.407

Table 5.7 The classification results of the foreground intersections at frame 1335 using the height matching.

Region	1a	1b	2a	2b	3a	3b
Camera *a*	Up	Op	Op	Lp	Up	Op
Camera *b*	Lp	Op	Op	Up	Up	Up
Label	Ph	Ob	Ob	Ph	Cv	Oc



(a) (b)

Figure 5.14 Classification results of the intersection regions at frame 1335 using height matching, (a) in the overlaid foreground projection image, and (b) in the foreground intersection image.

5.7.3 Evaluations

To show the performance, the phantom removal algorithm using height matching has been tested over 142 frames. The classification results are compared with the ground truth. Table 5.8 and Table 5.9 show the performance evaluation of the height matching in comparison with the ground truth. 786 intersection regions in 142 frames are classified into four categories: object regions, phantom regions, covered regions and occluded regions.

Table 5.8 Performance evaluation of the classification using height matching.

		Classification Results with Height Matching				Number of the Ground Truth
		Object Regions (Ob)	Phantom Regions (Ph)	Covered Regions (Cv)	Occluded Regions (Oc)	
Ground Truth	Object Regions (Ob)	304	0	9	6	319
	Phantom Regions (Ph)	0	307	2	5	314
	Covered Regions (Cv)	0	0	112	0	112
	Occluded Regions (Oc)	1	0	0	40	41
Number of classification		305	307	123	51	786

In Table 5.8, the confusion matrix of the classification results is given, along with the ground truth and the classification results. For each category, let *GT* and *CR* be the ground-truth numbers and actual classification numbers of that category. The false negatives (missed detections), *FN,* are the intersection regions which belong to that category but misclassified as the other category. The false positives (*FP*) or false alarms are the intersection regions which belong to the other category but are misclassified as that category. The false negative rate (R_{FN}) is obtained as the ratio

between the number of the false negatives and the number of ground truths. The false positives rate (R_{FP}) is the ratio between the number of the false positives and the ground-truth number.

$$R_{FN} = FN / GT \tag{5.5}$$

$$R_{FP} = FP / GT$$

The false negative rate and the false positive rate of the classification with height matching are shown in Table 5.9. For the object region, the number of ground truths is 319, in which 304 are correctly identified. 9 object regions which are misclassified as covered regions and 6 object regions which were misclassified as occluded regions are the false negatives. The false positive is the occluded region which is misclassified as an object region. The false negative rate is 4.70% and the false positive rate is 0.31%.

Table 5.9 The classification errors with height matching.

	False Negative Rate R_{FN} (%)	False Positive Rate R_{FP} (%)
Object Regions	4.70	0.31
Phantom Regions	2.23	0.00
Covered Regions	0.00	9.82
Occluded Regions	2.44	26.83

5.8 Summary

In this chapter, a height matching algorithm was proposed to identify the false-positive detections or phantoms using the geometrical information. Since the height matching is carried out in each camera view, intersection regions in the top view are warped back into the individual camera views. Based on the normalised distances between the centroids of the warped back patches and the foot points of the corresponding foreground regions, the nearest-neighbourhood algorithm is used to identify the matched patch for each foreground region. The position analysis is further applied to classify other patches in a single camera view. Finally, the classification results from both camera views are incorporated to classify the intersection regions in the top view. The experimental results have shown that this algorithm can robustly classify the intersection regions in the top view. The limitation of this algorithm is that the foreground segmentation error is assumed to

be relatively low. When the foreground segmentation error is high, a high threshold Th_d should be applied in the height matching, which increases the misclassification. Furthermore, when two or more objects are very close to each other in one camera view, the warped back patches of these objects may be close to the feet of the same object simultaneously. In this case, colour cues will be employed to identify the correct match. This is discussed in the next chapter.

6 PHANTOM REMOVAL WITH HEIGHTS AND COLOUR CUES

In the previous chapter, a height matching algorithm was proposed to identify the false-positive detections, which was based on the geometry between the individual camera views. However, when two or more objects are close to each other in one camera view, the warped back patches of these objects may be close to the feet of the same pedestrian in another camera view. This brings difficulties to the Nearest-Neighbourhood based height matching, when there exist homography estimation errors and foreground detection errors. On the other hand, colour is a strong cue to distinguish between objects. The colours of foreground regions in the individual camera views can be used to identify whether each intersection region in the top view is due to the same object or not. In this thesis, the colours of each foreground region were used to build an appearance model and a colour matching algorithm based on the Mahalanobis distance was applied to calculate the similarity of two foreground regions in the colour.

6.1 Colour Spaces

A colour space is the colour coordinate system to represent the colours of pixels in the image. They can be divided into three categories: physics and technic-based colour space (RGB, CMY, YUV, YC_BC_r and $I_1I_2I_3$), Uniform colour space (CIELAB and CIELUV), and perception-based colour space (HSI and HSV) [97]. Zhang et al. [98] investigated various colour spaces such as RGB, Lab, HSV and log-RGB. In this research, the RGB space, rgb space and HSI space were used in the colour matching algorithm.

6.1.1 The RGB and rgb Models

According to the laws of colourimetry, any colour can be created by red, green and blue, and the combined colour of these three colours is unique. Therefore, the most commonly used colour space is the RGB colour space. This colour space is popular because all other kinds of colour representations can be derived from the RGB

colour space by using linear or nonlinear transformations. Although the RGB colour space is powerful in colour display, it does not meet the requirements in colour-based region segmentation because the three components R, G and B are highly correlated and can simultaneously vary with illumination changes.

A normalized rgb colour model is sometimes used to remove the correlation among the three colour components. However, this method is sensitive to the image noise in the dark or black regions in an image. The rgb model is derived from the RGB model as follows:

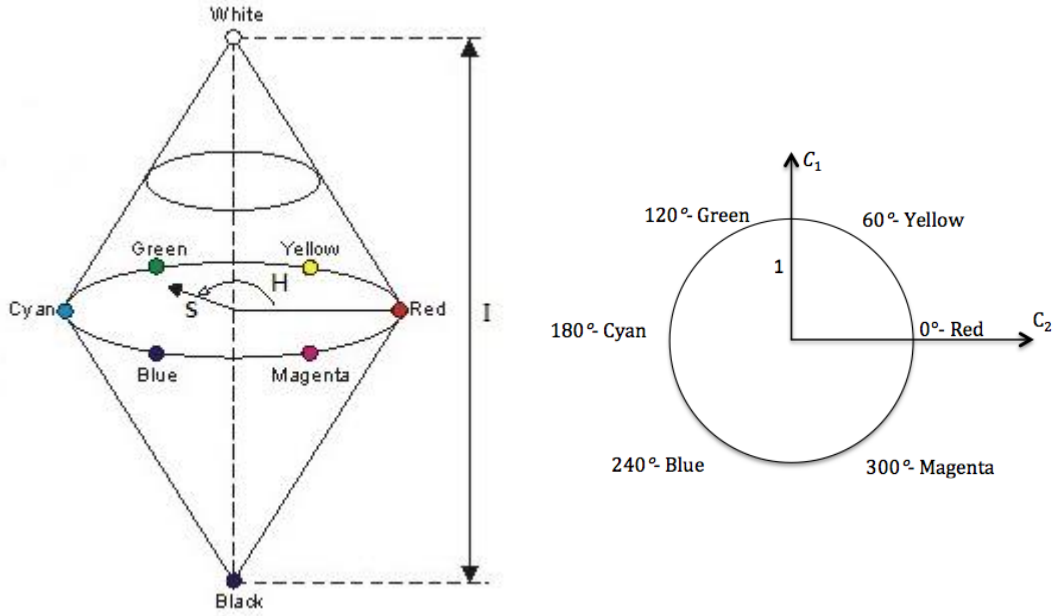
$$r = R / (R + G + B) \tag{6.1}$$

$$g = G / (R + G + B)$$

$$b = B / (R + G + B)$$

6.1.2 The HSI Model

The HSI (hue-saturation-intensity) model is also widely used in image processing. The hue component describes the basic colour and the saturation component determines the purity of the colour. The intensity indicates the brightness of an image. The three dimensions of HSI colour space is illustrated in Figure 6.1 (a). In Figure 6.1 (b), the two-dimensional C_1-C_2 space is used to illustrate how the hue space works. Points on the unit circle correspond to different colours. The hue value of a point on the unit circle is the positive angle from C_2 axis going anticlockwise along the unit circle to that point.



(a) (b)

Figure 6.1 HSI colour space, (a) three dimensions and (b) distribution of hue values.

The HSI model is derived from the RGB model as follows:

$$H = arctan\left(\frac{\sqrt{3}(\mathrm{G} - \mathrm{B})}{(\mathrm{R} - \mathrm{G}) + (\mathrm{R} - \mathrm{B})}\right) \tag{6.2}$$

$$\mathrm{I} = \frac{(\mathrm{R} + \mathrm{G} + \mathrm{B})}{3}$$

$$\mathrm{S} = 1 - \frac{\min(\mathrm{R}, \mathrm{G}, \mathrm{B})}{\mathrm{I}}$$

6.2 Colour Matching Methods

The methods to calculate the colour similarity of two regions can be divided into two categories: template matching and statistical matching. In the template matching, each pixel in one image is compared with the corresponding pixel in the other image. Then the differences of all the pixels in the underlying region are averaged or summed. The statistical matching methods, such as colour histogram based matching and Mahalanobis distance based matching, compare the statistical properties of the pixel colours in two regions.

6.2.1 Template Matching

Template matching compares each pair of the pixels, which are at the same location of two different images, by using some distance measurement. Suppose that $\mathbf{R}_a(x, y)$ is the colour value of pixel (x, y) in a region in camera view a and $\mathbf{R}_b(x, y)$ is that in camera view b. The colour difference of these two regions is measured by the sum of squared colour distances over all the pixels:

$$D_{ab} = \frac{\sum_{(x,y)} \|\mathbf{R}_a(x, y) - \mathbf{R}_b(x, y)\|^2}{\sqrt{\left(\sum_{(x,y)} \|\mathbf{R}_a(x, y)\|^2\right)\left(\sum_{(x,y)} \|\mathbf{R}_b(x, y)\|^2\right)}} \tag{6.3}$$

which is normalised by the sum of the squared magnitudes of the colour values. This can reduce the influence of the different colour sensitivity of the cameras and the locations of the light source in different camera views. An alternative equation is written as:

$$D_{ab} = \frac{\sum_{(x,y)} \|\mathbf{R}_a(x, y) - \mathbf{R}_b(x, y)\|^2}{\sum_{(x,y)} \|\mathbf{R}_a(x, y) + \mathbf{R}_b(x, y)\|^2} \tag{6.4}$$

6.2.2 Histogram Based Colour Matching

For colour histogram matching, a number of measures for the distance between two colour histograms have been previously proposed. These measures follow either a vector-based approach or a probabilistic approach. In the vector approach [99], Euclidean intersection is applied to measure the distance between the two colour histograms by considering each histogram as a vector [16]. In the probabilistic approach, the colour histograms are thought of as probability density functions. Then the distance between the two probability density functions is calculated.

6.2.3 Mahalanobis Distance Based Colour Matching

By normalizing the distance of the means with the covariances, it is possible to obtain a more realistic distance measure between two regions. Given the colour values of the pixels in region a, the mean $\boldsymbol{\mu}_a$ and the covariance $\boldsymbol{\Sigma}_a$ of the values can be estimated. Let $\boldsymbol{\mu}_b$ and $\boldsymbol{\Sigma}_b$ be the mean and the covariance of the colour values of the pixels in region b, the Mahalanobis distance can be used to measure the similarity of the colours in these two regions:

$$D_{ab} = (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^{\mathrm{T}}(\boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b)^{-1}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b) \tag{6.5}$$

6.3 Appearance Matching

Since colour is a strong cue to differentiate objects, the colours of the foreground regions in individual camera views are utilized to identify whether two foreground projections from different camera views are due to the same object. The first step of colour matching is to generate the appearance model of each foreground region. Then, the colour similarity of two foreground regions, which intersect in the top view, is measured according to the Mahalanobis distance of their appearance models.

6.3.1 Torso Regions

A torso region is defined as the part of a foreground region, which is within a specified range of heights. The torso region is used because it has a large area and can provide stable colour cues and discriminative features to distinguish between pedestrians. This is the reason why the warped back patches in the individual camera views were used in the previous work but are replaced with the torso region here [100] [101].

6.3.2 Appearance Models

The appearance model of each torso region is built by using the colours of all the pixels in the torso region. To handle the multiple colours in the torso region, the appearance model is developed by using the Gaussian mixture model. If $\boldsymbol{x}_i$ is the d dimensional colour vector of a pixel in the torso region, the colour vectors of N pixels are denoted by $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$. Let K be the number of Gaussian distributions used in the Gaussian mixture model, the Gaussian mixture model is denoted as:

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{6.6}$$

where

$$\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)} \tag{6.7}$$

The K-means algorithm [102] and the Expectation-maximization (EM) algorithm [103] are widely used to find the parameters of the probability density functions in a Gaussian mixture model. The aim of the K-means algorithm is to rapidly find the natural clusters of the data. It is achieved by minimizing the sum of the squared distances of the pixel values to its cluster centres. The K-means algorithm can be described as follows:

1. The number of the clusters, *K*, is selected by the user.
2. For each cluster in *K* clusters, the location of centre $\boldsymbol{\mu}_k$ is assigned randomly. $\boldsymbol{\Sigma}_k = \sigma^2 \mathbf{I}_d$ and $\pi_k = {}^1/_K$ are fixed.
3. Given *N* data points, the distances between the *N* points and the individual cluster centres $\boldsymbol{\mu}_k$ are calculated separately. Each point is associated to the cluster centre which has the minimum distance with that point and is labelled as:

$$z_i = \arg\min_k \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_k \right\|^2 \tag{6.8}$$

4. The new cluster centres of the *K* clusters are calculated.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i:z_i=k} \boldsymbol{x}_i \tag{6.9}$$

5. Return to step 3 until convergence.

The EM algorithm can be used iteratively to find an optimal parameter vector $\boldsymbol{\theta}^t = \{\pi_k^t, \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t\}_{k=1}^K$ of the Gaussian Mixture Model (GMM), where t represents the t-th iteration of the EM algorithm [103]. The parameter vector $\boldsymbol{\theta}^{\mathbf{1}}$ is initialized by the results of the K-means algorithm. The probability of the colour vector of a pixel belonging to the k-th distribution, k = 1,2,…,K, is defined as π_k. π_k lies in the region [0,1] in the EM algorithm but it is either 0 or 1 in the K-means algorithm.

The parameters $\pi_k^t, \boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t$ for $t \geq 2$ are re-estimated using an expectation step (E-step) and a maximization step (M-step). For the E-step, the probability that the colour vector $\boldsymbol{x}_i$ is from the k-th distribution is calculated by:

$$r_{ik}^t = \frac{\pi_k p\left(\boldsymbol{x}_i | \boldsymbol{\theta}_k^{t-1}\right)}{\sum_{k'} \pi_{k'} p\left(\boldsymbol{x}_i | \boldsymbol{\theta}_{k'}^{t-1}\right)} \tag{6.10}$$

In the M-step, the prior, the mean vector and the dispersion matrix of each distribution in the t-th iteration are recalculated:

$$\pi_k^t = \frac{1}{N}\sum_i r_{ik}^t = \frac{r_k^t}{N} \tag{6.11}$$

$$\boldsymbol{\mu}_k^t = \frac{\sum_i r_{ik}^t \boldsymbol{x}_i}{r_k^t} \tag{6.12}$$

$$\boldsymbol{\Sigma}_k^t = \frac{\sum_i r_{ik}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k^t)^T (\boldsymbol{x}_i - \boldsymbol{\mu}_k^t)}{r_k^t} \tag{6.13}$$

When the log-likelihood function of the GMM reaches the local maximum, the iteration t stops increasing.

$$\ell\left(\boldsymbol{\mu}_{k,}\boldsymbol{\Sigma}_k\right) = \sum_k \sum_i r_{ik} \log \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) p(\boldsymbol{x}_i|\boldsymbol{\theta}_k) \tag{6.14}$$

In this chapter, since the clustering of the pixels in a torso region is based on the hue component, the special cyclic property of the hue is considered. Let U_1 and U_2 be two hue values, for example $U_1 = 0$ (red) and $U_2 = 5\pi/3$ (magenta). The difference between these two hue values is $5\pi/3$ which is larger than $\pi/3$, the absolute angle from U_1 going clockwise along the unit circle to U_2. To avoid such inconsistency, the distance between two hue values is defined as follows:

$$d(U_1, U_2) = \begin{cases} |U_1 - U_2| & (|U_1 - U_2| \leq \pi) \\ 2\pi - |U_1 - U_2| & (|U_1 - U_2| > \pi) \end{cases} \tag{6.15}$$

For the HSI colour space represented in a 3D-conic coordinate system, standard statistical formulae can only be used to calculate the saturation and intensity values. Since the hue is an angular value, circular statistical descriptors are used to calculate the mean and variance of a cluster [104]. Given n hue values $U_i,\ i = 1, \ldots, n$, the mean value is:

$$\overline{U} = arc\tan\left(\frac{B}{A}\right) \tag{6.16}$$

where:

$$A = \sum_i \cos U_i \tag{6.17}$$

$$B = \sum_i \sin U_i$$

The circular variance of the hue differs from the standard statistical variance in being limited to the range [0, 1] and is defined as:

$$V = 1 - \frac{\sqrt{A^2 + B^2}}{n} \tag{6.18}$$

Since the hue values of the red colour are around 0 degree, if the clustering algorithm is applied to the original hue values directly, the red colours will be clustered into different groups which are separated by 0 degrees. In the clustering process, the circular property of the hue component is considered to avoid over-clustering of the colour pixels around 0 degrees. Given a set of hue values for the pixels in a torso region, the dividing degree of the circular hue value should be changed to a location where there is no dominant hue value. When the mean value of those hue values, U_c, is calculated [105], the direction of C_2 axis is changed to that of the angle $U_c + \pi$.

The hue components of the original pixels are changed into the new coordinates. Then, the K-means algorithm and the EM algorithm are applied to cluster the pixels in the torso region into different Gaussian distributions. Given a foreground region F_i^a, the colour appearance of the torso region A_i^a is modeled by K Gaussian distributions: $\mathcal{N}(\pi_{i,n}^a, \mu_{i,n}^a, \Sigma_{i,n}^a), n \in [1, K]$, where $\pi_{i,n}^a$, $\mu_{i,n}^a$ and $\Sigma_{i,n}^a$ are the weight, mean and covariance of the *n-th* Gaussian distribution. The K Gaussians are ordered according to the magnitudes of the weights and $\pi_{i,1}^a$ is the greatest weight.

6.3.3 Colour Matching

After the appearance model of each torso region is represented by a mixture of Gaussian distributions, the Mahalanobis distance is used to measure the colour similarity between any two foreground regions. Here the Mahalanobis distance rather than Battacharrya distance is used, because the former is appropriate for Gaussian distributions while the latter is for any distributions. Given another foreground region F_j^b, its torso region A_j^b can also be modeled by K Gaussian distributions: $\mathcal{N}(\pi_{j,m}^b, \mu_{j,m}^b, \Sigma_{j,m}^b), m \in [1, K]$, where $\pi_{j,m}^b$, $\mu_{j,m}^b$ and $\Sigma_{j,m}^b$ are the weight, mean and covariance of the *m-th* Gaussian distribution respectively.

A cross matching method can be used to calculate the Mahalanobis distance between A_i^a and A_j^b. Since the Gaussian distributions in each GMM are ranked in a descending order, the first distribution is always the dominant distribution in the GMM. Ideally, only the dominant distribution from each GMM should be involved in the colour matching. However, it is often necessary to consider some non-dominant distributions in the colour matching, when the underlying torso region has a lack of a dominant colour or its dominant colour is partly occluded by a non-dominant

colour in another camera view. An example for the former scenario is a textured T-shirt. An example of the latter scenario is a red T-shirt in one camera view, which is partly occluded by an arm in the other view. The distributions used in the cross matching are decided by the weights of the individual distributions which must be above a threshold. For the torso region A_i^a, the Gaussian distributions involved in the cross matching, which are called significant distributions, are represented by a set:

$$N_i^a = arg \min_{n \in [1,K]} \{\pi_{i,n}^a : \pi_{i,n}^a \geq T_g\} \tag{6.19}$$

The colour matching between the torso region A_i^a in camera view a and the torso region A_j^b in camera view b is carried out in three steps. In the first step, the Mahalanobis distances between the dominant distribution $\mathcal{N}(\pi_{i,1}^a, \mu_{i,1}^a, \Sigma_{i,1}^a)$ of A_i^a and all the significant distributions of A_j^b, $\mathcal{N}(\pi_{j,m}^b, \mu_{j,m}^b, \Sigma_{j,m}^b), m \in [1, N_j^b]$, are calculated:

$$c_{i,j,m}^a = (\mu_{i,1}^a - \mu_{j,m}^b)^{\mathrm{T}} (\Sigma_{i,1}^a + \Sigma_{j,m}^b)^{-1} (\mu_{i,1}^a - \mu_{j,m}^b) \tag{6.20}$$

In the second step, the Mahalanobis distances between the dominant distribution of A_j^b and all the significant distributions of A_i^a are calculated. The result is denoted as $c_{i,j,n}^b, n \in [1, N_i^a]$. Then the Mahalanobis distances between A_i^a and A_j^b is a combination of $\mathrm{c}_{i,j}^a = \{c_{i,j,m}^a\}_{m \in [1,N_j^b]}$ and $\mathrm{c}_{i,j}^b = \{c_{i,j,n}^b\}_{n \in [1,N_i^a]}$:

$$\mathrm{c}_{i,j}^{a,b} = \mathrm{c}_{i,j}^a \cup \mathrm{c}_{i,j}^b \tag{6.21}$$

where $\mathrm{c}_{i,j}^{a,b} = \{\mathrm{c}_{i,j,k}^{a,b}\}_{k \in [1,L]}$; the number of the Mahalanobis distances L is $(N_i^a + N_j^b - 1)$.

Then, the minimum value in $\{\mathrm{c}_{i,j,k}^{a,b}\}_{k \in [1,L]}$ is thought of as the colour distance between the pair of colour appearance models:

$$\mathrm{c}_{i,j}^{a,b} = \min_{k \in [1,L]} (\mathrm{c}_{i,j,k}^{a,b}) \tag{6.22}$$

The smaller $\mathrm{c}_{i,j}^{a,b}$ is, the more likely that the two foregrounds are from the same object. This is built on the assumption that the two foreground regions F_i^a and F_j^b are visible or partly visible in both camera views. If the two foreground regions can be visible in the two camera views, a high $\mathrm{c}_{i,j}^{a,b}$ indicates these two foreground regions are due to different objects. When either object is fully occluded in one camera view, the colour of that object is lost and is replaced by the colour of another

object which occludes that object. Then, the colour matching will lead to a wrong result. Therefore, the colour matching should be combined with a height matching in the phantom removal.

6.4 Phantom Removal Based on Heights and Colours

Since height matching is uncertain for adjacent pedestrians and colour matching cannot handle occluded pedestrians, they are combined to improve the robustness of the foreground intersection classification. Figure 6.2 shows a flowchart of the proposed phantom removal algorithm based on both heights and colours. Firstly, the foreground regions detected in each camera view are warped into a virtual top view according to the homography mapping for a plane at some height. The intersection regions indicate all the possible regions that contain real objects or phantoms. By assuming the pedestrians are standing upright, the intersection regions can be thought as the positions that objects touch the ground plane.

As the matching is carried out in each camera view, the intersection regions in the top view are warped back to the individual camera views according to the ground-plane homography. For each foreground region in a camera view, the warped back patches corresponding to the same foreground region are grouped into a patch set for that foreground region. Height matching and colour matching are applied successively to identify whether each warped back patch in the patch set can match that foreground region.

The height matching is based on the position analysis between each foreground region and the warped back patches corresponding to that foreground region. The position analysis is derived from the observation that if an intersection region of the foreground projections from different camera views contains a real object, the warped back patch of that intersection region by using the ground-plane homography will be located at the foot area of that foreground region. If more than one warped back patches are matched to the same foreground region in the height matching, they are further classified in the colour matching, this differs from the method presented in the previous chapter.

The colour matching is based on the Mahalanobis distance between the colours of a pair of foreground regions each from a different camera view and intersecting in the top view. After a position analysis is applied to the patch classification in a single view, the classification results from both camera views are integrated to classify the foreground intersection regions in the top view.

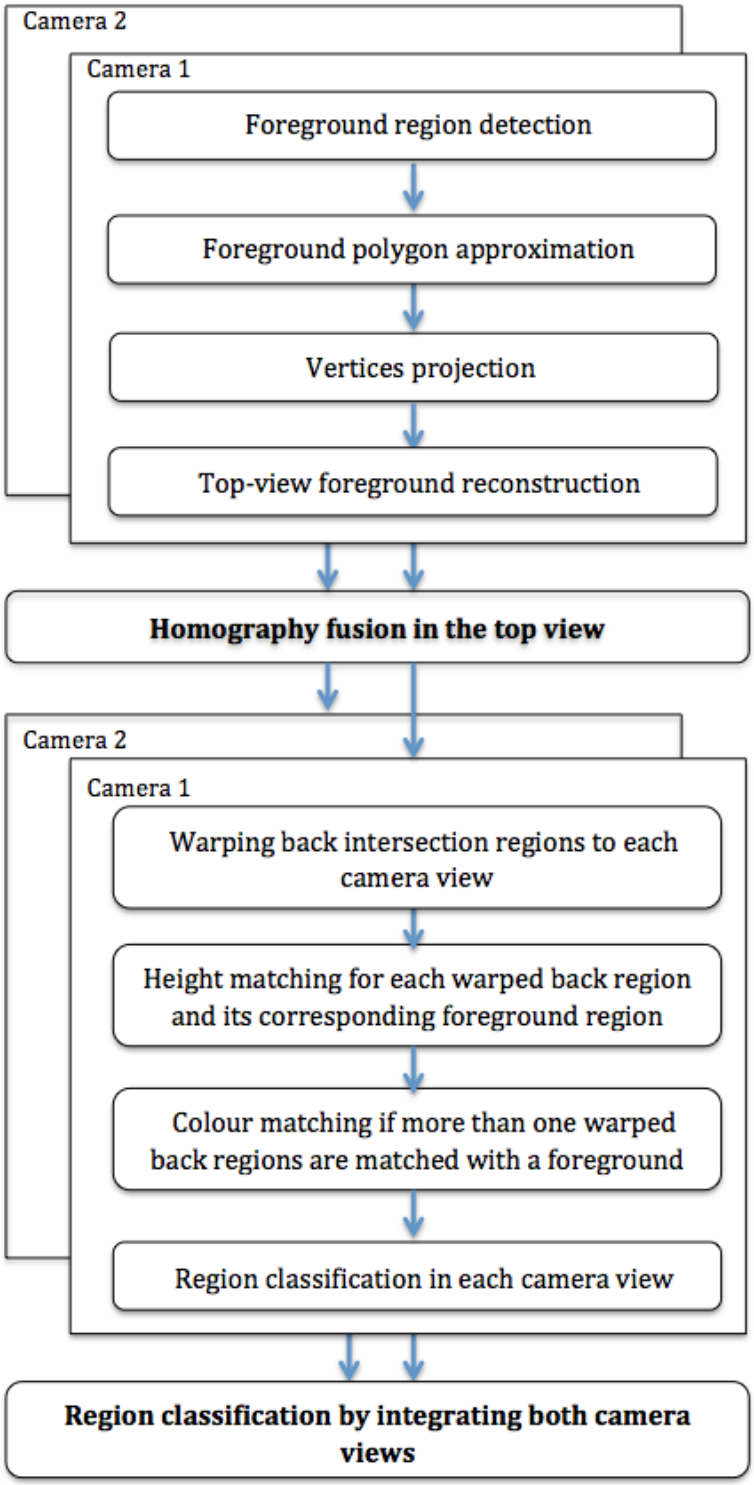


Figure 6.2 A flowchart of the proposed phantom removal algorithm based on heights and colours.

6.4.1 Height Matching

In the height matching method, the normalized distances of a warped back patch set are $\{d_{i,j}^a\}_{j\in[1,J]}$ for the *i*-th foreground region in camera view *a*. The number J_i^a of the patches which have normalized distances within a threshold is:

$$J_i^a = \#\{P_{i,j}^a : |d_{i,j}^a| \le Th_d, \quad j \in [1,J]\} \tag{6.23}$$

Based on the value of J_i^a, the height matching can be divided into three pathways. If $J_i^a = 1$, there is only one matched patch in $(P_{i,j}^a)_{j\in[1,J]}$; the normalized distance $d_{i,j}^a$ of that matched patch $P_{i,j}^a$ is selected as the matched height of the foreground region F_i^a and $d_i^a = d_{i,j}^a$. The matched height will be used to decide upper patches and lower patches. If $J_i^a = 0$, the matched height of foreground region F_i^a is set to zero and $d_i^a = 0$. If $J_i^a > 1$, the J_i^a patches will be further classified in the colour matching.

6.4.2 Colour Matching

The Mahalanobis distance between the appearance models of a pair of foreground regions reflects the likelihood that these two foreground regions, each in a different camera view, are coming from the same object. For a set of warped back patches $(P_{i,j}^a)_{j\in[1,J_i^a]}$ in camera view *a*, the Mahalanobis distances of these patches are $(c_{i,j}^{ab})_{j\in[1,J_i^a]}$. The patch that has the least Mahalanobis distance is identified as the matched patch.

$$j_{min} = \arg\min_{j\in[1,J_i^a]} (c_{i,j}^a) \tag{6.24}$$

Then, its normalized distance $d_{i,j_{min}}^a$ is used as the matched height of F_i^a.

6.4.3 Patch Classification in a Single View

Position analysis is applied to the patch classification in a single view. In the height matching and colour matching, for a patch set of a foreground region in the camera view, the object patch (**Op**) which matches the foreground region is identified and the matched height is determined. The normalized distances of the other patches in the patch set are compared with the matched height of the object patch to decide whether these patches are above or below the object patch in that camera view. If the normalized distance of a patch is greater than the matched height, then that

patch is identified as an upper patch (**Up**), which corresponds to an intersection region behind that for an object. If the normalized distance of a patch is less than the matched height, then that patch is identified as a lower patch (**Lp**), which corresponds to a foreground intersection region in front of that for an object. In this way, the patches for each foreground region are classified into the object patch, upper patches and lower patches.

6.4.4 Region Classification in the Top View

The matching and classification results of the warped back patches in the individual camera views are integrated to identify whether each intersection region in the top view contains an object or not. Table 6.1 summarized the classification of the intersection regions in the top view.

Table 6.1 The classification of the intersection regions in the top view.

Camera View *a* / Camera View *b*	Op	Up	Lp
Op	Ob	Oc	Ph
Up	Oc	Cv	Ph
Lp	Ph	Ph	Ph

The details of the phantom pruning algorithm using height matching and colour matching are described as Algorithm 3. Algorithm 2 only uses the nearest-neighborhood algorithm to identify the matched patch for each foreground region in the height matching. When two or more objects are very close to each other in one camera view, the warped back patches of these objects may be close to the feet of the same foreground region. In Algorithm 3, colour matching was combined with height matching when more than one warped back patches in a patch set are located at the foot area of a foreground region (see line 7-17 in **Algorithm 3**).

Algorithm 3: Phantom pruning using height matching and colour matching

1:	**for** all camera views **do**
2:	each intersection region $P_{i,j}^{t}$ of foreground projections in the top view are warped back to the camera view by using the homography for the ground plane;

3:	**for** all foreground regions in the camera view (using a as the index of the camera) **do**
4:	the patch set $\{P_{i,j}^a\}_{j\in[1,J]}$ of F_i^a is generated;
5:	**for** all the warped back patches in $\{P_{i,j}^a\}_{j\in[1,J]}$ **do**
6:	calculate the normalized distance $d_{i,j}^a$;
7:	count $J_i^a = \#\{P_{i,j}^a: \lvert d_{i,j}^a\rvert \le Th, j \in [1,J]\}$
8:	**if** $J_i^a = 0$ **then**
9:	the matched height $d_i^a = 0$;
10:	**else if** $J_i^a = 1$ **then**
11:	$d_i^a = d_{i,j}^a$;
12:	**else**
13:	**for** all $P_{i,j}^a$ that satisfies $\lvert d_{i,j}^a\rvert < Th,\ j \in [1, J_i^a]$ **do**
14:	calculate the Mahalanobis distance $c_{i,j}^{a,b}$ between A_i^a and A_j^b;
15:	**end for**
16:	Patch $P_{i,j_{min}}^a$, where $j_{min} = arg\min_{j\in[1,J_i^a]}(c_{i,j}^{a,b})$, is identified as the object patch for F_i^a; $d_i^a = d_{i,j_{min}}^a$;
17:	**end if**
18:	**If** $d_{i,j}^a == d_i^a$, **then** $P_{i,j}^a$ is labeled as Object Patch (Op);
19:	**else if** $d_{i,j}^a > d_i^a$, **then** $P_{i,j}^a$ is labeled as Upper Patch (Up);
20:	**else** $P_{i,j}^a$ is labeled as Lower Patch (Lp);
21:	**end if**
22:	**end for**
23:	**end for**
24:	Classification of foreground intersection regions based on integration of the results from all camera views.
25:	**end for**

6.5 Experimental Results

This chapter focused on the identification of the false-positive pedestrian detections from the intersection regions of the foreground projections, which is based on the homography mapping for a plane parallel to and one meter above the ground plane. The algorithms proposed to identify the false–positive detections were tested using the campus dataset. The experiments at an early stage are also discussed.

6.5.1 Phantom Removal with Colour Matching

The colour matching method uses the colours of foreground regions in the individual camera views to identify whether each intersection region in the top view is due to the same object or not. Figure 6.3 shows the procedure of the phantom removal algorithm using colour cues at frame 1320. Figure 6.3 (a)-(d) are the original images and the results of foreground detection in the two camera views. Although there are three pedestrians in each camera view, the pedestrian wearing a black jacket and the pedestrian wearing a blue jacket adhere to each other in Figure 6.3 (c), which is caused by the morphological closing operation in the foreground detection. The foreground regions are labelled with 1 and 2 in camera view *a* and a to c in camera view *b*. Figure 6.3 (e) shows the overlaid foreground projections from the two camera views to the top view with the homography for a plane at a height of one meter. The intersection regions of those foreground projections are shown in Figure 6.3 (f). Each intersection region in the top view is given a label corresponding to its parent foreground regions in the two camera views. Figure 6.3 (g) and (h) show the torso regions of the foregrounds in the two camera views. The torso regions are used to build the appearance models in the colour matching. The region at 55% to 80% of the height of each foreground region is thought of as the torso region, and the pixel colours in that region are used to build the appearance model.

To handle the multiple colours caused by textured patterns and adhering pedestrians, the pixels in each torso region are grouped into three clusters using the K-means algorithm and the EM algorithm. The colour clustering process was based on the RGB colour space. Figure 6.4 shows the clustering results in the two camera views at frame 1320. For the red torso region with label 1 in Figure 6.3 (g), the weights of the three clusters are 0.659, 0.2850 and 0.0555. The mean colours for these three clusters are shown in Figure 6.4(a)-(c). The over clustering of the red colour results from the sensitivity of the RGB colour space to illumination variations.

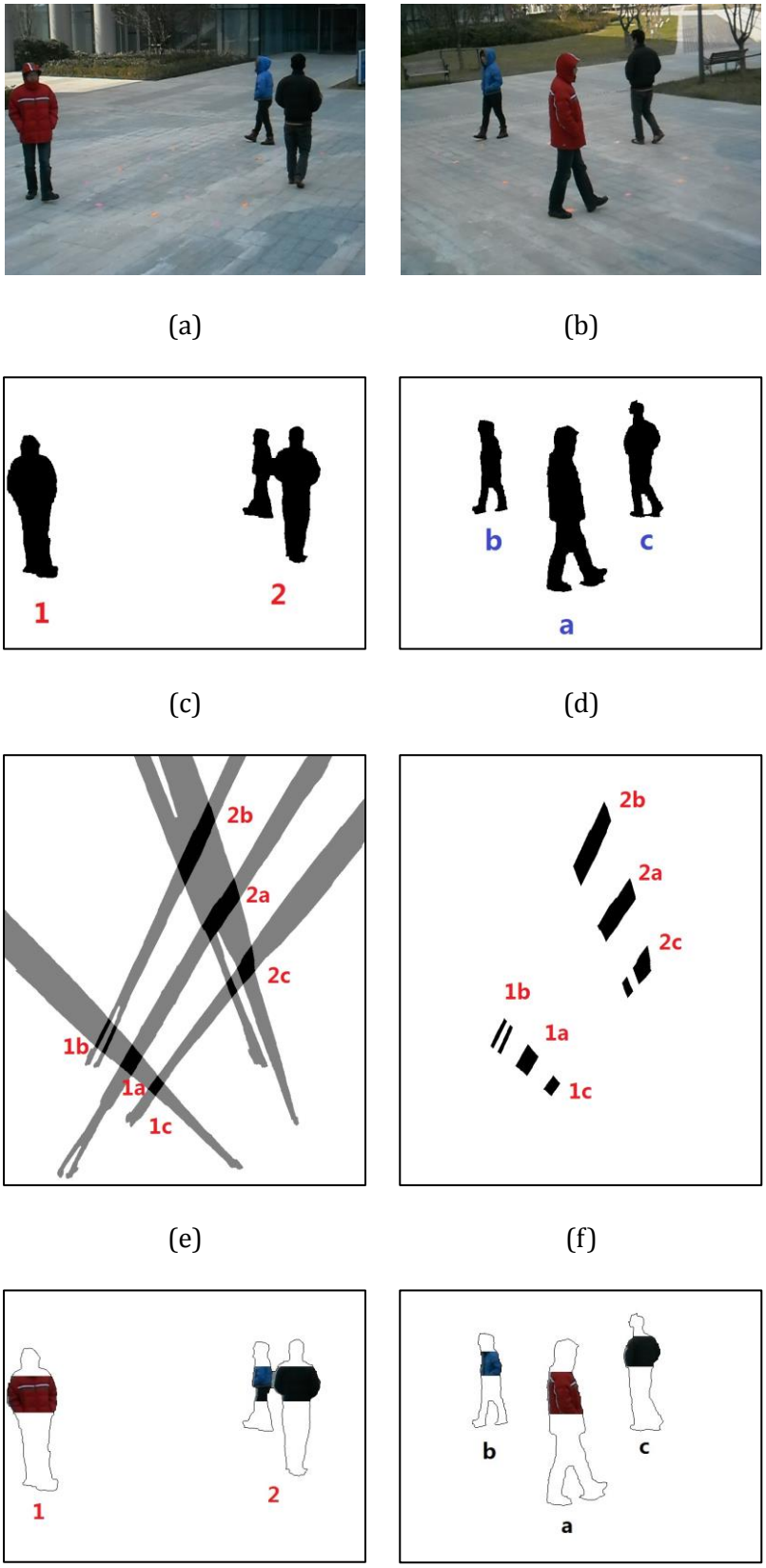
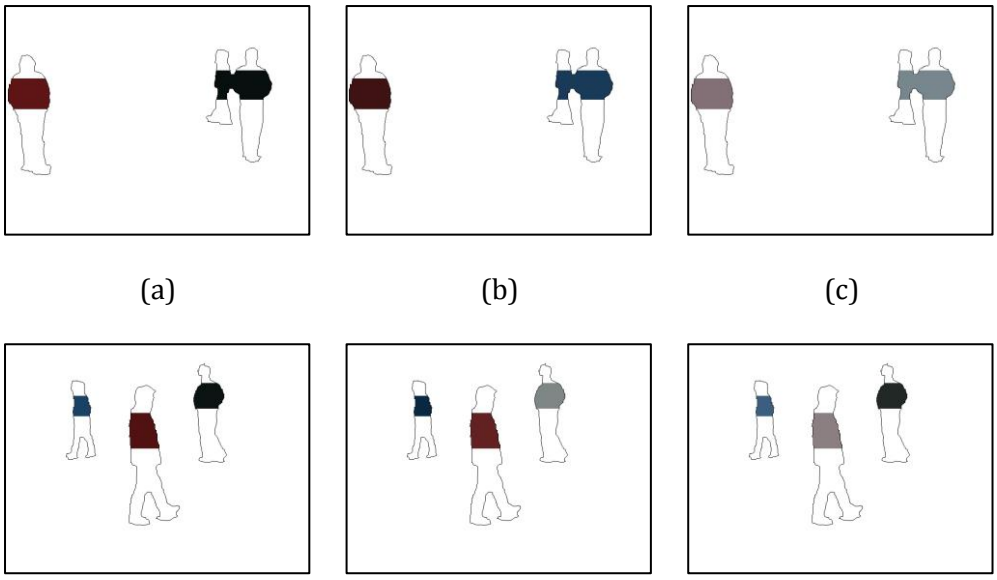


Figure 6.3 The process of the phantom removal algorithm using colour cues at frame 1320, (a)-(d) the original images and the foreground images in two camera views, (e) the overlaid foreground projections in the top view, (f) the intersection regions in the top view, and (g)(h) the torso regions in the two camera views.



(a) (b) (c)

(d) (e) (f)

Figure 6.4 The colour clustering results of the pedestrian with the red jacket at frame 1320, (a)(b)(c) the three clusters in camera view *a*, (a) the mean colour of the cluster with a weight 0.659, (b) the mean colour of the cluster with a weight 0.285, (c) the mean colour of the cluster with a weight 0.0555, and (d)(e)(f) the three clusters in camera view *b*.

To demonstrate the sensitivity to illumination, the mean values of the three clusters were transformed from the RGB space into the HSI space. Since these three clusters are red, the transformed means of these three clusters have similar hue values. Table 6.2 illustrates the clustering results of the pedestrian with a red coat in Figure 6.3 (g) in terms of the RGB colour space and the transformed values in the HSI colour space. It is noted that the hue component in the HSI colour space is very stable in the presence of illumination changes on the red coat, which is in contrast to the RGB space.

Table 6.2 The clustering results of the pedestrian with a red coat in Figure 6.3 (g) in terms of the RGB space and the transformed values in the HSI space.

Cluster	Weight	B	G	R	H	S	I
1	0.055	119.375	112.346	130.548	337.488	0.070	120.756
2	0.286	18.871	17.508	63.289	358.500	0.473	33.223
3	0.659	22.024	20.262	95.370	358.822	0.558	45.885
Average		26.529	24.588	88.177	358.493	0.470	46.400

Therefore, the colour clustering process is based on the hue component in the HSI colour space. Figure 6.5 shows the colour distributions of each torso region in the hue-and-saturation plane for the two camera views at frame 1320. Each green star represents the mean value of the hue components of each cluster and the saturation is set to the mean value of the saturation components for all the clusters. Table 6.3 illustrates the clustering results of the torso regions in both camera views on the basis of the hue component in the HSI colour space. The clustering results from the HSI colour space are much better than those from the RGB colour space.

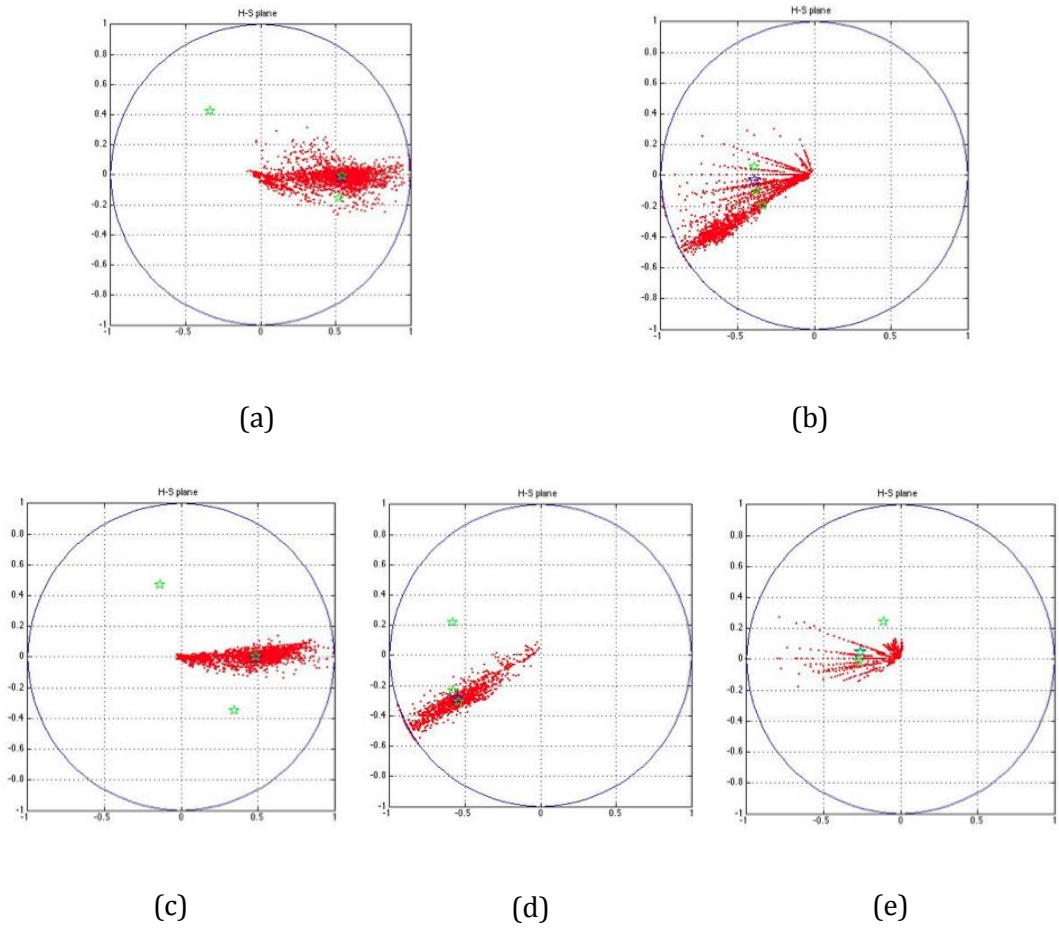


Figure 6.5 The colour distributions of the torso regions in the hue-and-saturation plane at frame 1320, (a) the red torso region in camera view *a*, (b) the blue and black torso region in camera view *a*, (c) the red torso region in camera view *b*, (d) the blue torso region in camera view *b*, and (e) the black torso region in camera view *b*.

When the appearance model of each foreground region in the individual camera views is built, using equations (6.19)-(6.22), the Mahalanobis distance of any two appearance models is calculated. As the number of the Gaussian distributions involved in the colour matching is determined by the weights of the

individual distributions which must be over the threshold T_g, the colour matching results of partly occluded objects or the smaller object in two adhering objects are influenced by T_g. For example, foreground region 2 in Figure 6.3 (c) corresponds to foreground regions b and c in Figure 6.3 (d). In Table 6.3, the mean value of the first distribution for foreground region 2 in camera view *a* is 171.895 which remains consistent with 171.001 which is the mean value of the first distribution for foreground region c in camera view *b*. For the first distribution for foreground region b in camera view *b*, the mean value is 209.562, which corresponds to the dominant colour of the blue torso region.

Table 6.3 The clustering results of the torso regions in both camera views, the data in bold indicates the hue value of each selected cluster in the colour matching.

		Cluster	1	2	3
Camera View *a*	Foreground Region 1	Weight	0.887	0.104	0.009
		Hue	**358.880**	338.416	109.766
	Foreground Region 2	Weight	0.571	0.264	0.165
		Hue	**171.895**	195.049	**209.978**
Camera View *b*	Foreground Region a	Weight	0.910	0.072	0.018
		Hue	**0.651**	354.926	264.298
	Foreground Region b	Weight	0.792	0.183	0.025
		Hue	**209.562**	202.956	162.632
	Foreground Region c	Weight	0.442	0.440	0.118
		Hue	**171.001**	183.395	113.805

In camera view *a*, the blue torso region adheres to the black torso region and only accounts for a small part of the whole "torso" region. Since the lower body of the pedestrian with the blue jacket is included in the torso region, the weight of the distribution corresponding to the blue jacket is low (0.165). Therefore, the threshold T_g should not be set very high to cope with partly occluded objects or adhering objects. In the following experiments, the threshold T_g was set to 0.10. The distributions which have weights over this threshold are involved in the cross matching. If the threshold T_g is set to 0.20, the distribution corresponding to the blue pedestrian in camera value *b* (hue = 209.562) will mismatch with the distribution corresponding to the black jacket in camera value *a* (hue = 171.895). If the threshold is set too low, the distribution which corresponds to noise will be

included in the matching. For example, the colour of arms for different pedestrians in two camera views will be matched. The threshold T_g is set empirical in this thesis.

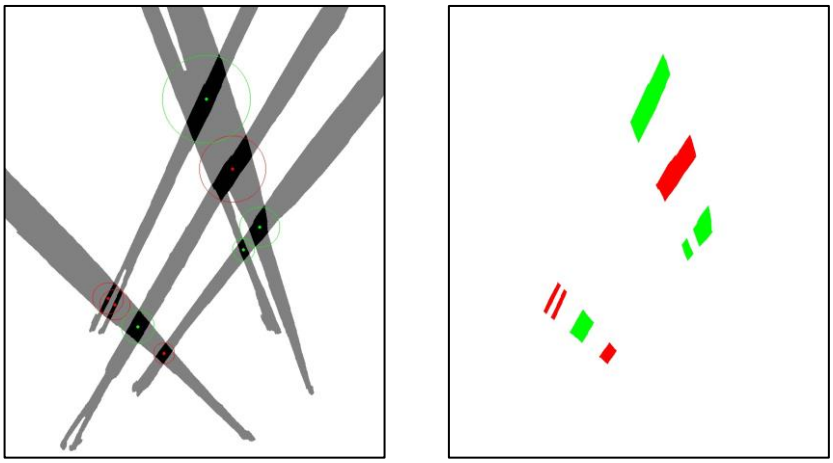
Since colour matching is carried out in each camera view, the intersection regions in the top view are warped back into the individual camera views to generate the patch set of each foreground region. Table 6.4 shows the colour matching results for the two foreground regions in camera view *a*. The results for camera view *b* are shown in Table 6.5. In each patch set, the patch which has the minimal Mahalanobis distance less than the threshold Th_c is identified as the matched patch. Since the Mahalanobis distances of the unmatched intersection regions are usually much greater then 1000, Th_c was set to 1000 in the experiments. The warped back patches 1a and 2b in Table 6.4 and the warped back patches a1, b2 and c2 in Table 6.5 are identified as the matched patches. The corresponding intersection regions 1a, 2b and 2c indicate the locations of the three pedestrians. To visualize the colour matching results, each intersection region in the top view is filled with a different colour, in which red indicates phantom regions and green indicates object regions.

Table 6.4 The colour matching results in camera view *a*, the data in bold indicates the matched foreground region in each patch set.

Foreground Region in Camera View *a*	Foreground Region in Camera View *b*	Mahalanobis Distance
1	**a**	**48.77**
	b	408345.00
	c	127256.00
2	a	291535.00
	b	**3.04**
	c	9.01

Table 6.5 The colour matching results in camera view *b*, the data in bold indicates the matched foreground region in each patch set.

Foreground Region in Camera View *b*	Foreground Region in Camera View *a*	Mahalanobis Distance
a	**1**	**48.76**
	2	291535.00
b	1	408345.00
	2	**3.04**
c	1	127256.00
	2	**9.01**



(a) (b)

Figure 6.6 Classification results of the intersection regions at frame 1320, (a) in the overlaid foreground projection image and (b) in the foreground intersection image.

The phantom removal algorithm using colour matching has been tested over the 142 sampled frames. The classification results are compared with the ground truth. Table 6.6 and Table 6.7 show the performance evaluation of the colour matching in comparison with the ground truth. The 786 intersection regions across the 142 frames are classified as either object regions or phantom regions.

Table 6.6 Classification results with colour matching (HSI) when compared with ground truth.

		Classification Results with Colour Matching		Number of the Ground Truth
		Object Regions (Ob)	Phantom Regions (Ph)	
Ground Truth	Object Regions (Ob)	350	10	360
	Phantom Regions (Ph)	6	420	426
Total number		356	430	786

For each category, the false negatives (missed detections) are the intersection regions which belong to that category but are misclassified as the other category. The false positives (false alarms) are the intersection regions which belong to the other category but are misclassified as the underlying category; The false negative rate (R_{FN}) is the ratio between the number of the false negatives and the ground-truth number. The false positives rate (R_{FP}) is the ratio between the number of the false positives and the number of that ground-truth number. The false negative rate and the false positive rate of the classification with the colour matching are shown in Table 6.7. The ground-truth number of object regions is 360, in which 350 are correctly identified. Since 10 object regions are misclassified as phantom regions, the false negative rate is 2.78%. There are 6 phantom regions which are misclassified as object regions. Therefore, the false positive rate is 1.67%.

Table 6.7 The false negative rate and the false positive rate of the classification with colour matching.

	False Negative Rate R_{FN} (%)	False Positive Rate R_{FP} (%)
Object Regions	2.78	1.67
Phantom Regions	1.41	2.35

During these experiments, 16 frames have been misclassified as intersection regions. Most of the misclassification cases in colour matching occur from the

following two scenarios. When one object is fully occluded by another object in one camera view, the colour of the occluded object is lost and replaced by the colour of the object in front of it in the colour matching. This gives rise to a false negative (missed detection).

When one object is partly occluded by another object or two objects adhere to each other because of the morphological operations in the foreground detection stage, the two objects are detected as a single foreground region. The appearance model of the occluded object or the object, which is of smaller size due to being further from the camera, in the two adhering objects is influenced by the position of these two objects. For the occluded object, if its visible portion is too small, then it may be excluded from the cross colouring matching, because its weight in the foreground region is less than T_g. For the adhering objects, as was discussed in the previous chapter, the object which is closer to the camera is located under the object which is farther from the camera, when the camera is viewing downwards. The height of the foreground region for these two objects is greater than that for a single object. Therefore, the foreground region from 55% to 80% of the height of that foreground region is not corresponding to the torso regions of those two objects. For example, the torso region of foreground region 2 in Figure 6.3 (g) contains two pedestrians. As the pedestrian with a blue jacket is located farther from the camera, the lower-body region is included in the "torso" region of the foreground.

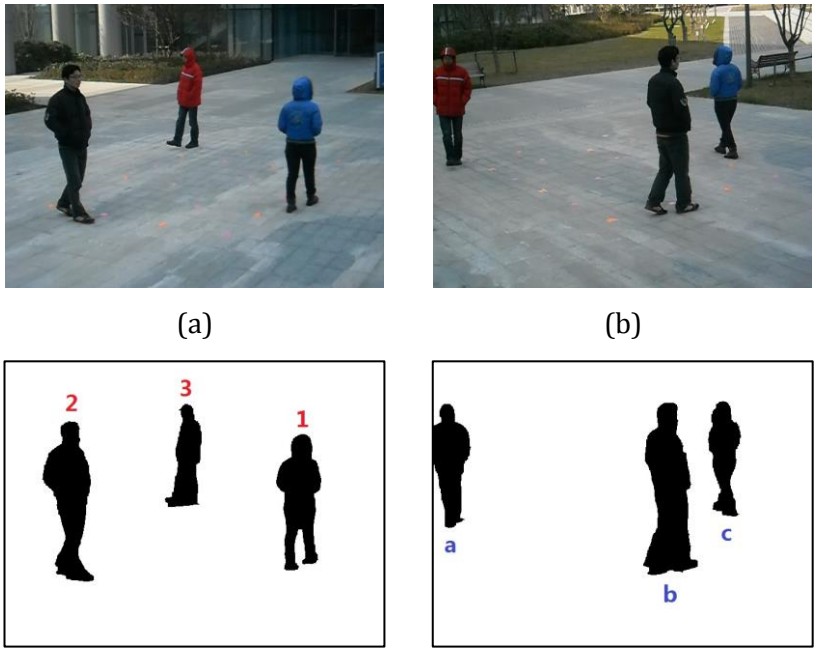
To evaluate the performance, the colour matching algorithm using RGB colour space was also tested. Table 6.8 shows the performance evaluation of the colour matching (RGB) in comparison with the ground truth. The false negative rate and the false positive rate of the classifications with colour matching using RGB space are higher than that using HSI space.

Table 6.8 The false negative rate and the false positive rate of the classification with colour matching using RGB space.

	False Negative Rate R_{FN} (%)	False Positive Rate R_{FP} (%)
Object Regions	3.89	2.22
Phantom Regions	1.88	3.29

6.5.2 Phantom Removal Based on Heights and Colours

Since the height matching algorithm has difficulties for adhering pedestrians and the colour matching algorithm cannot handle occluded pedestrians, they can be combined to improve the robustness of classification. Figure 6.7 shows the procedure of the phantom removal algorithm using the height matching and colour matching at frame 1200. Figure 6.7 (a)-(d) are the original images and the results of foreground detection in the two camera views. In each camera view, there are three pedestrians which are labelled with 1 to 3 in camera view *a* and are labelled with a to c in camera view *b*. Figure 6.7 (e) shows the overlaid foreground projections from the two camera views to the top view with the homography for a plane at a height of one meter. Their intersection regions in the top view are shown in Figure 6.7 (f). Figure 6.7 (g) and (h) are the warped back patches overlaid in the original camera views. Each intersection region or warped back patch is given a label to indicate the corresponding foreground regions in the two camera views. The torso regions in the two camera views are illustrated in Figure 6.7 (i) and (j).



(a) (b)

(c) (d)

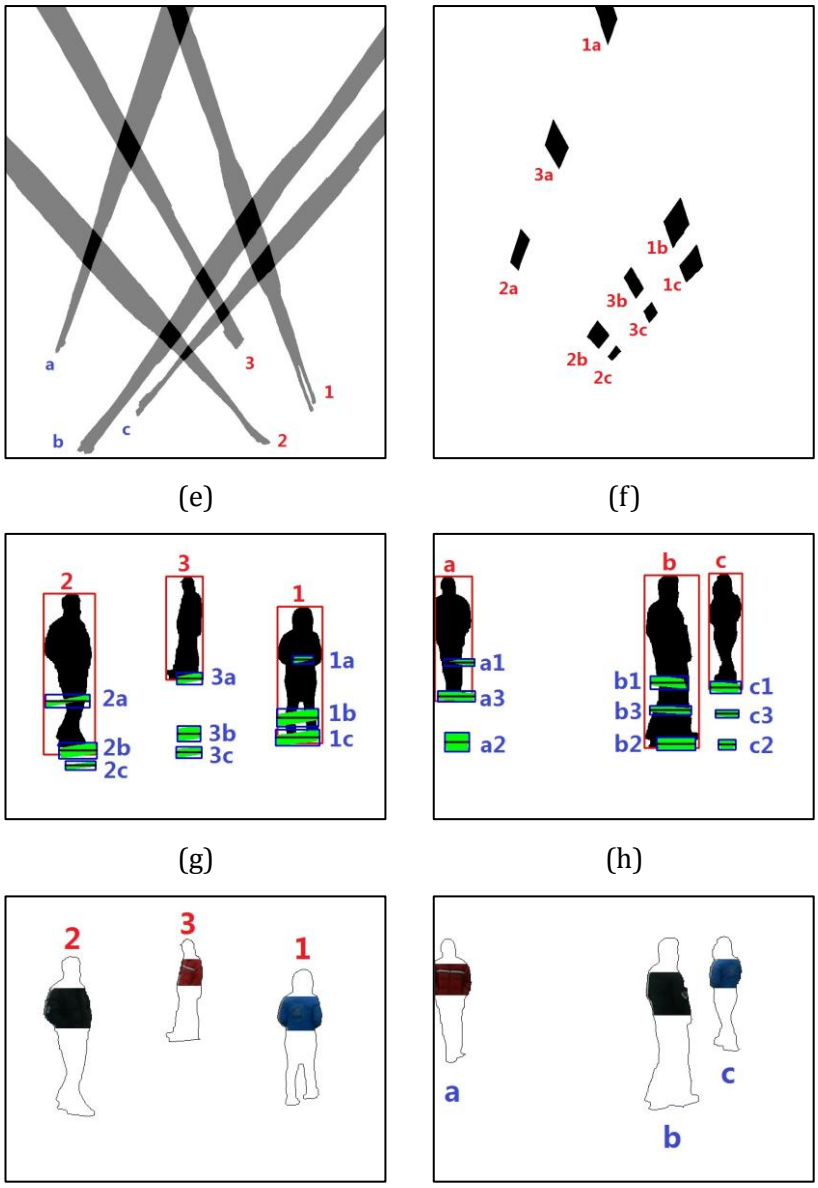


Figure 6.7 The process of phantom removal using the height matching and colour matching at frame 1200, (a)(b) the original images, (c)(d) the foreground regions in two camera views, (e) the overlaid foreground projections in the top view, (f) the intersection regions in the top view, (g)(h) the warped back patches in the two camera views, and (i)(j) the torso regions in the two camera views.

Table 6.9 and Table 6.10 show the results of the height matching and colour matching for the warped back patches in the two camera views. For foreground region 1 in camera view *a*, it is related to three warped back patches labelled with 1a, 1b and 1c. Patch 1c, which has the minimal normalized distance less than the threshold 0.1, is identified as an object patch. Patches 1a and 1b which have normalized distances larger than that for patch 1c are recognized as upper patches. For foreground region 3 in camera view *a*, patch 3a is identified as an object patch. Patches 3b and 3c are identified as lower patches because their normalize distances

are less than that for patch 3a. Since warped back patches 2b and 2c have the normalized distances less than the threshold 0.1, the colour matching is applied to further identify which may contain a real object. Then, patch 2b which has a lower colour distance is selected as the object patch of foreground region 2 in camera view *a*. The other patches in the two camera views can be classified using just the height matching.

Table 6.9 Height matching and colour matching at frame 1200 in camera view *a*, the data in bold indicates the matched foreground region and its corresponding normalized distance and colour distance in each patch set.

Foreground Region in Camera View *a*	Foreground Region in Camera View *b*	Normalized Distance	Colour Distance	Classification Result
1	a	0.613	3540780.00	Up
	b	0.187	11784.40	Up
	c	**0.043**	16.32	Op
2	a	0.332	460419.00	Up
	b	**0.026**	**22.41**	Op
	c	**-0.067**	4742.66	Lp
3	**a**	**0.012**	179.88	Op
	b	-0.523	499446.00	Lp
	c	-0.701	1.371490.00	Lp

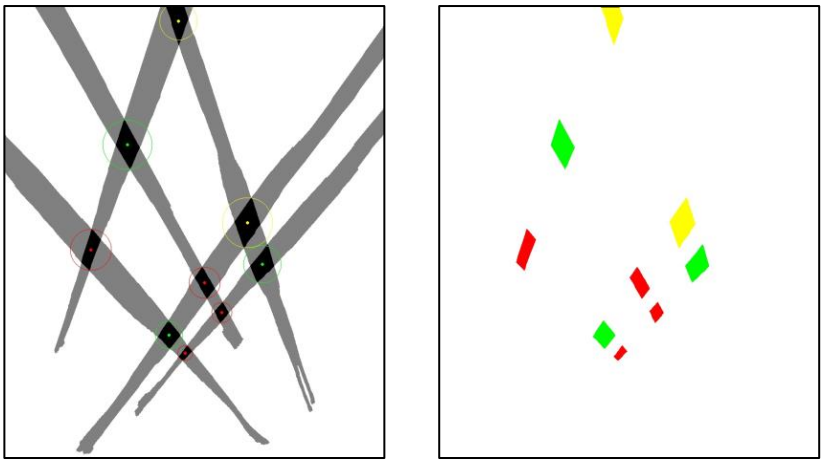
Table 6.10 Height matching and colour matching at frame 1200 in camera view *b*, the data in bold indicates the matched foreground region and its corresponding normalized distance and colour distance in each patch set.

Foreground Region in Camera View *b*	Foreground Region in Camera View *a*	Normalized Distance	Colour Distance	Classification Result
a	1	0.310	3540780.00	Up
	2	-0.329	460419.00	Lp
	3	**0.038**	179.88	Op
b	1	0.378	11784.40	Up
	2	**0.024**	22.41	Op
	3	0.220	499446.00	Up
c	1	**0.015**	16.32	Op
	2	-0.477	4742.66	Lp
	3	-0.210	1371490.00	Lp

The classification results in the two camera views are combined to make a final decision according to Table 6.1. Table 6.11 shows the classification results of the intersection regions, which are the same as those using the only height matching. To visualize the classification results, in Figure 6.8, each intersection region in the top view is filled with a different colour, in which red indicates phantom regions, green indicates object regions, yellow is for covered regions that are not be visible in both camera views and blue is for the regions that are occluded in one view but are visible in the other view.

Table 6.11 Classification results for the foreground intersections at frame 1200 using both height matching and colour matching.

Region	1a	1b	1c	2a	2b	2c	3a	3b	3c
Label	Cv	Cv	Ob	Ph	Ob	Ph	Ob	Ph	Ph



(a) (b)

Figure 6.8 Classification results of the intersection regions at frame 1200 using both height matching and colour matching, (a) in the overlaid foreground projection image and (b) in the foreground intersection image.

However, in some frames, the phantom removal using the height matching and colour matching provides more robust classification results than that only using the height. Figure 6.9 shows the procedure of the phantom removal algorithm using height matching and colour matching at frame 2115. Figure 6.9 (a)-(d) are the original images and the results of foreground detection in the two camera views. In each camera view, there are three pedestrians which are labelled with 1 to 3 in camera view *a* and are labelled with a to c in camera view *b*. Figure 6.9 (e) shows the overlaid foreground projections from the two camera views to the top view with the homography for a plane at a height of one meter. Their intersection regions in the top view are shown in Figure 6.9 (f). Figure 6.9 (g) and (h) are the warped back patches overlaid in the original camera views. Each intersection region or warped back patch is given a label to indicate the corresponding foreground regions in the two camera views. The torso regions in the two camera views are illustrated in Figure 6.9 (i) and (j).

(a) (b)

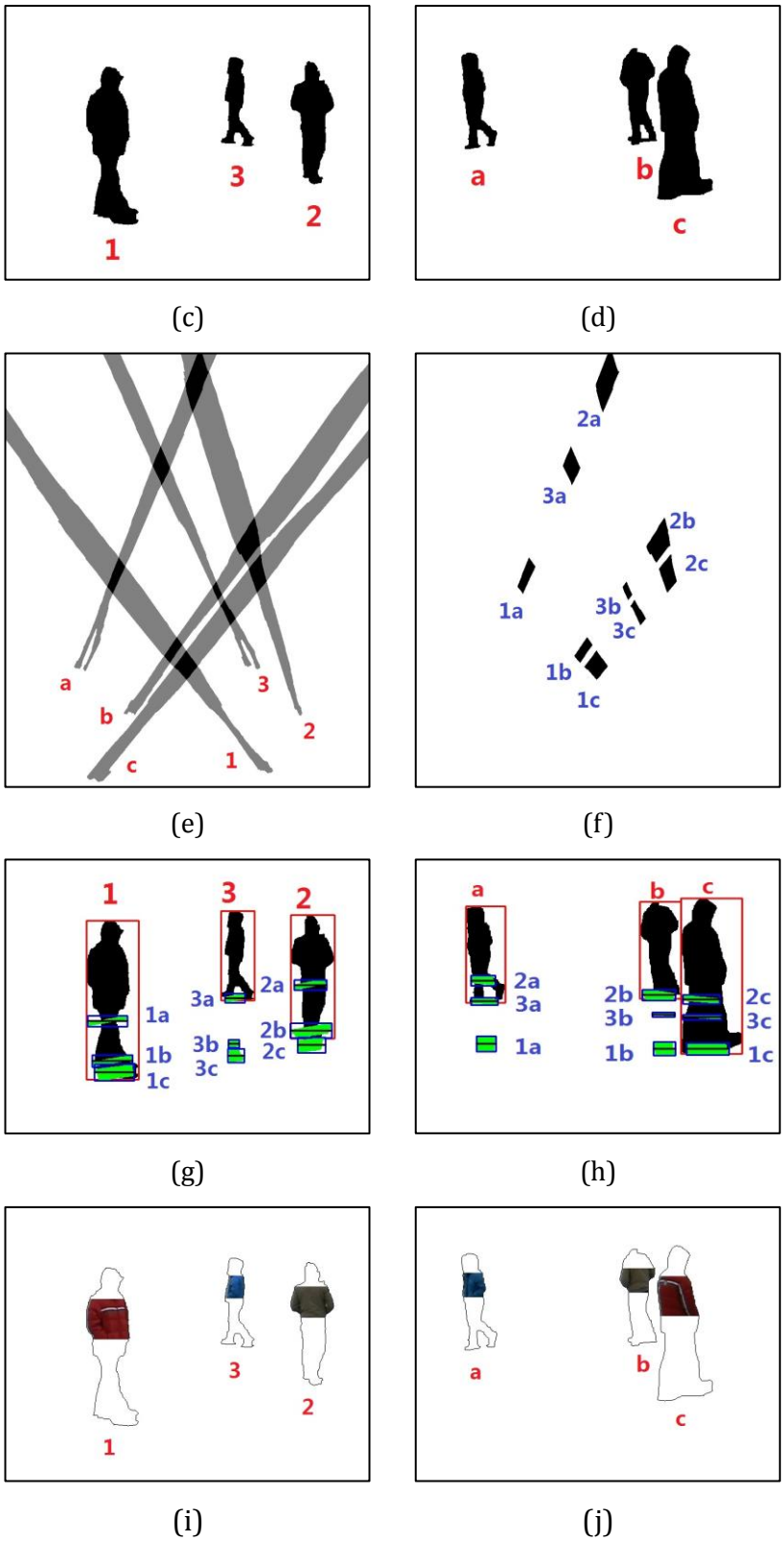


Figure 6.9 The process of the phantom removal using height matching and colour matching at frame 2115, (a)(b) the original images, (c)(d) the foreground regions in two camera views, (e) the overlaid foreground projections in the top view, (f) the intersection regions in the top view, (g)(h) the warped back patches in the two camera views, and (i)(j) the torso regions in two camera views.

Table 6.12 Height matching and colour matching at frame 2115 in camera view *a*, the data in bold indicates the matched foreground region and its corresponding normalized distance, colour distance and classification result in each patch set.

Foreground Region in Camera View *a*	Foreground Region in Camera View *b*	Normalized Distance	Colour Distance	Classification Result with Height and Colour Matching	Classification Result with Height Matching
1	a	0.369	653579.00	Up	Up
	b	0.118	22646.50	Up	Up
	c	**0.047**	38.21	Op	Op
2	a	0.433	200528.00	Up	Up
	b	**0.065**	**1.51**	**Op**	Up
	c	**-0.050**	15675.20	Lp	**Op**
3	a	**0.025**	35.27	Op	Op
	b	-0.484	250558.00	Lp	Lp
	c	-0.618	291744.00	Lp	Lp

Table 6.12 and Table 6.13 show the results of the height matching and colour matching for the warped back patches in the two camera views. For foreground region 1 in camera view *a*, it is related to three warped back patches labelled with 1a, 1b and 1c. Patch 1c, which has the minimal normalized distance less than the threshold 0.1, is identified as an object patch. Patches 1a and 1b which have normalized distances larger than that for patch 1c are recognized as upper patches. For foreground region 3 in camera view *a*, patch 3a is identified as an object patch. Patches 3b and 3c are identified as lower patches. The normalized distances of warped back patches 2b and 2c are less than the threshold 0.1. Then, patch 2c is a lower patch which is correctly identified using the additional colour matching because the colour distance of that patch is higher (15675.20). Patch 2b is identified as the object patch using the height and colour matching, which remains consistent with the ground truth. For the height matching using the nearest-neighbourhood algorithm, patch 2c (-0.050) is classified as the object patch in the height matching because its absolute normalized distance is lower than that for patch 2b (0.065) and patch 2b is classified as the upper patch. Foreground detection errors, homography estimation errors and very closed objects are the main causes of misclassification in the height matching. Foreground detection errors, homography estimation errors and two or more very closed objects are the main causes of misclassification in the

height matching. The other patches in the two camera views can be classified using the height matching only.

Table 6.13 Height matching and colour matching at frame 2115 in camera view *b*, the data in bold indicates the matched foreground region and its corresponding normalized distance, colour distance and classification result in each patch set.

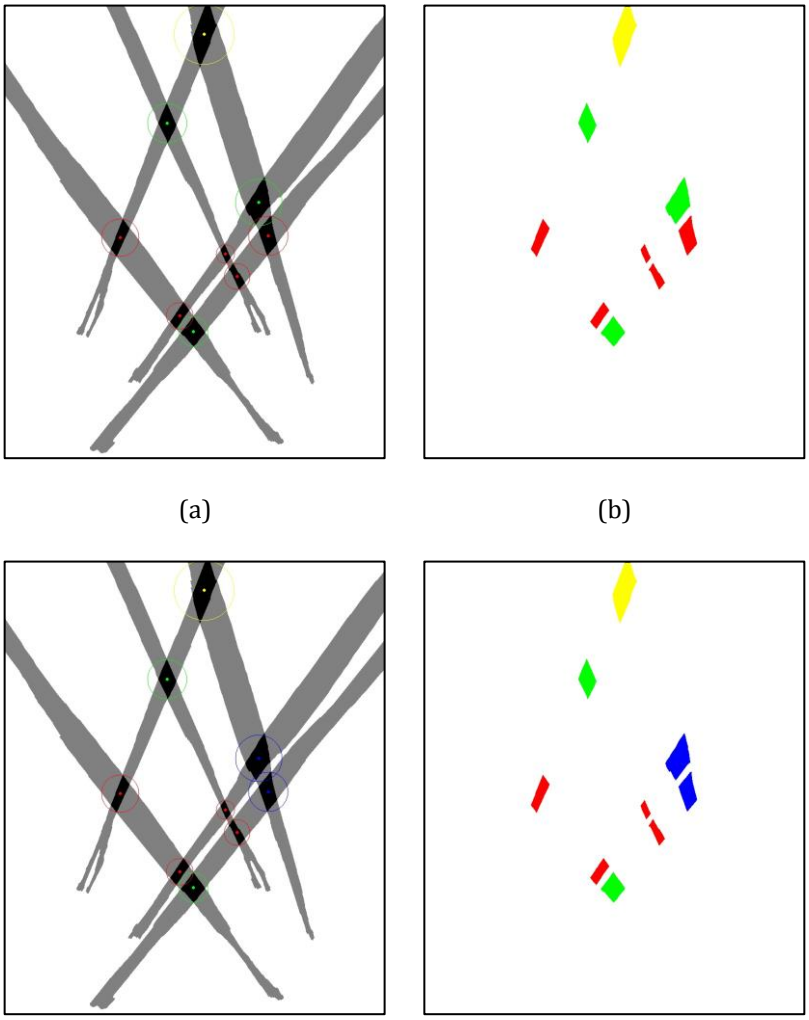
Foreground Region in Camera View *b*	Foreground Region in Camera View *a*	Normalized Distance	Colour Distance	Classification Result with Height and Colour Matching	Classification Result with Height Matching
a	1	-0.424	653579.00	Lp	Lp
	2	0.229	200528.00	Up	Up
	3	**0.018**	35.27	Op	Op
b	1	-0.515	22646.50	Lp	Lp
	2	**0.041**	1.51	Op	Op
	3	-0.164	250558.00	Lp	Lp
c	1	**0.033**	38.21	Op	Op
	2	0.350	15675.20	Up	Up
	3	0.237	291744.00	Up	Up

Table 6.14 Classification results of the foreground intersections at frame 2115 using both the height matching and colour matching and only the height matching.

Region	1a	1b	1c	2a	**2b**	**2c**	3a	3b	3c
Label with Height and Colour Matching	Ph	Ph	Ob	Cv	**Ob**	**Ph**	Ob	Ph	Ph
Label with Height Matching	Ph	Ph	Ob	Cv	**Oc**	**Oc**	Ob	Ph	Ph

The classification results in the two camera views are combined to make a final decision according to Table 6.1. Table 6.14 shows the classification results of the intersection regions using both the height matching and colour matching and the height matching only. Intersection regions 2b and 2c are misclassified as the occluded regions using the height matching. The classification results of these two regions using both the height matching and colour matching are correct. To

visualize the classification results, each intersection region in the top view is filled with a different colour in Figure 6.10.



(a) (b)

(c) (d)

Figure 6.10 Classification results of the intersection regions at frame 2115 using both height matching and colour matching and only the height matching, (a) (b) visualized classification results in the overlaid foreground projection image and in the foreground intersection image using both height matching and colour matching and (c)(d) visualized classification results in the overlaid foreground projection image and in the foreground intersection image using height matching.

The phantom removal algorithm by using height matching and colour matching has been tested over the 142 sampled frames. Table 6.15 and Table 6.16 show the performance evaluation of the phantom removal algorithm by using height matching and colour matching. The classification results are compared with ground truth. The 786 intersection regions from 142 frames are classified into four categories: object regions, phantom regions, covered regions and occluded regions.

Table 6.15 Performance evaluation of the classification using the height matching and colour matching.

		Classification Results with Height and Colour Matching				Number of the Ground Truth
		Object Regions (Ob)	Phantom Regions (Ph)	Covered Regions (Cv)	Occluded Regions (Oc)	
Ground Truth	Object Regions (Ob)	307	0	10	2	319
	Phantom Regions (Ph)	0	309	5	0	314
	Covered Regions (Cv)	0	0	112	0	112
	Occluded Regions (Oc)	0	0	0	41	41
Total number of Classification		307	309	127	43	786

Table 6.16 The classification errors with height matching and colour matching.

	False Negative Rate R_{FN} (%)	False Positive Rate R_{FP} (%)
Object Regions	3.76	0.00
Phantom Regions	1.59	0.00
Covered Regions	0.00	13.39
Occluded Regions	0.00	4.88

In Table 6.15, the confusion matrix of the classification results is given, along with the ground truth measurement. Using equation (5.5), the false negative rate and the false positive rate of each category were calculated and the results are shown in Table 6.16. The ground-truth number of object regions is 319, where 307 are correctly identified. 10 object regions are misclassified as covered regions

because pedestrians in these object regions cannot be visible in both camera views. 2 object regions are misclassified as occluded regions. Since no region is misclassified as an object region, the false negative rate is 3.76% and the false positive rate is 0.00%.

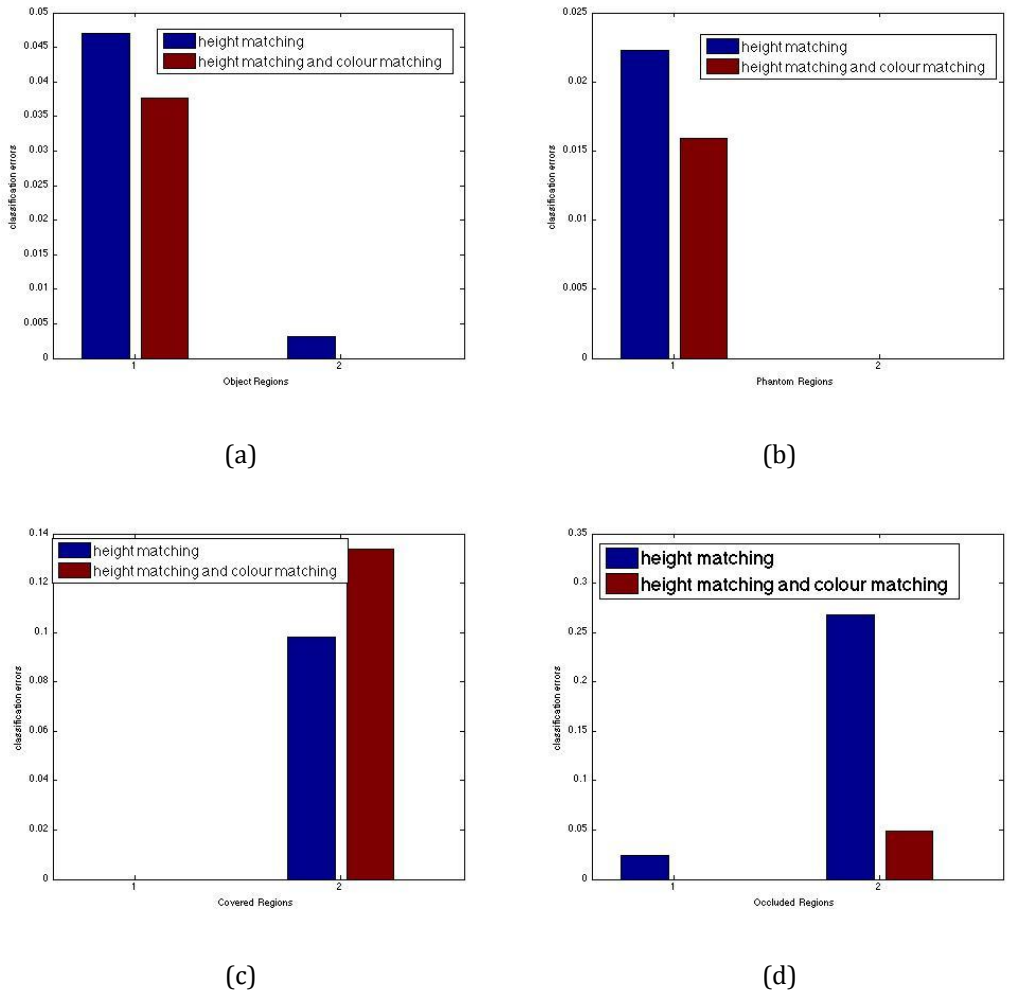


Figure 6.11 Comparison of classification errors between height matching and height and colour matching, (a) Object Regions, (b) Phantom Regions, (c) Covered Regions and (d) Occluded Regions.

To evaluate the classification errors, the height matching and colour matching algorithm has been compared with the height matching algorithm proposed in the previous chapter. Figure 6.11 shows the comparison results using Table 5.9 and Table 6.15. In each subfigure, "1" in the horizontal axis corresponds to the False Negative Rate and "2" illustrates the false positive rate. Each blue bar shows the result of height matching and the red bar represents the result for the height and colour matching. Figure 6.11(a)(b)(c)(d) correspond to the false negative rate and false positive rate of the object regions, phantom regions, covered regions and occluded regions respectively. Except the false positive rate of the covered regions

in the height and colour matching is higher than that for the height matching, the errors of the height matching are higher than that for the height and colour matching, which means that the height and colour matching provides more accurate classification results.

The computation costs of the height matching, colour matching and height and colour matching are shown in Table 6.17. Each algorithm has been tested over the 142 sampled frames. After generating the intersection regions in the top view, the total time to classify each intersection regions in the 142 sampled frames with different algorithms are listed in Table 6.17. The height matching method needs 0.11 s to process 1 frame, which is the fastest method of the proposed three algorithms. The computation cost of the height and colour matching is slightly higher than the height matching (0.16 s per frame). The colour matching runs slowly, which needs 0.47 s to process 1 frame.

Table 6.17 Computation costs of the height matching, colour matching and height and colour matching.

	Height Matching	Colour Matching	Height and Colour Matching
Total Time (s)	15.2	67.44	22.83
Time per Frame (s)	0.11	0.47	0.16

6.5.3 Discussions

In the initial research, to handle the over clustering problem, the covariance of the pixel colours in each torso region was used to decide whether a region needs to be divided into multiple clusters. The process was as follows:

1. Given a region, calculate the covariance of the pixel colours in that region.
2. If the covariance is greater than a threshold, that region is decided to have multiple colour distributions. The number of clusters is set to 3.
3. Using the K-means algorithm and the EM algorithm, the colours of that region are clustered into 3 clusters. Then, the mean and covariance of each cluster are calculated.

4. If the mahalanobis distance of any two clusters is less than a threshold, the two clusters are merged. Steps 3 and 4 are repeated until no two cluster can be merged.

This process can partly control the over-clustering using the RGB colour space.

6.6 Summary

In this chapter, a colour-matching based phantom removal algorithm was initially proposed. The appearance model of each foreground region was built on the colour clustering results with the hue component. The Mahalanobis distance was used to measure the colour similarity of every two appearance models. Given each intersection region in the top view, the colour matching result is determined by the Mahalanobis distance of the torso regions from the two corresponding foreground regions. From the experiential results, the phantom removal algorithm based on the colour matching can provide an acceptable classification results. The false negative rate and the false positive rate of the classifications with colour matching using RGB space are higher than that using HSI space. The limitation of the colour matching is that the threshold T_g is set empirical. If T_g is too low, the distribution which corresponds to noise will be included in the matching. For example, the colour of arms for different pedestrians in two camera views will be matched. However, the colour matching for phantom removal cannot handle the scenarios when objects are occluded in one camera view. Therefore, colour matching was combined with height matching in the phantom removal. From the experiential results, the phantom removal algorithm based on the height matching and the colour matching is more robust than that based only on the height matching. However, there is a slightly increased computational cost (0.05 s per frame). The limitation of the proposed colour matching is that the black and white regions are not modelled properly by HSI, which should be overcome in the future.

7 CONCLUSIONS AND FUTURE WORK

In this thesis, robust moving object detection using information fusion from multiple cameras has been investigated. To accelerate the homography transformation of foreground regions from the camera view to the top view, each foreground region is represented by a polygon and only the polygon vertices are transformed. In order to identify the intersection regions of no-corresponding foreground projections in the top view, a height matching algorithm based on the geometry between the individual camera views and a colour matching algorithm based on the Mahalanobis distance between the colour statistics of the two foreground regions are proposed. To improve the robustness of classification, the height matching and the colour matching are combined in the foreground intersection classification.

According to the experimental results in this thesis, the following conclusions are drawn:

- Unlike the pixelwise homography mapping which projects binary foreground images, the contour of each foreground region is approximated with a polygon and only the polygon vertices are transmitted and projected. The foreground projections are rebuilt according to the projected polygons in the reference view. The experimental results have shown that this method can be run in real time and generate results similar to those using foreground images.
- A phantom removal algorithm based on colour matching has been developed to identify the intersection regions of no-corresponding foreground projections in the top view. After using the K-means algorithm and the EM algorithm to build the appearance model of each foreground region, the Mahalanobis distance of the two appearance models was used to represent the likelihood that these two foreground regions are due to the same object. However, the results of the colour matching cannot handle occluded objects in one camera view.

- A height matching algorithm based on the geometrical information was proposed to identify whether an intersection region is due to the foreground regions of the same object. The intersection regions are classified into four categories. The intersection regions which cannot be visible in both camera views are identified as covered regions. Experimental results have shown that this algorithm can robustly classify the intersection regions in the top view. However, when two or more objects are adjacent to each other in one camera view, the warped back patches of these objects may be close to the feet of the foreground region simultaneously, which brings about uncertainty in the classification. Furthermore, the foreground segmentation error is assumed to be relatively low. A high foreground segmentation error needs a high threshold Th_d in the height matching, which leads to more misclassifications.
- Height matching and colour matching were combined to improve the robustness of the foreground intersection classification. The robustness of this algorithm has been illustrated by experiments. To evaluate the classification errors, the height and colour matching algorithm has been compared with the height matching algorithm. Except the false positive rate of the covered regions in the height and colour matching is higher than that for the height matching, the errors of the height matching are higher than that for the height and colour matching, which means that the height and colour matching provides more accurate classification results. The computation cost of the height and colour matching is higher than the height matching (0.16 s per frame). The colour matching is very slow, and needs 0.47 s to process 1 frame.

Although the proposed methods achieve promising results with respect to the classification of false-positive detections in the top view, some aspects need to be further investigated:

- In this research, only two cameras are used in the phantom removal algorithm. When the number of pedestrians increases, phantoms will be generated more frequently. However, the pedestrians are more likely to be grouped in the foreground detection. The proposed method cannot work reliably when the pedestrians are grouped in both camera views. In this situation more cameras should be considered. In future, this algorithm can be extended to include more than two cameras. Then, the experimental results can be compared with works proposed in [59] and [58].

- Currently, the classification results of the warped back patches in the individual camera views are integrated according to Table 6.1. Some machine learning methods can be applied to the classification.
- Since the foreground detection in a single view is not the main focus of this thesis, the foreground segmentation error is assumed to be relatively low. When the foreground segmentation error rate is high, the classes from different cameras which are grouped using a different classification approach. For example, the intersection region, which corresponds to an upper patch in one camera view and a lower patch in another camera view, can be classified as an object region. In this thesis, this kind of intersection regions is classified as phantom regions.
- In the colour matching, only the hue component is used. In future, the intensity and the saturation components in the HIS colour space can be used simultaneously to cope with black or dark regions.

APPENDIX

A. 1 Publication List

[1] M. Xu, J. Ren, D. Chen, J. Smith, and G. Wang, "Real-time detection via homography mapping of foreground polygons from multiple cameras," in *Proceedings of the IEEE International Conference on Image Processing*, 2011, pp. 3593-3596.

[2] M. Xu, J. Ren, D. Chen, J. S. Smith, and Z. Liu, "A Multiview Approach to Robust Detection in the Presence of Cast Shadows," in *Proceedings of the International Conference on Image and Graphics*, 2011, pp. 494-499.

[3] J. Ren, M. Xu, and J. S. Smith, "Pruning phantom detections from multiview foreground intersection," in *Proceedings of the IEEE International Conference on Image Processing*, 2012, pp. 1025-1028.

[4] J. Ren, M. Xu, and J. S. Smith, "A colour statistical approach to phantom pruning in multi-view detection," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 756-761.

[5] M. Xu, J. Ren, D. Chen, J. S. Smith, Z. Liu, and T. Jia, "Robust object detection with real-time fusion of multiview foreground silhouettes," *Optical Engineering,* vol. 51, pp. 047202-1-047202-13, 2012.

[6] M. Xu, L. Lu, T. Jia, J. Ren, and J. S. Smith, "Cast shadow removal in motion detection by exploiting multiview geometry," in *Proceedings of the International Conference on Systems, Man, and Cybernetics*, 2012, pp. 762-766.

BIBLIOGRAPHY

[1] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE,* vol. 89, pp. 1456-1477, 2001.

[2] H. Aghajan and A. Cavallaro, *Multi-camera networks: principles and applications*: Academic Press, 2009.

[3] T. Collins, Robert, G. Lipton, Alan, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 745-756, 2000.

[4] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, pp. 505-519, 2009.

[5] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 34, pp. 334-352, 2004.

[6] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding,* vol. 81, pp. 231-268, 2001.

[7] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding,* vol. 104, pp. 90-126, 2006.

[8] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters,* 2013, 34(1): 3-19.

[9] M. Valera and S. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proceedings-Vision, Image and Signal Processing,* vol. 152, pp. 192-204, 2005.

[10] Q. Cai and J. K. Aggarwal, "Automatic tracking of human motion in indoor scenes across multiple synchronized video streams," in *Proceedings of the International Conference on Computer Vision*, 1998, pp. 356-362.

[11] O. Javed, S. Khan, Z. Rasheed, and M. Shah, "Camera handoff: tracking in multiple uncalibrated stationary cameras," in *Proceedings of the Workshop on Human Motion*, 2000, pp. 113-118.

[12] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, 2.

[13] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous multicamera tracking on embedded smart cameras," *EURASIP Journal on Embedded Systems,* vol. 2007, pp. 35-35, 2007.

[14] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, pp. 1355-1360, 2003.

[15] G. P. Stein, "Tracking from multiple view points: Self-calibration of space and time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, 1.

[16] M. Xu, J. Orwell, L. Lowey, and D. Thirde, "Architecture and algorithms for tracking football players with multiple cameras," *IEE Proceedings-Vision, Image and Signal Processing,* vol. 152, pp. 232-241, 2005.

[17] J. Kang, I. Cohen, and G. Medioni, "Continuous tracking within and across camera streams," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. I-267-I-272 vol. 1.

[18] W. Du and J. Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Proceedings of the Asian Conference of Computer Vision*, 2007, pp. 365-374.

[19] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 663-671, 2006.

[20] J. Black and T. Ellis, "Multi camera image tracking," *Image and Vision Computing,* vol. 24, pp. 1256-1267, 2006.

[21] S. M. Khan, P. Yan, and M. Shah, "A homographic framework for the fusion of multi-view silhouettes," in *Proceedings of the International Conference on Computer Vision*, 2007, pp. 1-8.

[22] A. Mittal and L. S. Davis, "Unified multi-camera detection and tracking using region-matching," in *Proceedings of the IEEE Workshop on Multi-Object Tracking*, 2001, pp. 3-10.

[23] A. Mittal and L. S. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," in *Proceedings of the European Conference on Computer Vision*, 2002, pp. 18-33.

[24] J. Berclaz, F. Fleuret, and P. Fua, "Principled Detection-by-Classification from Multiple Views," in *Proceedings of the Conference on Computer Vision Theory and Applications*, 2008, pp. 375-382.

[25] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1806-1819, 2011.

[26] R. Eshel and Y. Moses, "Tracking in a dense crowd using multiple cameras," *International journal of computer vision,* vol. 88, pp. 129-143, 2010.

[27] O. Faugeras, *Three dimensional computer vision: A geometric viewpoint*: the MIT Press, 1993.

[28] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*: Cambridge Univ Press, 2000.

[29] I. Mikic, S. Santini, and R. Jain, "Video processing and integration from multiple cameras," in *Proceedings of the Image Understanding Workshop, Morgan-Kaufman, San Francisco*, 1998, 6.

[30] S. L. Dockstader and A. M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," *Proceedings of the IEEE,* vol. 89, pp. 1441-1455, 2001.

[31] Y. Li, A. Hilton, and J. Illingworth, "A relaxation algorithm for real-time multiple view 3D-tracking," *Image and vision computing,* vol. 20, pp. 841-859, 2002.

[32] D. Focken and R. Stiefelhagen, "Towards vision-based 3-d people tracking in a smart room," in *Proceedings of the IEEE International Conference on Multimodal Interfaces*, 2002, pp. 400-405.

[33] A. Mittal and L. S. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," *International Journal of Computer Vision,* vol. 51, pp. 189-203, 2003.

[34] R. Pflugfelder and H. Bischof, "People tracking across two distant self-calibrated cameras," in *Proceedings of the Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 393-398.

[35] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras," in *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2001, pp. 91-96.

[36] Q. Cai and J. Aggarwal, "Tracking human motion using multiple cameras," in *Proceedings of the International Conference on Pattern Recognition*, 1996, pp. 68-72.

[37] T.-H. Chang, S. Gong, and E.-J. Ong, "Tracking Multiple People Under Occlusion Using Multiple Cameras," in *Proceedings of the British Machine Vision Conference*, 2000, pp. 1-10.

[38] T.-H. Chang and S. Gong, "Tracking multiple people with a multi-camera system," in *Proceedings of the IEEE Workshop on Multi-Object Tracking*, 2001, pp. 19-26.

[39] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proceedings of the Workshop on Motion and Video Computing*, 2002, pp. 169-174.

[40] J. Berclaz, F. Fleuret, and P. Fua, "Multi-camera tracking and atypical motion detection with behavioral maps," in *Proceedings of the European Conference on Computer Vision*, 2008, pp. 112-125.

[41] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 758-767, 2000.

[42] T. T. Santos and C. H. Morimoto, "Multiple camera people detection and tracking using support integration," *Pattern Recognition Letters,* vol. 32, pp. 47-55, 2011.

[43] K. Kim and L. S. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 98-109.

[44] D. Arsic, E. Hristov, N. Lehment, B. Hornler, B. Schuller, and G. Rigoll, "Applying multi layer homography for multi camera person tracking," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1-9.

[45] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst, "Sparsity driven people localization with a heterogeneous network of cameras," *Journal of Mathematical Imaging and Vision,* vol. 41, pp. 39-58, 2011.

[46] J. Orwell, P. Remagnino, and G. A. Jones, "Multi-camera colour tracking," in *IEEE Workshop on Visual Surveillance*, 1999, pp. 14-21.

[47] E. D. Cheng and M. Piccardi, "Disjoint track matching based on a major color spectrum histogram representation," *Optical Engineering,* vol. 46, pp. 047201-047201-14, 2007.

[48] R. Bowden and P. KaewTraKulPong, "Towards automated wide area visual surveillance: tracking objects between spatially-separated, uncalibrated views," in *Proceedings of IEE Vision, Image and Signal*, 2005, pp. 213-223.

[49] A. Gilbert and R. Bowden, "Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity," in *proceedings in ECCV* 2006, pp. 125-136.

[50] U. Park, A. Jain, I. Kitahara, K. Kogure, and N. Hagita, "Vise: Visual search engine using multiple networked cameras," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, pp. 1204-1207.

[51] F. Porikli, "Inter-camera color calibration by correlation model function," in *Proceedings in International Conference on Image Processing*, 2003, pp. II-133-6 vol. 3.

[52] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proceedings of the International Conference on Computer Vision*, 2003, pp. 952-957.

[53] S. Sternig, T. Mauthner, A. Irschara, P. M. Roth, and H. Bischof, "Multi-camera multi-object tracking by robust hough-based homography projections," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1689-1696.

[54] M. Liem and D. M. Gavrila, "Multi-person tracking with overlapping cameras in complex, dynamic environments," in *Proceedings of the British Machine Vision Conference*, 2009, 38(3): 199-218.

[55] D. B. Yang, H. H. González-Baños, and L. J. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Proceedings of the International Conference on Computer Vision*, 2003, pp. 122-129.

[56] M. Liem and D. M. Gavrila, "Multi-person localization and track assignment in overlapping camera views," in *Pattern Recognition*, ed: Springer, 2011, pp. 173-183.

[57] X. Tong, T. Yang, R. Xi, D. Shao, and X. Zhang, "A Novel Multi-planar Homography Constraint Algorithm for Robust Multi-people Location with Severe Occlusion," in *Proceedings of theInternational Conference on Image and Graphics*, 2009, pp. 349-354.

[58] P. Peng, Y. Tian, Y. Wang, and T. Huang, "Multi-camera Pedestrian Detection with Multi-view Bayesian Network Model," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 1-12.

[59] W. Ge and R. T. Collins, "Crowd detection with a multiview sampler," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 324-337.

[60] Y. Abdel-Aziz and H. M. K. m. a, "Direct linear transformation into object shape coordinates in close-range photogrammetry," in *Proceedings of Symposium Close-Range Photogrammetry*, 1971, pp. pp.1-18.

[61] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1986.

[62] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1330-1334, 2000.

[63] K.-Y. Wong, P. R. S. Mendonca, and R. Cipolla, "Camera calibration from surfaces of revolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, pp. 147-161, 2003.

[64] P. Beardsley and D. Murray, "Camera calibration using vanishing points," in *Proceedings of the British Machine Vision Conference*, 1992, pp. 416-425.

[65] R. Cipolla, T. Drummond, and D. P. Robertson, "Camera Calibration from Vanishing Points in Image of Architectural Scenes," in *Proceedings of the British Machine Vision Conference*, 1999, pp. 382-391.

[66] F. Lv, T. Zhao, and R. Nevatia, "Self-calibration of a camera from video of a walking human," in *Proceedings of the International Conference on Pattern Recognition*, 2002, pp. 562-567.

[67] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1513-1518, 2006.

[68] Z. Zhang, M. Li, K. Huang, and T. Tan, "Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.

[69] J. Wright, A. Wagner, S. Rao, and Y. Ma, "Homography from coplanar ellipses with application to forensic blood splatter reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern* 2006, pp. 1250-1257.

[70] H. Zeng, X. Deng, and Z. Hu, "A new normalized method on line-based homography estimation," *Pattern Recognition Letters,* vol. 29, pp. 1236-1244, 2008.

[71] M. Brown and D. G. Lowe, "Recognising panoramas," in *Proceedings of the International Conference on Computer Vision*, 2003, pp. 1218-1225.

[72] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey vision conference*, 1988, p. 50.

[73] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, pp. 91-110, 2004.

[74] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*: Thomson-Engineering, 2007.

[75] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 580-593, 1997.

[76] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, pp. 381-395, 1981.

[77] S. Choi, T. Kim, and W. Yu, "Performance Evaluation of RANSAC Family," in *Proceedings of the British Machine Vision Conference*, 2009.

[78] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision,* vol. 40, pp. 123-148, 2000.

[79] M. Xu, J. Ren, D. Chen, J. S. Smith, Z. Liu, and T. Jia, "Robust object detection with real-time fusion of multiview foreground silhouettes," *Optical Engineering,* vol. 51, pp. 047202-1-047202-13, 2012.

[80] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 679-698, 1986.

[81] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing,* vol. 20, pp. 647-655, 2002.

[82] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *The International Journal for Geographic Information and Geovisualization,* vol. 10, pp. 112-122, 1973.

[83] J. L. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision,* vol. 12, pp. 43-77, 1994.

[84] D. Meyer, J. Denzler, and H. Niemann, "Model based extraction of articulated objects in image sequences for gait analysis," in *Proceedings of the International Conference on Image Processing*, 1997, pp. 78-81.

[85] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1998, pp. 8-14.

[86] Y. Kameda and M. Minoh, "A human motion estimation method using 3-successive video frames," in *Proceedings of the International conference on virtual systems and multimedia*, 1996, pp. 135-140.

[87] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE,* vol. 90, pp. 1151-1163, 2002.

[88] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. II-65-II-72 vol. 2.

[89] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 831-843, 2000.

[90] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 780-785, 1997.

[91] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, 2.

[92] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, ed: Springer, 2002, pp. 135-144.

[93] S. Suzuki, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing,* vol. 30, pp. 32-46, 1985.

[94] J. E. Hershberger and J. Snoeyink, *Speeding up the Douglas-Peucker line-simplification algorithm*: University of British Columbia, Department of Computer Science, 1992.

[95] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker, "A characterization of ten hidden-surface algorithms," *ACM Computing Surveys (CSUR),* vol. 6, pp. 1-55, 1974.

[96] M. Xu, J. Ren, D. Chen, J. Smith, and G. Wang, "Real-time detection via homography mapping of foreground polygons from multiple cameras," in *Proceedings of the IEEE International Conference on Image Processing*, 2011, pp. 3593-3596.

[97] A. Koschan and M. Abidi, *Digital color image processing*: Wiley. com, 2008.

[98] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, "Shape and appearance context modeling," in *Proceedings of the International Conference on Computer Vision*, 2007, pp. 1-8.

[99] S.-H. Cha and S. N. Srihari, "On measuring the distance between histograms," *Pattern Recognition,* vol. 35, pp. 1355-1370, 2002.

[100] J. Ren, M. Xu, and J. S. Smith, "Pruning phantom detections from multiview foreground intersection," in *Proceedings of the IEEE International Conference on Image Processing*, 2012, pp. 1025-1028.

[101] J. Ren, M. Xu, and J. S. Smith, "A colour statistical approach to phantom pruning in multi-view detection," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 756-761.

[102] S.Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory,* vol. 28, pp. 129-137, 1982.

[103] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological),* pp. 1-38, 1977.

[104] N. I. Fisher, *Statistical analysis of circular data*: Cambridge University Press, 1995.

[105] C. Zhang and P. Wang, "A new method of color image segmentation based on intensity and hue clustering," in *Proceedings of the International Conference on Pattern Recognition*, 2000, pp. 613-616.