UNIVERSITY OF LIVERPOOL

Global Optimisation of Multi-Camera Moving Object Detection

Thesis submitted in accordance with the requirements of the

University of Liverpool for the degree of

Doctor in Philosophy

by

Yuyao Yan

October 2018

Department of Electrical Engineering & Electronics

ABSTRACT

An important task in intelligent video surveillance is to detect multiple pedestrians. These pedestrians may be occluded by each other in a camera view. To overcome this problem, multiple cameras can be deployed to provide complementary information, and homography mapping has been widely used for the association and fusion of multi-camera observations. The intersection regions of the foreground projections usually indicate the locations of moving objects. However, many false positives may be generated from the intersections of non-corresponding foreground regions.

In this thesis, an algorithm for multi-camera pedestrian detection is proposed. The first stage of this work is to propose pedestrian candidate locations on the top view. Two approaches are proposed in this stage. The first approach is a top-down approach which is based on the probabilistic occupancy map framework. The ground plane is discretized into a grid, and the likelihood of pedestrian presence at each location is estimated by comparing a rectangle, of the average size of the pedestrians standing there, with the foreground silhouettes in all camera views. The second approach is a bottom-up approach, which is based on the multi-plane homography mapping. The foreground regions in all camera views are projected and overlaid in the top view according to the multi-plane homographies and the potential locations of pedestrians are estimated from the intersection regions.

In the second stage, where we borrowed the idea from the Quine-McCluskey (QM) method for logic function minimisation, essential candidates are initially identified, each of which covers at least a significant part of the foreground that is not covered by the other candidates. Then non-essential candidates are selected to cover the remaining foregrounds by following a repeated process, which alternates between merging redundant candidates and finding emerging essential candidates. Then, an alternative approach to the QM method, the Petrick's method, is used for finding the minimum set of pedestrian candidates to cover all the foreground regions. These two methods are non-iterative and can greatly increase the computational speed. No similar work has been proposed before. Experiments on benchmark video datasets have demonstrated the good performance of the proposed algorithm in comparison with other state-of-the-art methods for pedestrian detection.

Key Words: Video surveillance, Object detection, Multicamera, Homography, Logic minimisation.

ACKNOWLEDGEMENTS

First of all, I would like to thank my primary supervisors Dr. Ming Xu. Without his guidance, I cannot make progresses in my research in the past several years. His academic rigour and innovative thinking inspire me along the way of my research. His guidance has helped me in all the time of this research and in writing of this thesis. Furthermore, I want to thank my co-supervisor Prof. Jeremy S. Smith for his patience and support in supervising me in my research. His enthusiasm in the research was contagious and motivational for me, even during tough times in my Ph.D. study.

I would like to thank Xi'an Jiaotong-Liverpool University for providing a Ph.D. scholarship under Grant PGRS-12-02-07 and the support from the National Natural Science Foundation of China (NSFC) under Grant 60975082.

Moreover, I would like to thank my colleagues for their help and support in my research: Mr. Daoman Hu, Mr. Jie Yang, Mr. Tao Huang, Mr. Mo Shen, Mr. Jin Xi, Ms. Aizhen Zhang and Mr. Chenhui Lu.

Finally, I would like to thank my family for their continuous support. Especially, I would like to thank my wife Xi Yang who supports me not only in life but also in my research. Without her help and support, I cannot insist on my research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Computer vision (CV) is a discipline that deals with how a computer can be made for automatically perceiving and understanding images or videos. In computer vision, as well as artificial intelligence, intelligent visual surveillance is an active field which aims to automatically analyse and interpret the object behaviours such as object detection, tracking, classification and event detection [1].

Compared with the traditional video surveillance systems, intelligent video surveillance is designed to assist or replace human for the video analysis in both online or offline applications. Online intelligent video surveillance is used to automatically recognise scenes or objects, track specific objects, and respond to specific events. An example is the tracking of pedestrians or the alert of violence accidents. Compared with the traditional online video surveillance, an online intelligent video surveillance system no longer requires an observer to spend time observing the monitor and responding for the events, which effectively saves the labour and also reduce false positives and false negatives. Offline intelligent video surveillance can organise and analyse recorded video sequences to obtain statistical results or generate the index for the objects and events which provides faster retrieval of date, which often uses statistical and pattern recognition methods. Different from the traditional offline video surveillance, an offline intelligent video surveillance system can obtain information faster and more accurate. Intelligent video surveillance systems have now been deployed in many applications such as the crime prevention, public security, sporting events and traffic monitoring.

A typical video surveillance system consists of distributed cameras. Each of these cameras monitors a particular area. There is no or only a small overlapping field of view between every two cameras. However, this system is flawed that, in single-camera video surveillance, occlusions between moving objects or moving

objects occluded by a static obstacle (such as trees, billboards) will adversely affect the detection results. To avoid the effect of occlusions, multi-camera video surveillance was introduced. Multi-camera video surveillance refers to use multiple cameras capturing video on the same scene at different locations and to monitor this scene by analysing video signals from all cameras. The advantage of multi-camera video surveillance is that it increases the viewing angle which provides more sufficient and reliable data for the video analysis and processing; the information fusion of multiple cameras makes the detection result more robust; the use of multiple cameras can determine the three-dimensional position of a moving object.

Since moving objects in video surveillance can be mainly classified into two categories: vehicles and pedestrians. Vehicles are solid objects that are easily modelled and detected. However, pedestrians are not easy to be detected due to their different actions, grouping or occlusions. This research focuses on the pedestrian detection using multiple cameras.

1.1 Aims and Objectives

When working with multiple camera views, a classical method, called homography, has been widely used for the association and fusion of multi-camera observations. In early works, the measurements, features or tracks were extracted in individual camera views and then integrated to obtain the global estimates, which makes this approach vulnerable to dynamic occlusion and grouping [2, 3]. For example, Hu et al. [3] projected the principal axis of each pedestrian from one camera view to another and selected the intersection of every two correlated principal axes as the pedestrian location. However, it is not trivial to reliably extract such axes, when pedestrians occlude each other or are in groups in a single view. An efficient way to solve this problem is that the individual cameras no longer extract features but provide foreground silhouettes to the fusion centre.

There are two benchmark works in this trend. Khan and Shah [4] projected the foreground likelihoods, from individual camera views, to a reference view by using

ground-plane homographies and identified the heavily overlapped regions as the potential locations of pedestrians. Fleuret et al. [5] discretised the ground plane into a grid and modelled each pedestrian as a rectangle of the average size of pedestrians standing at a location. Then by using a Probabilistic Occupancy Map (POM), the probability of pedestrian presence at each location is calculated by seeking evidence from the foreground silhouettes in all camera views.

Although both methods add robustness to pedestrian detection, they still have drawbacks. In Khan and Shah's method [4], the foreground projections of different pedestrians, each from a different camera view, may falsely intersect in the reference view, which gives rise to phantoms. Deploying more cameras or placing the cameras at a high elevation [6] may reduce the number of phantoms, but these approaches are not always suitable for most real scenarios. Furthermore, each of the overlapped regions is a range of locations, rather than an accurate location. This becomes obvious when multiple pedestrians are overlapping in the same foreground region in an individual camera view. Based on Khan and Shah's method, Utasi and Benedek [7, 8] utilized the height information of pedestrians with pixel-level features to achieve a better performance, and Liu et. al. [9] converted foreground regions in each camera view by vertical line segments to speed up the algorithm and used geometry-based rule to filter out false detection. However, the results of these methods will be deteriorated by the broken foregrounds. In the POM method [5], the locations which are close to pedestrians may have high occupancy probabilities, even if they do not contain a pedestrian. Moreover, the foreground pixels for a pedestrian were considered to be uniformly distributed within the rectangle, which is not accurate. Based on the POM method, Alahi et al. [10] modelled pedestrian detection as a linear inverse problem, and Peng et al. [11] used the multiview Bayesian networks to analyse the occlusion relationship between potential pedestrians. However, these methods rely on iterative algorithms to update the occupancy likelihood of each location, which is time-consuming.

Given the background as above, there are two major challenges in this research. The first challenge is to develop techniques which can eliminate phantoms in

multi-camera pedestrian detection. The second challenge is to develop a quick and effective multi-camera pedestrian detection method which can fully utilise the foreground silhouettes.

In this thesis, an algorithm for multi-camera pedestrian detection is proposed, which can be divided into two stages. In the first stage, two efficient approaches are introduced to propose the pedestrian candidates which include real pedestrians and phantoms. In the second stage, the pedestrians are further identified from the candidates by using a logic minimisation approach. Figure 1-1 shows an overview structure of the proposed algorithm.

In the first stage, the pedestrian candidates are proposed by fusing the foreground silhouettes in each camera view. Two approaches are proposed for this stage. The first one is based on the POM framework and the second one is based on the Khan and Shah's method [4]. Both the approaches use a template to model the foreground pixels and background pixels of the pedestrians based on the foreground silhouettes. The joint occupancy likelihood of each pedestrian candidate is calculated by taking into account the template matching response and the head/foot observability. In the second approach, the top of head and the local maxima of template matching response are also used to find the accurate locations of pedestrians. In the proposed approaches, the joint occupancy likelihood is only calculated once, which can greatly improve the computational speed.

In the second stage, the pedestrians are further identified from the candidates by using a logic minimisation approach. The Quine-McCluskey (QM) method, which is originally used for logic function minimisation, is borrowed to identify pedestrians and phantoms from the candidates. To implement the QM method, each foreground region is decomposed into sub-regions according to the overlapping relationship of the candidate boxes associated with that foreground region. Then a prime candidate chart is built and simplified to find real pedestrians. Furthermore, an alternative approach to the QM method, the Petrick's method, is used for finding the minimum set of pedestrian candidates to cover all the foreground sub-regions of interest. Since these two methods are based on logic operations, their computational time is

only a few milliseconds. By using the logic minimisation approach, pedestrians and phantoms can be effectively identified from pedestrian candidates.

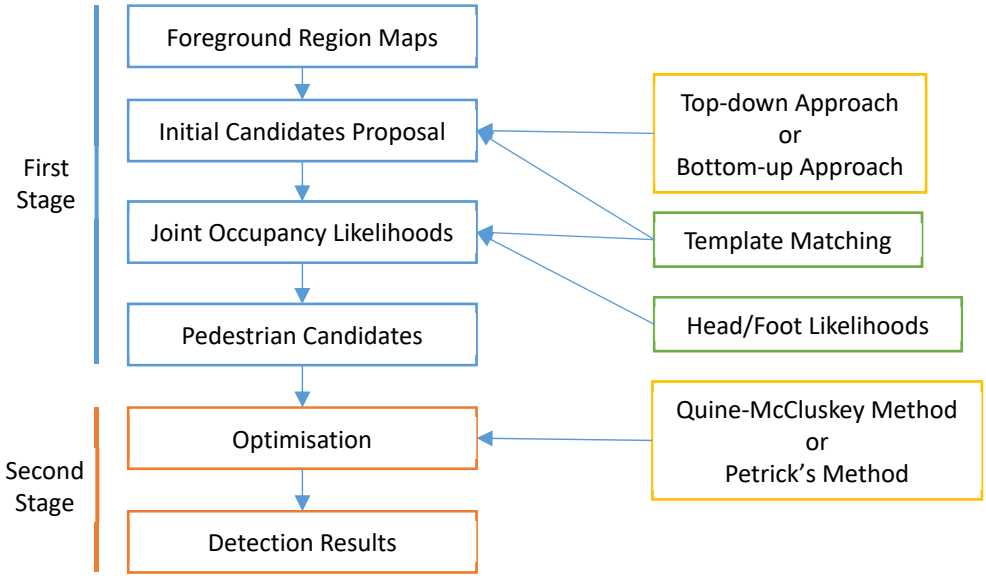


Figure 1-1 Overview of the proposed algorithm.

1.2 Contributions

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

1) A global optimisation approach is proposed, which transforms the optimisation problem into a logic minimisation problem. Based on the Quine-McCluskey method, a prime candidate chart is used which greatly reduces the search space for an optimised solution for multiview pedestrian detection.
2) The Petrick's method, which was used for the logic function simplification, is used to find the minimum set of pedestrian candidates to cover all the foreground sub-regions of interest. This method greatly increases the computational speed. No similar work has been proposed before.
3) Two approaches for multi-camera pedestrian detection are proposed by fusing the high-level information from multiple camera view. The top-down approach is based a discretized top view and extracts the corresponding foreground information of each location from all camera views. In the bottom-up approach, the foregrounds from all camera views are projected into the top view to make

the decision.

4) Template matching is used to calculate the occupancy likelihood in each camera view. The template considers not only the contribution of foreground pixels but also that of background pixels, which can differentiate pedestrians and phantoms more robustly.
5) The observability of a pedestrian's head and feet is considered, which is based on the foreground silhouette within the candidate box. This method can enlarge the difference between pedestrians and phantoms.

1.3 Structure of This Thesis

The remainder of this thesis is organized as follows: In Chapter 2, literature on single-camera and multi-camera video surveillance are reviewed and discussed. Chapter 3 describes the two approaches to propose pedestrian candidates based on foreground silhouettes. In Chapter 4, two logic minimisation approaches based on the Quine-McCluskey method and the Petrick's method are proposed to identify pedestrians and phantoms from the pedestrian candidates. Chapter 5 shows the qualitative and quantitative evaluation on the proposed methods. Conclusions and the future work are presented in Chapter 6.

Chapter 2

Literature Review

Since intelligent video surveillance has been one of the most active research areas in computer vision, there are a large number of publications in this field [12-16]. In this chapter, a survey of the research in the scope of this thesis is presented. It begins with a review of single-camera video surveillance systems. Then the researches on multi-camera video surveillance systems are reviewed. Based on the degree of information fusion from multiple camera views, the existing methods are divided into three categories: low-level information fusion, intermediate-level information fusion and high-level information fusion. The methods of high-level information fusion are introduced in details. Finally, the existing algorithms for multiview pedestrian detection are introduced.

2.1 A Review of Single-Camera Video Surveillance

Early single camera video surveillance is mainly based on the analysis and tracking of foreground blobs in moving object detection. Some researchers modelled the foreground blobs as joint connected body parts [17] or parameterised shapes [18]. These methods require the classification of moving objects before tracking to assign a specific model to that type of objects. However, there existing un-modelled objects or non-rigid objects which are difficult to model. Using some uninterpreted low-level visual features, e.g. the centroid, bounding box, colour histogram or intensity template of each foreground region, will get better results in the tracking [19-24]. These methods use tracking information combined with low-level features to solve the dynamic occlusion in single-camera video surveillance. However, they are based on the background subtraction which is sensitive to illumination variation or camera shaking.

With the development of machine learning techniques, the methods using

artificially extracted features combined with classifiers have been widely used in single-camera object detection. These methods use a sliding window to scan the image, compare the enclosed image with an image pyramid, and use a classifier to determine whether each window encloses a specific class of objects. By using Haar-like features [25, 26], Edgelet features [27], Shapelet features [28] or Histograms of Gradients (HOG) features [29], the feature-based methods obtain good results in pedestrian detection. At this stage, since no single feature can outperform the HOG features, the researchers focus on the improvement of the HOG features [30] and combine it with other features [31].

This type of methods has good detection results for the objects which have minor occlusions or deformations, but it often fails in severe occlusions. In order to cope with such deformation and occlusion, part-based detection methods have been proposed. Each part of an object is detected separately so that the object can still be detected even if the deformation or occlusion occurs. Based on [32], Felzenszwalb et al. [33, 34] proposed a Deformable Part Model (DPM), in which the HOG features of each part of pedestrians are trained separately and the corresponding position of each part is trained to fit a distribution. This model is widely used to detect occluded pedestrians [35].

In recent years, deep learning has achieved excellent performances in single-camera object detection [36, 37]. In particular, there exist the detection methods based on object proposals [38], such as R-CNN [39, 40], Fast R-CNN [41] and Faster R-CNN [42]. In these methods, object proposals are used to replace the image pyramid to improve efficiency. At the same time, an end-to-end detection method has been proposed, which can significantly improve the detection speed with limited loss of accuracy. This family of approaches includes YOLO [43], YOLO2 [44] and SDD [45], etc.

On the other hand, none of the above methods can correctly detect objects that are severely occluded without using tracking information. Typically, when cameras are at eye-level, severe occlusion cannot be avoided in the crowds in high density. The solution to this problem lies in multi-camera video surveillance.

2.2 A Review of Multi-Camera Video Surveillance

Placing multiple cameras at different locations to monitor a scene can provide a broader field of view and collect sufficient observations in video analysis. For example, when the dynamic occlusion of moving objects occurs in a camera view, the involved objects may be not occluded in other camera views. Using multiple cameras can improve the accuracy and robustness in information fusion. In addition, the 3D position of an object in the air can be determined from the intersection of a pair of image rays [46], each of which comes from a different camera view. This is impossible by using a single camera. Depending on the degree of information fusion, the existing multi-camera surveillance systems can be categorised as low-level information fusion, intermediate-level information fusion and high-level information fusion.

2.2.1 Low-Level Information Fusion

The first category of multi-camera video surveillance systems is the methods which switch the tracking of objects across camera views [47-51]. In these methods, each camera detects and tracks moving objects separately. When a tracked object goes beyond the field of view (FOV) of the current camera, it is switched to another camera. Therefore, multiple cameras are used to extend the limited field of view of a single camera. The existing researches are focused on when to switch the camera, which camera is optimal for an object, and how to establish the correspondence of the objects between cameras.

In [47], each object is tracked in a single-camera view and switched to another camera when the system predicts that the current camera will no longer have a good view. The good view is defined as the camera view which has a high-confidence match, between the object in this camera view and the object passed from the original camera, and will observe the object over the greatest number of frames in future. The matching confidence is evaluated by extracting features from the upper human bodies.

Javed et al. [48] proposed a method to calculate the FOV borders, of each camera, in other camera views, for a set of uncalibrated cameras. The borders are called field of view lines which are used to trigger the handoff between two camera views. When an object is close to a field of view line, the correspondence of this object between two camera views is established. The advantage of this method is that there is no feature matching step which is difficult in widely separated cameras.

Quaritsch et al. [49] designed a migration region in the overlapping FOV of two camera views. Before an object moves out of the FOV, it enters the migration region which triggers the handover of the tracker to the next camera view where the tracker can continue tracking the object.

In these methods, the same object is only detected and tracked by one camera at the same time. During the switch of the cameras, the exchange of information is very limited. Only the parameters of the tracker or the features of the tracked object are passed. These methods cannot use multiple cameras to solve the problems caused by the lack of information from a single-camera view, e.g. the dynamic occlusion. Therefore, this method of camera switching is classified as low-level information fusion.

2.2.2 Intermediate-Level Information Fusion

When the spatial correspondence in the overlapping FOV of multiple cameras is obtained, the tracking trajectory or extracted features of the same object in different camera views can be associated and fused to obtain a global estimate. The extracted features include but are not limited to, bounding boxes [52], centroids [46], principal axes [53] and colours [54].

The correspondence of the objects between camera views can be obtained by calibrating the cameras [55] or estimating the homography between camera views. In [56], the trajectories of objects in different camera views are used to estimate the ground-plane homography among the camera views. By using the homography mapping, the tracking is more accurate by fusing the same object's trajectories in different camera views. Khan and Shah [57] obtained the ground-plane homography,

from each camera view to a reference top view, by using camera calibration and fused the detection results of multiple camera views on the top view to estimate the location of each object. In [2], the measurement uncertainty of the objects in each camera view is considered. When the measurements in a camera view are projected to a top view, the uncertainty changes according to the distances from the objects to the camera. By considering the uncertainty in multi-camera information fusion, the locations of the objects are estimated more accurately.

Kang et al. [52] projected the bounding box of each detected object from one camera view to other camera views. According to the relationship between the projected bounding boxes from other camera view and the bounding boxes detected in a camera view, the occurrence that an object is partially occluded by a static obstacle or two objects are merged into the same bounding box can be detected. This method has a good result when the objects are in a low density and the occlusion relationship is relatively simple.

The principal axis is another good feature in pedestrian detection. Kim and Davis [53] used colour information to segment each detected foreground region into different pedestrians, where each pedestrian has a colour model. Then, the principal axis of each pedestrian in a camera view is obtained and projected to the top view. The location of each pedestrian is detected by finding the intersections of all the corresponding principal axes. Since the colour segmentation as above is time-consuming, Hu et al. [3] segmented each foreground silhouette using a vertical projection histogram. The principal axes are calculated from the segmented foreground regions and then projected to the top view through homography mapping. Different from [53], Hu et al. fused the principal axes on the top view without the prior knowledge that which principal axes belong to the same pedestrian. Du and Piater [58] further integrated particle filters into the principal axis-based method, which makes the detection results more robust.

Using epipolar lines as the feature to fuse the observations of objects can offer measurements in 3D space. Chang and Gong [59] calculated the epipolar line of the top of each pedestrian's head in other camera views as the height observation of

the pedestrian. This information is combined with colours to identify the pedestrians. In [60] and [61], the foreground is segmented into sub-regions based on colours. The sub-regions are then matched across pairs of views by using colours. The midpoints of each matched pair in different camera views are projected on the top view as epipolar lines. By analysing the intersections of the epipolar lines, the pedestrian can be detected. Black and Tim [46] calculated the epipolar line of the centroid of each foreground region in world coordinates and identified the intersections of these epipolar lines as 3D locations of pedestrians.

These methods are classified as intermediate-level information fusion, because they attempted to integrate the features, which are extracted from the individual camera views, to solve dynamic occlusions on the basis of the assumption that dynamic occlusion would not occur simultaneously in different cameras. Since the features are extracted from the individual camera views, these methods are vulnerable to the occlusion and grouping of pedestrians.

2.2.3 High-Level Information Fusion

In recent ten years, the third category of multi-camera video surveillance systems has been favoured by video surveillance community, in which a single camera no longer provides the extracted features but the foreground bitmap information to the fusion centre. These systems can be divided into two classes: bottom-up methods and top-down methods [62].

The Bottom-up method is to project the foreground likelihood or foreground silhouettes into the top view by using the ground-plane homography. In the top view, the location of each object is determined by the analysis of the overlaid foreground projections [63, 64]. This method was firstly proposed by Elfes [65] who projected the camera view of a moving robot to the reference ground plane. Otsuka and Mukawa [66] used an elliptical cylinder to approximate each pedestrian and projected the foreground region to the top view as a visual cone. The occlusion relationship is then analysed from the intersections of the visual cones in the top view.

Khan and Shah [4] projected the foreground likelihood maps of the individual camera views into a reference camera view by using the ground-plane homography and multiplied them together. By thresholding the overlaid foreground likelihood map, the locations of the pedestrians can be detected and then warped back to the individual views. The advantage of this method is the foreground pixels of the front pedestrians can still support the pedestrian who is hidden behind when an occlusion occurs. Once the foreground pixels belonging to the hidden pedestrian are observed in other camera views, that pedestrian can be detected. However, this may lead to false positives (phantoms) in detection since the foreground pixels belonging to non-corresponding pedestrians in different camera views may be falsely intersected when the number of cameras is limited. Also, this method is sensitive to broken foreground regions. Applying the multi-plane homography mapping [67], which projects and overlays the foreground likelihood maps of individual camera views onto the top view according to the homographies of a set of planes parallel to the ground plane and at different heights, can relieve the effect of the broken foregrounds.

Eshel and Moses [6, 68] projected the foreground intensities from multiple camera views to a reference view using the head-plane homographies. Then pixel-wise correlation is carried out in these foreground intensity projections. Pedestrians can be detected at those locations where the projected intensities from multiple camera views are highly correlated. This method can detect pedestrians by using the cameras hanging above the heads of pedestrians. It may fail when eye-level cameras are used, because this method is vulnerable to occlusion. In addition, it depends too much on the dissimilarity of the colours between pedestrians.

Ge et al. [69] fused foreground silhouettes on a top view and generated the occupancy likelihood rays in each camera view, which are based on the polar coordinates with the location of the camera as the origin. The pedestrian locations are sampled by using the occupancy likelihood rays as the proposed distributions in the Markov Chain Monte Carlo method. In this method, the pedestrians are

modelled as cylinders in the world coordinate and projected to the individual camera views as a rectangle. The objective of this method is to find the optimal pedestrian locations so that the generated rectangles can interpret the foreground silhouettes well.

Utasi and Benedek [7, 8] projected the foreground silhouettes of each camera view to the top view, according to the homographies of parallel planes at different heights, to estimate the locations and heights of pedestrians. Similar to [69], they modelled pedestrians as cylinders and then extended the classical Bayesian Marked Point Process (MPP) [70] to the 3D space to generate a finite number of the cylinders which fit the foreground silhouettes well. The advantage of this method is the usage of the height information of pedestrians based on pixel-level features, but the drawback is such features are sensitive to broken foregrounds, which may lead to missed detection.

In [71] and [9], the foreground regions in each camera view are approximated by vertical line segments which are then projected to a top view. The positions of pedestrians are inferred by counting the number of intersections of the projected line segments in the top view. By considering the physical shape and size of the foreground silhouette of a pedestrian, geometry-based rules are applied to filter out phantoms. This method is a faster implementation of Khan and Shah's method [67] and is at least one hundred times faster than the latter.

In the top-down method, the ground plane is divided into a grid, and the occupancy probabilities at each location of the grid are estimated based on the back-projection of some kind of generative models in each of multiple camera views [5, 10, 72]. The generative models, which are usually based on the average height and width of pedestrians, are compared with the foreground silhouettes [5] or features [72] to validate the estimated pedestrians' locations.

Fleuret et al. [5] calculated a probabilistic occupancy map (POM) in the ground plane which is discretized into a grid. Each pedestrian is modelled as a rectangle of the average size of pedestrians standing at a location. They minimised the Kullback-Leibler divergence between the approximated occupancy probabilities of

each location and the posterior distributions observed from foreground silhouettes. Then they updated the approximated occupancy probabilities iteratively to find the optimal rectangles which cover more foreground pixels and fewer background pixels in all camera views.

In [72], for each location in the grid, the corresponding rectangular sub-images are extracted from each camera view. Then, a classifier is trained to recognise pedestrians in sub-images and generate a classification score map for each camera view independently. Finally, these maps are combined into a single detection score map by using a tree response model which can handle occlusions.

Alahi et al. [10] modelled pedestrian detection as a linear inverse problem which is regularized by using a sparse binary occupancy vector. Then an iterative process was undertaken to find the optimal occupancy vector which contains the minimum number of non-zero elements and fits the multi-view silhouettes. After that, a real-time version based on the same framework is proposed by using a greedy optimisation algorithm based on set covering [73].

Peng et al. [74] used the Multiview Bayesian Networks (MvBN) to analyse the occlusion relationship between potential pedestrians to obtain the position of real pedestrians. They then expanded their research by implementing the self-adaptive heights of pedestrians [11], which makes this method more robust. In [11], each candidate rectangle can independently shift within a small range in the camera views to cope with inaccurate camera calibration or synchronisation.

Recently, some deep-learning based methods have been proposed [75-77]. Similar to [5], the location of each pedestrian in [75] is obtained by minimising the difference between the generated rectangles and the foregrounds proposed by a Convolutional Neural Network (CNN). They used high-order CRF terms to model potential occlusions, which improve the robustness. Based on a similar framework to [72], a CNN-based [76] work was proposed, in which the samples of the partly occluded pedestrians are manually generated in the training stage to enhance the robustness of the detector which is based on deep neural networks. However, there has been a limited success to train a complete multi-view processing model. It may

be caused by the lack of large-scale multi-camera datasets which are annotated. Therefore, the great potential of the deep leaning technique has not been fully revealed in multi-view pedestrian detection.

2.3 A Review of Phantom Removal

In the information fusion of a multi-camera system, no matter whether the bottom-up or the top-down approach is used or not, it will inevitably generate false positive detections (phantoms). In the bottom-up approach, the foreground projection of one pedestrian may falsely intersect with that of another pedestrian. The intersections of non-corresponding foregrounds lead to phantoms in pedestrian detection. In the top-down approach, the locations close to pedestrians may have high occupancy probabilities, even if they do not contain a pedestrian. Deploying more cameras or placing the cameras at a high elevation may reduce the number of phantoms, but these methods are not always suitable for most realistic scenarios. Therefore, developing techniques to eliminate phantoms has become a challenging task.

Significant research has been undertaken to avoid the generation of phantoms. Existing methods usually resort to temporal coherence, geometric constraints and colour cues. The temporal approach copes with phantoms in the tracking process. As it is noted that phantoms appear from nowhere and are often unsteadily detected. Therefore, the temporal coherence of each foreground intersection region is checked over some time [78, 79]. If a candidate cannot survive over that time period, in tracking, then it is classified as a phantom. Liem and Gavrila [79] proposed that a new object can only appear from the border of the overlapping FOVs; those initially detected in the middle of the overlapping FOV are phantoms. Similar tracking processes were carried out in [5, 67]. Arsic and Hristov [80] warped each intersection region from the top view back into each camera view and checked if it is occluded by foreground regions in all camera views. If this is true, a tracker is used to identify if it is a phantom or an occluded object. Since the temporal approach is closely related to multi-camera multi-target tracking which is another active

research field in video surveillance, it is not the focus of this thesis.

The geometric approach is based on the comparison of heights and sizes of foreground intersection regions between phantoms and pedestrians. Khan and Shah [67] extended their early work [4] by projecting the foreground likelihoods to a reference view with the homographies of a set of parallel planes and selected the most heavily overlapping areas as pedestrian locations. This approach can reduce the number of phantoms and is the foundation of the research work in [9, 62, 80]. In [78], Yang et al. found that the size of a phantom is often smaller than the minimum pedestrian size if the viewing rays intersect behind a pedestrian in the top view. Eshel and Moses [6] used cameras looking downwards and found that if the viewing rays from two cameras intersect behind a true object, the phantoms are lower than the true object; also taller phantoms occur when the rays intersect in front of true objects. By limiting the heights of pedestrians within an appropriate range, they could remove many, but not all, phantoms.

The colour approach is built on the assumption that the intersecting foreground regions from multiple views are correlated in their colours if they correspond to the same object. Eshel and Moses [6] applied the pixel-wise intensity correlation between aligned frames in a reference view to remove phantoms. Ren et al. [81, 82] proposed an appearance model and a colour matching algorithm to identify phantoms, in which the Mahalanobis distance is used to measure the colour similarity of two intersecting foreground regions. These methods are vulnerable to the occlusion between pedestrians, which leads to more false negatives in the detection.

Phantom removal is sometimes thought of as an optimisation problem. In [5] and [10], generative models and iterative optimisation approaches were used to find the optimal solutions, which can reduce the generation of phantoms. The difference between these methods is the kind of generative models (rectangles [5] or a dictionary [10]) and the way they decide the occupancy in each point of the grid (the probabilistic occupancy map inferred from background subtraction masks [5] or the sparsity constrained binary occupancy map [10]). In [8] and [70], Gibbs sampling

is used to estimate the number and locations of pedestrians in a crowd. In these methods, each pedestrian is modelled as a cylinder which covers a specific region on the ground and inhibited to be generated around the location of another pedestrian. A similar assumption was used in [10] as the Repulsive Spatial Sparsity (RSS) which sets a minimum distance between pedestrians. Peng et al. [11, 74] proposed Multiview Bayesian Networks to prune phantoms on preliminary results obtained from [5]. They modelled each pedestrian as a rectangle similar to [5] and analysed the occlusion relationship among such rectangles using a Bayesian network model.

Chapter 3

Proposition of Pedestrian Candidates

In this thesis, the algorithm for multi-camera pedestrian detection is divided in two stages: proposition of pedestrian candidates and logic function minimisation. The first stage of this work is to propose pedestrian candidate locations in a top view. Two approaches are proposed for this stage. The first approach is a top-down approach which is based on the probabilistic occupancy map framework. The ground plane is discretized into a grid and the likelihood of pedestrian presence at each location is estimated by comparing a rectangle, of the average size of the pedestrians standing there, with the foreground silhouettes in all camera views. The second approach is a bottom-up approach which is based on the multi-plane homography mapping. The foreground regions in all camera views are projected and overlaid in the top view according to the multi-plane homographies and the potential locations of pedestrians are estimated from the intersection regions of the foreground projections.

This chapter is organized as follows: Firstly, the foreground segmentation in individual camera views is introduced. Secondly, the estimation of the homographies for parallel planes is described. Thirdly, a top-down approach based on the probabilistic occupancy map is described. Finally, a bottom-up approach based on the multi-plane homography mapping is proposed.

3.1 Foreground Segmentation

The proposed algorithm begins with the foreground segmentation, which is an essential process in video surveillance systems. It aims to separate moving objects from a background image in each frame. Optical flow [83, 84], temporal differencing [85, 86], and background subtraction [87, 88] are three typical methods in the foreground segmentation. Optical flow can be divided into the dense optical flow

and sparse optical flow. The dense optical flow calculates the velocity of every pixel in the image, which is time-consuming. The sparse optical flow can reduce the computation cost by only calculating the velocity of feature points. However, the computation cost of the sparse optical flow method is still higher than those of the temporal differencing and background subtraction methods. The temporal differencing method calculates the pixel-wise differences between two or three consecutive frames to generate the foreground map in the video sequence. This method has a very low computation cost and can adapt to a dynamic environment quickly. However, when an object moves slowly, a large foreground area will be falsely detected as background due to the similar pixel values; when an object moves too fast, a false-positive foreground area will be detected. In the background subtraction method, a background image was built. Each new frame is subtracted from the background image, and the foreground is obtained by thresholding the subtraction result. Since the foreground pixels are identified according to the pixel-wise 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.

In this thesis, background subtraction is used for the foreground segmentation in each camera view. In the proposed algorithm, a Gaussian mixture model (GMM) [88] is used to model the colour value of each pixel, $\boldsymbol{p} = [p_r, p_g, p_b]^T$, by a mixture of K_g Gaussian distributions. This is to cope with the variations of the background. The k-th multivariate Gaussian distribution with the d-dimensional (currently $d = 3$) observations $\boldsymbol{p}$ is as follows:

$$N(\boldsymbol{p}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{p} - \boldsymbol{\mu_k})^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{p} - \boldsymbol{\mu_k})\right] \quad (3.1)$$

where the mean is denoted by $\boldsymbol{\mu}_k$ and the covariance matrix is $\boldsymbol{\Sigma}_k$. Since each component of the GMM has a weight w_k and $\sum_{k=1}^{K_g} w_k = 1$, the probability of observing values $\boldsymbol{p}$ is defined as the following equation:

$$P(\boldsymbol{p}) = \sum_{k=1}^{K_g} w_k N(p; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (3.2)$$

This model is updated every frame. Each incoming frame is compared with this model, and the pixels with low probability are classified as foreground. When an incoming pixel $\boldsymbol{p}_t(u, v)$ has a small standard deviation (less than 2.5) from the $k-th$ distribution, where t represents the current frame, the parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and w_k are then updated with a fixed learning rate $\alpha \in (0,1)$.

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha \tag{3.3}$$

$$\boldsymbol{\mu}_{k,t} = (1-\alpha)\boldsymbol{\mu}_{k,t-1} + \alpha \tag{3.4}$$

$$\boldsymbol{\Sigma}_{k,t} = (1-\alpha)\boldsymbol{\Sigma}_{k,t-1} + \alpha\left\|\boldsymbol{p_t} - \boldsymbol{\mu}_{k,t-1}\right\|^2 \tag{3.5}$$

The means and covariances of other distributions remain the same, and the weight of each distribution is normalized by the sum of the new K weights. When an incoming pixel fails to match any distribution, a new distribution is built to replace the distribution which has the least weight. The mean is initialized by the pixel value, and the weight and the covariance are initialized with small values [88].

After connected component analysis, the foreground pixels are transformed into a foreground region map $F^c \in \{0,1\}^{W\times H}$, where $c \in [1, C]$, C is the number of cameras and $W \times H$ is the image resolution.

Figure 3-1 shows an example of the foreground segmentation. Figure 3-1 (a) and (b) are two camera views from the PETS2009 CC dataset [89], and the foreground region map of them are shown in Figure 3-1 (c) and (d). In Figure 3-1 (c), the fluttering ribbon around pedestrians is falsely detected as foreground.

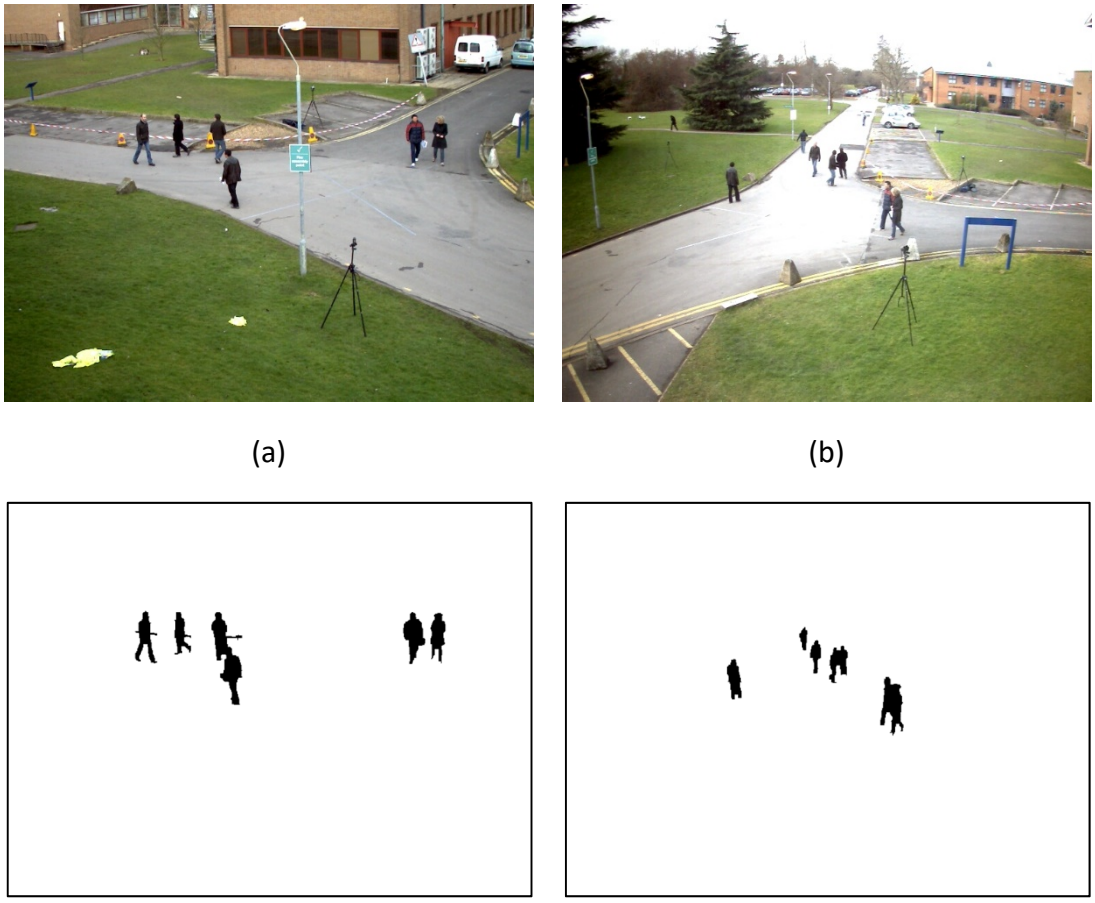


Figure 3-1 An example of foreground segmentation: (a) (b) two camera views and (c) (d) the foreground region maps.

3.2 Homography Mapping

Planar homography is defined by a 3 x 3 transformation matrix between a pair of captured images of the same plane from two camera views. Let $\boldsymbol{u}$ and $\boldsymbol{u}'$ be the image coordinates of a point on such a plane in the two views. They are associated by the homography matrix $\mathbf{H}$ as follows:

$$\tilde{\boldsymbol{u}}' \cong \mathbf{H}\tilde{\boldsymbol{u}} \tag{3.6}$$

where $\cong$ denotes the equivalence defined up to scale and the vectors with a tilde represent their homogeneous coordinates that are often shown with the number 1 in the last row. The homography matrix can be estimated by camera calibration as explained in Section 3.2.2.

3.2.1 Camera Calibration

Camera calibration is an essential operation in measuring the 3D world, which provides a mechanism to build the relationship between a point in the 3D world and a point in a 2D image. When a camera model is selected, both the intrinsic parameters (such as focal length, principal points, skew coefficients and distortion coefficients) and the extrinsic parameters (such as the position of the camera centre and the orientation of the camera in world coordinates) should be estimated.

In many computer vision applications, the pinhole camera model has been widely used [90]. 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. Therefore, by using this model, a point in the 3D world coordinate is projected to the 2D image coordinate in two steps. In the first step, the world coordinate system is aligned to the camera coordinate system with a translation vector $\mathbf{T}$ and a rotation matrix $\mathbf{R}$. The matrix which contains $\mathbf{T}$ and $\mathbf{R}$ is called extrinsic calibration matrix. In the second step, the point is projected from the camera coordinate system to the image coordinate by using the intrinsic calibration matrix, which is constituted by the focal length of the lens f, scale factors m_x and m_y, principal point C_x and C_y, and skew coefficient Γ. Let $\boldsymbol{X} = [x, y, z]^T$ be a point in 3D space and $\boldsymbol{u}^c = [u^c, v^c]^T$ be the corresponding point in camera view c, the transformation that maps $\boldsymbol{X}$ to $\boldsymbol{u}^c$ is as follows:

$$\widetilde{\boldsymbol{u}}^c \cong \begin{bmatrix} fm_x & \Gamma & C_x & 0 \\ 0 & fm_x & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \widetilde{\boldsymbol{X}} \tag{3.7}$$

where the first matrix at the right side of the equation is the intrinsic calibration matrix and the second matrix is the extrinsic calibration matrix.

3.2.2 Homography Estimation

The estimated intrinsic and extrinsic parameters of a camera were used to build a 3x4 projection matrix $\boldsymbol{M}$. As the homography transformation is a special variation of

the projective transformation, $\boldsymbol{M}$ can be used to determine the homography matrix for a specific plane. The transformation that maps $\boldsymbol{X}$ to $\boldsymbol{u}^c$ can be rewritten as follows:

$$\widetilde{\boldsymbol{u}}^c \cong \boldsymbol{M}\widetilde{\boldsymbol{X}} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, \boldsymbol{m}_4]\widetilde{\boldsymbol{X}} \tag{3.8}$$

By assuming that points $\boldsymbol{X}$ and $\boldsymbol{u}^c$ are on the ground plane, then point $\boldsymbol{X}$ is denoted as $\boldsymbol{X}_0 = [x, y, 0]^T$, where subscript 0 denotes the ground plane. Eq. (3.8) can be rewritten as:

$$\widetilde{\boldsymbol{u}}_0^c \cong [\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, \boldsymbol{m}_4]\widetilde{\boldsymbol{X}}_0 = [\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_4]\widetilde{\boldsymbol{x}}^t \tag{3.9}$$

where $\widetilde{\boldsymbol{x}}^t$ is the homogeneous world coordinate of the point $\boldsymbol{x}^t = [x, y]^T$ in the top view. Eq. (3.10) reveals the homography, between camera view c and the top view, for the ground plane.

$$\mathbf{H}_0^{t,c} = \left(\mathbf{H}_0^{c,t}\right)^{-1} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_4] \tag{3.10}$$

The world coordinate of point $\boldsymbol{x}^t$ in the top view can be converted to the image coordinate $\boldsymbol{u}^t$ in the top view by using the linear transformation:

$$\boldsymbol{x}^t = \lambda(\boldsymbol{u}^t - \overline{\boldsymbol{u}}^t) \tag{3.11}$$

where λ is a scale factor to determine the resolution of the top-view image and $\boldsymbol{u}^t$ is the image coordinate of the origin of the world coordinates used in the camera calibration.

Then let us consider a plane parallel to the ground and at a height of h. $\boldsymbol{X}_h = [x, y, h]^T$ is a point on such a plane. Point $\boldsymbol{X}_h$ corresponds to the same point $\boldsymbol{x}^t = [x, y]^T$ in the top view as $\boldsymbol{X}_0$. The projection of point $\boldsymbol{X}_h$ to camera view c is denoted as:

$$\widetilde{\boldsymbol{u}}_h^c \cong [\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, \boldsymbol{m}_4]\widetilde{\boldsymbol{X}}_{\boldsymbol{h}} = [\boldsymbol{m}_1, \boldsymbol{m}_2, h\boldsymbol{m}_3 + \boldsymbol{m}_4]\widetilde{\boldsymbol{x}}^t \tag{3.12}$$

Eq. (3.13) reveals the homography, between camera view c and the top view, for the plane at the height of h:

$$\mathbf{H}_0^{t,c} = [\boldsymbol{m}_1, \boldsymbol{m}_2, h\boldsymbol{m}_3 + \boldsymbol{m}_4] = \mathbf{H}_0^{t,c} + [\mathbf{0}|h\boldsymbol{m}_3] \tag{3.13}$$

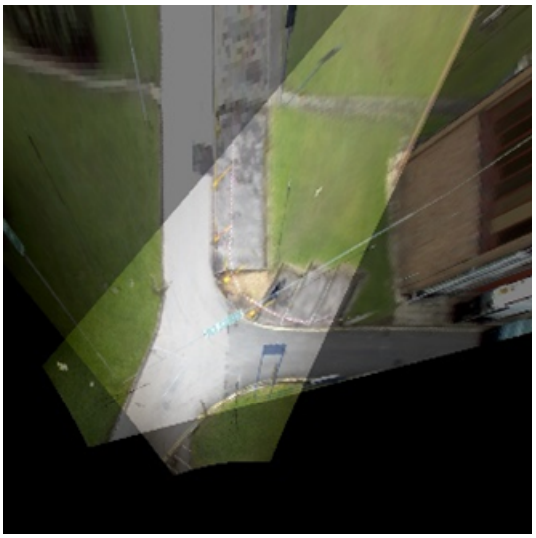
where $[\mathbf{0}]$ is a 3 x 2 zero matrix. Point $\mathbf{x}^t$ can be converted into image coordinates in the top view by using Eq. (3.11).

Figure 3-2 shows an example of the ground-plane homography. Figure 3-2 (a) and (b) are the two background images of view 1 and 2 from the PETS2009 CC

dataset [89]. The intrinsic and extrinsic calibration matrices of each camera view are provided by the dataset. The ground-plane homography from each camera view to the top view is estimated by using the camera calibration matrices. Figure 3-2 (c) shows the synthetic top view by overlaying the projections from Figure 3-2 (a) and (b).

(a)

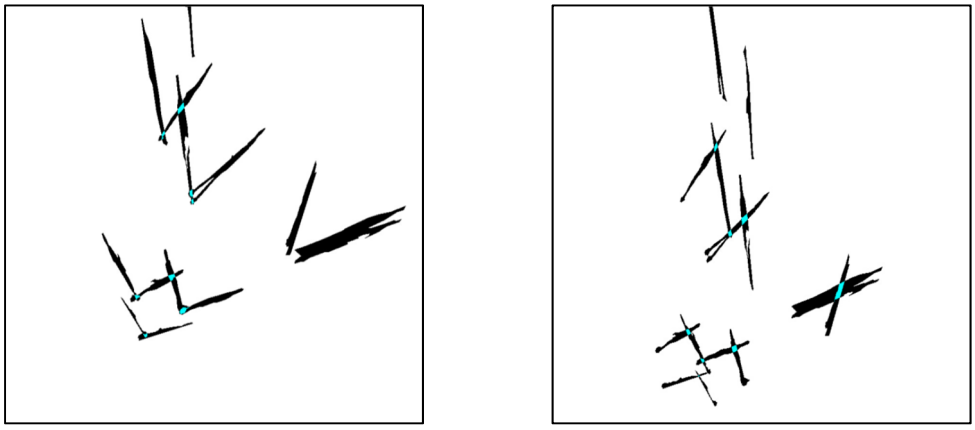
(b)



(c)

Figure 3-2 An example of the ground-plane homography: (a) (b) two camera views, and (c) the homography projection and fusion of the two camera views in the top view.

Figure 3-3 illustrates the homographic projections of the foregrounds from the two camera views to the top view, where the ground plane is used as the reference plane in Figure 3-3 (a), and a waist-plane, at 1.2 meters height above the ground plane, is used in Figure 3-3 (b). By using the ground-plane homography, the foreground projections of each pedestrian from different camera views are intersected at the feet, and they are intersected at the waist by using waist-plane homography.



(a) (b)

Figure 3-3 Homographic projections of the foregrounds from the two camera views to the top view: (a) The projected foregrounds in the top view by using the ground-plane homography. (b) The projected foregrounds using the waist-plane homography. The blue regions represent intersections.

3.3 Top-Down Approach

The goal of this approach is to detect a priori unknown number of pedestrians in multi-camera views. The problem can be formulated as finding the most probable pedestrian locations given a set of detected foreground silhouettes. The ground plane is discretised into a 2-D grid and each location in the grid is considered as the potential location of a pedestrian. Therefore, the objective of this approach is transformed to deducing the locations, occupied by pedestrians, which present such

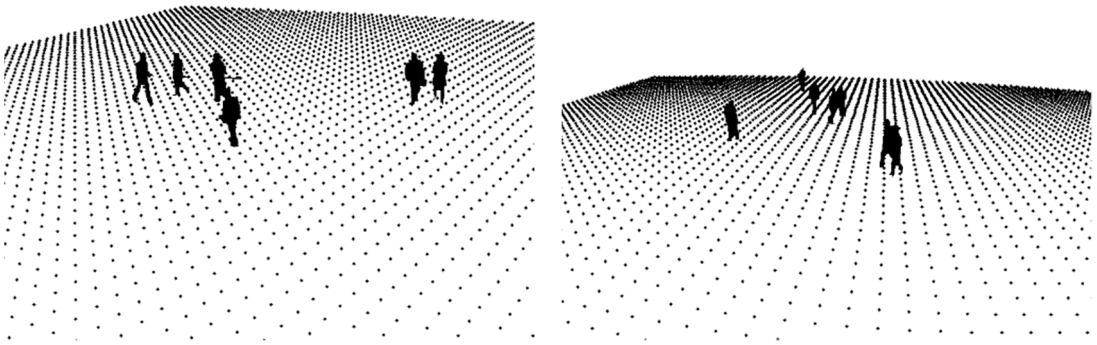
foreground silhouettes in multi-camera views. This section is organized as follows: Firstly, a basic method is introduced to divide the ground plane into a grid. Secondly, an approach is presented to estimate an occupancy likelihood map, and then each local maximum in the map is considered as the potential location of a pedestrian. Thirdly, the observations of the head and feet of each pedestrian candidate in the individual camera views are used to enhance the occupancy likelihoods. Finally, a computation reduction method is proposed to reduce the computation time of this approach.

3.3.1 Occupancy Likelihood Maps

The grid is generated by discretising the ground plane in the top view, and each location is warped to the multiple camera views by using ground-plane homographies. Therefore, each discrete location in the top view is associated with a corresponding location in each of the multiple camera views. Suppose the area of interest on the ground is discretized into a grid of G locations. The i-th location ($i \in [1, G]$) in the top view is associated with its corresponding location (u_i^c, v_i^c) in camera view c through the ground-plane homography $\mathbf{H}_0^{\mathrm{t,c}}$. Figure 3-4 shows an example of foreground silhouettes and the discrete locations on the ground in two camera views. Black regions represent detected foreground silhouettes; black dots are the grid warped from the top view. The grid resolution influences both the accuracy and computational cost of multi-view pedestrian detection.

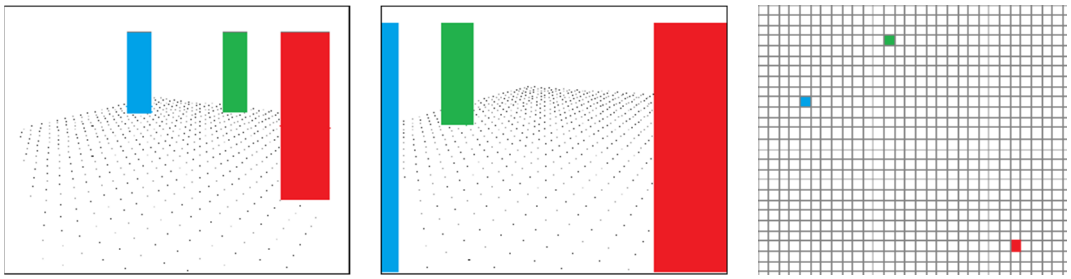
In a calibrated multi-camera system, when a location in the top view is selected, the size of a pedestrian standing at the corresponding location in each camera view can be approximatively represented by a rectangle of the average height and width of the pedestrians standing there. By using the homography $\mathbf{H}_{h_a}^{t,c}$ for the plane at the average height h_a of pedestrians, the i-th location in the top view is mapped to the top of the head of a pedestrian, standing at (u_i^c, v_i^c) and of average height, in camera view c. Therefore, the average height H_i^c and width $W_i^c = \alpha H_i^c$ of the pedestrians standing at the i-th location of camera view c can be obtained. Figure 3-5 is an example of such rectangles, in which different pedestrians are represented

in different colours. Figure 3-5 (a) and (b) show the rectangles which have different sizes at different locations in the same view, and Figure 3-5 (c) shows the locations of the rectangles in the top view.



(a) (b)

Figure 3-4 Foreground silhouettes and the discretised ground plane.



(a) (b) (c)

Figure 3-5 A schematic diagram of approximated rectangles: (a) (b) the same rectangles in two views, and (c) the locations of these rectangles in a top view.

Suppose the events that a pedestrian appears at a specific location are independent. F^c represents the binary foreground region map provided by camera c. R_i^c is the synthetic image obtained by putting a filled rectangle at the i-th location of an empty background image for camera view c and $R_i^c \in \{0,1\}^{W \times H}$, where W and H are the width and height of the image. The foreground pixels contained in R_i^c are represented by A_i^c as follows:

$$A_i^c = F^c \otimes R_i^c \tag{3.14}$$

where $\otimes$ denotes pixel-wise multiplication.

Let $\mathrm{L_i} \in \{0,1\}$ be the event that a pedestrian is present at the i-th location, the posteriori probability of event L_i occurring is proportional to the likelihood of having observations $A_i^1, A_i^2, \dots, A_i^C$, given event L_i happens:

$$P\left(L_i \middle| A_i^1, A_i^2, \dots, A_i^C\right) \propto P\left(A_i^1, A_i^2, \dots, A_i^C \middle| L_i\right) P(L_i) \tag{3.15}$$

According to the conditional independence, the first term of the right side of Eq. (3.15) can be written as:

$$P\left(A_i^1, A_i^2 \dots, A_i^C \middle| L_i\right) = \prod_{c=1}^{C} P(A_i^c | L_i) \tag{3.16}$$

Therefore, at the i-th location, the posteriori probability of event L_i happening is proportional to the joint likelihood of all the observations:

$$P\left(L_i \middle| A_i^1, A_i^2 \dots, A_i^C\right) \propto \prod_{c=1}^{C} P(A_i^c | L_i) \tag{3.17}$$

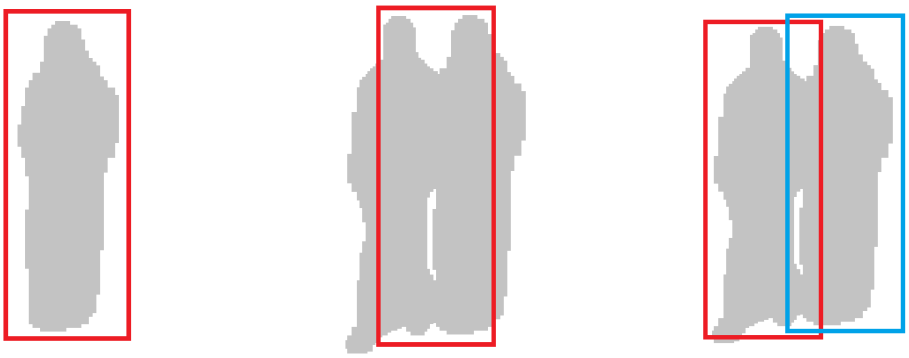
At the i-th location of each camera view, three independent observations are derived, from the foreground pixels A_i^c within the rectangle in R_i^c, to measure how the foreground pixel distribution in the rectangle resembles the silhouette of a pedestrian. The three observations are the template matching response t_i^c, the foot position f_i^c and the head position h_i^c. Considering the conditional independence between the three measurements on the foregrounds, we have:

$$\begin{aligned} & P\left(L_i \middle| A_i^1, A_i^2 \dots, A_i^C\right) \\ & \propto \prod_{c=1}^{C} P(t_i^c, f_i^c, h_i^c | L_i) \\ & = \prod_{c=1}^{C} [P(t_i^c | L_i) P(f_i^c | L_i) P(h_i^c | L_i)] \end{aligned} \tag{3.18}$$

The calculation of $P(t_i^c|L_i)$, $P(f_i^c|L_i)$ and $P(h_i^c|L_i)$ will be introduced in the following sub-sections.

3.3.2 Template Matching Responses

In the proposed method, an important step is to estimate the occupancy likelihood of each location. In the simplest case shown in Figure 3-6 (a), the most intuitive approach is to put a rectangle at the corresponding location in that camera view and calculate the foreground ratio, the ratio of the foreground pixels to all the pixels, within the rectangle. Figure 3-6 (b) shows a more complicated scenario, where a rectangle lodged between two pedestrians can also output a high foreground ratio. However, it is a phantom.



(a) (b) (c)

Figure 3-6 Examples of the detection result by using foreground ratio: (a) When background pixels are located around the foreground pixels within a rectangle, the candidate is likely to be a pedestrian; (b) when many background pixels are located in the middle of a rectangle, the candidate is likely to be a phantom; (c) the correct rectangles for this foreground region.

To solve this problem, researchers have proposed some iterative approaches. Fleuret et al. [5] proposed a generative model to evaluate the difference between the estimated rectangles and the foregrounds. Peng et al. [11] suggested that the foreground pixels close to the vertical central axis of a rectangle should be given greater weights. Another method was proposed by Alahi et al. [10], in which a half-ellipsoid sitting on a cylinder is used to accurately model the silhouette of each pedestrian, since they noted that the foreground pixels on the left and right sides of

a pedestrian's head often do not belong to that pedestrian. In all of these methods, optimised solutions are iteratively estimated by finding the minimum number of pedestrian models, such as rectangles, which cover most of the foreground pixels. The correct result is shown in Figure 3-6 (c). These approaches are often effective but time-consuming [62]. Therefore, a non-iterative approach is proposed to efficiently calculate the occupancy likelihood at each location.

In fact, not only the foreground pixels close to the vertical central axis within a rectangle should have greater weights, but the background pixels close to the left and right edges of the rectangle should also be given some weights. Let r_i^c be the rectangle at the i-th location in camera view c and is defined by $\{u_i^c, v_i^c, H_i^c, W_i^c\}$, where u_i^c and v_i^c are the horizontal and vertical coordinates of the i-th location in camera view c; H_i^c and W_i^c are the height and width of region r_i^c. r_i^c is defined as:

$$r_i^c = \{(u, v) | u \in [u_i^c - W_i^c/2,\ u_i^c + W_i^c/2], v \in [v_i^c, v_i^c + H_i^c]\} \qquad (3.19)$$

By assuming that the foreground pixels for a pedestrian candidate are ideally located in the middle two-thirds of the candidate box, a template for pedestrian matching within r_i^c is defined as:

$$T_i(u, v) = 1 - 3|u|/W_i^c \qquad (3.20)$$

where $u \in [-W_i^c/2, W_i^c/2]$ and $v \in [0, H_i^c]$. Suppose that foreground pixels have positive values and background pixels have negative values, the template is used to reward the foreground pixels when $|u| < W_i^c/3$ and to reward the background pixels when $W_i^c/3 < |u| \leq W_i^c/2$. Figure 3-7 shows the template. When $u = 0$, the weight of the template reaches the maximal value 1. There are two zero-crossings at $|u| = W_i^c/3$, which are the expected borders between foreground and background pixels for a pedestrian. When $|u| = W_i^c/2$, the weight reaches the minimum value -0.5.

Therefore, the template matching response at the i-th location in camera view c is as follows:

$$t_i^c = \sum_{u=-W_i^c/2}^{W_i^c/2} \sum_{v=0}^{H_i^c} [A_i^c(u_i^c + u, v_i^c + v) - 1/2] \times T_i^c(u, v) \qquad (3.21)$$

where $A_i^c \in \{0,1\}^{W \times H}$ is the foreground map masked by R_i^c. $A_i^c(u_i^c + u, v_i^c + v)$ is used to align each pixel in the candidate box with the corresponding pixel in the template. $A_i^c(u_i^c + u, v_i^c + v) - 1/2$ shifts the foreground map values from $\{0,1\}$ to $\{-1/2,1/2\}$, which means the value of foreground pixels is changed to $1/2$ and that of background pixels is changed to $-1/2$. Finally, the normalized occupancy likelihood is obtained by using its maximum response, $t_{i,max}^c$ and its minimum response $t_{i,min}^c$:

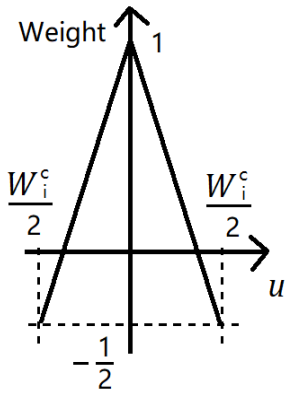


Figure 3-7 The template for pedestrian matching.

$$P(t_i^c \,|L_i) = \frac{t_i^c - t_{i,min}^c}{t_{i,max}^c - t_{i,min}^c} \qquad (3.22)$$

The template matching has the maximum response, when $A_i^c(u, v) = 1$ for $|u - u_i^c| \in [0, W_i^c/3]$, $v - v_i^c \in [0, H_i^c]$ and $A_i^c(u, v) = 0$ otherwise. That is, the central part of the template is covered by foreground pixels and the border part is covered by background pixels. The maximum value is:

$$t_{i,max}^c = \sum_{u=-W_i^c/2}^{W_i^c/2} \sum_{v=0}^{H_i^c} |T_i^c(u, v)| \,/2 \qquad (3.23)$$

The template matching has the minimum response, when $A_i^c(u, v) = 1$ for $|u - u_i^c| \in [W_i^c/3, W_i^c/2]$, $|v - v_i^c| \in [0, H_i^c]$ and $A_i^c(u, v) = 0$ otherwise. That is, the central part of the template is covered by background pixels and the border part is covered by foreground pixels. The minimum value is:

$$t_{i,min}^{c} = -t_{i,max}^{c} \tag{3.24}$$

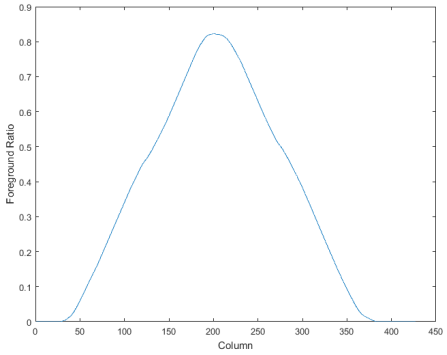
Therefore, Eq. (3.22) can be rewritten as:

$$P(t_i^c \mid L_i) = \frac{t_i^c + t_{i,max}^c}{2t_{i,max}^c} \tag{3.25}$$

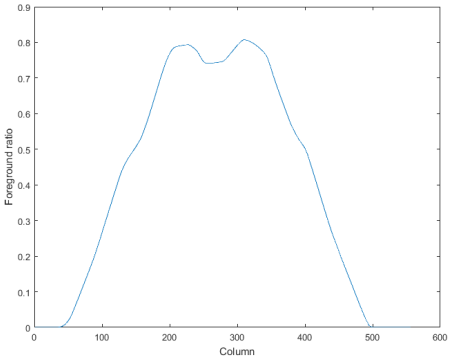
Figure 3-8 shows a comparison of the template matching method and the foreground ratio method. Figure 3-8 (a) and (b) show the foreground silhouettes of side-by-side pedestrians. Figure 3-8 (a) contains two pedestrians and Figure 3-8 (b) contains three pedestrians. Some background pixels appear between every two pedestrians. When a candidate box is shifted from left to right, the foreground ratio at each location is calculated. The foreground ratio results of Figure 3-8 (a) and (b) are shown in Figure 3-8 (c) and (d). By using the foreground ratio, the side-by-side pedestrians cannot be identified because it only focuses on how many foreground pixels are within the candidate box but does not care where they are. When a template is shifted in the same way to calculate the template matching response, the result is shown in Figure 3-8 (e) and (f). There is a significant punishment in the template matching response when background pixels appear in the middle of the template; the template matching response has a high value when foreground pixels are concentrated in the middle of the candidate box and background pixels are around them. The two pedestrians in Figure 3-8 (a) and three pedestrians in Figure 3-8 (b) are correctly detected as the peaks in the template matching response. It is worth noting that, if a candidate box only contains background pixels, the template matching response approximates to 0.2, which is not the minimum value. The minimum value occurs in the case where background pixels are in the middle of the candidate box and foreground pixels are around them. Therefore, two valleys appear at the left and right sides of the foreground silhouettes, as shown in Figure 3-8 (e) and (f).

(a)

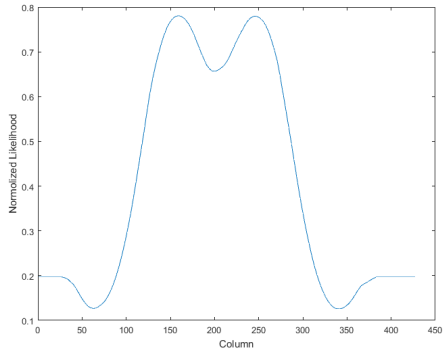
(b)



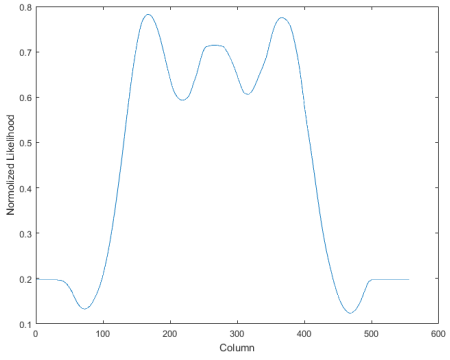
(c)



(d)



(e)



(f)

Figure 3-8 A comparison of the template matching method and the foreground ratio method: (a) a foreground region of two pedestrians, (b) a foreground region of three pedestrian, (c) the foreground ratios of (a), (d) the foreground ratios of (b), (e) the template matching response of (a), and (f) the template matching response of (b).

By finding each local maximum which reaches a threshold on the occupancy likelihood map generated by the template matching response, the locations of pedestrian candidates can be obtained in the top view, as shown in Figure 3-9. When the threshold is set too height, the candidates for short pedestrians or the pedestrians which have broken foregrounds will be filtered out. However, when the threshold is set too low, too many candidates will be generated, which brings difficulty to the next step for pedestrian identification. The threshold should be increased with the camera numbers, because the information from additional camera views makes occupancy likelihoods more confident. The candidates include real pedestrians as well as phantoms. These phantoms are often caused by the foreground support from non-corresponding pedestrians in different views and may have high occupancy likelihoods.

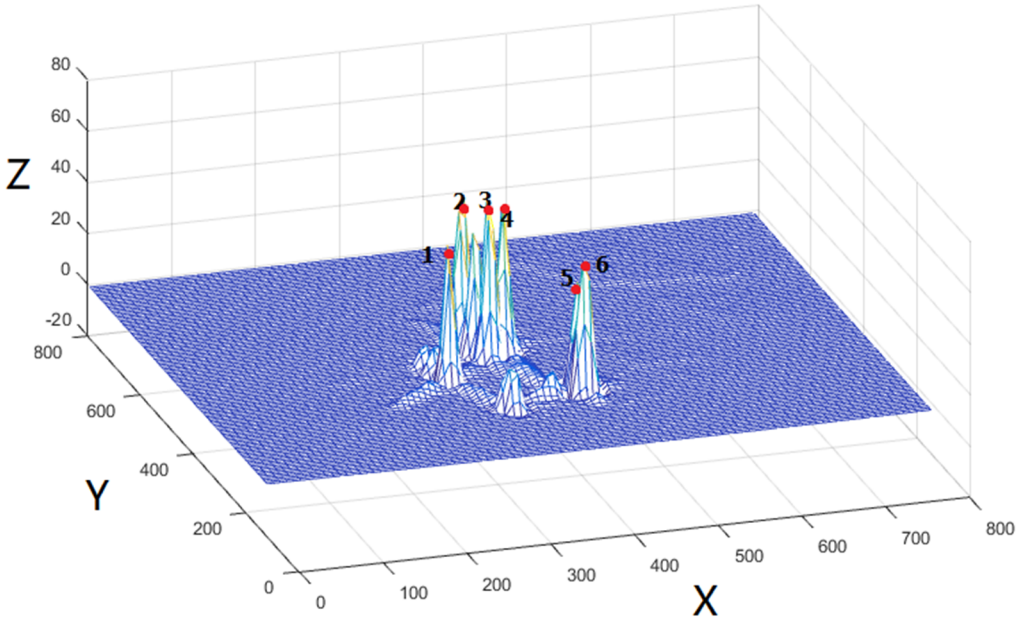


Figure 3-9 The joint occupancy likelihood map generated by the template matching response of Figure 3-4. X and Y axes represent top-view coordinates on the ground plane; Z axis represents the joint occupancy likelihoods.

3.3.3 Repulsive Spatial Sparsity

By finding each local maximum which reaches a threshold in the occupancy likelihood map, the locations of pedestrian candidates can be obtained and recorded as a set $\boldsymbol{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$, where N is the number of pedestrian candidates and each element $\boldsymbol{z}_n = [u_n, v_n]^T$ represents the corresponding coordinate in the top view. However, it does not enforce a certain form of spatial sparsity desired. For example, the algorithm allows two (or more) candidates to be very close to each other. It causes some low local maxima around a larger local maximum. To avoid this situation, a constraint is added, in which every two candidates must be farther from each other than a spatial distance related to the minimum space occupied by a person. This is called Repulsive Spatial Sparsity (RSS)[10].

Suppose $\boldsymbol{z}_n,\ \boldsymbol{z}_k \in \boldsymbol{z}$ and $n \neq k$. Mathematically, the RSS is defined as the distance between every two elements must be larger than a threshold τ:

$$d_{n,k} = \|\boldsymbol{z}_n - \boldsymbol{z}_k\|_2 > \tau \tag{3.26}$$

The value of τ approximates the average width of a standing person. Algorithm 3.1 is used to filter set $\mathbf{z}$ in terms of the Repulsive Spatial Sparsity (RSS).

Algorithm 3.1 Repulsive Spatial Sparsity Filtering

Input: set $\boldsymbol{z}$ for the locations of pedestrian candidates

Output: $\boldsymbol{s}$ − RSS set of $\boldsymbol{z}$

1: Initialize the output set $\boldsymbol{s} = \emptyset$.

2: Find the k-th location, which has the largest occupancy likelihood in **z**.

3: Move $\boldsymbol{z}_k$ from set $\mathbf{z}$ to set **s**.

4: For all remaining $\boldsymbol{z}_n \in \boldsymbol{z}$, if $d_{n,k} < \tau$, remove $\boldsymbol{z}_n$ from set $\boldsymbol{z}$.

5: Repeat steps 2-4 until $\boldsymbol{z} = \emptyset$.

6: **Return** set $\boldsymbol{s}$

3.3.4 Foot and Head Positions

After the RSS filtering, the survived candidates often have high occupancy likelihoods and keep spaces from each other. Then, additional observations from the head and feet within each candidate box are utilised in the occupancy likelihoods to further discriminate pedestrian candidates.

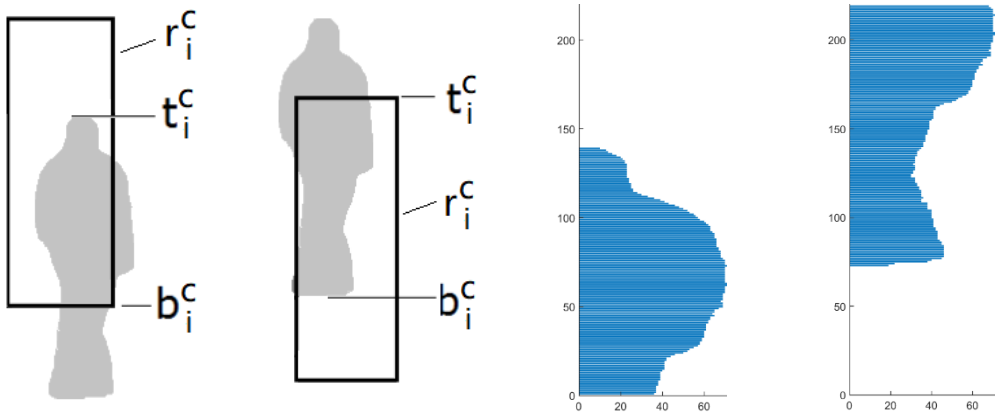
As shown in Figure 3-10, the grey regions represent foreground pixels. r_i^c represents a candidate box. f_i^c and h_i^c represent the vertical coordinates of the observed feet and head of a pedestrian, respectively. If a candidate box only frames the upper body of a pedestrian (see Figure 3-10 (a)), f_i^c is equal to the vertical coordinate of the bottom of the candidate box, and h_i^c is within the candidate box. On the other hand, if a candidate box only frames the lower body of a pedestrian (see Figure 3-10 (b)), h_i^c is equal to the vertical coordinate of the top of the candidate box, and f_i^c is within the candidate box. Due to the measurement errors in foreground extraction and the variation of pedestrians' heights, the observations of the vertical positions of the feet and head, within the rectangle r_i^c, are Gaussian distributed:

$$f_i^c \sim N\left(v_i^c, \left(\beta_f H_i^c\right)^2\right) \tag{3.27}$$

$$h_i^c \sim N(v_i^c + H_i^c, (\beta_h H_i^c)^2) \tag{3.28}$$

where the standard deviations are defined in proportion to the average height H_i^c of the pedestrians at the i-th location in camera view c, and $\beta_f, \beta_h \in (0,1)$.

h_i^c and f_i^c are estimated by using the horizontal projection histogram of the foreground pixels within the rectangle r_i^c. Figure 3-10 (c) and (d) show the horizontal projection histograms of Figure 3-10 (a) and (b), respectively. To calculate h_i^c, the histogram is scanned from top to bottom. When five consecutive rows contain a sufficient number of foreground pixels, which is greater than one tenth of the candidate box width, the vertical coordinate for the first of these five rows is recorded as h_i^c. This step is to filter artefacts and noise in the foreground detection. Similarly, f_i^c can be estimated by scanning the histogram projection form bottom to top.



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

Figure 3-10 A schematic diagram of the variables related to the observations of the head and feet of a pedestrian: (a) a candidate box only frames the upper body of a pedestrian, (b) a candidate box only frames the lower body of a pedestrian, (c) the horizontal projection histogram of (a), and (d) the horizontal projection histogram of (b).

Such a foot position is actually the one closest to the bottom of the candidate box, since the foregrounds at higher locations in the candidate box are the potential foot position of the pedestrian who is standing at the i-th location but is hidden behind the others (see Figure 3-10 (b)). Therefore, suppose the tail probability on the Gaussian distribution is denoted by:

$$Q_G(x) = \int_{\mathrm{x}}^{\infty} P_G(t)dt \tag{3.29}$$

where $P_G(t)$ is the probability density function for $N(0,1)$, the likelihood of such a foot observation can be expressed as:

$$\mathrm{P}(f_i^c|L_i) = Q_G\left(\frac{f_i^c - v_i^c}{\beta_f H_i^c}\right) \tag{3.30}$$

Similarly, the extracted head position as above is actually the one closest to the top of the candidate box, since the foregrounds at lower locations in the candidate box are the potential head position of the pedestrian who is standing in the i-th location but is in front of the others (see Figure 3-10 (a)). The likelihood for the head

observation is expressed as:

$$\mathrm{P}(h_i^c|L_i) = Q_G\left(\frac{v_i^c + H_i^c - h_i^c}{\beta_h H_i^c}\right) \tag{3.31}$$

In this implementation, the foot and head likelihoods are doubled so as to change their ranges from $[0,0.5]$ to $[0,1]$.

By using the observations from heads and feet of each pedestrian in multiple camera views, the joint occupancy likelihoods of pedestrian candidates are more discriminative and more robust. A thresholding operation is then applied to the sparse occupancy likelihood map. Only the local maxima with their joint occupancy likelihoods over a threshold P_T are handed over to the global optimisation stage.

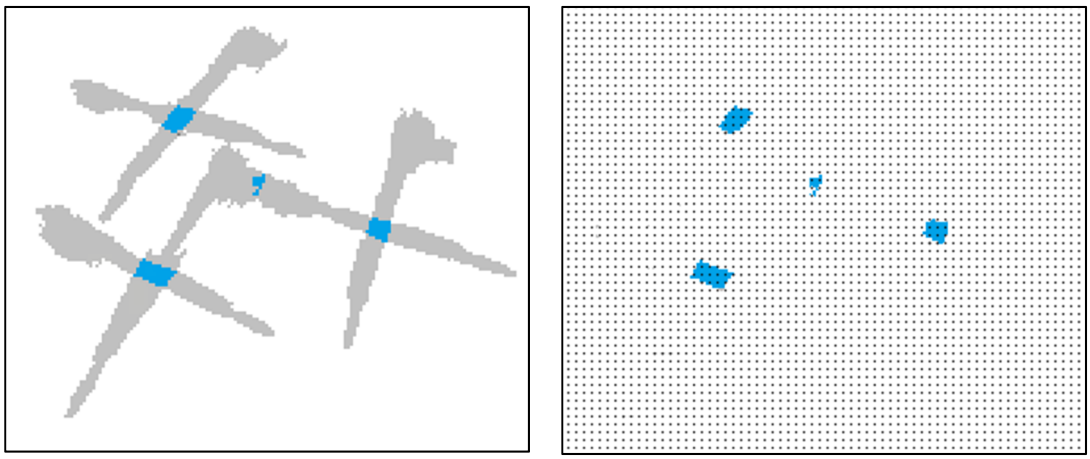
3.3.5 Computation Reduction

If multiple cameras are used, or the grid is very dense, the grid resolution becomes an issue in Eq. (3.17). It affects the computational time of the proposed algorithms. Therefore, the following sections will detail a method on computation reduction.

In multi-camera pedestrian detection, a sound way is to project the foregrounds from multiple camera views to the top view, using ground-plane homographies, and find the foreground intersections. In the top view, the foreground projections from different camera views and belonging to the same object will intersect at the location where the object touches the ground. Therefore, the intersections of foreground projections from multiple camera views are used to reduce the search space for pedestrians. However, as shown in Figure 3-3 (a), if the feet of a pedestrian are lost in the foreground detection in either camera view, there is no foreground intersection for that pedestrian, which leads to a false negative. On the other hand, the observation of a pedestrian's torso is more robust than that of the legs and feet. Even if the result of foreground detection is not perfect, there is still a foreground intersection at the waist of each pedestrian by using the waist-plane homography. Therefore, in this work, the foregrounds are projected to the top view by using the homographies for the waist plane. The foreground intersection map based on the waist-plane homographies is shown in Figure 3-11

(a).

The foreground intersection regions can be obtained by thresholding the foreground intersection map. Figure 3-11 (b) shows the foreground intersection regions on the gird and the dots are the discretised locations. Only the locations within the top-view foreground intersection regions are considered to calculate the occupancy likelihoods. However, there are two challenges when this method is implemented. Firstly, when more than two cameras are used, different locations on the top view are covered by different numbers of cameras. Therefore, the threshold of each location should be decided by the number of cameras covering that location. Secondly, broken foreground projections may lead to broken intersection regions. When a location belonging to a pedestrian is not exactly covered by the broken intersection region, this location will be filtered out and a missed detection occurs. To cope with these two challenges, a robust method for computational reduction is proposed.



(a) (b)

Figure 3-11 Foreground intersections on the top view: (a) The top view foreground intersection map. (b)The top view intersection in the grid.

Suppose $V_i^c \in \{0,1\}$ denotes if the i-th location is within the FOV of camera c. $V_i^c = 1$ if this is true. Then the set and number of the cameras, which cover the i-th location, are $\{c|V_i^c = 1\}$ and $C_i^V = \sum_{c=1}^{C} V_i^c$ respectively. For each camera

view, say camera view c, if $V_i^c = 1$, the i-th location is warped to $(u_{i,h_w}^c, v_{i,h_w}^c,)$ in camera c by using the homography $\mathbf{H}_{h_w}^{t,c}$ for a plane at the waist height h_w, as shown in Figure 3-12. Then the foreground ratio Υ_i^c in the row, at the height of v_{i,h_w}^c within the rectangle r_c^i is calculated. If $\Upsilon_i^c > \Upsilon$, the i-th location is thought of as having foreground supports from camera view c and sets $F_c^i = 1$; otherwise, $F_c^i = 0$. Then the number of the camera views, which have foreground supports to the i-th location, is $C_i^F = \sum_{c=1}^{C} F_i^c$. The set of the locations, which have overlapping foreground supports from multiple views, is $\{i | C_i^F \geq 1\}$.

Since the foreground likelihood $P(A_i^c|L_i) \leq 1$, the joint occupancy likelihood of a location which has foreground supports from more cameras tends to be lower than that of a location which has foreground observations from less cameras. To balance the variation in the number of cameras covering different locations, the joint occupancy likelihood is calculated by extracting the $\mathrm{C_i^V}$-th root, as follows:

$$P(L_i | A_i^1, A_i^2 \dots, A_i^C) = \left(\prod_{\{c | V_i^c = 1\}} P(A_i^c | L_i) \right)^{1/\mathrm{C_i^V}} \tag{3.32}$$

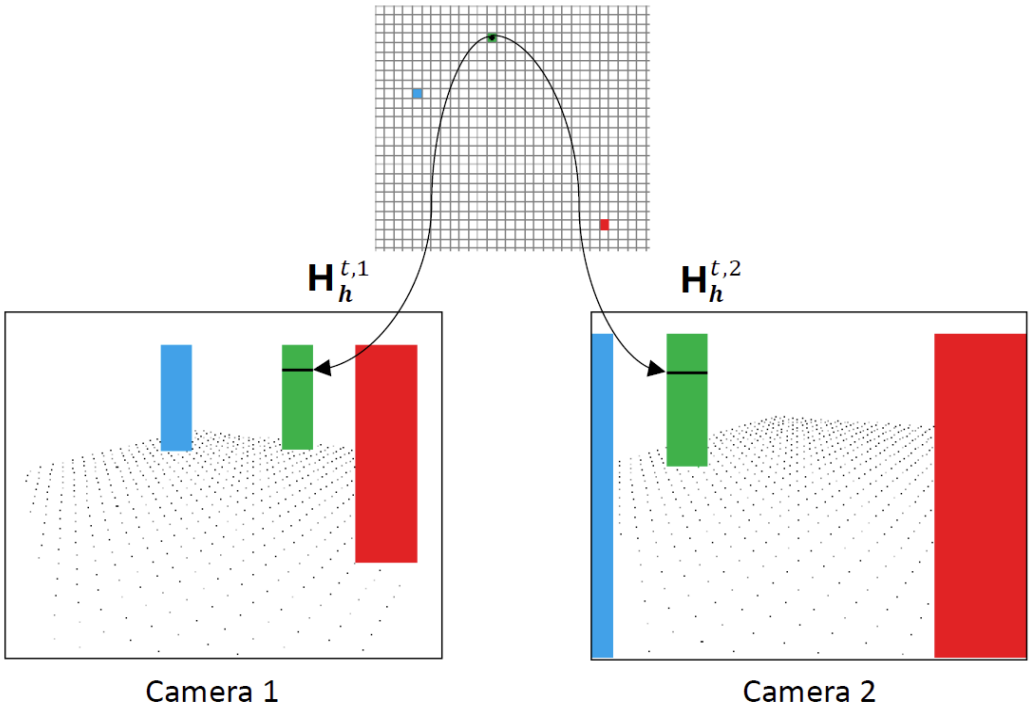


Figure 3-12 Homography mapping, from the top view to both camera views, for a plane at a height.

where $i \in \{k|C_k^F \geq \max(2, C_k^V - 1)\}$; otherwise, it is defined as zero to build a sparse occupancy map. The condition of $i \in \{k|C_k^F \geq \max(2, C_k^V - 1)\}$ is a trade-off between two targets: to filter out false positives and to avoid false negatives in the detection. That is, to focus on the locations which have foreground supports from as many cameras as possible, and to tolerate foreground detection failures or static occlusion in some camera view, since the locations, which have foreground observations from more than one camera, may be either pedestrians' locations or phantoms. If there are only two camera views, then we have to rely on the foreground observations from both cameras.

3.4 Bottom-Up Approach

In this approach, Khan and Shah's method [67] is used to select the locations, in the top view, which may be occupied by pedestrians. The foreground regions, which are extracted in the individual camera views, are projected and overlaid in the top view according to the homographies of a set of planar planes parallel to the ground and at different heights. The intersection regions with heavily overlapped foregrounds are the potential areas where pedestrians may present. However, each of these intersection regions is a range of locations, rather than an accurate location. This becomes obvious when multiple pedestrians are overlapping in the same foreground region in an individual camera view. Therefore, it is necessary to estimate the intersection points for foreground projections, which is carried out in a three-stage process. In the first stage, the essential intersection regions in the multi-camera, multi-plane foreground fusion map are identified and warped back to the individual camera views; In the second stage, the tops of heads and the local maxima of a template matching response, which are horizontally close to a specific warped intersection region, of the corresponding foreground region in a camera view are identified; In the third stage, the vertical line segments, passing the tops of heads and/or the local maxima of the template matching responses, are projected into the top view using homography mapping. The intersections of these line segments are determined as foreground intersection points. The details of this

process are described as follows.

3.4.1 Foreground Intersection Regions

Suppose the foreground region map extracted in camera view c is $F^c \in \{0,1\}^{W\times H}$, where $c \in [1, C]$, C is the number of cameras and $W \times H$ is the image resolution. The foreground region maps of multiple camera views are then projected to and overlaid in the top view, according to the homographies of one of K planes parallel to the ground and at a series of heights h_k, where $k \in [1, K]$. One has:

$$F_k^t = \sum_{c=1}^{C} \mathbf{H}_{h_k}^{\boldsymbol{c,t}} (F^c) \tag{3.33}$$

where $F_k^t \in [0, C]^{W\times H}$. Then the foreground projections at the K heights are further overlaid:

$$F^t = \sum_{k=1}^{K} F_k^t \tag{3.34}$$

An intuitive way to identify the potential locations for pedestrians is to threshold such an overlaid foreground projection map. However, this may become intractable, since pedestrians have different heights and a very low threshold may lead to large foreground intersection regions when pedestrians are crowded. Therefore, Algorithm 3.2 is used to identify the local maxima (high plateaus) in the overlaid foreground projection map. In Algorithm 3.2, the pseudo codes in lines 1-12 are used to identify the local maxima (high plateaus) of the foreground projection map at a single height; those in lines 13-22 are used to identify the maxima (high plateaus) of the foreground projection map at K heights. $k1$ and $k2$ corresponds to the minimum number and the typical number of parallel planes which intersect the height of a pedestrian.

Algorithm 3.2 Segmentation of Foreground Region Intersections

Input: Overlaid foreground maps F_k^t in the top view, $k \in [1, K]$;

Output: A foreground intersection region map;

for each of the K heights, say the k-th height, **do**
  for c = C to 2 **do,** $F_{k,c}^t = \{(u,v)|F_k^t(u,v) = c\}$; **end for**
  for c = C to 3 **do**
    for each intersection region in $F_{k,c}^t$ **do**
      Calculate its centroid;
      If the centroid is within an intersection region in $F_{k,c-1}^t$ **then**
        Remove that intersection region in $F_{k,c-1}^t$;
      end if
    end for
  end for
  Generate the essential intersection regions, i.e. $F_k^t = \bigcup_{c=2}^{C} F_{k,c}^t$;
end for
Overlay the essential intersection regions at K heights $F^t = \sum_{k=1}^{K} F_k^t$;
for k = k1 and k2, where $\mathrm{k1}, \mathrm{k2} \in [1, \mathrm{K}]$ and $\mathrm{k1} < \mathrm{k2}$, **do**
  $F_{(k)}^t = \{(u,v)|F^t(u,v) = k\}$;
end for
for each intersection region in $F_{(k2)}^t$ **do**
  Calculate its centroid;
  If the centroid is within an intersection region in $F_{(k1)}^t$ **then**
    Remove that intersection region in $F_{(k1)}^t$;
  end if
end for
Determine intersection regions as $I^t = \bigcup_{k=k1,k2} F_{(k)}^t$;

Figure 3-13 shows an example of the generation of the essential intersection regions at a height. In Figure 3-13 (a) the foregrounds are projected from three camera views. Two of the camera views contain two pedestrians in their FOV

individually, and the third camera only contains one pedestrian in its FOV. If the threshold is set to two layers, a large intersection will be generated, as shown in Figure 3-13 (b). However, if the threshold is set to three layers, the intersection of the pedestrian which only appears in two camera view is lost, as shown in Figure 3-13 (c). Therefore, by analysing the inclusion relation between each intersection regions generated by using different threshold, the essential intersection regions which are the local maxima (high plateaus) can be found.

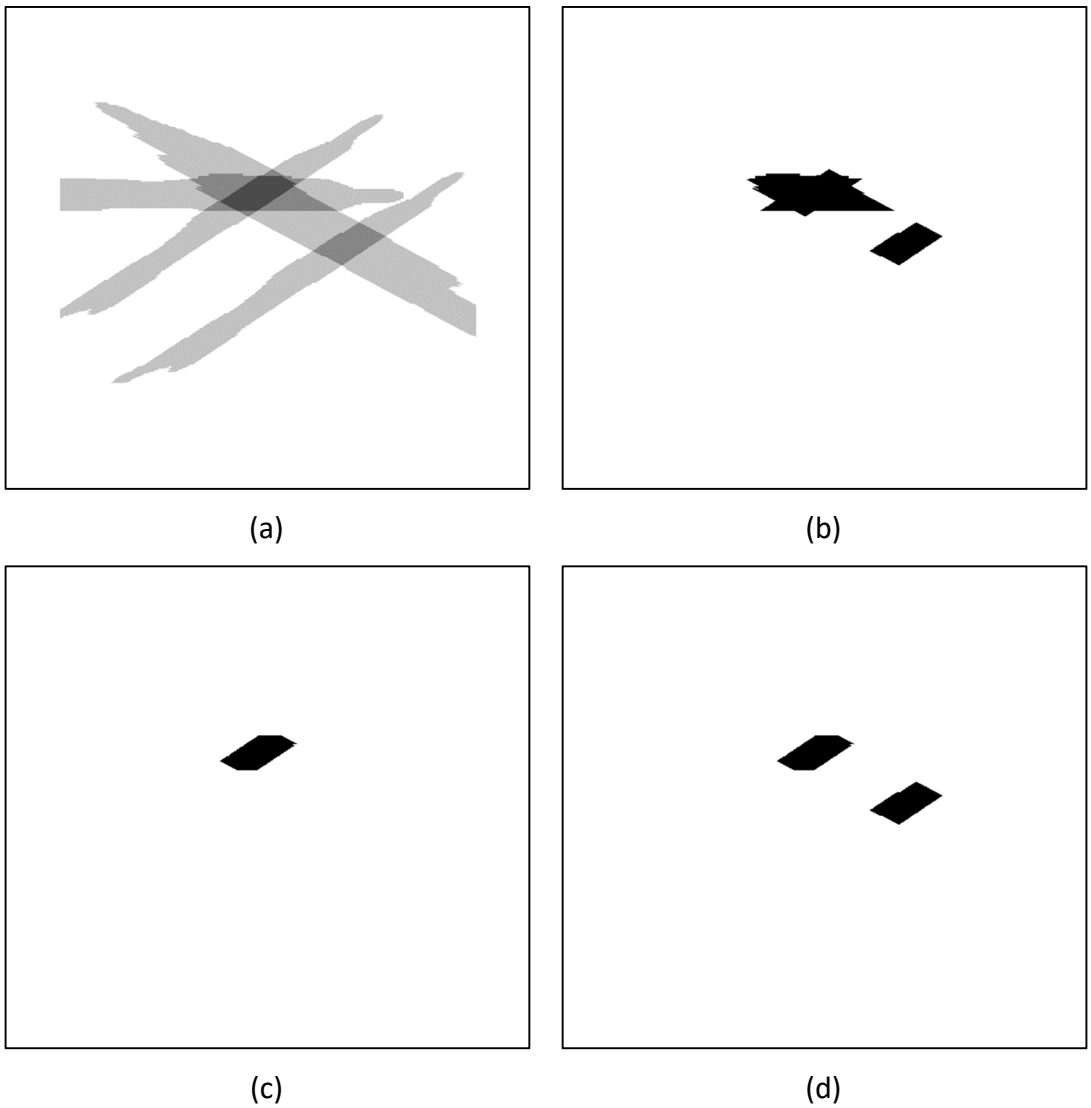


Figure 3-13 An example of the generation of the essential intersection regions at a height: (a) overlaid foreground projection map at a height, (b) after threshold (a) by 2 layers, (c) after threshold (a) by 3 layers, and (d) the essential intersection regions which are the local maxima (high plateaus).

3.4.2 Top of Head Detection and Template Matching

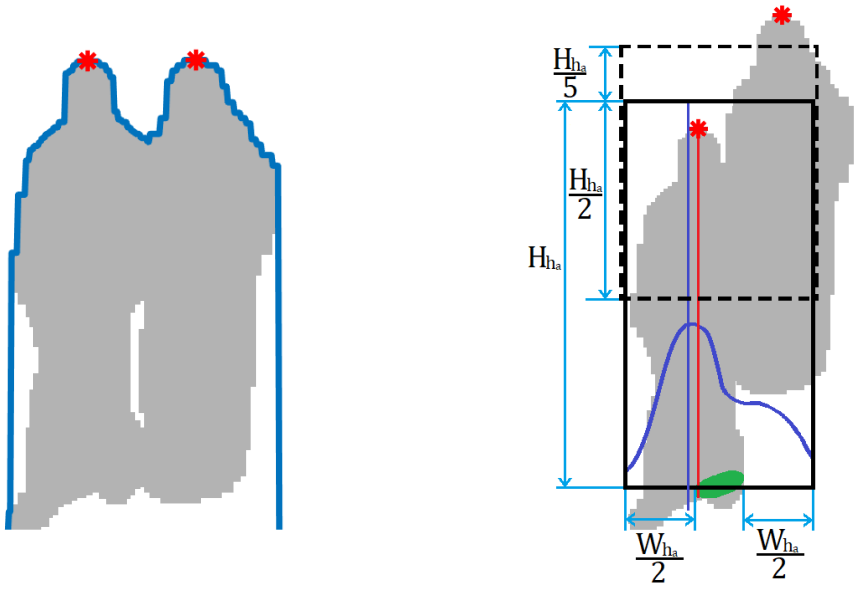
For a given foreground intersection region extracted from step 3.4.1, its centroid and contour are warped back to the individual camera views according to the ground-plane homography. The warped intersection region for a pedestrian is usually located at the bottom of the corresponding foreground region in an individual camera view. The warped regions, each of which is well below the bottom of the corresponding foreground region, are phantoms in front of pedestrians in the top view and are therefore filtered out. The warped regions, each of which is above the bottom of the corresponding foreground region, are either phantoms or pedestrians standing behind other pedestrians in the top view. They are further analysed in joint occupancy likelihoods and global optimisation.

When the centroid and contour of each intersection region are warped back to an individual camera view, a rectangular model for pedestrians is built, which has the average size of the pedestrians standing on the warped centroid. Then the tops of heads and the local maxima of a template matching response are estimated from the foreground silhouette enclosed by this rectangle.

To identify the tops of heads of the potential pedestrians standing on the warped centroid, a search window is built. The horizontal range of the search window is extended from the left/right borders of the warped intersection region by the half average width of pedestrians standing at that warped centroid, and the vertical coordinates are corresponding to 50% to 120% the height range of the rectangular model. This is to cope with lower or taller pedestrians. Then for each column of such a search window, a scan for foreground pixels is carried out from top to bottom. If ten consecutive foreground pixels are found for a column, the vertical coordinate of the first identified foreground pixel is recorded. Therefore, the foreground protrusions in the horizontal direction are removed in this way. Then the local maxima of the recorded vertical coordinates correspond to the tops of heads for potential pedestrians.

Figure 3-14 (a) shows a schematic diagram of the top of head detection. The red

asterisks denote the detected tops of heads. The range of the search window is shown as the rectangle of black dashed lines in Figure 3-14 (b), where the green region represents the warped intersection region, and H_{h_a} and W_{h_a} represent the average height and width of pedestrians standing at the warped centroid. In Figure 3-14 (b), only the lower top of head is associated with the warped intersection region.



(a) (b)

Figure 3-14 Schematic diagrams of the top of head detection (a) and the template matching (b).

In the template matching, the template is the same as that used in our top-down approach. It has the same size of the rectangular model, which is the average size of pedestrians standing on the warped centroid. This template is shifted along the horizontal axis passing the warped centroid and within a limited range which is the same with the horizontal range of the search window. The local maxima of the template matching response for each intersection region are then identified. Some local maxima may be aligned well with an identified top of head, since they correspond to the same pedestrian. Therefore, the local maxima of the template match response, which is within a horizontal distance of 1/10 width of the

rectangular model, are removed. Figure 3-14 (b) shows a schematic diagram of the template matching, in which the middle axis of the template is shifted within the rectangle of solid black lines. The response of the template and the local maxima are shown in dark blue. The local maxima in this schematic diagram should be removed because the horizontal distance between it and the associated top of head is too small.

3.4.3 Foreground Intersection Points

For each foreground intersection region in the top view, it has a warped back region in each of the multiple camera views. The vertical lines passing the tops of heads or the survived local maxima of the template matching response are projected to the top view. The intersection points of such projected lines, each of which is from a different camera view, are the locations of potential pedestrians. If there are two camera views only, it is equivalent to the intersection of the two line projections in the top view. If there are more than two camera views, the foreground intersection point in the top view is determined as the point which has the least sum of squared distances to all the line projections from multiple camera views and for the same foreground intersection region.

Suppose the sum of squared distances, D, from a point (x, y) to a set of C lines $a_k x + b_k y + c_k = 0$ is defined as:

$$D(x, y) = \sum_{k=1}^{C} \left(\frac{a_k x + b_k y + c_k}{\sqrt{a_k^2 + b_k^2}} \right)^2 \tag{3.35}$$

Where $k \in [1, C]$, and C is the number of the cameras. The intersection point is therefore defined as:

$$(x_I, y_I) = arg \min_{(x,y)} D(x, y) \tag{3.36}$$

Taking the partial derivatives to x and y respectively, and letting them equal to zero, one has:

$$\frac{\partial \mathrm{D}}{\partial \mathrm{x}} = 2 \sum_{k=1}^{C} \frac{a_k (a_k x + b_k y + c_k)}{a_k^2 + b_k^2} = 0 \tag{3.37}$$

$$\frac{\partial \mathrm{D}}{\partial \mathrm{y}} = 2\sum_{k=1}^{C} \frac{b_k(a_k x + b_k y + c_k)}{a_k^2 + b_k^2} = 0 \tag{3.38}$$

x and y is the solution of the following equations:

$$\begin{aligned} &\sum_{k=1}^{C} \frac{a_k^2}{a_k^2 + b_k^2} x + \sum_{k=1}^{C} \frac{a_k b_k}{a_k^2 + b_k^2} y + \sum_{k=1}^{C} \frac{a_k c_k}{a_k^2 + b_k^2} = 0 \\ &\sum_{k=1}^{C} \frac{a_k b_k}{a_k^2 + b_k^2} x + \sum_{k=1}^{C} \frac{b_k^2}{a_k^2 + b_k^2} y + \sum_{k=1}^{C} \frac{b_k c_k}{a_k^2 + b_k^2} = 0 \end{aligned} \tag{3.39}$$

If one sets:

$$\begin{aligned} A_1 &= \sum_{k=1}^{C} \frac{a_k^2}{a_k^2 + b_k^2} \quad B_1 = \sum_{k=1}^{C} \frac{a_k b_k}{a_k^2 + b_k^2} \quad C_1 = \sum_{k=1}^{C} \frac{a_k c_k}{a_k^2 + b_k^2} \\ A_2 &= \sum_{k=1}^{C} \frac{a_k b_k}{a_k^2 + b_k^2} \quad B_2 = \sum_{k=1}^{C} \frac{b_k^2}{a_k^2 + b_k^2} \quad C_2 = \sum_{k=1}^{C} \frac{b_k c_k}{a_k^2 + b_k^2} \end{aligned} \tag{3.40}$$

The solution is:

$$\begin{aligned} x &= \frac{B_1 C_2 - B_2 C_1}{A_1 B_2 - A_2 B_1} \\ y &= \frac{A_1 C_2 - A_2 C_1}{A_2 B_1 - A_1 B_2} \end{aligned} \tag{3.41}$$

Finally, the foreground intersection points are warped back to each camera view by using the ground-plane homography, and a rectangular model for pedestrians is built, which has the average size of the pedestrians standing on the warped point. The template matching response, head likelihood and foot likelihood for each foreground intersection point in each camera view are calculated based on the same algorithms in the top-down approach. Then, the RSS filter is used to ignore the points which are within a distance from a point with a large joint occupancy likelihood. The survived points become pedestrian candidates.

Chapter 4

Logic Minimisation Approaches

The joint occupancy likelihood is derived separately for each pedestrian candidate. To encode the interaction such as occlusion and grouping between pedestrians, a global optimisation process is carried out for the multi-view pedestrian localisation. Mathematically, it is to solve the following theoretical optimisation problem [10]:

$$\arg\min\|S\|_0 \qquad s.t. \sum_{c=1}^{C}\|F^c - f^c(S)\|_2^2 < \xi \tag{4.1}$$

where $S \in \{0,1\}^{N_S}$ is a vector of candidates, N_S denotes the total number of candidates, and ξ is a threshold. $f^c(S)$ is a binary map when drawing the filled corresponding candidate boxes of proposed S in camera view c. The problem is to use the least number of candidate boxes to cover most foreground pixels and fewer background pixels in all camera views.

Two approaches are proposed in this chapter. These methods utilize the global information of candidate boxes across different camera views. In the first approach, we borrowed the idea from the Quine-McCluskey method [91] [92] which has been used for the minimisation of Boolean functions. In the second approach, an alternative approach to the QM method, the Petrick's method [93], is used for finding the minimum set of pedestrian candidates to cover all the foreground sub-regions of interest. These two methods can quickly find a solution and are suitable for programming. Karnaugh map [94] is also a popular method for minimisation of Boolean functions. It was not chosen in this thesis because it is a chart-based method which is an intuitive method for human but is not suitable for programming. Moreover, when the number of items in the Boolean functions increases, this method will become very complicated.

4.1 Quine-McCluskey Method

The Quine-McCluskey (QM) method was designed for the minimisation of a Boolean function. When this method is used for pedestrian detection, essential candidates are initially identified, each of which covers at least a significant part of the foreground that is not covered by the other candidates. Then non-essential candidates are selected to cover the remaining foregrounds by following an iterative process, which alternates between merging redundant candidates and finding emerging essential candidates. The tabular form of this method makes it readily implemented by a computer programme.

4.1.1 Foreground Decomposition

To facilitate the use of the Quine-McCluskey method, each foreground region is decomposed into sub-regions according to the overlapping relationship of all the candidate boxes associated with that foreground region. The foreground decomposition must make each sub-region as large as possible while ensuring that there is no transition on the overlapping candidate boxes inside the sub-region. Each sub-region must be big enough and contain a significant portion of foreground pixels. A prime candidate chart is introduced to select a minimum set of pedestrian candidates to cover all the foreground sub-regions of interest, which is similar to a prime implicant chart in the Quine-McCluskey method for finding the minimum set of prime implicants to cover all of the minterms.

Suppose there are N pedestrian candidates surviving in the occupancy likelihood filtering. The filled rectangles of these candidates are summed up in an image of size $W \times H$, with weights in powers of 2, in each camera view (say camera c):

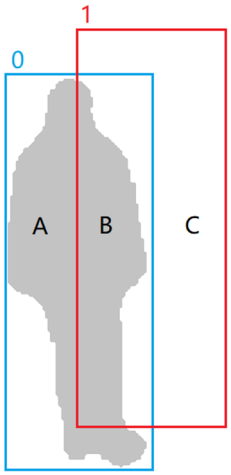
$$B^c = \sum_{n=0}^{N-1} (2^n \times R_n^c) \tag{4.2}$$

where n is the index which is different from the original index of the G locations. Since each sub-region is the overlap of a specific combination of candidate boxes, it

has a unique decimal code in B^c. Such a code corresponds to a N-bit binary code, in which the rightmost bit is bit-0 (the least significant bit). A one in bit-n indicates this sub-region is covered by candidate box n; otherwise, bit-n is zero. By scanning image B^c along with F^c, two histograms with N bins are generated. Each bin reports the pixel number and foreground pixel number in a sub-region, respectively. Then the subregions, which are too small or contain few foreground pixels, are filtered out. It is worth to mention that if there are too many candidate boxes provided in the previous step, the foreground will be decomposed into too many small areas. Then these small areas will be filtered out in the prime candidate chart. The lack of the remaining foreground regions will lead to false detections or missed detections.

A simple example of the prime candidate chart is shown in Figure 4-1. In Figure 4-1 (a), the foreground is decomposed into three sub-regions by two candidate boxes. Candidate 0 in blue is a true pedestrian and candidate 1 in red is a phantom. The three sub-regions are labelled with A, B, and C. To generate the prime candidate chart, each sub-region has a 2-bit binary code which is shown in Figure 4-1 (b). Sub-region A with a binary code 01 is only covered by candidate box 0; sub-region B with a binary code 11 is covered by both candidate boxes 0 and 1; sub-region C with a binary code 10 is only covered by candidate box 1. Then, the prime candidate chart is obtained by using these binary codes. Figure 4-1 (c) is the corresponding prime candidate chart. The foreground sub-regions in all the camera views are listed across the top of the chart, and the pedestrian candidates are listed down the left-hand side. If a sub-region is covered by a candidate box, then a cross is put at the intersection; otherwise, a plus sign is put at the intersection. Each column lists the binary code of a sub-region, in which a plus sign represents a zero and a cross represents a one. The top bit in each column corresponds to the rightmost bit of the corresponding binary code. The bottom bit in each column corresponds to the left-most bit of the binary code. For example, sub-region A has a binary code ’01’, which becomes ‘+X’ when being read from bottom to top in the prime candidate chart in Figure 4-1 (c). In Figure 4-1, since sub-region C does not contain sufficient

foreground pixels, the corresponding column should be removed to simplify the prime candidate chart.



(a)

Sub-region	Decimal code	Binary code	Candidates covering this region
A	1	01	0
B	3	11	1, 0
C	2	10	1

(b)

Candidate	Sub-region		
	A	B	C
0 (BLUE)	X	X	+
1 (RED)	+	X	X

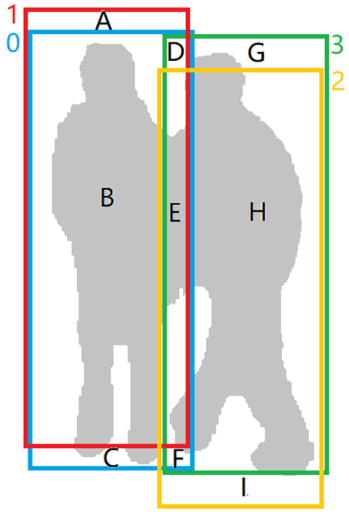
(c)

Figure 4-1 A simple example of prime candidate charts: (a) decomposition of a foreground region into sub-regions, (b) the binary codes of the sub-regions, and (c) the prime candidate chart.

Figure 4-2 shows a more complicated example. Two pedestrians are walking side-by-side and four candidate boxes are generated for them, which is a typical example in multi-view pedestrian detection. Figure 4-2 (a) is the decomposition of the foreground region. More sub-regions are generated when four candidate boxes are involved. Figure 4-2 (b) shows the binary codes of the sub-regions. A 4-bit binary code is used for each sub-region divided by of these four candidate boxes. Figure 4-2 (c) is the prime candidate chart. In this chart, sub-regions A and I are invalid because they do not contain sufficient pixels; sub-regions D and F are invalid because the area of them are too small.

Figure 4-3 is an example which shows the foreground decomposition and prime candidate chart when two camera views are used. Figure 4-3 (a) and (b) show two pedestrians being observed by two cameras. These two cameras are opposite to each other. The pedestrian closer to the camera in Figure 4-3 (a) is farther from the

camera in Figure 4-3 (b). In this case, the two pedestrians are merged into a foreground region in both camera views and three candidate boxes are associated with them. Figure 4-3 (c) shows the binary codes of the sub-regions in the two camera views. The decimal code and binary code of a sub-region can be reused in the other camera view because they are generated separately in each camera view. Figure 4-3 (d) is the corresponding prime candidate chart. In this chart, sub-region A, D, E, F, G and I are invalid. Sub-region F contains a few foreground pixels, but it is still determined to be invalid because it does not contain a significant portion of foreground pixels.



(a)

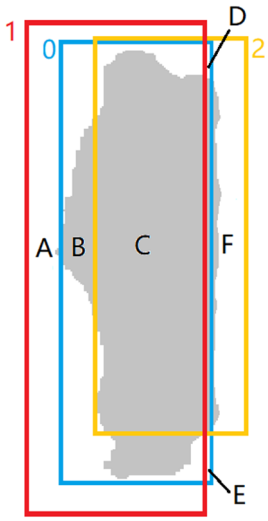
Sub-region	Decimal code	Binary code	Candidates covering this region
A	2	0010	1
B	3	0011	0, 1
C	1	0001	0
D	11	1011	0, 1, 3
E	15	1111	0, 1, 2, 3
F	13	1101	0, 2, 3
G	8	1000	3
H	12	1100	2, 3
I	4	0100	2

(b)

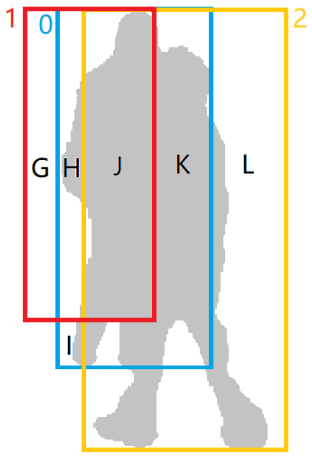
Candidate	Sub-region								
	A	B	C	D	E	F	G	H	I
0 (BLUE)	+	X	X	X	X	X	+	+	+
1 (RED)	X	X	+	X	X	+	+	+	+
2 (YELLOW)	+	+	+	+	X	X	+	X	X
3 (GREEN)	+	+	+	X	X	X	X	X	+

(c)

Figure 4-2 Another example of prime candidate charts: (a) decomposition of a foreground region into sub-regions, (b) the binary codes of the sub-regions, and (c) the prime candidate chart, in which sub-regions A, D, F and I are invalid and are therefore removed afterwards.



(a)



(b)

Camera view	Sub-region	Decimal code	Binary code	Candidates covering this region
1	A	2	010	1
	B	3	011	0, 1
	C	7	111	0, 1, 2
	D	5	101	0, 2
	E	1	001	0
	F	4	100	2
2	G	2	010	1
	H	3	011	0, 1
	I	1	001	0
	J	7	111	0, 1, 2
	K	5	101	0, 2
	L	4	100	2

(c)

Candidate	Sub-region											
	Camera view 1						Camera view 2					
	A	B	C	D	E	F	G	H	I	J	K	L
0 (BLUE)	+	X	X	X	X	+	+	X	X	X	X	+
1 (RED)	X	X	X	+	+	+	X	X	+	X	+	+
2 (YELLOW)	+	+	X	X	+	X	+	+	+	X	X	X

(d)

Figure 4-3 An example of prime candidate charts in two camera views: (a) (b) decomposition of the foreground of two pedestrians in two camera views, (c) the binary codes of the sub-regions, and (d) the prime candidate chart in which sub-regions A, D, E, F, G and I are invalid.

4.1.2 Updating of a Prime Candidate Chart

The prime candidate chart describes the occlusion relationship between the candidate boxes in all camera views. This chart can be updated to identify real pedestrians and phantoms. Algorithm 4.1 and Algorithm 4.2 are used to update the prime candidate chart, in which the functions are defined in Algorithm 4.1 and the main body is described in Algorithm 4.2. The inputs of this algorithm are the original prime candidate chart and a list of joint occupancy likelihoods for all the candidates. The output is a prime candidate chart with the roles of all the candidates assigned.

As shown in Algorithm 4.1, function FINDESSENTIAL is used to identify essential candidates, each of which covers at least a foreground sub-region that is not covered by other candidates. If a given column in the prime candidate chart contains only one X, the corresponding candidate is identified as an essential candidate and labelled as a pedestrian. The X's in the same row and in the columns, which corresponds to the sub-regions covered by this candidate, are replaced by plus signs.

Function MERGE is used to merge redundant X's, which aims to use a minimum set of candidates to cover all the sub-regions. If there is any candidate with its sub-regions fully contained in another candidate, then the contained candidate is removed. If two candidates cover exactly the same sub-regions, the one with a lower joint occupancy likelihood is removed.

The updating procedure of a prime candidate chart is divided into four steps, as shown in Algorithm 4.2. Although this algorithm seems somewhat lengthy, it usually terminates after the first two steps. The remaining steps are designed to cope with the most complicated scenarios which rarely occur. Step 1 is used to filter out the invalid sub-regions. Step 2 is used to find essential candidates which are then labelled with 'PEDESTRIAN'. The X's in the corresponding row and columns are removed afterwards. If there are no X's left in the chart, then the algorithm terminates. Step 3 is used to merge redundant candidates, each of which is contained by another candidate or covers the same sub-regions as another

candidate. Such candidates are not initially redundant but may become redundant when some of their sub-regions are also covered by essential candidates and are removed with the essential candidates. With the redundant candidates removed, it may leave a single X in some columns, then the corresponding candidates become essential candidates and are labelled with ‘PEDESTRIAN’. After their corresponding rows and columns are removed, some candidates may become redundant. Then an iterative process is run between functions MERGE and FINDESSENTIAL until no redundant candidates can be found. If there are still X’s in the chart at this stage, these X’s must be in a cyclic form. That is, each remaining column has more than one X and no row is contained in another row. In this case, step 4 is used to find alternative solutions on a trial basis.

In step 4, a column with the least number of X’s is selected. Then an X in this column is selected as a trial row and the other X’s in the same column are temporarily removed in a cloned chart. Accordingly, the candidate which covers the selected X becomes an essential candidate. This is followed by a process similar to steps 2 and 3. The essential candidates identified in this process are labelled with ‘TRIAL’. Then the next X in the same column is selected as a trial row and the same process is carried out in another cloned chart, which leads to another set of ‘TRIAL’ candidates. This process is repeated until each of the X’s in the selected column has been tested as a trial row. Finally, the set of ‘TRIAL’ candidates with the maximum joint occupancy likelihoods are accepted. The chart is updated according to the corresponding cloned chart and the ‘TRIAL’ candidates are labelled with ‘PEDESTRIAN’.

Algorithm 4.1 Function definition

```
1:  function FINDESSENTIAL(Q, STATUS)
2:      % Q: a prime candidate chart
3:      % STATUS: the role assigned to a candidate
4:      for each column (sub-region) in Q do
5:          if it contains only one X then
6:              The candidate is labelled as STATUS
7:              The X's in this row are removed
8:              The X's in the columns covered by this candidate are removed
9:          end if
10:     end for
11:     return Q
12: end function
13:
14: function MERGE(Q, P)
15:     % Q: a prime candidate chart
16:     % P: a list of joint occupancy likelihoods
17:     Flag=FALSE
18:     for each row (candidate) in Q do
19:         if its X's are the same as another row then
20:             The X's for the candidate with a lower P value are removed
21:             Flag=TRUE
22:         else if its X's are contained by another row then
23:             The X's in this row are removed
24:             Flag=TRUE
25:         end if
26:     end for
27:     return [Q, Flag]
28: end function
```

Algorithm 4.2 The update of a prime candidate chart

Input: A prime candidate chart Q;

Input: A list of joint occupancy likelihoods P;

Output: The prime candidate chart Q with assigned status for each candidate;

```
% step 1: filtering
for each column do
    Remove the X's in this column if the sub-region is invalid
end for

% step 2: essentializing
Q=FINDESSENTIAL(Q,'PEDESTRIAN')
if no X's are left in Q then
    return Q
end if

% step 3: merging
repeat
    [Q, Flag]=MERGE(Q,P)
    Q=FINDESSENTIAL(Q, 'PEDESTRIAN')
until Flag==FALSE

% step 4: grouping
while there are X's in Q do
    for all columns that still contain X's do
        Find a column with the minimum number of X's
    end for
    for each of the X's in the selected column do
        Q'=Q
        The other X's in the selected column in Q' are removed
        Q'=FINDESSENTIAL(Q', 'TRIAL')
```

```
27:             repeat
28:                 [Q', Flag]=MERGE(Q',P)
29:                 Q'=FINDESSENTIAL(Q', 'TRIAL')
30:             until Flag==FALSE
31:             Backup Q'
32:             Multiply the P values for all 'TRIAL' candidates
33:         end for
34:         for all the X's in the selected column do
35:             Select the X with the maximum product of P values
36:             Q=Q'
37:             Replace 'TRIAL' with 'PEDESTRIAN'
38:         end for
39:     end while
40:     return Q
```

4.1.3 Examples of Quine-McCluskey Method

Three examples on how Algorithm 4.2 is running are shown in Figure 4-4, Figure 4-5 and Figure 4-6. The prime candidate chart in Figure 4-4 (a) is the same as Figure 4-1 (c). In Figure 4-4 (b) the invalid sub-region C is removed and the X in the corresponding column is replaced with a plus sign. After this step, there is a single X in column A. Candidate 0, the candidate for cell 0A (row 0 and column A), is identified as an essential candidate and labelled with a circle for 'PEDESTRIAN'. Row 0 is then removed by replacing all the X's with plus signs. Column B covered by candidate 0 is also removed. Then, the chart is updated as Figure 4-4 (c), in which no X's are left and the algorithm terminates. Candidate 0 is correctly identified as a pedestrian and candidate 1 is identified as a phantom.

Candidate	Status	Sub-region		
		A	B	C
0		X	X	+
1		+	X	X

(a)

Candidate	Status	Sub-region		
		A	B	C
0	o	X	X	+
1		+	X	+

(b)

Candidate	Status	Sub-region		
		A	B	C
0	o	+	+	+
1		+	+	+

(c)

Figure 4-4 The updating of a prime candidate chart: (a) the original chart, which is the same as Figure 4-1 (c), (b) after step 1 when an invalid sub-region is removed, and (c) after step 2 when essential candidate 0 is removed.

The prime candidate chart in Figure 4-5 (a) is the same as Figure 4-3 (c). In Figure 4-5 (b) the invalid sub-regions A, D, F and I are removed. In this step, there are a single X in columns C and G. Candidate 0 and 3 are identified as essential and labelled with circles for 'PEDESTRIAN'. Row 0 and 3 are then removed. Columns B, E and H covered by these two candidates are also removed. These lead to the chart as shown in Figure 4-5 (c), in which no X's are left. Candidates 0 and 3 are correctly identified as pedestrians and other candidates are identified as phantoms.

The prime candidate chart in Figure 4-6 (a) is the same as Figure 4-3 (d), which is the prime candidate chart for two camera views. In the chart, sub-regions A to F belong to camera view 1 and sub-regions G to L belong to camera view 2. In Figure 4-6 (b) the invalid sub-regions A, D, E, F, G and I are removed. Candidate 2 is identified as essential and labelled with 'PEDESTRIAN', because column L is only covered by this candidate. Row 2 and the corresponding columns C, J and K are removed. Figure 4-6 (c) shows the remaining X's after step 2. Candidate 0 and 1 cover the same sub-regions in columns B and H. Suppose the joint occupancy likelihood of candidate 0 is greater than that of candidate 1, candidate 1 is then merged into candidate 0, as shown in Figure 4-6 (d). Since there is a single X in columns B and H, candidate 0 becomes an essential candidate and is labelled with

'PEDESTRIAN'. Then the chart is updated as in Figure 4-6 (e) and the algorithm terminates. Candidates 0 and 2 are correctly identified as pedestrians and candidate 1 is identified as a phantom.

Candidate	Status	Sub-region								
		A	B	C	D	E	F	G	H	I
0		+	X	X	X	X	X	+	+	+
1		X	X	+	X	X	+	+	+	+
2		+	+	+	+	X	X	+	X	X
3		+	+	+	X	X	X	X	X	+

(a)

Candidate	Status	Sub-region								
		A	B	C	D	E	F	G	H	I
0	o	+	X	X	+	X	+	+	+	+
1		+	X	+	+	X	+	+	+	+
2		+	+	+	+	X	+	+	X	+
3	o	+	+	+	+	X	+	X	X	+

(b)

Candidate	Status	Sub-region								
		A	B	C	D	E	F	G	H	I
0	o	+	+	+	+	+	+	+	+	+
1		+	+	+	+	+	+	+	+	+
2		+	+	+	+	+	+	+	+	+
3	o	+	+	+	+	+	+	+	+	+

(c)

Figure 4-5 The updating of a prime candidate chart: (a) the original chart, which is the same as Figure 4-2 (c), (b) after step 1 when invalid sub-regions A, D, F and I are removed and candidate 0 and 3 are identified as essentials, and (c) after step 2 when essential candidates are removed.

Candidate	Status	Sub-region											
		A	B	C	D	E	F	G	H	I	J	K	L
0		+	X	X	X	X	+	+	X	X	X	X	+
1		X	X	X	+	+	+	X	X	+	X	+	+
2		+	+	X	X	+	X	+	+	+	X	X	X

(a)

Candidate	Status	Sub-region											
		A	B	C	D	E	F	G	H	I	J	K	L
0		+	X	X	+	+	+	+	X	+	X	X	+
1		+	X	X	+	+	+	+	X	+	X	+	+
2	o	+	+	X	+	+	+	+	+	+	X	X	X

(b)

Candidate	Status	Sub-region											
		A	B	C	D	E	F	G	H	I	J	K	L
0		+	X	+	+	+	+	+	X	+	+	+	+
1		+	X	+	+	+	+	+	X	+	+	+	+
2	o	+	+	+	+	+	+	+	+	+	+	+	+

(c)

Candidate	Status	Sub-region											
		A	B	C	D	E	F	G	H	I	J	K	L
0	o	+	X	+	+	+	+	+	X	+	+	+	+
1		+	+	+	+	+	+	+	+	+	+	+	+
2	o	+	+	+	+	+	+	+	+	+	+	+	+

(d)

Candidate	Status	Sub-region											
		A	B	C	D	E	F	G	H	I	J	K	L
0	o	+	+	+	+	+	+	+	+	+	+	+	+
1		+	+	+	+	+	+	+	+	+	+	+	+
2	o	+	+	+	+	+	+	+	+	+	+	+	+

(e)

Figure 4-6 The updating of a prime candidate chart: (a) the original chart, which is the same as Figure 4-3 (d), (b) after step 1 when invalid sub-regions A, D, E, F, G and I are removed, (c) after step 2 when essential candidate 2 is identified, (d) in step 3 when candidate 1 is merged into candidate 0, and (e) after step 3 when candidate 0 is identified as essential.

The next example is more challenging, as shown in Figure 4-7. The original prime candidate chart is shown in Figure 4-7 (a), in which candidate 0 is identified as an essential candidate and labelled with 'PEDESTRIAN' due to cell 0A. Figure 4-7 (b) is the chart after the removal of row 0 and columns A and B. There are no redundant candidates to merge at this stage. The remaining X's are in a cyclic form and each of the remaining columns contains two X's. Column C is selected for the trial. It contains two X's at 1C and 4C. Therefore, two cloned charts are made. In the first cloned chart as shown in Figure 4-7 (b), candidate 1 is considered as a trial row and labelled with an asterisk for 'TRIAL'. Then row 1 and columns C and F, covered by candidate 1, are removed, which leads to Figure 4-7 (c). Since candidates 2 and 4 are contained by candidate 3, the MERGE function gives rise to Figure 4-7 (d). As there is a single X in columns D and E, candidate 3 becomes an essential candidate and is labelled with 'TRIAL'. In the second cloned chart as shown in Figure 4-7 (e), candidate 4 is considered as a trial row and labelled with 'TRIAL'. Then row 4 and columns C and D, covered by candidate 4, are removed, which leads to Figure 4-7 (f). Since candidates 1 and 3 are contained by candidate 2, the MERGE function gives rise to Figure 4-7 (g). As there is a single X in columns E and F, candidate 2 becomes an essential candidate labelled with 'TRIAL'. Therefore, there are two alternative solutions from the trials. One is candidates 1 and 3 labelled with 'TRIAL'. The other is candidates 2 and 4. Suppose the joint occupancy likelihood for the first set of 'TRIAL' candidates is higher than the second one and the first cloned chart is used to update the prime candidate chart, as shown in Figure 4-7 (h).

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	X	X	+	+	+	+
1		+	X	X	+	+	X
2		+	+	+	+	X	X
3		+	+	+	X	X	+
4		+	+	X	X	+	+

(a)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	X	+	+	X
2		+	+	+	+	X	X
3		+	+	+	X	X	+
4		+	+	X	X	+	+

(b)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	+	+	+	+
2		+	+	+	+	X	+
3		+	+	+	X	X	+
4		+	+	+	X	+	+

(c)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	+	+	+	+
2		+	+	+	+	+	+
3	*	+	+	+	X	X	+
4		+	+	+	+	+	+

(d)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	X	+	+	X
2		+	+	+	+	X	X
3		+	+	+	X	X	+
4	*	+	+	X	X	+	+

(e)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	+	+	+	X
2		+	+	+	+	X	X
3		+	+	+	+	X	+
4	*	+	+	+	+	+	+

(f)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	+	+	+	+
2	*	+	+	+	+	X	X
3		+	+	+	+	+	+
4	*	+	+	+	+	+	+

(g)

Candidate	Status	Sub-region					
		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	o	+	+	+	+	+	+
2		+	+	+	+	+	+
3	o	+	+	+	+	+	+
4		+	+	+	+	+	+

(h)

Figure 4-7 The updating of a prime candidate chart: (a) the original chart with essential candidate 0 being identified, (b) the first cloned chart with candidate 1 being selected as a trial row, (c) the first cloned chart with candidates 2 and 4 being contained by candidate 3, (d) the first cloned chart with candidate 3 becoming an essential candidate, (e) the second cloned chart with candidate 4 being selected as a trial row, (f) the second cloned chart with candidates 1 and 3 being contained by candidate 2, (g) the second cloned chart with candidate 2 becoming an essential candidate, and (h) the first cloned chart is used to update the prime candidate chart due to its higher joint occupancy likelihood.

4.2 Petrick's Method

In the previous section, prime candidate charts are used to identify pedestrians and phantoms from pedestrian candidates. In this section, an alternative approach to the QM method, the Petrick's method, is used for finding the minimum set of pedestrian candidates to cover all the foreground sub-regions of interest.

4.2.1 Petrick Functions

To facilitate the Petrick's method, each foreground region is also decomposed into sub-regions, as in the QM algorithm. Based on B^c in Eq. (4.2), the binary code for the s-th sub-region is recorded as $b^s = (b_1^s b_2^s \dots b_N^s)$, where N is the number of pedestrian candidates. Each bit in the code is associated with a particular candidate and is initialized to zero. When a sub-region is covered by a candidate box, the corresponding bit is set to one. Suppose $\mathrm{P}_i \in \{0,1\}$ represents the event that the i-th candidate is a pedestrian ($\mathrm{P}_i = 1$) or a phantom ($\mathrm{P}_i = 0$), the Petrick function P^s for the s-th sub-region is shown as follows:

$$\mathrm{P}^s = \mathrm{P}_1 \cdot b_1^s + \mathrm{P}_2 \cdot b_2^s + \cdots + \mathrm{P}_N \cdot b_N^s \tag{4.3}$$

Since the Petrick function is a Boolean function, a dot in the function represents the logic AND operation and a plus sign represents the logic OR operation. A candidate P_i is removed from this function when the corresponding b_i^s is zero. The remaining candidates are the candidates covering this sub-region. The result $\mathrm{P}^s = 1$ represents at least one P_i's covers this sub-region. $\mathrm{P}^s = 0$ represents none of the candidates cover this sub-region.

Since each sub-region is thought of as a part of the foreground silhouette of a pedestrian, all the sub-regions should be covered by the candidates. To cover all the foreground regions, the Petrick functions of all the sub-regions should be one. The relationship is logic AND. Therefore, the Petrick function for a camera view, say camera view c, is defined as the product of the Petrick functions of all the sub-regions:

$$\mathrm{P}^{\mathrm{c}} = \mathrm{P}^{1,\mathrm{c}} \cdot \mathrm{P}^{2,\mathrm{c}} \cdot \ldots \cdot \mathrm{P}^{\mathrm{M}^{\mathrm{c}},\mathrm{c}} \tag{4.4}$$

where M^c is the number of the sub-regions in camera view c. Similar to the Petrick function for a sub-region, the Petrick function P^c is equal to one only when $\mathrm{P}^{s,c} = 1$ for $\mathrm{s} \in [1, \mathrm{M}^{\mathrm{c}}]$, that is, each sub-region in this camera view is covered by at least one candidate. Then, this function is extended to the multiple camera views, which is the product of the Petrick functions for all the camera views:

$$\mathrm{P} = \prod_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{P}^{\mathrm{c}} = \prod_{\mathrm{c}=1}^{\mathrm{C}} \prod_{\mathrm{s}=1}^{\mathrm{M}^{\mathrm{c}}} \mathrm{P}^{\mathrm{s,c}} \tag{4.5}$$

Some examples are shown in Figure 4-8. Figure 4-8 (a) is the same as Figure 4-1 (a) and Figure 4-8 (b) shows the valid sub-regions with their Petrick functions. In Figure 4-8 (a), two candidate boxes are associated with a foreground region and divide the foreground region into three sub-regions {A, B, C}. Sub-region C is invalid because it does not contain any foreground pixel. Therefore, the Petrick function for this example is:

$$\mathrm{P} = \mathrm{P}^{\mathrm{A}} \cdot \mathrm{P}^{\mathrm{B}} = \mathrm{P}_0 \cdot (\mathrm{P}_0 + \mathrm{P}_1) \tag{4.6}$$

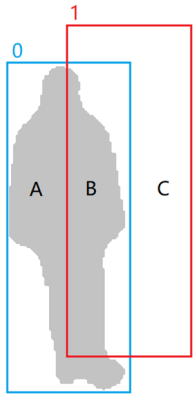
As shown in the Petrick function, P is one only when both P^{A} and P^{B} are one.

Figure 4-8 (c) is the same as Figure 4-2 (a), and Figure 4-8 (d) shows the valid sub-regions with their Petrick functions. Since sub-regions A, D, F and I are invalid, the Petrick function is as follows:

$$\begin{aligned} \mathrm{P} &= \mathrm{P}^{\mathrm{B}} \cdot \mathrm{P}^{\mathrm{C}} \cdot \mathrm{P}^{\mathrm{E}} \cdot \mathrm{P}^{\mathrm{G}} \cdot \mathrm{P}^{\mathrm{H}} \\ &= (\mathrm{P}_0 + \mathrm{P}_1) \cdot \mathrm{P}_0 \cdot (\mathrm{P}_0 + \mathrm{P}_1 + \mathrm{P}_2 + \mathrm{P}_3) \cdot \mathrm{P}_3 \cdot (\mathrm{P}_2 + \mathrm{P}_3) \end{aligned} \tag{4.7}$$

Figure 4-8 (e) is the same as Figure 4-3 (a) and (b), and Figure 4-8 (f) shows the valid sub-regions with their Petrick functions. In Figure 4-8 (e), two camera views are involved. Since the valid sub-regions B and C are in camera view 1 and sub-regions H, J, K and L are in camera view2, the Petrick function for this example is as follows:

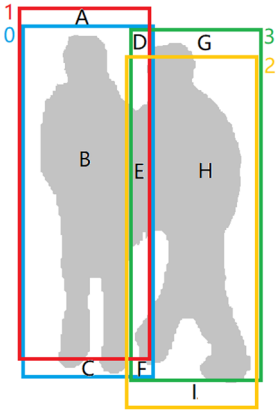
$$\begin{aligned} \mathrm{P} &= (\mathrm{P}^{\mathrm{B}} \cdot \mathrm{P}^{\mathrm{C}}) \cdot (\mathrm{P}^{\mathrm{H}} \cdot \mathrm{P}^{\mathrm{J}} \cdot \mathrm{P}^{\mathrm{K}} \cdot \mathrm{P}^{\mathrm{L}}) \\ &= (\mathrm{P}_0 + \mathrm{P}_1) \cdot (\mathrm{P}_0 + \mathrm{P}_1 + \mathrm{P}_2) \cdot (\mathrm{P}_0 + \mathrm{P}_1) \\ &\quad \cdot (\mathrm{P}_0 + \mathrm{P}_1 + \mathrm{P}_2) \cdot (\mathrm{P}_0 + \mathrm{P}_2) \cdot \mathrm{P}_2 \end{aligned} \tag{4.8}$$



(a)

Sub-region	Petrick function
A	$P^A = P_0$
B	$P^B = P_0 + P_1$

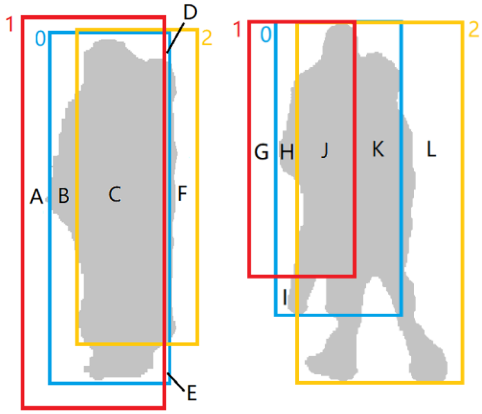
(b)



(c)

Sub-region	Petrick function
B	$P^B = P_0 + P_1$
C	$P^C = P_0$
E	$P^E = P_0 + P_1 + P_2 + P_3$
G	$P^G = P_3$
H	$P^H = P_2 + P_3$

(d)



(e)

Sub-region	Petrick function
B	$P^B = P_0 + P_1$
C	$P^C = P_0 + P_1 + P_2$
H	$P^H = P_0 + P_1$
J	$P^J = P_0 + P_1 + P_2$
K	$P^K = P_0 + P_2$
L	$P^L = P_2$

(f)

Figure 4-8 Examples of Petrick functions: (a), (c) and (e) decomposition of foreground regions into sub-regions; (b), (d), and (f) the Petrick function of each sub-region.

The Petrick functions of all camera views are multiplied to find the Boolean expression in minimised sum-of-products form, in which each product term corresponds to a selected set of candidate boxes covering all the sub-regions. In the Petrick's method for the minimisation of Boolean functions, the product term which contains the minimal number of candidates is chosen as the final solution. However, in the multi-view pedestrian detection, the product term which has the largest joint occupancy likelihood is chosen as the final solution.

4.2.2 Simplification of Petrick Functions

The Petrick function P is equal to one only if all the sub-regions in all camera views are covered by at least one candidate, which is similar to the prime candidate chart where all the columns (sub-regions) should be covered by at least one candidate. However, multiple sets of candidates can satisfy this condition, for example, the set including all the candidates. Hence, the next step is to reduce the Petrick function P to the minimum sum-of-products form to find the optimal solution. To simplify a Boolean expression, three basic rules in Boolean algebra are used:

$$\mathrm{K} + \mathrm{K} = \mathrm{K} \tag{4.9}$$

$$\mathrm{K} \cdot \mathrm{K} = \mathrm{K} \tag{4.10}$$

$$\mathrm{K} + \mathrm{K} \cdot \mathrm{L} = \mathrm{K} \tag{4.11}$$

In these equations, K and L can be either a Boolean variable or a sum-of-products term, which have a value of either one or zero. The plus sign denotes a logic OR operation and the dot denotes a logic AND operation. In addition, the distributive law and associative law can be used in the simplification. By multiplying out and applying Eq. (4.9), (4.10) and (4.11), the Petrick function can be simplified into the minimum sum-of-products form. Each product term in the result represents a solution, which is a set of candidates covering all the sub-regions.

By using the Boolean algebra rules repeatedly, the Petrick function in Eq. (4.6) is simplified:

$$\mathrm{P} = \mathrm{P}_0 \cdot (\mathrm{P}_0 + \mathrm{P}_1)$$

$$= P_0 \cdot P_0 + P_0 \cdot P_1$$

$$= P_0 + P_0 \cdot P_1$$

$$= P_0 \tag{4.12}$$

In the first step, the distributive law in Boolean algebra is used. In the second step, by using Eq. (4.10), $P_0 \cdot P_0$ is combined into P_0. In the third step, Eq. (4.11) is used and finally only P_0 remains. The result shows that candidate 0 is a pedestrian and candidate 1 is a phantom in Figure 4-8 (a). From this example, an important Boolean equation is derived:

$$K \cdot (K + L) = K \tag{4.13}$$

where K and L can be either a single Boolean variable or a sum-of-products term. This equation is similar to the 'FINDESSENTIAL' step in updating a prime candidate chart. When one sub-region is only covered by a single candidate, this candidate becomes an essential candidate, and all the sub-regions covered by this essential candidate are then removed. By using this equation, the simplification of a Petrick function significantly speeds up.

To implement the Petrick's method in a computer program, the Petrick function of each sub-region, which is a sum term in the whole Petrick function, is stored as a 'string' type variable, the size of which is adapted to the number of the candidates which cover that sub-region. Each variable in the Petrick function of a sub-region is encoded by an ASCII character stored in the string. Therefore, the same candidate covering different sub-regions is assigned with a unique ASCII code. The whole Petrick function is stored as an array of 'string's, and the size of the array is equal to the number of the sub-regions. Algorithm 4.3 shows a quick method to simplify a Petrick function. It is divided into four steps and usually terminates after the first step. The following steps are designed to cope with more complicated scenarios which rarely occur.

Algorithm 4.3 Simplification of a Petrick function

Input: A Petrick function P in product-of-sums form

Output: S - Simplified P in the minimised sum-of-products form

```
1:  S = ∅
2:  % step 1 find essential
3:  for each sum term in P do
4:      if it contains only one Boolean variable then
5:          Move this variable to S
6:          Remove any sum term which contains that variable (Eq. (4.13));
7:      end if
8:  end for
9:  % step 2 find group
10: for each sum term in P do
11:     Remove any other sum term which contains it (Eq. (4.13));.
12: end for
13: % step 3 multiply out
14: E is set to the first sum term P^1 in P.
15: For each sum term P^i in P, where i > 1 do
16:     Find all the product terms E = E · P^i;
17:     Merge the repeated variables in each product term in E (Eq. (4.10));
18:     Merge the repeated product terms in E (Eq. (4.9));
19:     for each product term in E do
20:         Remove any other product term which contains it (Eq. (4.11));
21:     end for
22: end for
23: % step 4 compare likelihood
24: Add the product term, with the largest joint occupancy likelihood in E, to S;
25: return S
```

4.2.3 Examples of Petrick's Method

The following example, which corresponds to Figure 4-8 (c), shows the simplification of the Petrick function by only using step 1 in Algorithm 4.3:

$$\begin{aligned} P &= (P_0 + P_1) \cdot P_0 \cdot (P_0 + P_1 + P_2 + P_3) \cdot P_3 \cdot (P_2 + P_3) \\ &= P_0 \cdot (P_0 + P_1 + P_2 + P_3) \cdot P_3 \cdot (P_2 + P_3) \\ &= P_0 \cdot P_3 \cdot (P_2 + P_3) \\ &= P_0 \cdot P_3 \end{aligned} \tag{4.14}$$

In this example, P_0 is first identified as an essential candidate and the sum terms $(P_0 + P_1)$ and $(P_0 + P_1 + P_2 + P_3)$, which contain P_0, are removed. Then, P_3 is identified as an essential candidate and the sum term $(P_2 + P_3)$, which contains P_3, is removed. Therefore, P_0 and P_3 are identified as pedestrians.

For more complicated scenarios, step 2 may be used. The following example is the simplification of the Petrick function of Figure 4-8 (e), in which step 2 is used:

$$\begin{aligned} P &= (P_0 + P_1) \cdot (P_0 + P_1 + P_2) \cdot (P_0 + P_1) \cdot (P_0 + P_1 + P_2) \cdot (P_0 + P_2) \cdot P_2 \\ &= (P_0 + P_1) \cdot (P_0 + P_1) \cdot P_2 \\ &= (P_0 + P_1) \cdot P_2 \\ &= P_0 \cdot P_2 + P_1 \cdot P_2 \end{aligned} \tag{4.15}$$

In this example, P_2 is first identified as an essential candidate and the sum terms $(P_0 + P_1 + P_2)$, $(P_0 + P_1 + P_2)$ and $(P_0 + P_2)$ are removed. Then, two $(P_0 + P_1)$'s are combined. In this step, the Petrick function is not in the sum-of-products form, and then the distributive law is applied. The result of this function has two product terms which are two set of solutions. Suppose the joint occupancy likelihood of candidate 0 is greater than that of candidate 1, then candidates 0 and 2 are identified as pedestrians. This situation corresponds to Figure 4-6 in the QM method. When candidate 2 is identified as an essential candidate and all the sub-regions covered by this candidate are removed. The remaining sub-regions are all covered by candidate 0 and candidate 1. Candidate 1 is merged into candidate 0

because it has a lower joint occupancy likelihood.

The next example of the Petrick function simplification is more challenging. The Petrick function corresponds to the prime candidate chart in Figure 4-7.

$$\begin{aligned} P &= P_0 \cdot (P_0 + P_1) \cdot (P_1 + P_4) \cdot (P_3 + P_4) \cdot (P_2 + P_3) \cdot (P_1 + P_2) \\ &= P_0 \cdot [(P_1 + P_4) \cdot (P_3 + P_4)] \cdot [(P_2 + P_3) \cdot (P_1 + P_2)] \\ &= P_0 \cdot (P_1 \cdot P_3 + P_3 \cdot P_4 + P_4) \cdot (P_1 \cdot P_2 + P_2 + P_1 \cdot P_3 + P_2 \cdot P_3) \\ &= P_0 \cdot (P_1 \cdot P_3 + P_4) \cdot (P_2 + P_1 \cdot P_3) \\ &= P_0 \cdot (P_1 \cdot P_2 \cdot P_3 + P_1 \cdot P_3 + P_2 \cdot P_4 + P_1 \cdot P_3 \cdot P_4) \\ &= P_0 \cdot (P_1 \cdot P_3 + P_2 \cdot P_4) \\ &= P_0 \cdot P_1 \cdot P_3 + P_0 \cdot P_2 \cdot P_4 \end{aligned} \tag{4.16}$$

In the first step, only P_0 is identified as an essential candidate and the sum term $(P_0 + P_1)$ is removed. Then, the associative law and distributive law are used to combine sum terms $(P_1 + P_4) \cdot (P_3 + P_4)$ and $(P_2 + P_3) \cdot (P_1 + P_2)$. After that, the Boolean algebra rules are repeatedly used to simplify the function into a sum-of-products form. The result shows this function has two sets of solutions. In the Petrick's method for the minimisation of Boolean functions, the product term which contains the minimal numbers of candidates is chosen as the final solution. However, in the multi-view pedestrian detection, the joint occupancy likelihood is used to select a set of candidates. The joint occupancy likelihood of each product term is calculated, and the product term which has the largest joint occupancy likelihood is chosen as the final result. This method tends to choose the product term which contains fewer candidates, because the occupancy likelihood of a candidate is usually less than one and the terms containing more candidates are punished in the calculation of joint occupancy likelihood. Once a product term is selected as the final result, the corresponding candidates are labelled as pedestrians.

Chapter 5

Experiments

In this section, the performance of the proposed algorithms is evaluated by using three benchmark datasets which are the PETS2009 City Centre (CC) dataset, PETS2009 S2L1 dataset and EPFL Terrace dataset. They contain multiple pedestrians in multiple calibrated camera views. In the experiments, the top-down approach is combined with the Quine-McCluskey Method and the bottom-up approach is combined with the Petrick's method.

5.1 Experimental Setup

Three benchmark datasets were used to evaluate the proposed approach. Table 5.1 shows a comparison of these datasets.

- The PETS2009 City Centre (CC) dataset is a famous and challenging benchmark dataset to evaluate the performance of multi-view pedestrian detection algorithms. It was captured in an outdoor environment and contains eight camera views (four far-field views and four eye-level views). Each camera view has 795 frames with a frame rate of 7 fps. C1, C2, C3 and C4 have a resolution of 768×576 pixels, and C5, C6, C7 and C8 have a resolution of 720×576 pixels. In each frame of this video sequence, as many as eight pedestrians appear in the overlapping fields of view and with significant occlusion. In the first experiment, two far-field views (C1 and C2) were selected. In the second experiment, two far-field views (C1 and C2) and one eye-level view (C5) were selected. An area-of-interest (AOI) of size 12.2 m $\times$ 14.9 m was used in both experiments. The AOI is completely visible from C1 and C2 but partially visible from C5. The challenge of this dataset is the static occlusions in C1 and inaccurate calibration of C5.

- The PETS2009 S2L1 dataset comes from the same cameras as those in the PETS2009 CC dataset. Different from the PETS2009 CC dataset, S2L1 only contains 7 camera views, in which C2 in the PETS2009 CC dataset is excluded. Each camera view has 795 frames with a frame rate of 7 fps. C1, C3 and C4 have a resolution of 768×576 pixels, and C5, C6, C7 and C8 have a resolution of 720×576 pixels. In the experiment, four camera views were used, including one far-field view (C1) and three eye-level views (C5, C6, and C8) with frequent and severe occlusions. The area-of-interest is the same as that in the PETS2009 CC dataset. It is completely visible from camera view 1 but partially visible from C5, C6 and C8. The challenge of this dataset is the inaccurate calibration of camera C5, C6, and C8.

- The EPFL Terrace dataset is a challenging benchmark dataset which contains 4 eye-level calibrated camera views in a small space on a terrace. The video sequence has 5000 frames with an image resolution of 360 × 288 pixels and a frame rate of 25 fps. It was recorded in a controlled outdoor environment with no background motion and static occlusion. An AOI was defined as a 5.3 m × 5.0 m rectangle. Since up to 8 pedestrians may appear inside the AOI at the same time, the scene is heavily crowded in some time periods. Compared with the PET2009 dataset, the EPFL Terrace dataset has a much higher occlusion rate. In the experiments, two views, three views and four views were selected to evaluate the proposed algorithm, respectively. The challenge of this dataset is the heavy occlusion between pedestrians and the poor foreground extraction due to the automatic white balance of the cameras.

In the experiments of the PETS2009 CC and S2L1 datasets, background subtraction was used for the foreground extraction in each camera view. Each incoming frame is compared with an adaptive background image, and the pixels of significant variation are classified as foreground. The colour of each pixel is

modelled by a Gaussian mixture model, and the Gaussian distributions are updated every frame. After connected component analysis, the foreground pixels are transformed into a foreground region map, which is further filtered by a morphological closing operation to bridge split body parts. The first 395 frames were used to generate the background model, and the remaining 400 frames were used to evaluate the performance of the proposed algorithm. The synthetic top-view images for the two PETS datasets are of 500 × 500 pixels. The average height of pedestrians was set to 1.7 metres in the experiments of the CC dataset and 2.0 metres in that of the S2L1 dataset. The average width of the pedestrians was set to 40% of the average height. The threshold for joint occupancy likelihoods was set as 0.4 in the experiments of the CC dataset and 0.35 in that of the S2L1 dataset. The height for the waist plane was set to 1.2 metres. The scale factor to determine the resolution of the top-view image was set to 50. In the bottom-up approach, a series of planes, which are used for the multi-plane homographies, are at heights from 70 cm to 170 cm with an increment of 20 cm.

Table 5.1 A comparison of the datasets used in experiments

Datasets	Numbers of cameras	Cameras
PETS2009 CC	2	2 far-field views
PETS2009 CC	3	2 far-field views + 1 eye-level view
PETS2009 S2L1	4	1 far-field view + 3 eye-level views
EPFL Terrace	2	2 eye-level views
EPFL Terrace	3	3 eye-level views
EPFL Terrace	4	4 eye-level views

In the experiments of EPFL Terrace dataset, due to the automatic white balance of the cameras, when the fields of view are small, pedestrians close to a camera can significantly change the grey level of the camera view. To cope with this problem, SuBSENSE [95] was used to extract foregrounds. SuBSENSE is a pixel-level

segmentation method using textural features and colour information to detect foregrounds. Therefore, it is not sensitive to the global grey level variation due to the automatic white balance and most illumination variations are ignored. Connected component analysis along with a morphological closing operation was also used in SuBSENSE. Even so, the quality of the foreground detection of the Terrace dataset is not as good as that of the PETS2009 datasets. Since the ground truth of one frame per second is available for this dataset, only the frame numbers at multiples of 25 were used to evaluate the proposed algorithm. The synthetic top-view images for this dataset are of 500 $\times$ 500 pixels. The average height of pedestrians was set to 2.0 metres in the experiments. The average width of the pedestrians was set to 35% of the average height. The threshold for joint occupancy likelihoods was set to 0.60 for two camera views and was set to 0.4 for three and four camera views. The height for the waist plane was set to 1.2 metres. The scale factor to determine the resolution of the top-view image was set to 30. In the bottom-up approach, a series of planes, which are used for the multi-plane homographies, are at heights from 90 cm to 190 cm with an increment of 20 cm.

5.2 Qualitative Results

5.2.1 Top-Down Approach with Quine-McCluskey Method

Three benchmark datasets were used to evaluate the top-down approach with Quine-McCluskey method. In this section, some examples were selected from these datasets to show the detection results, which include joint occupancy likelihoods for pedestrian candidates and the updating of prime candidate charts.

Results on the PETS2009 CC Dataset

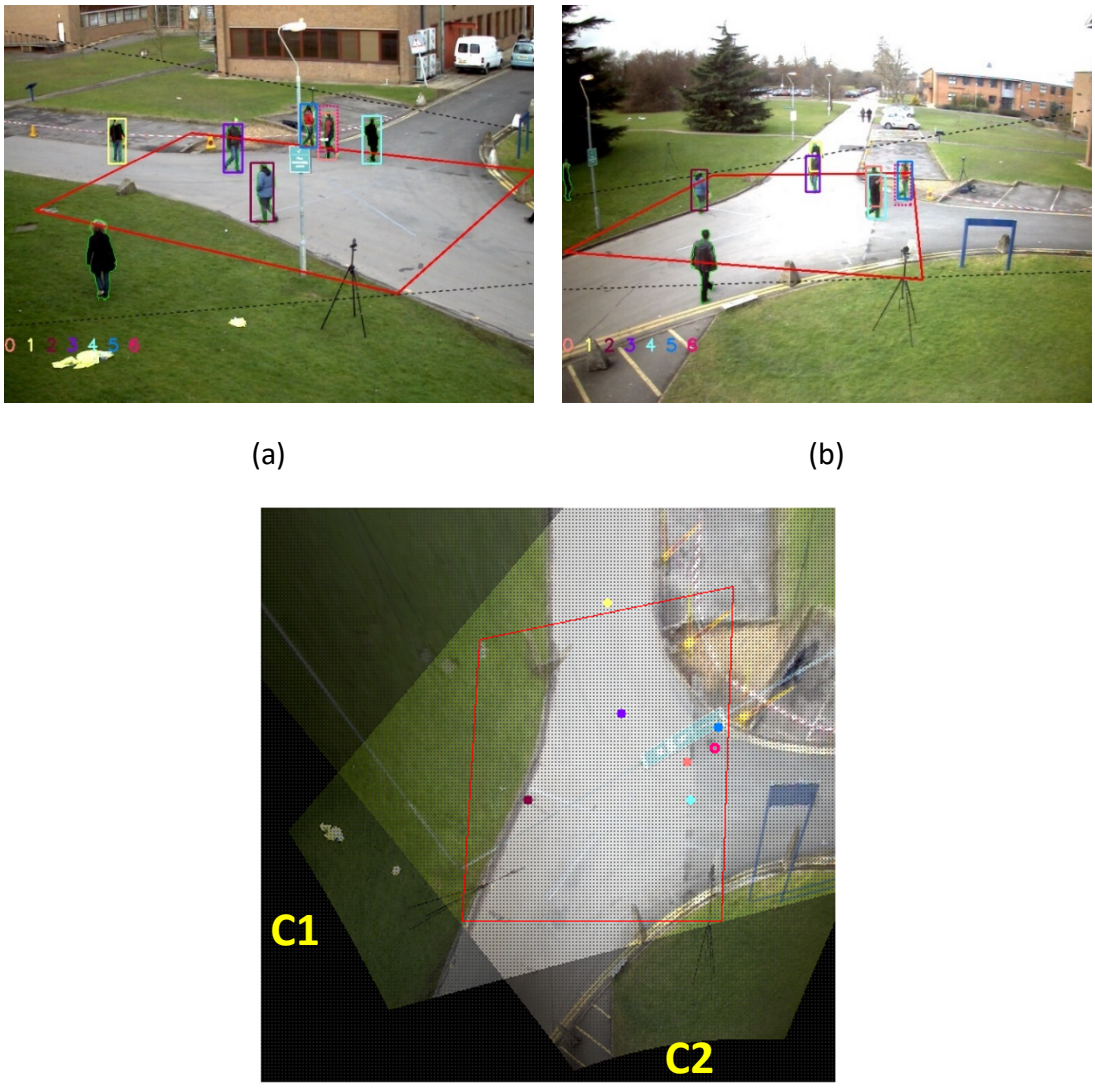
For the PETS2009 CC dataset, two experiments were carried out. In the first experiment, two far-field views (C1 and C2) were selected. In the second experiment, two far-field views (C1 and C2) and one eye-level view (C5) were selected. In the joint occupancy likelihoods and the prime candidate chart, the camera indices 1, 2,

and 3 are corresponding to camera views C1, C2, and C5 in the dataset.

Figure 5-1 shows the detection results at frame 689 on the PETS2009 CC dataset with two camera views, where Figure 5-1 (a) and (b) are the two camera views and Figure 5-1 (c) is a synthetic top view. This frame was selected because it is a simple example in which occlusion occurs in C2 only. The borderlines of the overlapping fields of view are shown as black dashed lines. The region surrounded by red lines is the area-of-interest (AOI). The detection result in the AOI was used to evaluate the performance of the proposed method. The contour of each foreground region is shown in green. Each candidate in the top view, along with its two corresponding candidate boxes in the two camera views, is shown in the same distinguished colour. The IDs (0 to 6) of these candidates are shown at the bottom of both camera views in the same distinguished colours. Each identified pedestrian is labelled with a dot in the top view and a rectangle of solid lines in both camera views, while each phantom is labelled with a circle in the top view and a rectangle of dashed lines in the camera views.

Figure 5-2 shows the joint occupancy likelihoods for the pedestrian candidates at frame 689. The left-most column is the IDs of the pedestrian candidates. In the first row, 1 and 2 are the indices of the two cameras. F is the template matching response $P(t_i^c|L_i)$, T is the head likelihood $P(h_i^c|L_i)$ and B is the foot likelihood $\mathrm{P}(f_i^c|L_i)$. JL is the joint occupancy likelihood, obtained by multiplying the occupancy likelihoods in all the camera views covering the i-th location and extracting the k_i-th root, where k_i is the number of the camera views covering the i-th location. In Figure 5-1 (a), pedestrian candidate 6, represented in magenta, has a predicted top higher than the top of its associated foregrounds in C1 and is therefore penalised by a low T value (0.768) - if this happens, it must correspond to a very short person; It also has a bottom much lower than the bottom of its associated foreground in C2 and is therefore penalised by a low B value (0.564) - if this happens, it must correspond to a pedestrian leaping in the air. Overall, candidate 6 has a low joint likelihood (0.496). Candidate 4, represented in light blue, fits well to its associated foregrounds in both camera views. In C2, the corresponding pedestrian is

merged with another pedestrian. Since the top of the candidate box is supported by foreground observations, there is no penalty on the head likelihood. Finally, this candidate has a higher joint occupancy likelihood (0.756).



(a) (b)

(c)

Figure 5-1 The detection results at frame 689 on the PETS2009 CC dataset: (a) (b) camera views C1 and C2 and (c) the synthetic top view. Each candidate in the top view, along with its corresponding candidate boxes in the two camera views, are in the same distinguished colour. The ID of each candidate is shown at the bottom of the two camera views. Each identified pedestrian is labelled with a dot in the top view and a rectangle of solid lines in the two camera views, while each phantom is labelled with a circle in the top view and a rectangle of dashed lines in the two camera views.

	1F	1T	1B	2F	2T	2B	JL
I0	0.845	1.000	1.000	0.888	1.000	1.000	0.866
I1	0.859	1.000	1.000	0.856	1.000	1.000	0.858
I2	0.836	0.977	1.000	0.853	1.000	1.000	0.835
I3	0.820	1.000	0.942	0.859	1.000	1.000	0.815
I4	0.824	0.917	1.000	0.805	1.000	0.938	0.756
I5	0.833	0.968	1.000	0.764	0.929	1.000	0.756
I6	0.813	0.768	1.000	0.698	1.000	0.564	0.496

Figure 5-2 The joint likelihoods for the pedestrian candidates at frame 689. I0 to I6 indicate the IDs of the candidates. In the first row, 1 and 2 are the indices of the two cameras. F, T and B are the template matching responses, head likelihoods and foot likelihoods, respectively. JL is the joint occupancy likelihood.

Table 5.2 shows the binary codes of the foreground sub-regions at frame 689. The foreground regions of each camera view are decomposed into sub-regions according to the overlapping relationship of all the candidate boxes. Each sub-region in a camera view has a unique binary code, in which a one in the n-th bit (starting from the rightmost bit 0) means this sub-region is covered by candidate box n, and a zero means this sub-region is not covered by candidate box n. In C1, sub-region 2, which has a binary code 000010, is only covered by candidate box 1 in yellow; Sub-region 65, which has a binary code 1000001, is covered by both candidate boxes 0 and 6, which corresponds to the upper part of candidate box 0 in orange or the lower part of the candidate box 6 in magenta. According to Table 5.2, a prime candidate chart at this frame is formed, as shown in Figure 5-3 (a).

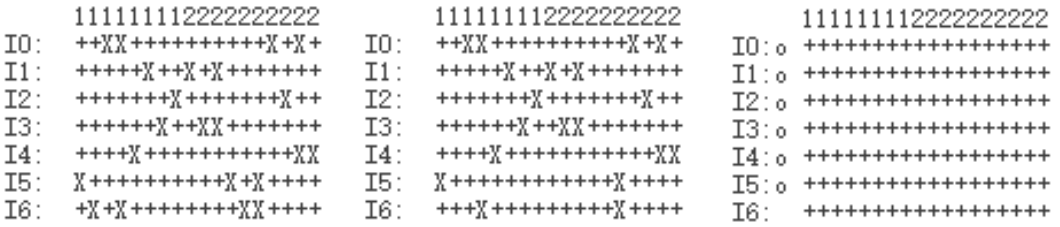
Figure 5-3 shows the prime candidate charts at frame 689. Down the left-hand side of the charts is the list of pedestrian candidates. If a candidate is identified as a pedestrian, then it is labelled with a circle. At the top of each chart, is the indices of the cameras. Each column represents the binary code of a sub-region, in which “+” represents a zero and X represents a one in the corresponding bit for a specific candidate. For example, the first sub-region which has a decimal code 32 and a binary code 0100000. A one in the bit 5 represents this sub-region is covered by candidate 5. Therefore, only an X is placed at the intersection with candidate 5 in

Table 5.2 Sub-regions at frame 689.

Camera	Decimal code	Binary code	Candidates covering this region
1	32	0100000	5
	64	1000000	6
	1	0000001	0
	65	1000001	0, 6
	16	0010000	4
	2	0000010	1
	8	0001000	3
	4	0000100	2
2	2	0000010	1
	8	0001000	3
	10	0001010	1, 3
	32	0100000	5
	64	1000000	6
	96	1100000	5, 6
	1	0000001	0
	4	0000100	2
	17	0010001	0, 4
	16	0010000	4

the first column of the original chart, as shown in Figure 5-3 (a). Figure 5-3 (b) is the chart after step 1, where invalid foreground sub-regions are removed. For example, the sub-region 64 is in the second column of the chart, which is only covered by candidate 6 in C1. It is the top part of candidate box 6 in magenta. This sub-region was removed because it does not contain enough foreground pixels. Figure 5-3 (c) is the chart after step 2 by removing essential candidates. In this step, candidates 0, 1, 2, 3, 4 and 5 are identified as essential candidates. Candidate 6 is not identified as a

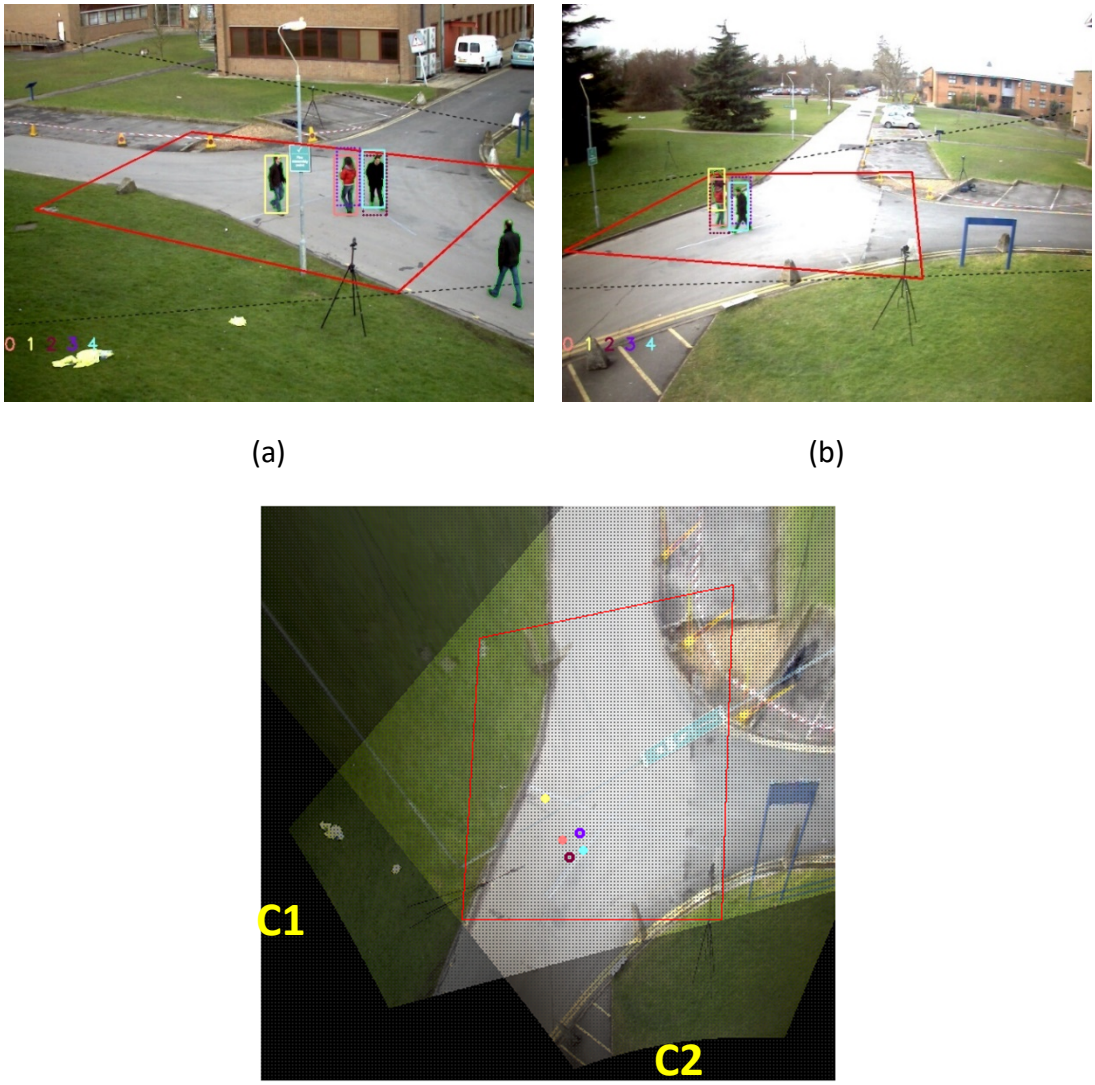
pedestrian, because all its sub-regions are already covered by essential candidates. After the Xs for essential candidates are replaced by plus signs, the links between candidate 6 and its sub-regions no longer exist, although it fits its associated foregrounds reasonably well in both camera views. This avoids more than one candidate being matched to the same pedestrian and keeps the number of identified pedestrians to the minimum. In this case all the candidates are identified without resorting to steps 3 and 4.



(a) (b) (c)

Figure 5-3 The prime candidate chart at frame 689: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified. If a candidate is identified as a pedestrian, then it is labelled with a circle.

Figure 5-4 shows the detection results at frame 465 on the PETS2009 CC dataset with two camera views. This frame was selected because the “merging” step in the QM algorithm was applied. Figure 5-5 shows the joint occupancy likelihoods for the pedestrian candidates at frame 465. Pedestrian candidates 2 and 3 are represented in dark red and purple respectively in Figure 5-4. Pedestrian candidate 2 has a bottom much lower than the bottom of its associated foreground in both camera views and is therefore penalised by low B values. Candidate 3 has a predicted top much higher than the top of its associated foreground in both camera views and is penalised by low T values. Both of them have lower joint occupancy likelihoods, but these are not enough to determine whether these two candidates are phantoms.



(a) (b)

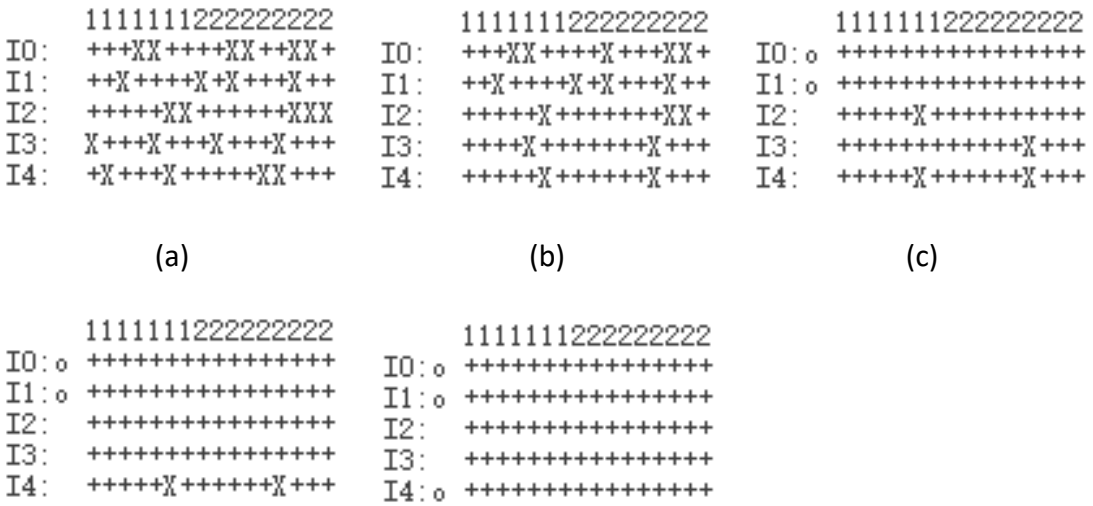
(c)

Figure 5-4 The detection results at frame 465 on the PETS2009 CC dataset: (a) (b) camera views C1 and C2 and (c) the synthetic top view.

	1F	1T	1B	2F	2T	2B	JL
I0	0.841	0.977	1.000	0.900	1.000	1.000	0.860
I1	0.865	1.000	0.948	0.807	0.966	1.000	0.800
I2	0.773	1.000	0.736	0.812	1.000	0.650	0.548
I3	0.822	0.744	1.000	0.728	0.600	1.000	0.517
I4	0.782	0.852	1.000	0.699	0.785	1.000	0.605

Figure 5-5 The joint likelihoods for the pedestrian candidates at frame 465.

Figure 5-6 is the prime candidate chart at frame 465. Figure 5-6 (a) shows the original chart. Figure 5-6 (b) is the chart after step 1, where invalid foreground sub-regions are removed. Figure 5-6 (c) is the chart after step 2 by removing essential candidates. In this step, candidates 0 and 1 are identified as essential candidates. Figure 5-6 (d) shows the chart with the two candidates 2 and 3 being merged into candidate 4. This leaves candidate 4 as an emerging essential candidate. In this step, candidates 2 and 3 are detected as phantoms because the critical sub-regions in these two candidates are fully covered by other candidates, and other invalid sub-regions in these two candidates have already been removed in step 1. Figure 5-6 (e) is the chart with candidate 4 being labelled as essential. At this stage, candidates 0, 1 and 4 are correctly identified as pedestrians and candidates 2 and 3 are identified as phantoms.



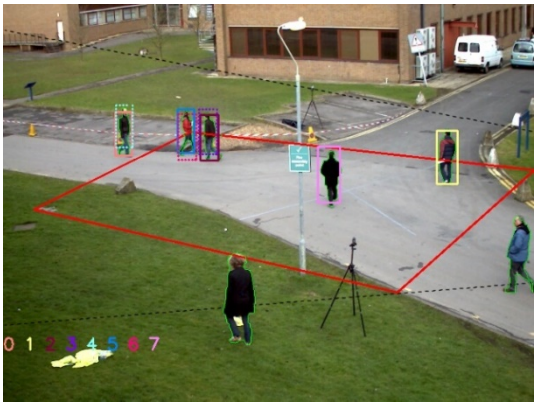
(a) (b) (c)

(d) (e)

Figure 5-6 The prime candidate chart at frame 465: (a) the original table, (b) after step 1 when invalid sub-regions are removed, (c) after step 2 when essential candidates are identified, (d) in step 3, candidates 2 and 3 are merged into candidate 4, and (e) after step 3 when candidate 4 is identified as essential.

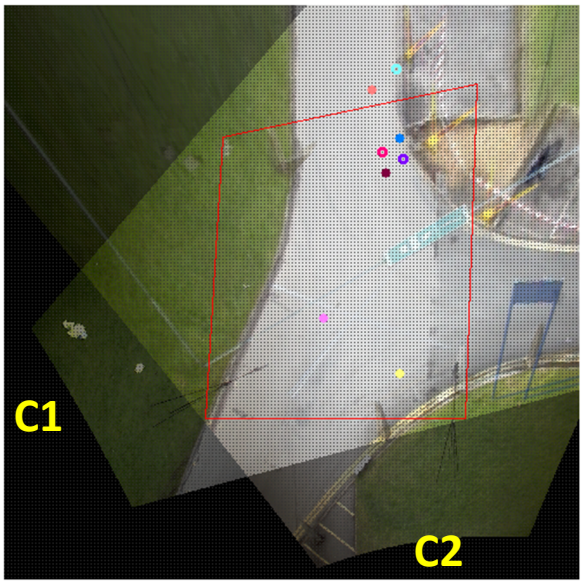
Figure 5-7 shows the detection results at frame 657 on the PETS2009 CC dataset with two camera views. This frame was selected because the proposed algorithm went through all the four steps in the prime candidate chart. Figure 5-8 shows the joint occupancy likelihoods for the pedestrian candidates at frame 657.

Figure 5-9 is the prime candidate chart at frame 657. Figure 5-9 (a) shows the original chart. Figure 5-9 (b) shows the chart after step 1 when invalid sub-regions are removed. Figure 5-9 (c) is the chart after step 2 by removing essential candidates 0, 1, and 7. Figure 5-9 (d) shows the chart with candidate 4 being merged into candidate 3. The remaining X's in the chart are in the cyclic form: each remaining column contains two X's, and no candidate is contained by another candidate. Therefore, a remaining column, column 8 which is covered by candidates 2 and 3, is selected for the trial. Two cloned charts are made. In the first cloned chart, shown in Figure 5-9 (e), candidate 3 is selected as a trial row and labelled with an asterisk for 'TRIAL'. The relevant row of candidate 3 and the columns covered by candidate 3 are removed. Figure 5-9 (f) shows the first cloned chart with the two contained candidates 2 and 5 being merged into candidate 6. This leaves candidate 6 as an emerging essential candidate. Figure 5-9 (g) is the chart with candidate 6 being labelled as 'TRIAL'. In the second cloned chart as shown in Figure 5-9 (h), candidate 2 is selected as a trial row and labelled with 'TRIAL'. The relevant row of candidate 2 and the columns covered by candidate 2 are removed. Figure 5-9 (i) shows the second cloned chart with the two contained candidates 3 and 6 being merged into candidate 5. This leaves candidate 5 as an emerging essential candidate. Figure 5-9 (j) is the chart with candidate 5 being labelled as 'TRIAL'. Figure 5-9 (k) shows the comparison of the joint likelihoods between these two groups of 'TRIAL' candidates. As the joint occupancy likelihood for candidates 2 and 5 is higher than that for candidates 3 and 6, the second cloned chart is accepted to update the prime candidate chart, in which the labels for candidates 2 and 5 are changed from 'TRIAL' to 'PEDESTRIAN' (see Figure 5-9 (l)). At this stage, candidate 0, 1, 2, 5 and 7 are correctly identified as pedestrians.



(a)

(b)

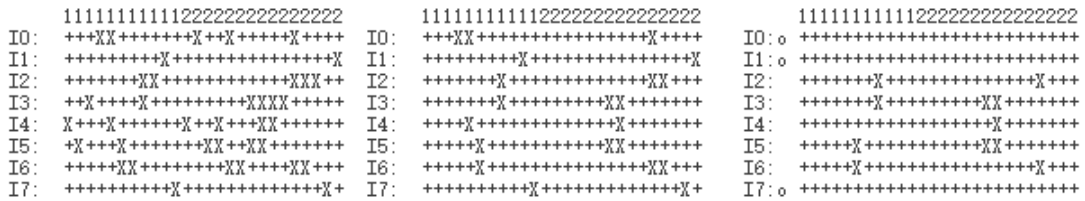


(c)

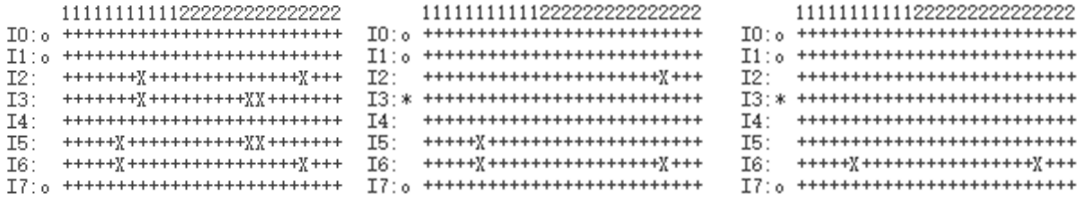
Figure 5-7 The detection results at frame 657 on the PETS2009 CC dataset: (a) (b) camera views C1 and C2 and (c) the synthetic top view.

	1F	1T	1B	2F	2T	2B	JL
I0	0.876	1.000	1.000	0.887	0.948	1.000	0.859
I1	0.872	1.000	1.000	0.837	1.000	0.945	0.831
I2	0.825	1.000	1.000	0.818	1.000	0.923	0.789
I3	0.829	0.863	1.000	0.757	1.000	0.764	0.643
I4	0.882	0.903	1.000	0.706	0.431	1.000	0.492
I5	0.695	0.885	1.000	0.803	0.849	1.000	0.648
I6	0.640	1.000	0.676	0.870	1.000	1.000	0.613
I7	0.800	0.848	1.000	0.644	0.737	1.000	0.568

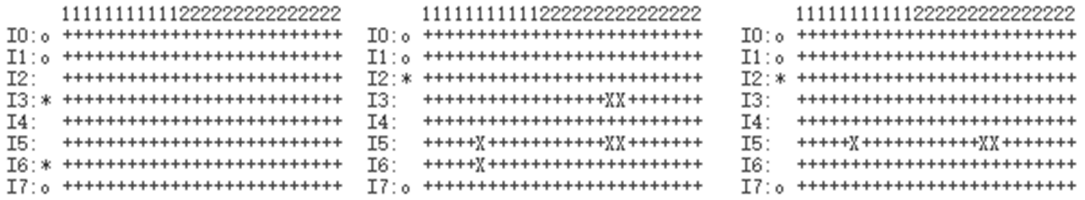
Figure 5-8 The joint likelihoods for the pedestrian candidates at frame 657.



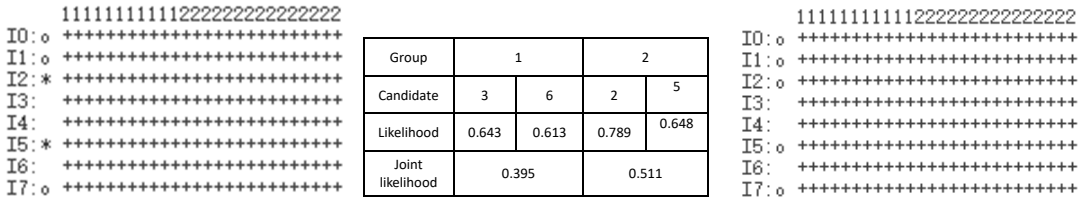
(a) (b) (c)



(d) (e) (f)



(g) (h) (i)

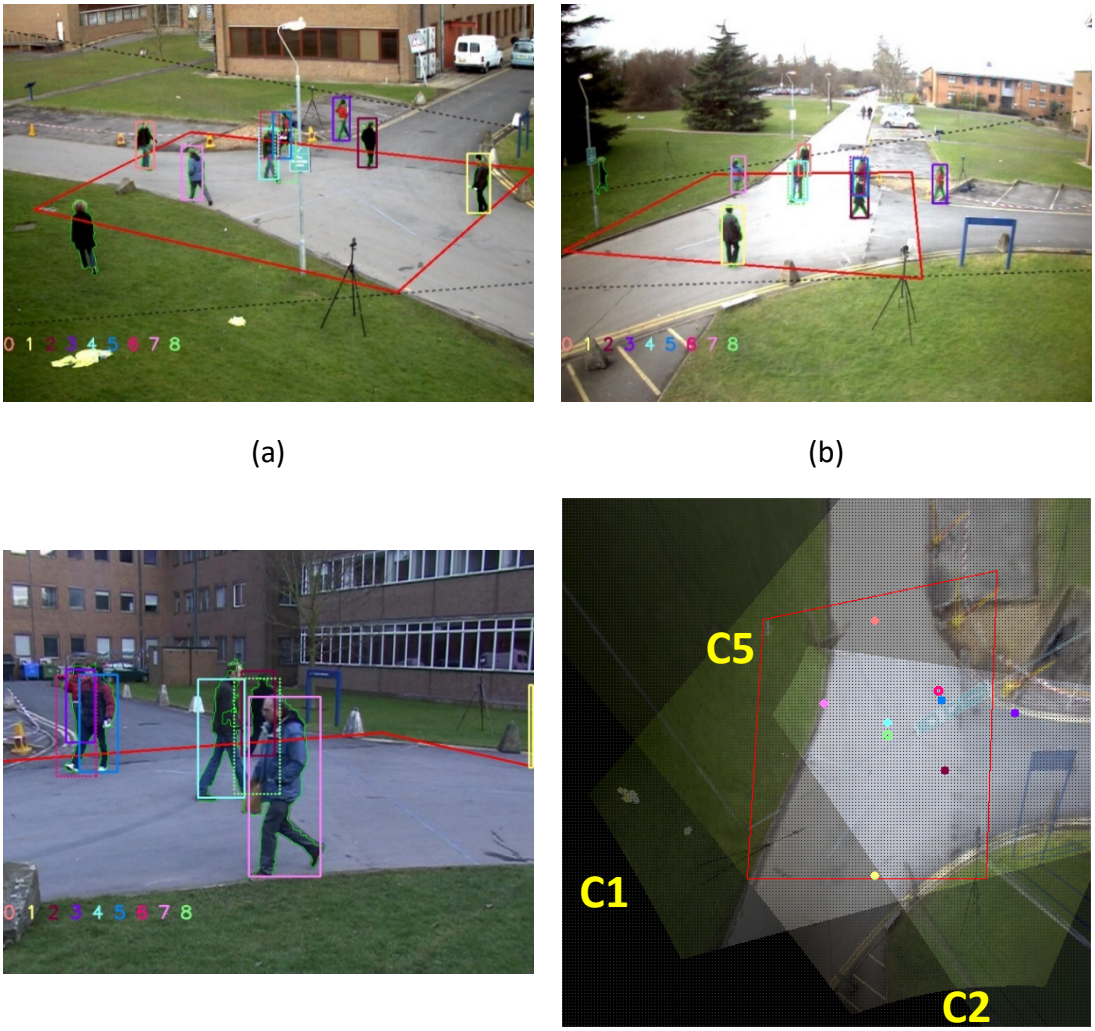


Group	1		2	
Candidate	3	6	2	5
Likelihood	0.643	0.613	0.789	0.648
Joint likelihood	0.395		0.511	

(j) (k) (l)

Figure 5-9 The prime candidate chart at frame 657: (a) the original table, (b) after step 1 when invalid sub-regions are removed, (c) after step 2 when essentials candidates are identified and removed, (d) after step 3 when candidate 4 is merged into candidate 3, (e) the first cloned chart with candidate 3 being a trial row, (f) the first cloned chart with candidates 2 and 5 being merged into candidate 6, (g) the first cloned chart with candidate 6 being identified as essential, (h) the second cloned chart with candidate 2 being a trial row, (i) the second cloned chart with candidates 3 and 6 being merged into candidate 5, (j) the second cloned chart with candidate 5 being identified as essential, (k) the comparison of the joint likelihoods between two groups, and (l) the updated chart according to the second cloned chart.

Figure 5-10 shows the detection results at frame 701 on the PETS2009 CC dataset with three camera views C1, C2 and C5. This frame was selected because an eye-level camera view (C5) was added. The additional camera view not only brings more observations but also disturbance. Due to the inaccurate camera calibration of the third camera view, the locations and heights of the pedestrians are not accurate. For example, the height of candidate box 5 is equal to the corresponding pedestrian in C1 and C2, but it is obviously lower than the pedestrian's head in C5. The foregrounds outside the candidate box may bring false positive detections.



(a) (b)

(c) (d)

Figure 5-10 The detection results at frame 701 on the PETS2009 CC dataset with three camera views: (a) (b) (c) camera views C1, C2 and C5, and (c) the synthetic top view.

Figure 5-11 shows the joint occupancy likelihoods for the pedestrian candidates at frame 701. The template matching response $P(t_i^c|L_i)$, head likelihood $P(h_i^c|L_i)$ and foot likelihood $P(f_i^c|L_i)$ of the third camera view are shown as 3F, 3T and 3B, respectively. When a pedestrian is beyond the field of view of a camera, all the F, T, B values for that camera view are set to one. That camera view is ignored in the calculation of the joint occupancy likelihood. For example, candidate 0 is invisible in the third camera view and its 3F, 3T and 3B values are all set to one. The joint occupancy likelihood of candidate 0 is the square root of the product of all the likelihoods.

Figure 5-12 is the prime candidate chart at frame 701. Figure 5-12 (a) shows the original chart. Figure 5-12 (b) is the chart after step 1, when invalid foreground sub-regions are removed. Figure 5-12 (c) is the chart after step 2 by removing essential candidates. In this step, the candidates 0, 1, 2, 3, 4, 5 and 7 are identified as pedestrians and the candidates 4 and 6 are identified as phantoms. All the pedestrians and phantom in this example are correctly detected.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	JL
I0	0.836	1.000	1.000	0.803	1.000	1.000	1.000	1.000	1.000	0.819
I1	0.824	1.000	1.000	0.813	1.000	1.000	1.000	1.000	1.000	0.819
I2	0.789	0.852	1.000	0.800	1.000	0.938	0.760	0.862	1.000	0.691
I3	0.761	0.937	1.000	0.647	0.931	0.931	0.845	1.000	1.000	0.697
I4	0.882	1.000	1.000	0.761	1.000	1.000	0.607	1.000	1.000	0.742
I5	0.618	1.000	0.890	0.905	0.961	1.000	0.703	1.000	0.969	0.688
I6	0.684	0.503	1.000	0.758	0.894	1.000	0.675	1.000	0.843	0.510
I7	0.749	0.926	1.000	0.667	0.893	1.000	0.701	1.000	1.000	0.662
I8	0.368	1.000	1.000	0.800	1.000	0.878	0.755	1.000	1.000	0.580

Figure 5-11 The joint likelihoods for the pedestrian candidates at frame 701.

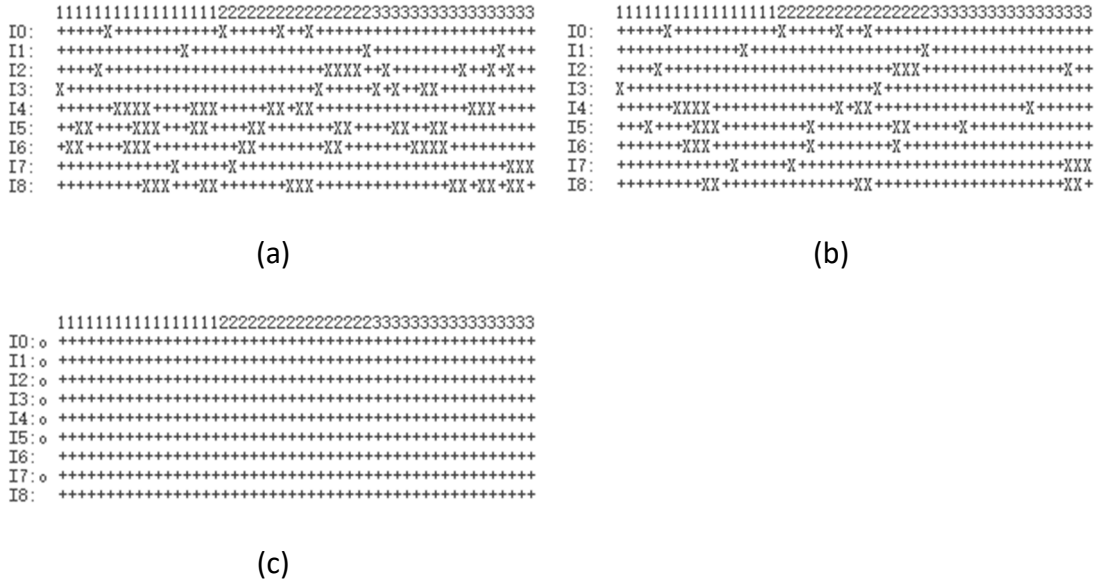
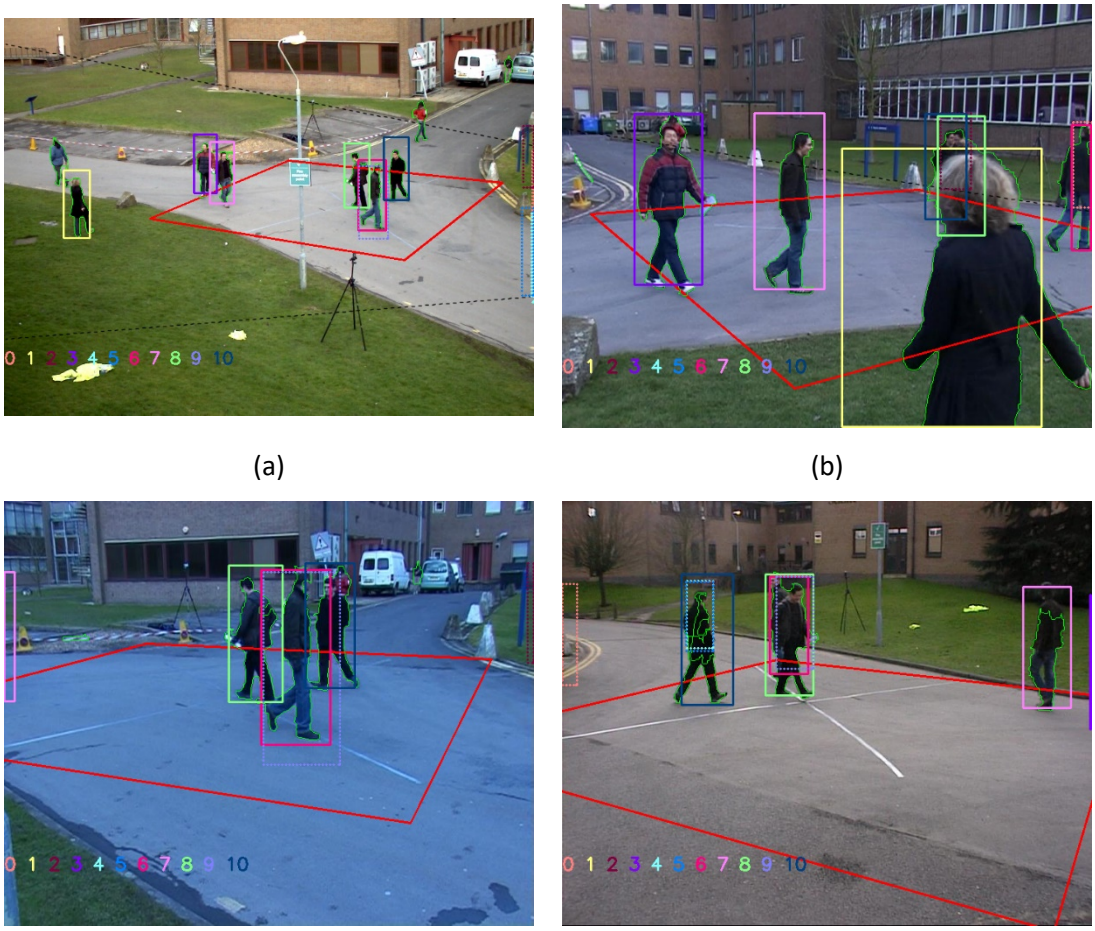


Figure 5-12 The prime candidate chart at frame 701: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified.

Results on the PETS2009 S2L1 Dataset

In the experiment on the PETS2009 S2L1 dataset, one far-field view (C1) and three eye-level views (C5, C6 and C8) were selected. Three eye-level camera views bring more occlusions. The tops of the heads of pedestrians in a camera view are always around the same horizontal line, and the foreground silhouettes of these pedestrians are often merged, which brings a challenge in the pedestrian detection. In the joint likelihoods and prime candidate chart, the camera indices 1, 2, 3 and 4 are corresponding to camera views C1, C5, C6 and C8 in the dataset.

Figure 5-13 shows the detection results at frame 730 on the PETS2009 S2L1 dataset. Because of the inaccurate camera calibration of C5, C6 and C8, the average height of the pedestrians was set to 2 m, and the width of the pedestrians was still set to 0.4 of the average height. That means the candidate boxes are higher and larger to make sure the whole body of a pedestrian is contained in a candidate box. For example, the top of candidate box 7 in pink is much higher than the head of the pedestrian in C1 and C5. However, in C8, the top of the same candidate box is lower than the pedestrian' head due to the inaccurate camera calibration.



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

Figure 5-13 The detection results at frame 730 on the PETS2009 CC dataset with four camera views: (a) (b) (c)(d) camera views C1, C5, C6 and C8, and (e) the synthetic top view.

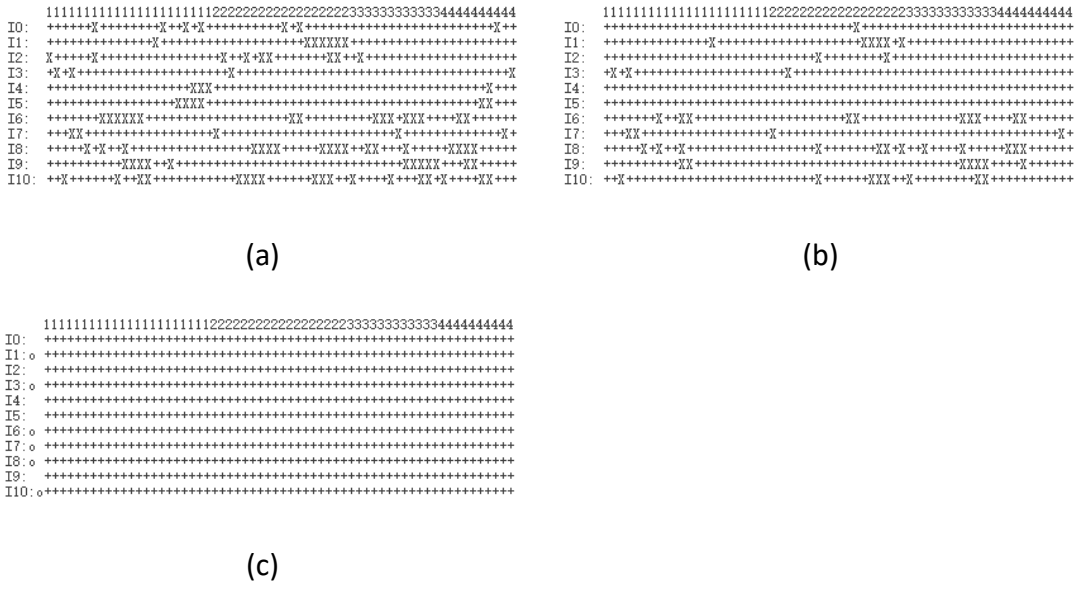
Figure 5-14 shows the joint occupancy likelihoods for the pedestrian candidates at frame 730. All the candidates have a lower template matching response, and most of the candidates are punished by low head likelihoods because of the large candidate box. The quality of the foreground detection also affects the joint occupancy likelihoods. The head and shoulder of the pedestrian corresponding to candidate 7 in C8 were not detected because their colour is the same as the vegetations behind. Therefore, this candidate is punished by a low head likelihood (0.623). The template matching response of candidate 10 in C8 is very low (0.402) because of the poor foreground detection.

Figure 5-15 is the prime candidate chart at frame 730. Figure 5-15 (a) shows the original chart. Figure 5-15 (b) is the chart after step 1, where invalid foreground sub-regions are removed. Figure 5-15 (c) is the chart after step 2 by removing essential candidates. In this step, candidates 1, 3, 6, 7, 8 and 10 are correctly identified as essential candidates

	1F	1T	1B	2F	2T	2B	3F	3T	3B	4F	4T	4B	JL
I0	1.000	1.000	1.000	0.790	0.916	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.724
I1	0.709	0.793	0.955	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.537
I2	1.000	1.000	1.000	0.679	0.602	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.408
I3	0.730	0.705	1.000	0.795	0.964	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.628
I4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.512	0.702	1.000	0.359
I5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.507	0.710	1.000	0.360
I6	0.698	0.904	0.925	1.000	1.000	1.000	0.605	1.000	0.890	0.822	1.000	1.000	0.637
I7	0.703	0.715	1.000	0.671	0.779	1.000	1.000	1.000	1.000	0.725	0.623	1.000	0.491
I8	0.683	0.724	1.000	0.737	0.780	1.000	0.717	0.837	1.000	0.811	1.000	1.000	0.610
I9	0.691	1.000	0.551	1.000	1.000	1.000	0.571	1.000	0.539	0.728	1.000	1.000	0.440
I10	0.695	0.633	1.000	0.724	0.719	1.000	0.697	0.948	1.000	0.402	0.735	0.934	0.452

Figure 5-14 The joint likelihoods for the pedestrian candidates at frame 730.

.



(a) (b)

(c)

Figure 5-15 The prime candidate chart at frame 730: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified.

Results on the EPFL Terrace Dataset

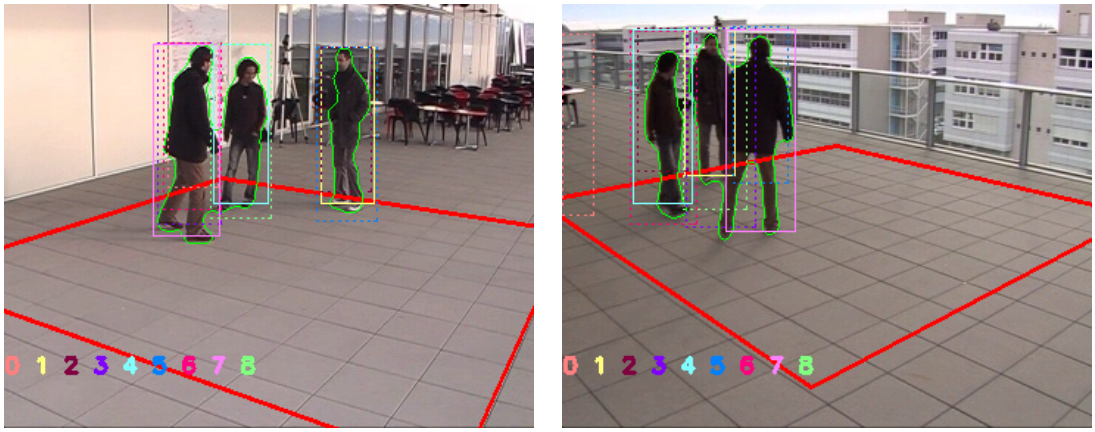
To evaluate the proposed method on the EPFL Terrace dataset, three experiments were designed. Two camera views, three camera views and four camera views were used in the experiments, respectively. By comparing this dataset with the PETS2009 CC dataset, the density of the pedestrians is obviously increased. Without a far-field view, the occlusion rate of the Terrace dataset is much higher than that of the PETS2009 CC dataset. Moreover, the poor foreground extraction is also a challenge. Due to the automatic white balance of the cameras, some background pixels were falsely detected as foregrounds. This may bring false alarms in pedestrian detection. In the joint likelihoods and prime candidate chart, the camera indices 1, 2, 3 and 4 are corresponding to camera views C0, C1, C2 and C3 in the dataset.

Figure 5-16 shows the detection results at frame 825 on the EPFL Terrace dataset with two camera views, where Figure 5-16 (a) and (b) are the two camera views (C0 and C1) and Figure 5-16 (c) is a synthetic top view. This frame was selected because it is a simple example to explain the features of this dataset. The region

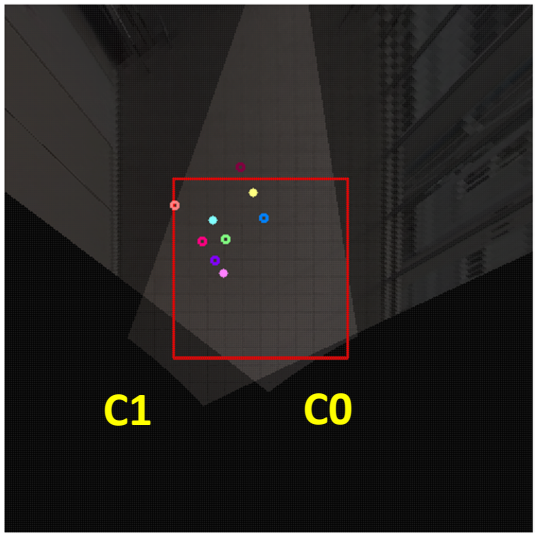
surrounded by red lines is the area-of-interest (AOI). Each candidate in the top view, along with its two corresponding candidate boxes in the two camera views, is shown in the same distinguished colour.

Figure 5-17 shows the joint occupancy likelihoods for the pedestrian candidates at frame 825. The left-most column is the IDs of pedestrian candidates. In the first row, 1 and 2 are the indices of the two cameras. F is the template matching response $P(t_i^c|L_i)$, T is the head likelihood $P(h_i^c|L_i)$ and B is the foot likelihood $P(f_i^c|L_i)$. JL is the joint occupancy likelihood, obtained by multiplying the occupancy likelihoods in the camera views covering this location and extract the k_i-th root, where k_i is the number of the camera views covering this location. Candidate 0 is not completely visible in C1. Therefore, its likelihoods for this camera view were set to one.

Figure 5-18 shows the prime candidate charts at frame 825. Figure 5-18 (a) shows the original chart. By comparing with the PETS2009 CC dataset, the Terrace dataset has a more complicated prime candidate chart because of the increase in the number of adjacent pedestrian candidates. Figure 5-18 (b) is the chart after step 1, where invalid foreground sub-regions are removed. The number of the invalid foreground sub-regions are obviously increased because more adjacent candidate boxes divide the foreground into more sub-regions. Most of these sub-regions are invalid because they are too small. Figure 5-18 (c) is the chart after step 2 by removing essential candidates. In this step, candidate 7 is identified as an essential candidate. Candidate 0 is not identified as a pedestrian, because all its sub-regions are already covered by essential candidates. After the Xs for essential candidates are replaced by plus signs, the links between candidate 0 and its sub-regions no longer exist. Figure 5-18 (d) shows the chart with the three contained candidates 3, 5 and 6 being merged into other candidates. This leaves candidates 1 and 4 as emerging essential candidates. Figure 5-18 (e) is the chart with candidates 1 and 4 being labelled as essential. At this stage, all the sub-regions are covered by the three essential candidates and they are identified as pedestrians.



(a) (b)



(c)

Figure 5-16 The detection results at frame 825 on the EPFL Terrace dataset with two camera views: (a) (b) camera views C0 and C1, and (c) the synthetic top view.

	1F	1T	1B	2F	2T	2B	JL
I0	0.856	0.954	1.000	1.000	1.000	1.000	0.817
I1	0.884	0.983	1.000	0.843	0.902	1.000	0.813
I2	0.908	0.981	1.000	0.805	0.702	1.000	0.709
I3	0.875	0.985	1.000	0.738	0.940	1.000	0.773
I4	0.828	0.853	1.000	0.766	0.755	1.000	0.639
I5	0.839	0.985	0.849	0.749	0.880	1.000	0.680
I6	0.871	0.984	1.000	0.707	0.803	0.887	0.657
I7	0.872	0.986	1.000	0.681	0.930	1.000	0.738
I8	0.766	0.864	0.899	0.762	0.936	1.000	0.651

Figure 5-17 The joint likelihoods for the pedestrian candidates at frame 825.

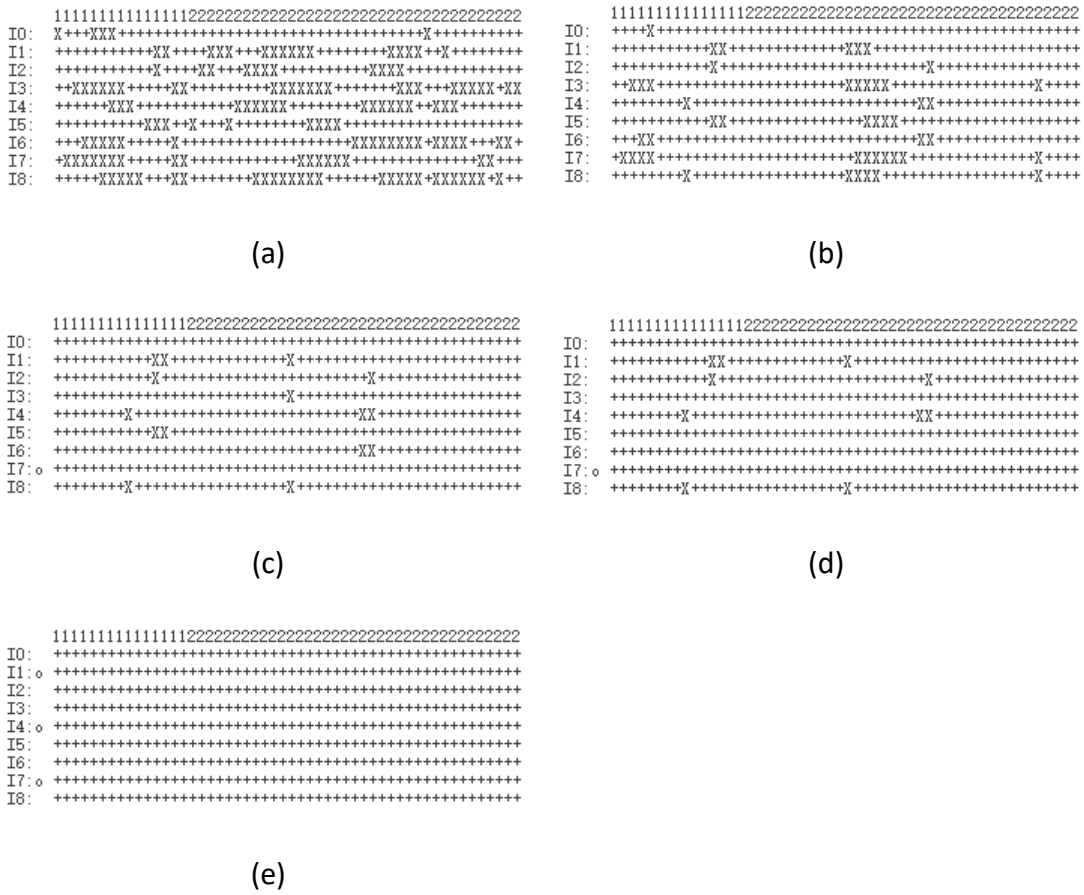
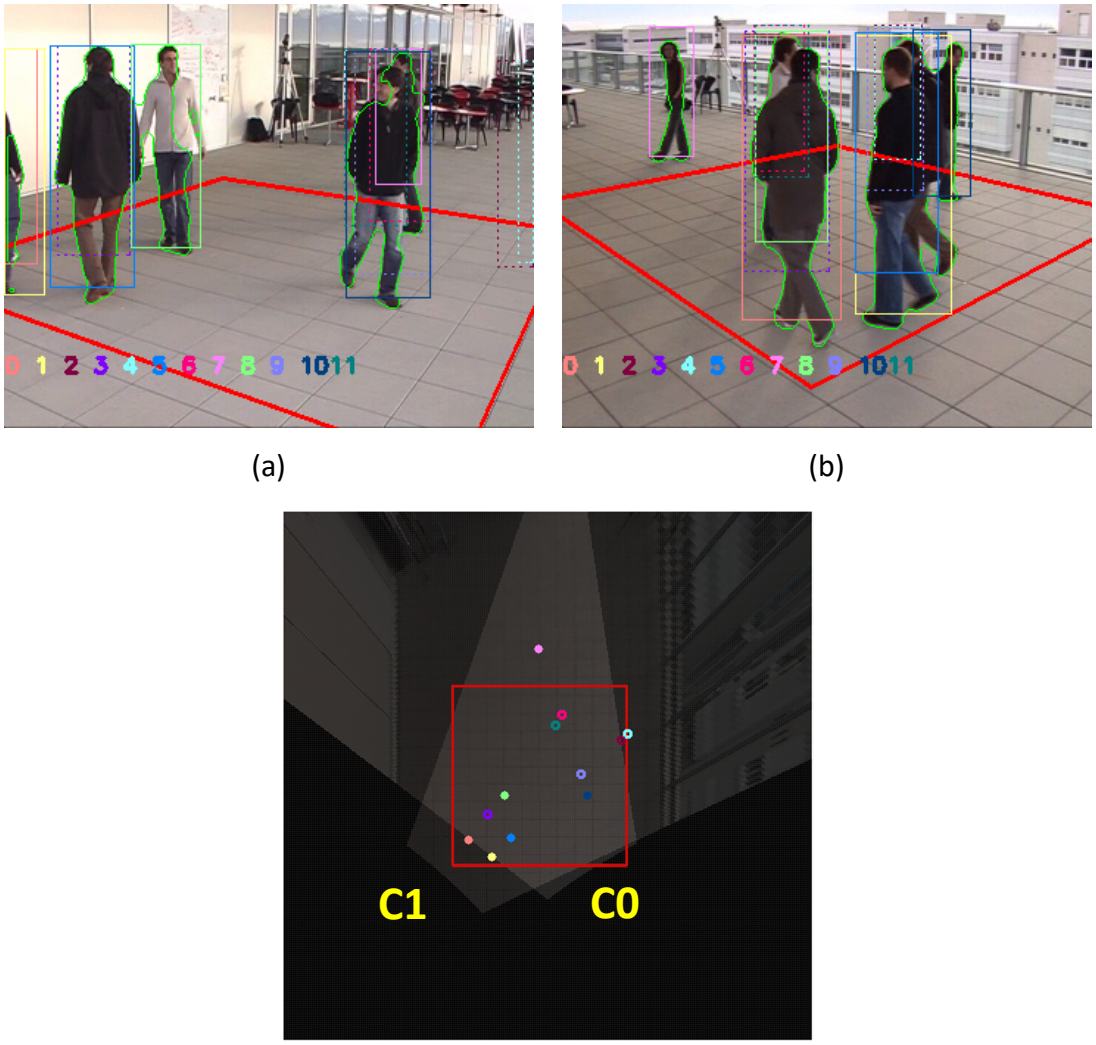


Figure 5-18 The prime candidate chart at frame 825: (a) the original table, (b) after step 1 when invalid sub-regions are removed, (c) after step 2 when essential candidate 7 is identified. (d) in step 3, candidates 3 and 5 are merged into candidate 1, and candidate 6 is merged into candidate 4, and (e) after step 3 when candidate 1 and candidate 4 are identified as essential candidates.

Figure 5-19 shows the detection results at frame 2350 on the EPFL Terrace dataset with two camera views C0 and C1, where Figure 5-19 (a) and (b) are the two camera views and Figure 5-19 (c) is a synthetic top view. This frame was selected because there are overlapped pedestrians in both camera views and a missed detection. Candidate 6, represented in magenta, is a real pedestrian but was recognised as a phantom. This pedestrian is hidden behind others in both camera views and it is rather difficult to identity him by human. The evidence of this pedestrian appearing in this scene is his leg and head in C0. However, his head is easy to be misunderstood as the head of candidate 7 who is behind candidate 6 in

C1. In C0, the foreground in candidate box 8 is poorly detected because the colour of the pedestrian's white shirt is very similar to that of the wall.



(a) (b)

(c)

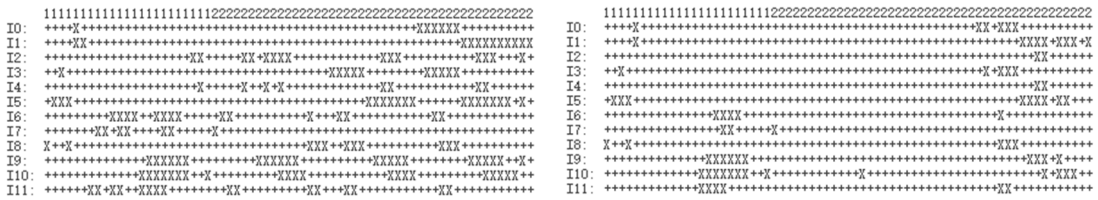
Figure 5-19 The detection results at frame 2350 on the EPFL Terrace dataset with two camera views: (a) (b) camera views C0 and C1, and (c) the synthetic top view.

Figure 5-20 shows the joint occupancy likelihoods for the pedestrian candidates at frame 2350. Candidates 0, 1, 2 and 4 are not completely visible in C0. Therefore, they obtained one in all the likelihoods for this camera view. The likelihood of candidate 6 is not obviously higher or lower than those for the other candidates. Figure 5-21 shows the prime candidate charts at frame 2350. Figure 5-21 (a) shows the original chart. Figure 5-21 (b) is the chart after step 1, where invalid foreground

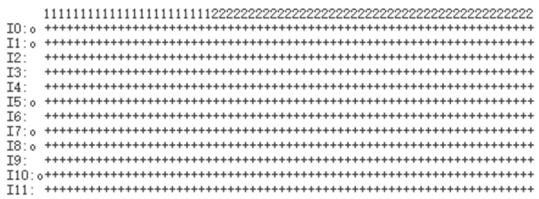
sub-regions are removed. In this step, all the sub-regions only belonging to candidate 6 have been removed because they are too small. The other sub-regions for candidate 6 are fully covered by candidate 10 in C0 and by candidates 0 or 8 in C1. Therefore, by removing essential candidates, candidate 6 along with its sub-regions no longer exist and it is recorded as a phantom. The result is shown in Figure 5-21 (c). At this stage, candidates 0, 1, 5, 7, 8 and 10 are correctly identified as essential candidates. Candidate 6 is falsely recognised as a phantom.

	1F	1T	1B	2F	2T	2B	JL
I0	1.000	1.000	1.000	0.913	1.000	1.000	0.913
I1	1.000	1.000	1.000	0.878	0.968	1.000	0.850
I2	1.000	1.000	1.000	0.789	0.847	1.000	0.668
I3	0.875	0.987	1.000	0.900	0.989	1.000	0.877
I4	1.000	1.000	1.000	0.784	0.842	1.000	0.660
I5	0.877	0.989	1.000	0.869	0.947	1.000	0.845
I6	0.861	0.964	1.000	0.798	0.900	1.000	0.772
I7	0.855	0.936	1.000	0.785	0.790	1.000	0.705
I8	0.751	0.987	1.000	0.882	0.972	1.000	0.797
I9	0.795	0.988	1.000	0.810	0.883	1.000	0.750
I10	0.781	0.989	1.000	0.716	0.860	1.000	0.690
I11	0.678	0.946	1.000	0.808	0.904	1.000	0.684

Figure 5-20 The joint likelihoods for the pedestrian candidates at frame 2350.



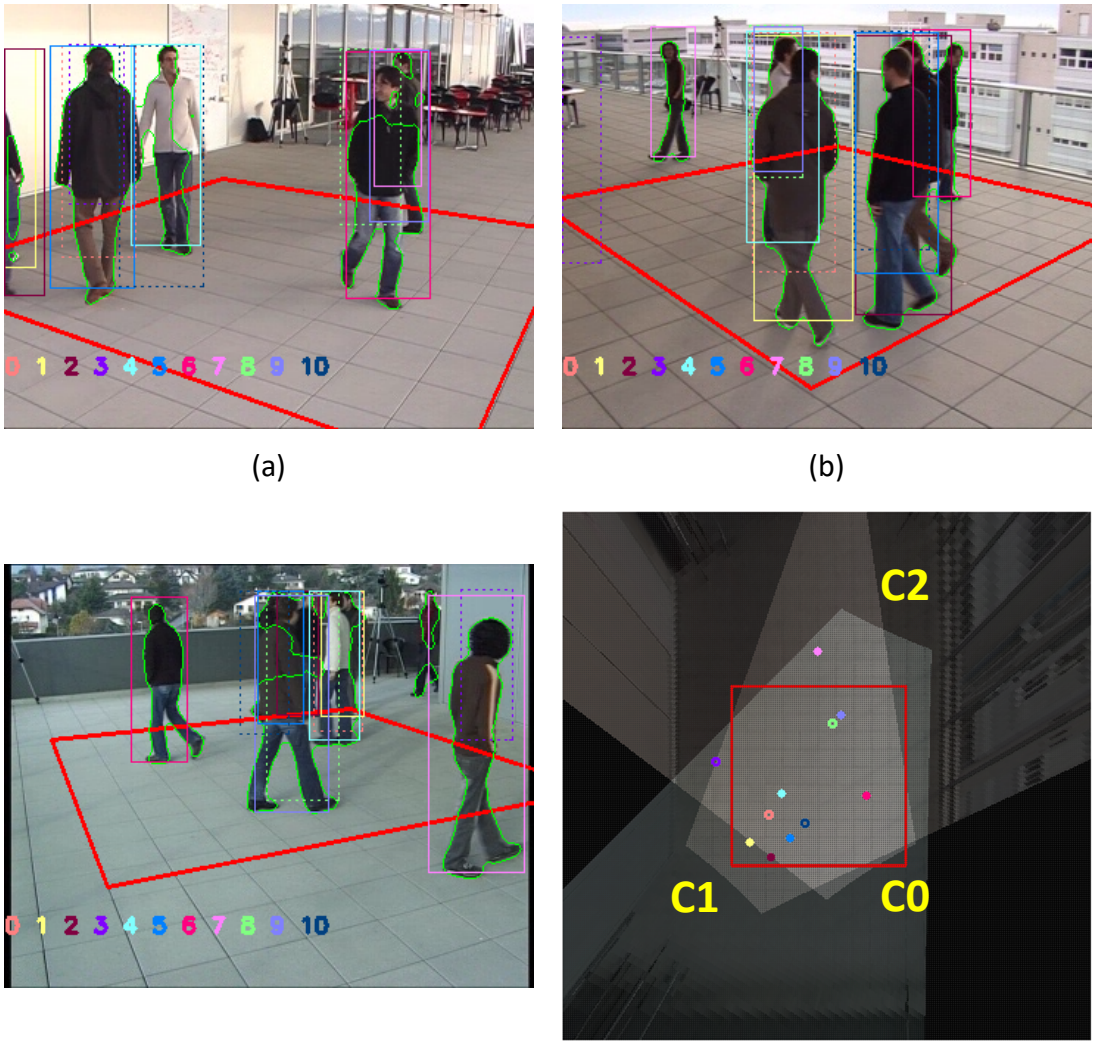
(a) (b)



(c)

Figure 5-21 The prime candidate chart at frame 2350: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified.

When more camera views were used, the problem of insufficient observations can be solved. Figure 5-22 shows the detection results at frame 2350 on the EPFL Terrace dataset with three camera views C0, C1 and C2, where Figure 5-22 (a), (b) and (c) are the three camera views and Figure 5-22 (d) is a synthetic top view. The additional camera view C2 is opposite to C1. The pedestrian missed in the detection by using two camera views was correctly detected as candidate 9. He is completely observed in C2.



(a) (b)

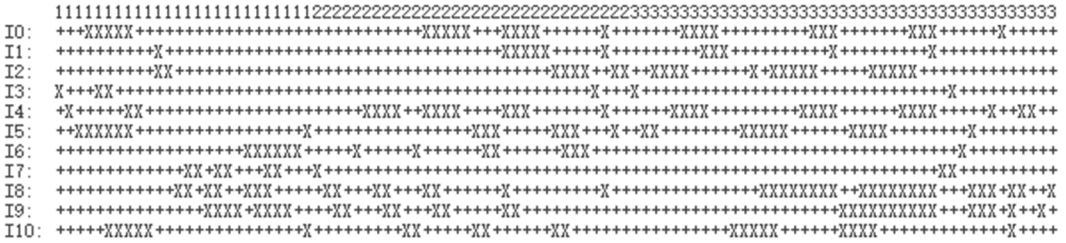
(c) (d)

Figure 5-22 The detection results at frame 2350 on the EPFL Terrace dataset with three camera views: (a) (b) (c) camera views C0, C1 and C2, and (d) the synthetic top view.

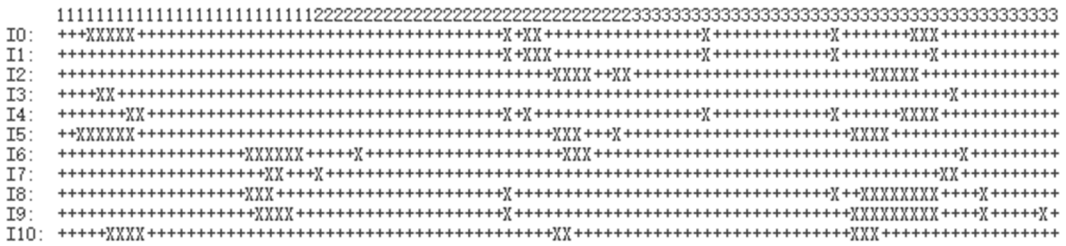
Figure 5-23 shows the joint occupancy likelihoods for the pedestrian candidates at frame 2350. Figure 5-24 shows the prime candidate charts at frame 2350. Figure 5-24 (a) shows the original chart. Figure 5-24 (b) is the chart after step 1, where invalid foreground sub-regions are removed. After step 2 when essential candidates are identified, candidates 4 and 10 are left as shown in Figure 5-24 (c). Since the joint likelihood of candidate 4 (0.803) is greater than that of candidate 10 (0.526), candidate 10 is removed (shown in Figure 5-24 (d)) and candidate 4 is identified as an essential candidate (shown in Figure 5-24 (e)). At this stage, candidates 1, 2, 4, 5, 6, 7 and 9 are correctly identified as pedestrians.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	JL
I0	0.872	0.987	1.000	0.897	0.989	1.000	0.878	1.000	1.000	0.875
I1	1.000	1.000	1.000	0.837	1.000	1.000	0.802	1.000	1.000	0.819
I2	1.000	1.000	1.000	0.873	0.968	1.000	0.692	0.977	1.000	0.756
I3	0.814	0.983	1.000	1.000	1.000	1.000	0.711	0.669	1.000	0.617
I4	0.733	0.987	1.000	0.866	0.972	1.000	0.852	1.000	1.000	0.803
I5	0.874	0.989	1.000	0.867	0.939	1.000	0.607	0.979	1.000	0.748
I6	0.672	0.989	1.000	0.710	0.860	1.000	0.829	0.897	1.000	0.670
I7	0.615	0.937	1.000	0.778	0.790	1.000	0.777	0.928	1.000	0.635
I8	0.623	0.900	1.000	0.775	0.904	1.000	0.698	1.000	1.000	0.650
I9	0.675	0.964	1.000	0.796	0.900	1.000	0.601	1.000	1.000	0.654
I10	0.604	1.000	0.528	0.848	0.921	1.000	0.583	1.000	1.000	0.526

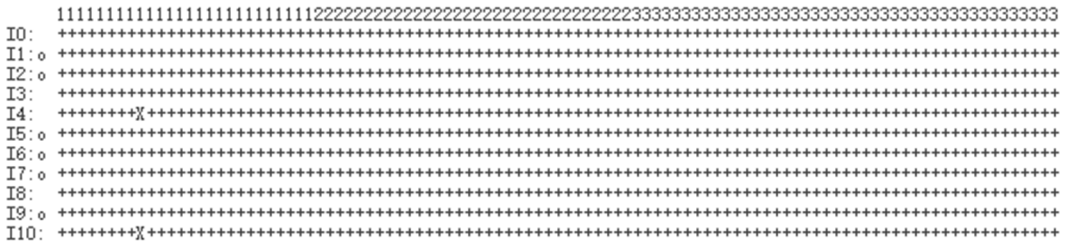
Figure 5-23 The joint likelihoods for the pedestrian candidates at frame 2350.



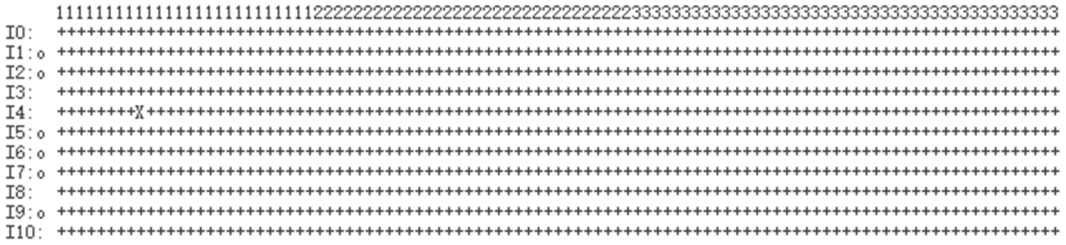
(a)



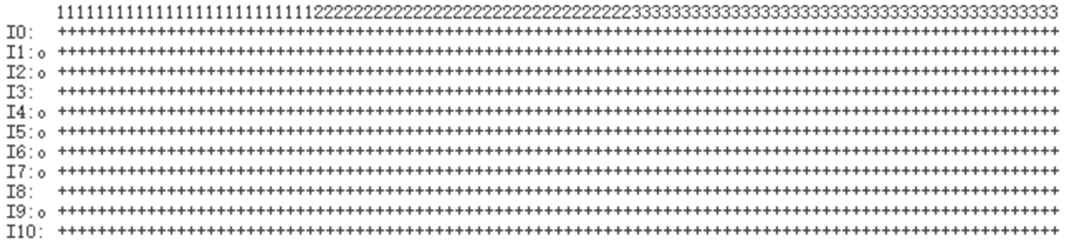
(b)



(c)



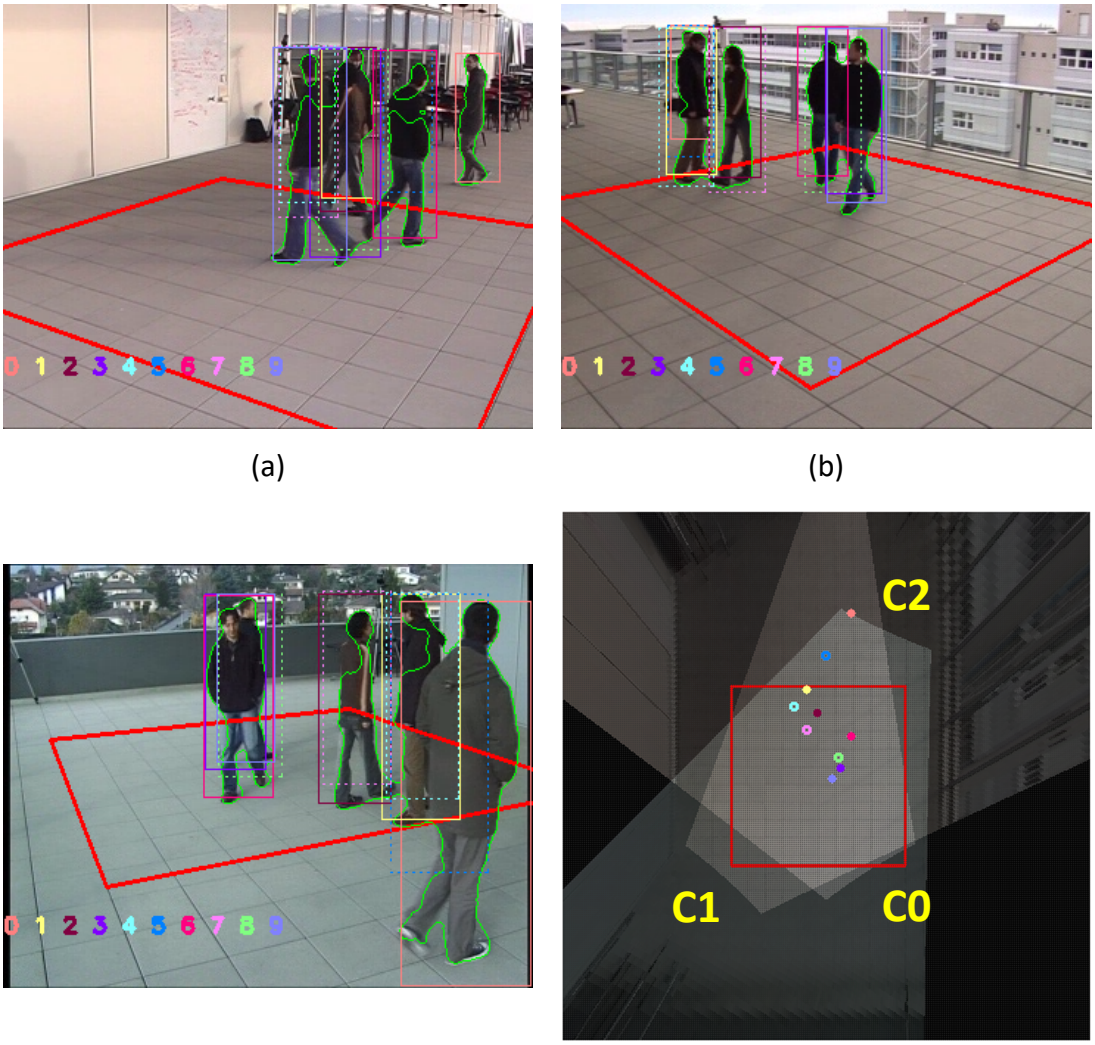
(d)



(e)

Figure 5-24 The prime candidate chart at frame 2350: (a) the original table, (b) after step 1 when invalid sub-regions are removed, (c) after step 2 when essential candidates are identified, (d) in step 3, candidate 10 is merged into candidate 4, and (e) after step 3 when candidate 4 is identified as an essential candidate.

Figure 5-25 shows the detection results at frame 1250 on the EPFL Terrace dataset with three camera views C0, C1 and C2, where Figure 5-25 (a), (b) and (c) are the three camera views and Figure 5-25 (d) is a synthetic top view. This frame was selected because there are overlapped pedestrians in all camera views and a false positive detection. Candidate 3 is a phantom but is falsely detected as a pedestrian because of foreground detection errors. In C0, the foreground at the bottom of candidate 3 is connected by the morphological closing operation, which makes candidate 3 be identified as essential.



(a) (b)

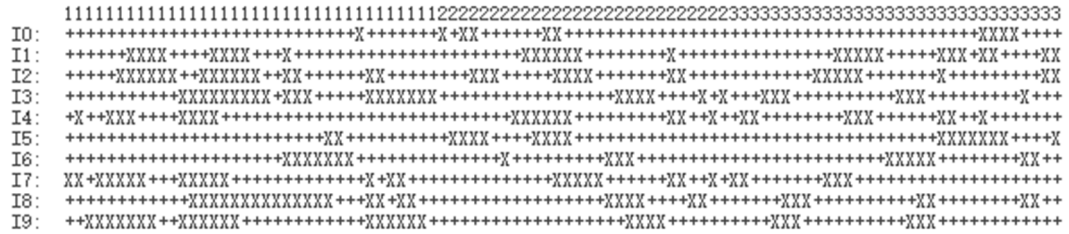
(c) (d)

Figure 5-25 The detection results at frame 1250 on the EPFL Terrace dataset with three camera views: (a) (b) (c) camera views C0, C1 and C2, and (d) the synthetic top view.

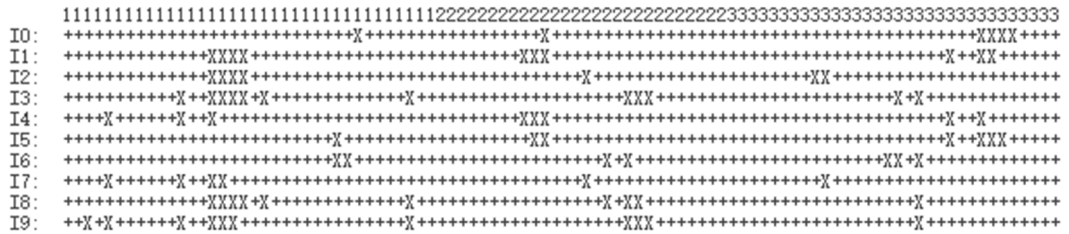
Figure 5-26 and Figure 5-27 show the joint occupancy likelihoods for the pedestrian candidates and the prime candidate charts at frame 1250. Figure 5-27 (a) shows the original chart. Figure 5-27 (b) is the chart after step 1, where invalid foreground sub-regions are removed. After step 2 when essential candidates are identified, candidates 1, 3, 4, 5 and 8 are left, as shown in Figure 5-27 (c). In this step, it can be found that candidates 3 and 8 are competing for a sub-region (in C0) which is only covered by these two candidates. This region is the region at the bottom right corner of candidate 3. If the foreground detection is accurate, this region will be ignored in step1 due to its low foreground ratio. Since the joint likelihood of candidate 3 (0.756) is greater than candidate 8 (0.712), candidate 8 is removed. Then candidate 5 is merged into candidate 4 or 1. Candidates 4 and 1 are competing for three sub-regions in C1. Since the joint likelihood of candidate 1 (0.826) is greater than candidate 4 (0.739), candidate 4 is removed. Therefore, only candidates 3 and 8 survive, as shown in Figure 5-27 (d). Finally, candidates 1 and 3 are identified as essential candidates. At this stage, candidates 0, 1, 2, 6 and 9 are correctly identified as essentials. Candidate 3, which is a phantom, is falsely identified as essential. The result is shown in Figure 5-27 (e).

	1F	1T	1B	2F	2T	2B	3F	3T	3B	JL
I0	0.726	1.000	1.000	0.920	0.899	1.000	1.000	1.000	1.000	0.775
I1	0.843	0.982	1.000	0.920	0.956	1.000	0.773	1.000	1.000	0.826
I2	0.840	0.984	1.000	0.819	0.754	1.000	0.778	0.851	1.000	0.697
I3	0.805	1.000	1.000	0.767	0.886	1.000	0.844	0.934	1.000	0.756
I4	0.754	0.893	1.000	0.818	0.960	1.000	0.774	0.987	1.000	0.739
I5	0.724	0.809	1.000	0.917	0.934	1.000	0.711	1.000	1.000	0.709
I6	0.697	0.861	1.000	0.790	0.801	1.000	0.847	0.950	1.000	0.674
I7	0.748	0.888	1.000	0.763	0.800	0.930	0.788	0.820	1.000	0.625
I8	0.802	1.000	1.000	0.710	0.884	1.000	0.766	0.936	1.000	0.712
I9	0.654	0.920	1.000	0.768	0.890	1.000	0.845	0.932	1.000	0.687

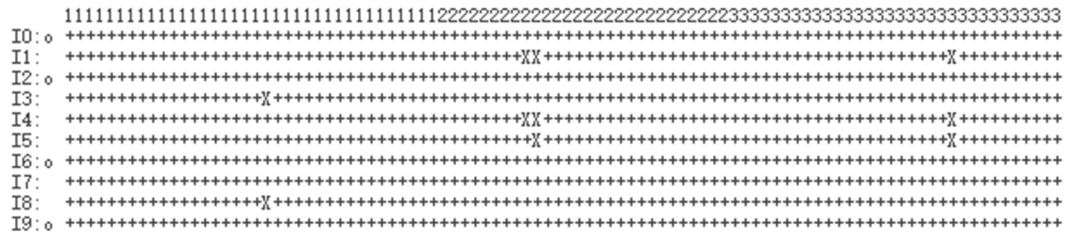
Figure 5-26 The joint likelihoods for the pedestrian candidates at frame 1250.



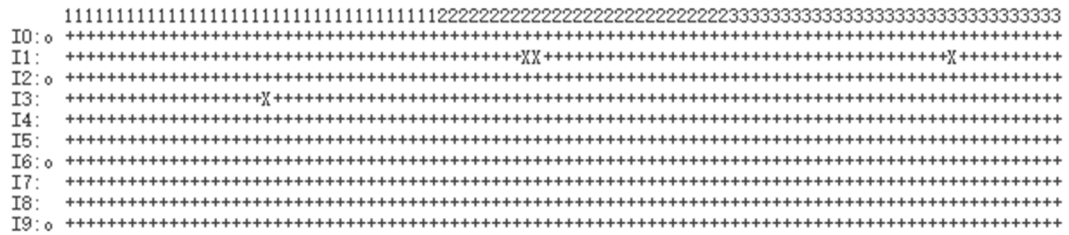
(a)



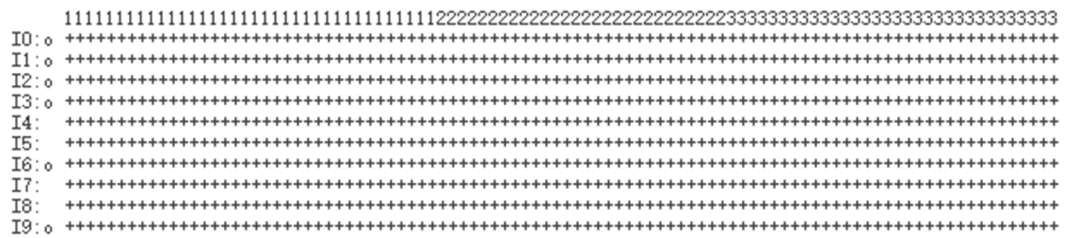
(b)



(c)



(d)



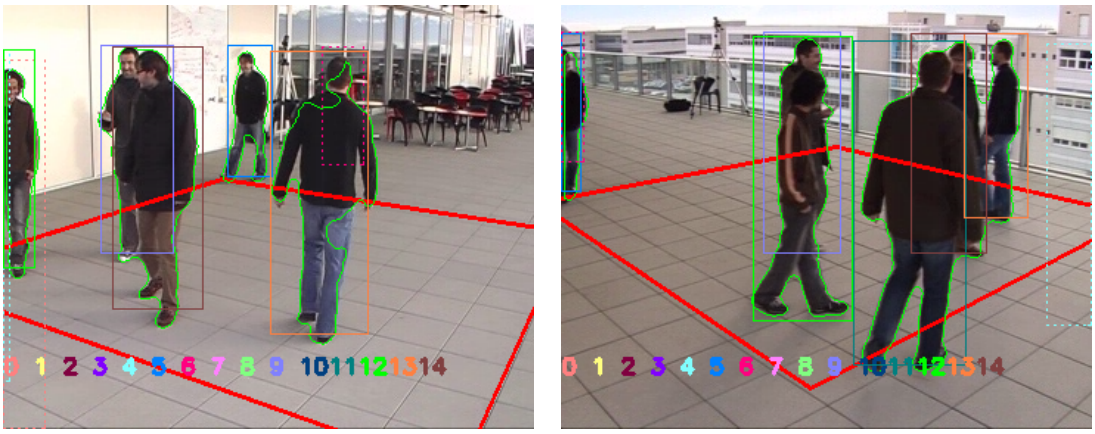
(e)

Figure 5-27 The prime candidate chart at frame 1250: (a) the original table, (b) after step 1 when invalid sub-regions are removed, (c) after step 2 when essential candidates are identified, (d) in step 3, candidates 4 and 5 are merged into candidate 1, and candidate 8 is merged into candidate 3, and (e) after step 3 when candidates 1 and 3 are identified as essential candidates.

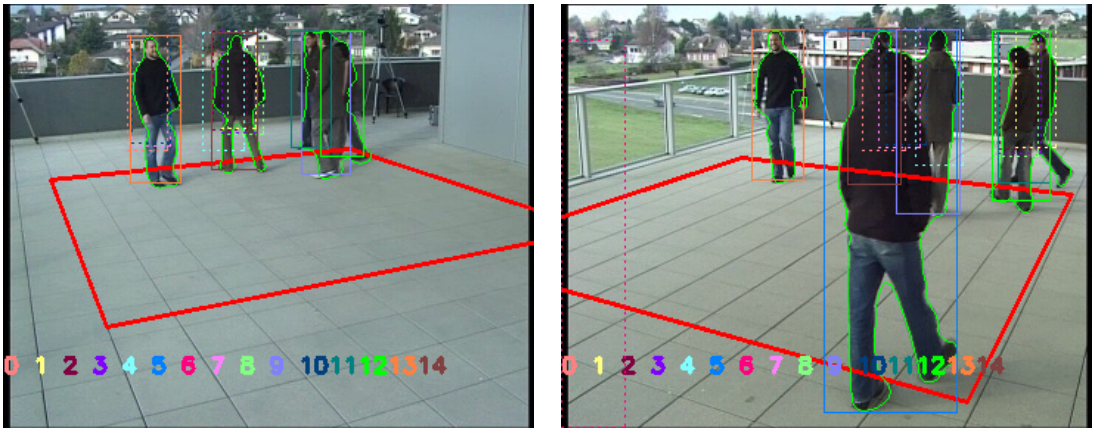
Figure 5-28 shows the detection results at frame 1475 on the EPFL Terrace dataset with four camera views C0, C1, C2 and C3, where Figure 5-28 (a), (b), (c) and (d) are the four camera views and Figure 5-28 (e) is a synthetic top view. The additional camera view C3 is opposite to C0. This frame was selected because 6 pedestrians appear in the scene with significantly overlapping. Candidate 9, which is a pedestrian, is merged with other pedestrians in all camera views and it is correctly detected. With the increased number of cameras, the number of phantoms decreases in the overlapping field of view of the four cameras. All the phantoms in this example are in the areas which are only covered by two cameras.

Figure 5-29 shows the joint occupancy likelihoods for the pedestrian candidates at frame 1475. The template matching responses, head likelihoods and foot likelihoods of candidate 12 in C0 and candidate 5 in C1 are equal to 1, because the candidate boxes are at the boundary of the camera views and only parts of the candidate boxes are visible. Therefore, the foreground observation of candidate 12 in C0 and candidate 5 in C1 are ignored and set to 1.

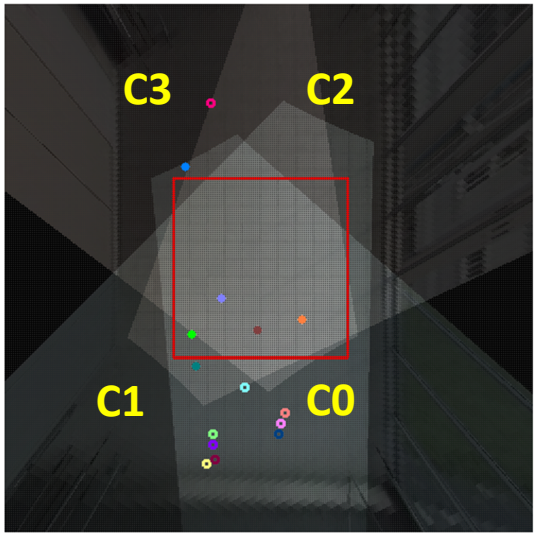
Figure 5-30 shows the prime candidate charts at frame 1475. Figure 5-30 (a) shows the original chart. Since phantoms mainly appear in C2 and C3, more candidate boxes are involved to decompose the foregrounds. The sub-regions in these two camera views are significantly more than those in C0 and C1. Figure 5-30 (b) is the chart after step 1, where invalid foreground sub-regions are removed. In this step, lots of sub-regions in C2 and C3 are removed, because the foregrounds in these two camera views are decomposed into fragmented sub-regions, which are invalid due to their small sizes. Figure 5-30 (c) is the chart after step 2, where essential candidates are identified. At this stage, candidates 5, 9, 11, 12, 13 and 14 are correctly detected as pedestrians.



(a) (b)



(c) (d)

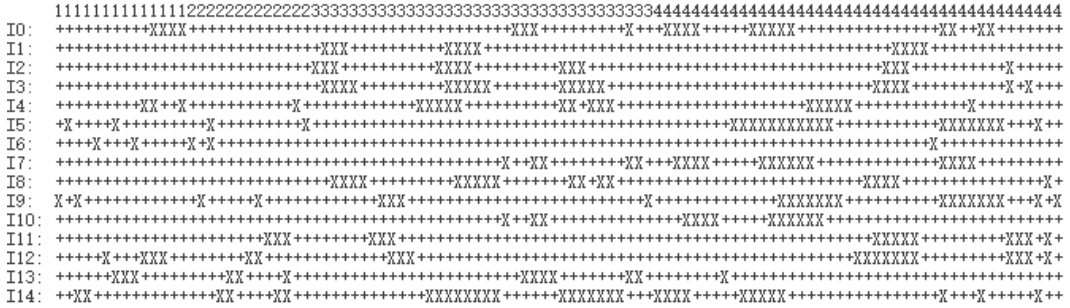


(e)

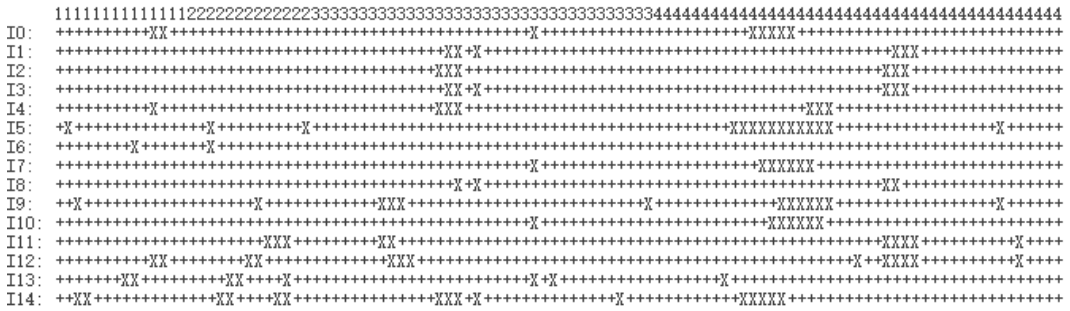
Figure 5-28 The detection results at frame 1475 on the EPFL Terrace dataset with four camera views: (a) (b) (c) (d) camera views C0, C1, C2 C3, and (e) the synthetic top view.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	4F	4T	4B	JL
I0	1.000	1.000	1.000	1.000	1.000	1.000	0.842	0.945	1.000	0.760	0.947	1.000	0.757
I1	1.000	1.000	1.000	1.000	1.000	1.000	0.767	0.911	1.000	0.825	0.946	1.000	0.738
I2	1.000	1.000	1.000	1.000	1.000	1.000	0.779	0.913	1.000	0.800	0.946	1.000	0.733
I3	1.000	1.000	1.000	1.000	1.000	1.000	0.749	0.915	1.000	0.804	0.947	1.000	0.722
I4	1.000	1.000	1.000	1.000	1.000	1.000	0.691	0.946	1.000	0.834	0.979	1.000	0.731
I5	0.740	0.894	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.774	0.993	0.969	0.701
I6	0.696	0.739	1.000	0.810	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.646
I7	1.000	1.000	1.000	1.000	1.000	1.000	0.843	0.924	1.000	0.665	0.947	1.000	0.700
I8	1.000	1.000	1.000	1.000	1.000	1.000	0.710	0.917	1.000	0.787	0.948	1.000	0.697
I9	0.869	0.987	1.000	0.922	0.935	1.000	0.854	1.000	1.000	0.809	0.986	1.000	0.842
I10	1.000	1.000	1.000	1.000	1.000	1.000	0.847	0.921	1.000	0.611	0.884	1.000	0.649
I11	1.000	1.000	1.000	0.736	0.991	1.000	0.830	1.000	1.000	0.814	0.958	1.000	0.779
I12	1.000	1.000	1.000	0.814	0.981	0.938	0.712	1.000	1.000	0.829	0.964	1.000	0.752
I13	0.744	0.870	1.000	0.757	0.923	1.000	0.883	0.956	1.000	0.879	0.982	1.000	0.758
I14	0.835	0.989	1.000	0.804	0.923	1.000	0.727	0.937	1.000	0.804	0.960	1.000	0.753

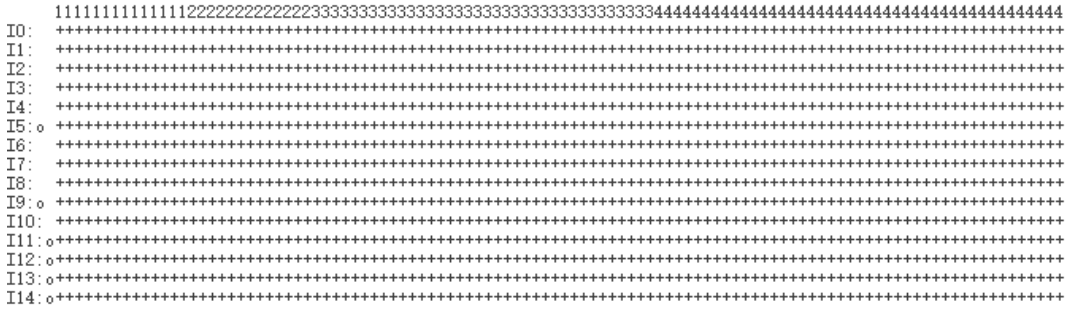
Figure 5-29 The joint likelihoods for the pedestrian candidates at frame 1475.



(a)



(b)



(c)

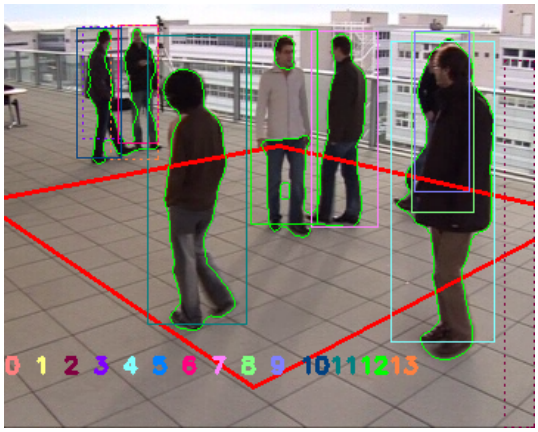
Figure 5-30 The prime candidate chart at frame 1475: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified.

Figure 5-31 shows the detection results at frame 3450 on the EPFL Terrace dataset with four camera views C0, C1, C2 and C3, where Figure 5-31 (a), (b), (c) and (d) are the four camera views and Figure 5-31 (e) is a synthetic top view. This frame was selected because 8 pedestrians appear in the scene and the foreground detection is poor in all the camera views.

Figure 5-32 shows the joint occupancy likelihoods for the pedestrian candidates at frame 3450. The template matching responses for candidates 10 and 13 in C0, candidate 12 in C1 and candidate 11 in C2 are very low due to the poor foreground detection. The head likelihood for candidate 8 in C0 is equal to one because the width of the foreground region at the top-right corner of candidate box 8 is larger than one-tenth of the candidate box width. However, this pedestrian can still be detected even if the foreground pixels at the top-right corner are missed in foreground detection. Because the observations of this candidate are good in other three camera views, which leads to a high joint occupancy likelihood.

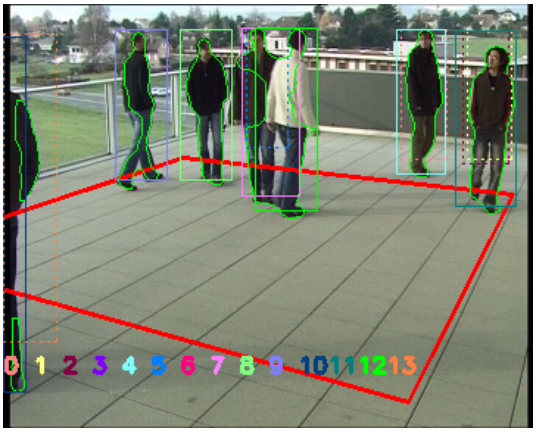
Figure 5-33 shows the prime candidate charts at frame 3450. Figure 5-33 (a) shows the original chart. Figure 5-33 (b) is the chart after step 1, where invalid foreground sub-regions are removed. Figure 5-33 (c) is the chart after step 2, where essential candidates are identified. Even when the scene is crowded and complicated, the correct result can be obtained by only identifying essential candidates. With the increased number of cameras, the essential candidates become easier to be identified because more observations bring more sub-regions only covered by one candidate.

(a)

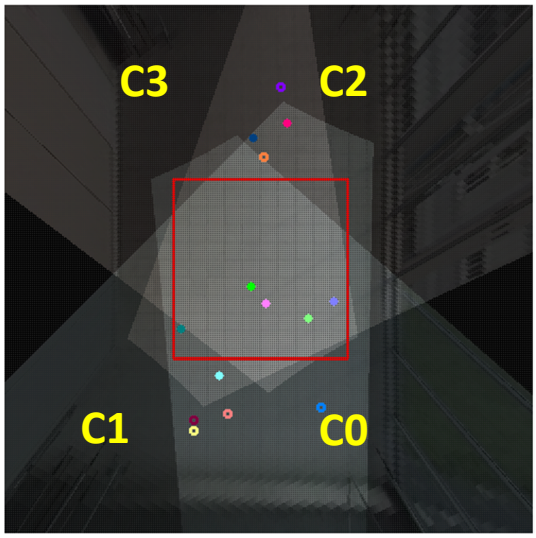


(b)

(c)



(d)

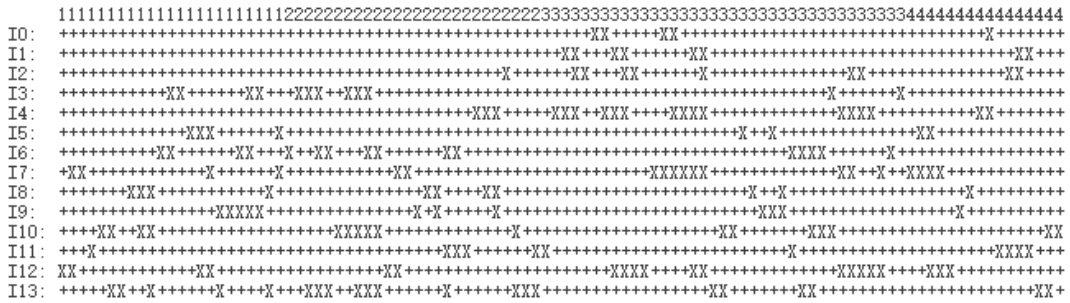


(e)

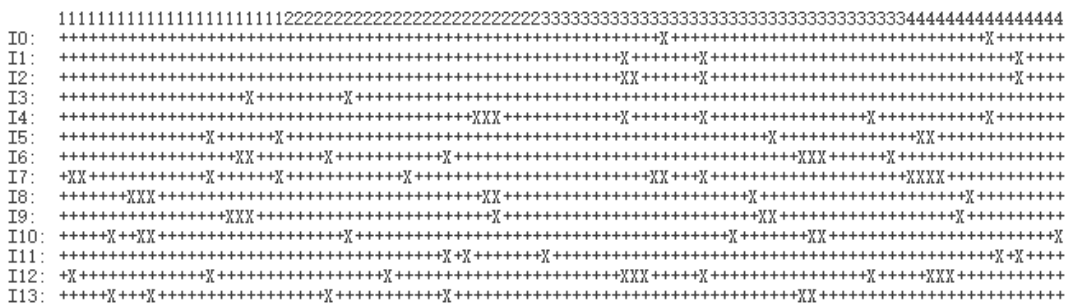
Figure 5-31 The detection results at frame 3450 on the EPFL Terrace dataset with four camera views: (a) (b) (c) (d) camera views C0, C1, C2, C3, and (e) the synthetic top view.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	4F	4T	4B	JL
I0	1.000	1.000	1.000	1.000	1.000	1.000	0.905	0.973	1.000	0.922	0.978	1.000	0.891
I1	1.000	1.000	1.000	1.000	1.000	1.000	0.864	0.971	1.000	0.814	0.754	1.000	0.718
I2	1.000	1.000	1.000	1.000	1.000	1.000	0.834	0.972	1.000	0.808	0.765	1.000	0.708
I3	0.786	1.000	1.000	0.850	0.945	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.794
I4	1.000	1.000	1.000	0.857	1.000	1.000	0.784	0.975	1.000	0.895	0.981	1.000	0.832
I5	1.000	1.000	1.000	1.000	1.000	1.000	0.782	0.846	1.000	0.688	1.000	1.000	0.675
I6	0.781	1.000	1.000	0.853	0.928	1.000	0.779	1.000	0.945	1.000	1.000	1.000	0.769
I7	0.738	0.989	1.000	0.860	0.949	1.000	0.857	0.982	1.000	0.783	0.984	1.000	0.788
I8	0.698	1.000	1.000	0.825	0.921	1.000	0.840	0.877	1.000	0.838	0.877	1.000	0.732
I9	0.817	1.000	1.000	0.826	0.895	1.000	0.836	0.896	1.000	0.712	0.941	1.000	0.742
I10	0.509	1.000	1.000	0.803	0.949	1.000	0.829	1.000	1.000	1.000	1.000	1.000	0.685
I11	1.000	1.000	1.000	0.748	0.779	1.000	0.524	0.833	1.000	0.841	0.836	1.000	0.564
I12	0.751	0.988	1.000	0.612	0.927	1.000	0.901	1.000	1.000	0.758	0.986	1.000	0.729
I13	0.467	1.000	1.000	0.684	0.950	0.880	0.743	1.000	1.000	1.000	1.000	1.000	0.583

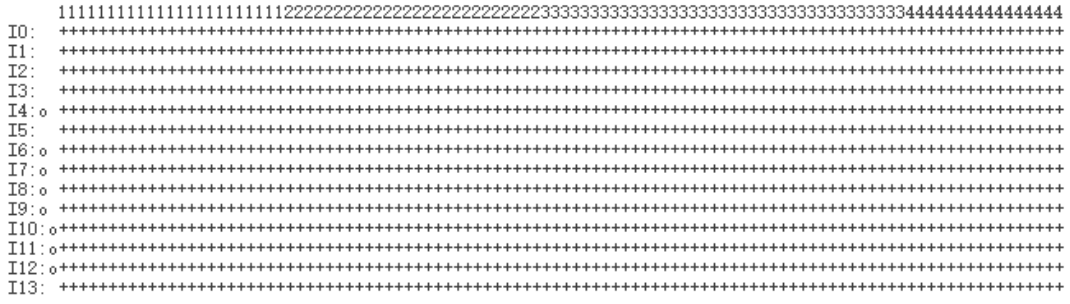
Figure 5-32 The joint likelihoods for the pedestrian candidates at frame 3450.



(a)



(b)



(c)

Figure 5-33 The prime candidate chart at frame 3450: (a) the original table, (b) after step 1 when invalid sub-regions are removed, and (c) after step 2 when essential candidates are identified.

5.2.2 Bottom-Up Approach with Petrick's Method

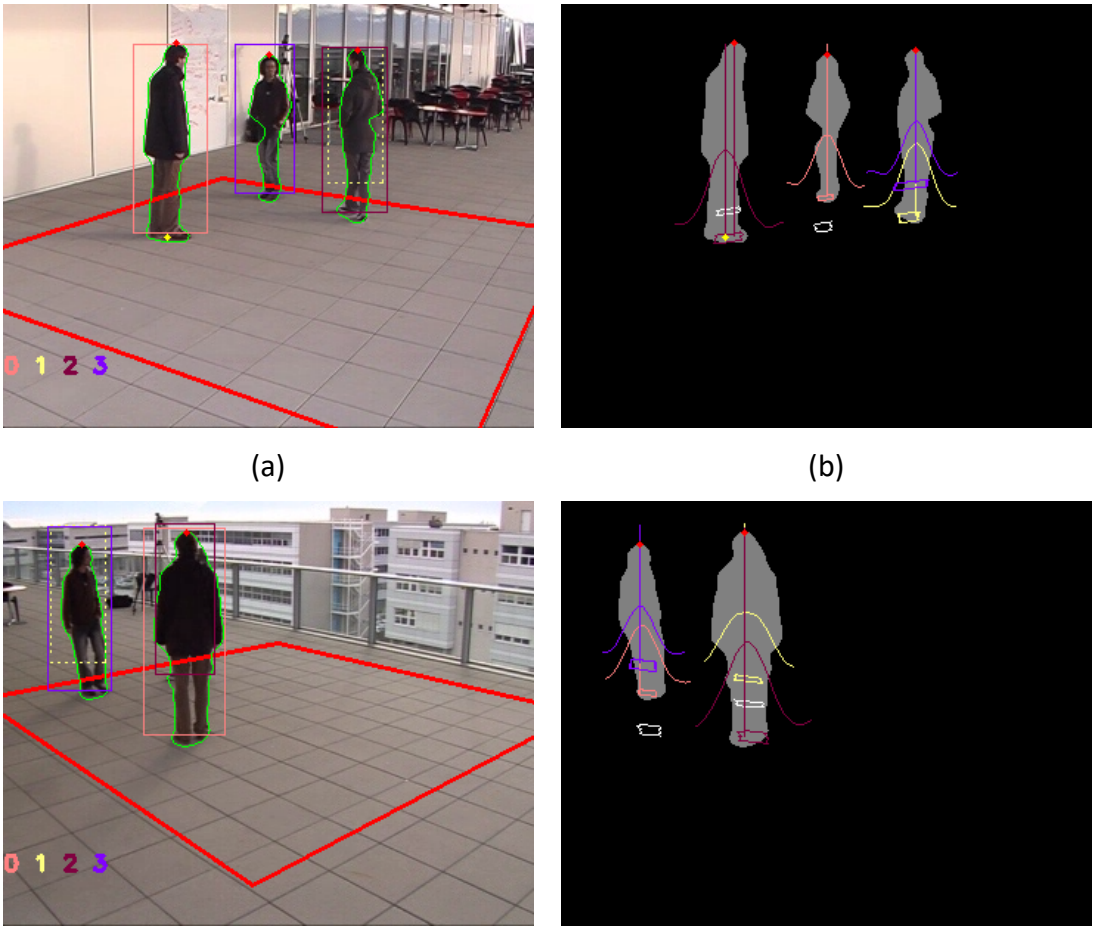
Two benchmark datasets were used to evaluate the bottom-up approach with the Petrick's method. In this section, some examples were selected from these datasets to show the detection results with the joint likelihoods for pedestrian candidates, the prime candidate charts and the simplification of Petrick Functions.

Results on the EPFL Terrace Dataset

For the EPFL Terrace dataset, three experiments were carried out. Two camera views, three camera views and four camera views were used in the experiments, respectively. In the joint likelihoods and prime candidate chart, the camera indices 1, 2, 3 and 4 are corresponding to camera views C0, C1, C2 and C3 in the dataset.

Figure 5-34 shows the detection results at frame 800 on the EPFL Terrace dataset with two camera views, where Figure 5-34 (a) and (c) are the two camera views (C0 and C1), Figure 5-34 (b) and (d) are the corresponding foreground region maps, and Figure 5-34 (e) is a synthetic top view. This frame was selected because it is a simple example to explain the features of the bottom-up approach and the Petrick's method.

In the camera views, the contour of each foreground region is shown in green. Each candidate in the top view, along with its two corresponding candidate boxes in the two camera views, is shown in the same distinguished colour. The IDs of these candidates are shown at the bottom of both camera views in the same distinguished colours. Each identified pedestrian is labelled with a dot in the top view and a rectangle of solid lines in both camera views, while each phantom is labelled with a circle in the top view and a rectangle of dashed lines in the camera views. In both camera views and foreground region maps, the tops of heads for each foreground region are shown as red dots and the survived local maximums of the template matching are shown as yellows. The vertical coordinate of the local maximums of the template matching are the same with that of the mass centre for the corresponding warped foreground intersection regions.



(a) (b)

(c) (d)

(e)

Figure 5-34 The detection results at frame 800 on the EPFL Terrace dataset with two camera views: (a) (c) camera views C0 and C1, (b) (d) foreground region maps of camera C0 and C1, and (e) a synthetic top view.

In the foreground region maps, the regions surrounded by white contours are the warped foreground intersection regions. The curves in different colours represent the template matching responses of the foreground intersection regions, and the vertical lines in the same colour represent the lines passing the tops of heads or the survived local maxima of the template matching response for the same intersection region. For pedestrian candidate 0 in C0, since the horizontal distance between the local maximum of the template matching response and the top of head is greater than 1/10 of the average width of pedestrians, the local maximum remains. On the other hand, in the same camera view, the local maximum of the template matching response of candidate 3 is removed because it is too close to the top of head in the horizontal coordinates.

In the top view, the line segments represent the projected lines from both camera views. The foreground intersection points are then identified by finding the intersection of every two projected line segments. The pedestrian candidates are shown as the colourful dots or circles are. The colour of each candidate box is different from that of the corresponding template matching response curve, e.g. candidate box 2 is in dark red but the corresponding template matching response curve is in yellow. This is because one foreground intersection region may be associated with more than one pedestrian candidate or no pedestrian candidate. For example, there should be two intersections of dark red lines, but one of them is filtered out by the RSS filter.

Figure 5-35 shows the joint occupancy likelihoods for the pedestrian candidates at frame 800. Figure 5-36 shows the results of the Petrick's method at this frame. Figure 5-36 (a) shows the corresponding prime candidate chart, in which the invalid sub-regions are already removed. Figure 5-36 (b) is the Petrick function, in which 0 to 3 represents the four candidates. Each sum term within parentheses represents a sub-region, and the numbers in the parentheses represent the IDs of the candidates covering that sub-region. Then, the solution of the Petrick function is derived by finding the essential candidates which covers a sub-region alone and removing sum terms containing an essential candidate. The result is shown in Figure 5-36 (c), in

which candidates 0, 2 and 3 are identified as pedestrians.

	1F	1T	1B	2F	2T	2B	JL
I0	0.866	1.000	1.000	0.898	0.958	1.000	0.863
I1	0.890	0.956	1.000	0.824	0.726	1.000	0.714
I2	0.853	0.965	1.000	0.827	0.884	1.000	0.776
I3	0.785	0.841	1.000	0.797	0.769	1.000	0.636

Figure 5-35 The joint occupancy likelihoods for the pedestrian candidates at frame 800.

```
     11112222
I0:  X+++++XX
I1:  +++X+X++
I2:  ++XX+++X
I3:  +X++XX++
```

(a)

Petrick function:

P = (0)(3)(2)(1+2)(3)
    (1+3)(0)(0+2)

(b)

First step — find essential candidates:

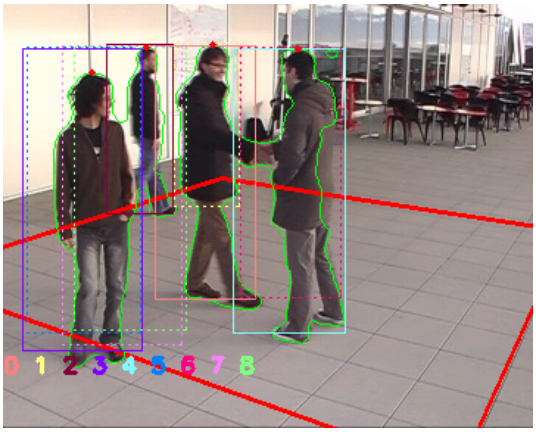
P = (0)(3)(2)

(c)

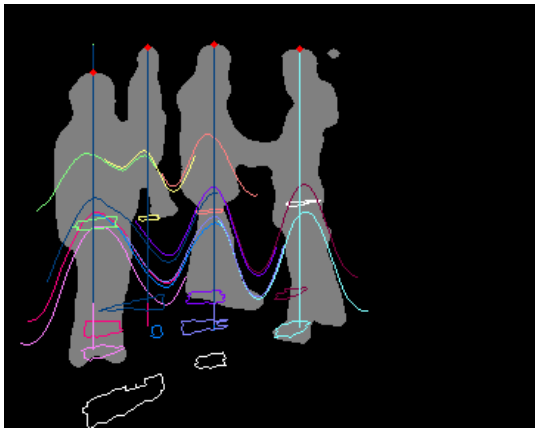
Figure 5-36 The results of the Petrick's method at frame 800: (a) the prime candidate chart, (b) the Petrick function, and (c) the simplification of the Petrick function.

When three camera views were used, the estimation of foreground intersection points is more complicated. It is no longer to find the intersection of every two lines but to calculate a point which has the minimal sum of squared distances from three projected lines. Figure 5-37 shows the detection results at frame 975 on the EPFL Terrace dataset with three camera views C0, C1 and C2, where Figure 5-37 (a), (c) and (e) are the three camera views, Figure 5-37 (b), (d) and (f) are the corresponding foreground region maps and Figure 5-37 (g) is a synthetic top view. The additional camera view is opposite to C1. In C0, pedestrian candidate 3 in purple only has one vertical line passing the top of head, since the corresponding local maximum of the template matching response is removed. In each of C1 and C2, two vertical lines are associated with this pedestrian candidate. One is the local maximum of the template matching response and the others are three tops of heads detected within the search window. Therefore, there are four foreground intersection points associated with the corresponding foreground intersection region, which is shown in the top of Figure 5-37 (g). As the result, one of them is identified as pedestrian and other three are identified as phantoms by using Petrick's method.

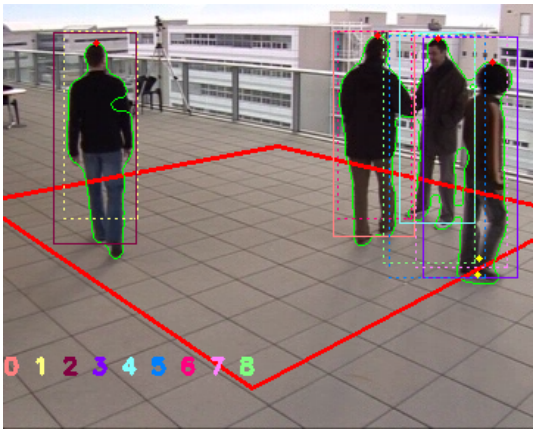
Figure 5-38 shows the joint occupancy likelihoods for the pedestrian candidates at frame 975 and Figure 5-39 shows the result of the Petrick's method at this frame. Figure 5-39 (a) shows the prime candidate chart and the corresponding Petrick function is shown in Figure 5-39 (b). Figure 5-39 (c) shows the simplification of the Petrick function, in which the algorithm goes through most of the steps. In the first step, candidates 0, 2 and 4 are detected as essential, and the sub-regions covered by candidates 3, 5, 7 and 8 are left. After finding the groups, the term (3 + 7) is remaining. This Boolean function represents that it has two sets of solutions, and the only difference is choosing candidate 3 or 7. Therefore, by comparing the joint likelihood between these two candidates, the final result will be the solution containing candidates 0, 2, 3 and 4, as shown in Figure 5-39 (d).



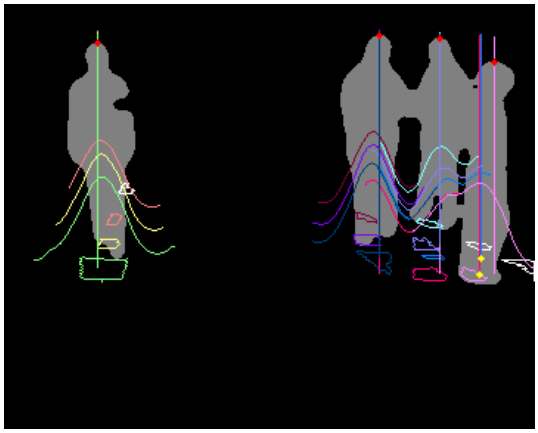
(a)



(b)

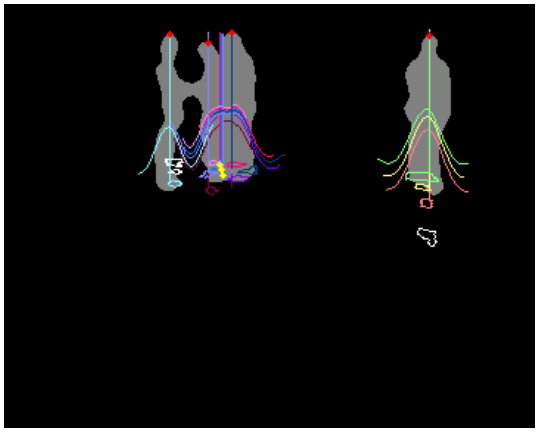


(c)

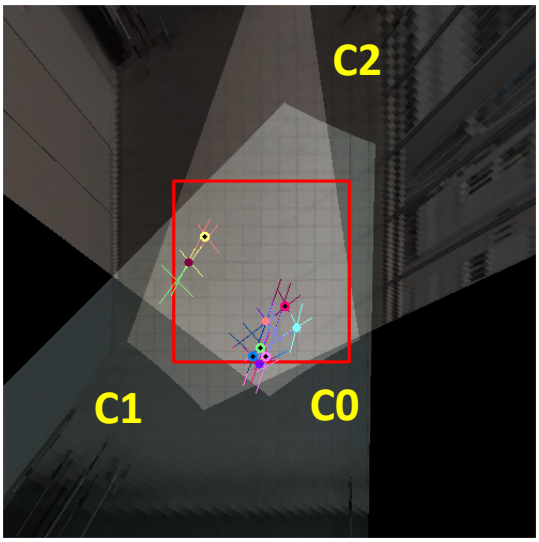


(d)

(e)



(f)



(g)

Figure 5-37 The detection results at frame 975 on the EPFL Terrace dataset with three camera views: (a) (c) (e) camera views C0, C1 and C2, (b) (d) (f) foreground region maps of camera views C0, C1 and C2, and (d) the synthetic top view.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	JL
I0	0.829	1.000	1.000	0.882	0.958	1.000	0.865	1.000	1.000	0.846
I1	0.914	1.000	1.000	0.837	0.857	1.000	0.691	0.931	0.587	0.628
I2	0.756	0.966	1.000	0.810	0.889	1.000	0.862	0.924	1.000	0.748
I3	0.759	0.845	1.000	0.756	0.976	1.000	0.814	1.000	1.000	0.728
I4	0.737	0.990	1.000	0.785	0.939	1.000	0.757	1.000	1.000	0.741
I5	0.790	0.826	1.000	0.596	0.963	1.000	0.835	1.000	1.000	0.679
I6	0.767	0.988	1.000	0.887	0.952	1.000	0.499	1.000	0.841	0.645
I7	0.655	0.980	1.000	0.682	0.974	1.000	0.751	1.000	1.000	0.685
I8	0.568	0.990	1.000	0.537	0.974	1.000	0.822	1.000	1.000	0.623

Figure 5-38 The joint likelihoods for the pedestrian candidates at frame 975.

```
     111111111111122222222223333333
I0:  XX+XXX+++X+++XX++XX++++++XXXX++X
I1:  X+++XX++++++++++X+++++++X+++++++
I2:  ++XX++++X++++++XX+++++++X+++++++
I3:  ++++++XXX+++X+++++++XXXX+XX+++++
I4:  +++++++++XXX+++++++XX++++++++XX+
I5:  ++++++XXX+++++++++XXXX+X+XXXX+++
I6:  ++++++++++X+++X++XX++++++X++++XX
I7:  ++XXX+XXX+++X++++++XXXXX+X++++++
I8:  ++XXXX+XX++++++++XXXX++X+XXX++++
```

(a)

```
P = (0+1)(0)(2+7+8)(0+2+7+8)(0+1+7+8)
    (0+1+8)(3+5+7)(3+5+7+8)(2+3+5+7+8)(0+4)
    (4+6)(4)(3+7)(0)(0+6)
    (2)(1+2)(0+6+8)(0+5+6+8)(4+5+7+8)
    (3+4+5+7+8)(3+5+7)(3+7)(3+5+7+8)(1+2)
    (0+3+5+6+7+8)(0+3+5+8)(0+5+8)(0+5)(4)
    (4+6)(0+6)
```

(b)

```
First step — find essential candidates:

P = (0)(4)(2)
    (3+5+7)(3+5+7+8)(3+7)(3+5+7)(3+7)
    (3+5+7+8)


Second step — find groups:

P = (0)(4)(2)
    (3+7)
```

(c)

```
Third step — multiply out:

Fourth step — compare likelihood:

3 Joint Likelihood:0.728
7 Joint Likelihood:0.685

P = (0)(4)(2)(3)
```

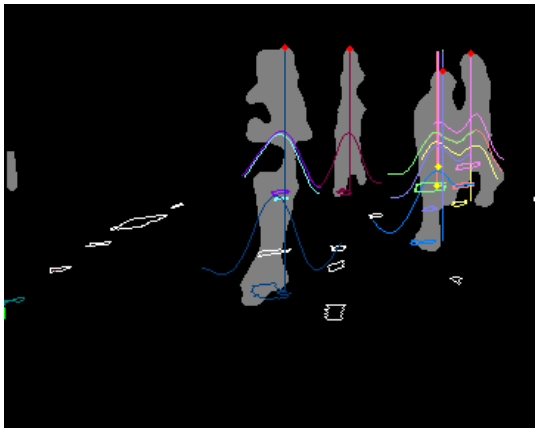
(d)

Figure 5-39 The results of the Petrick's method at frame 975: (a) the prime candidate chart, (b) the Petrick function, (c) the simplification of the Petrick function and (d) the comparison of the joint occupancy likelihoods of two solutions.

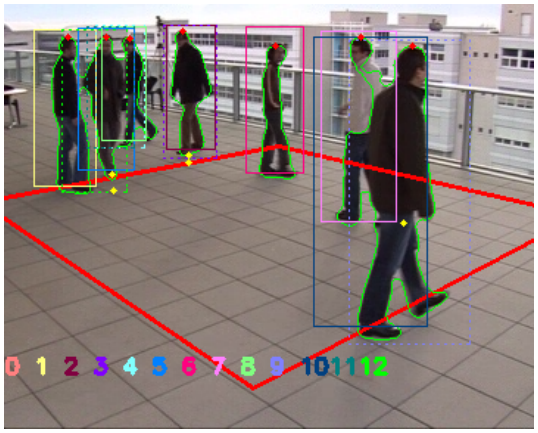
Figure 5-40 shows the detection results at frame 2025 on the EPFL Terrace dataset with four camera views C0, C1, C2 and C3, where Figure 5-40 (a), (c), (e) and (g) are the four camera views, Figure 5-40 (b), (d), (f) and (h) are the corresponding foreground region maps and Figure 5-40 (i) is a synthetic top view. The additional camera view C3 is opposite to C0. This frame was selected because seven pedestrians appear in the scene with significantly overlap and the location of one pedestrian is not accurate. The candidate 8 in light green is biased to the left in C1 (Figure 5-40 (c)). The reason is in C2 (Figure 5-40 (e)), there are two head points corresponding to this pedestrian, but none of them are the ground truth. Therefore, the algorithm chooses the head point on the right side as the head point of this pedestrian in this camera view, and the correct location was filtered out by the RSS filter, because it has a lower joint likelihood.

Figure 5-41 shows the joint occupancy likelihoods for the pedestrian candidates at frame 2025. Figure 5-42 shows the Petrick's method at the same frame. Figure 5-42 (a) shows the prime candidate chart. Figure 5-42 (b) is the Petrick function, and Figure 5-42 (c) is the solution derived by simplifying the Petrick function. Figure 5-42 (d) shows a comparison of the joint occupancy likelihoods of two set of solutions. In this case, candidates 1, 2, 5, 6, 7 and 10 are correctly detected. The detected location of candidate 8 in green is not accurate.

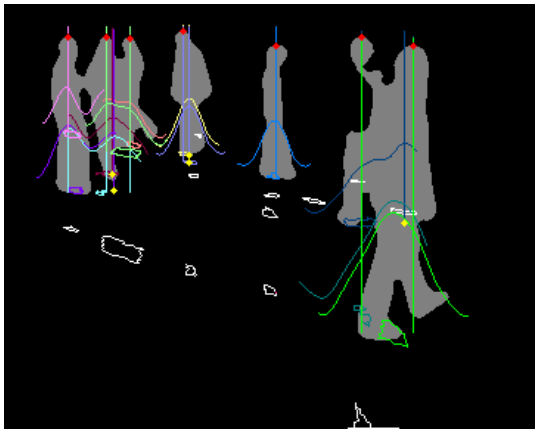
(a)



(b)

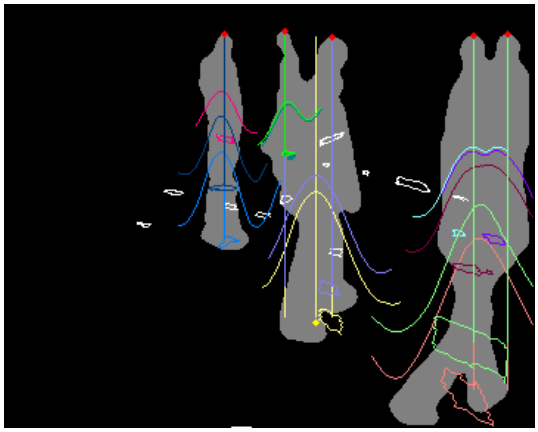


(c)

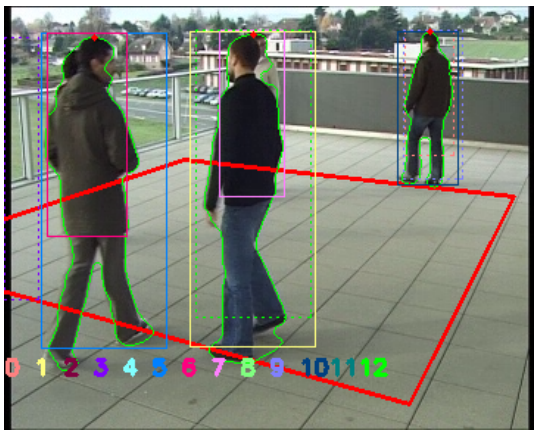


(d)

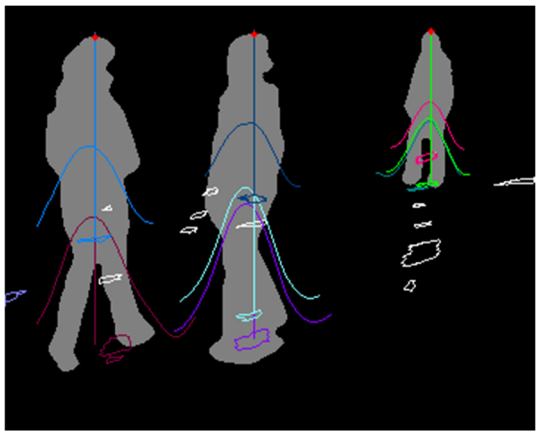
(e)



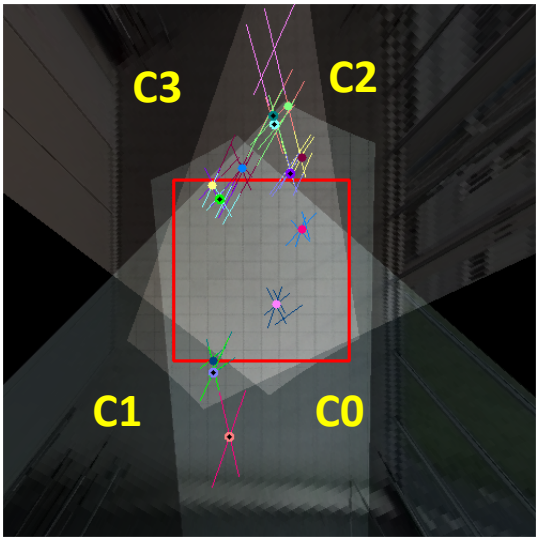
(f)



(g)



(h)



(i)

Figure 5-40 The detection results at frame 2025 on the EPFL Terrace dataset with four camera views: (a) (c) (e) (g) camera views C0, C1, C2 and C3, (b) (d) (f) (h) foreground region maps of camera views C0, C1, C2 and C3, and (i) the synthetic top view.

	1F	1T	1B	2F	2T	2B	3F	3T	3B	4F	4T	4B	JL
I0	1.000	1.000	1.000	1.000	1.000	1.000	0.853	0.943	1.000	0.854	0.976	1.000	0.819
I1	0.807	0.939	1.000	0.782	0.887	1.000	0.831	0.972	1.000	0.839	0.982	1.000	0.769
I2	0.718	1.000	1.000	0.835	0.907	1.000	0.893	1.000	1.000	1.000	1.000	1.000	0.786
I3	0.786	0.906	1.000	0.809	0.912	0.912	0.836	1.000	1.000	1.000	1.000	1.000	0.737
I4	0.778	0.698	1.000	0.715	0.809	1.000	0.898	1.000	1.000	1.000	1.000	1.000	0.656
I5	0.805	0.959	1.000	0.705	0.854	1.000	0.857	0.988	1.000	0.749	0.963	1.000	0.730
I6	0.665	0.800	1.000	0.726	0.746	1.000	0.846	1.000	1.000	0.885	0.957	1.000	0.674
I7	0.752	0.988	1.000	0.592	0.924	1.000	0.910	0.981	1.000	0.873	0.947	1.000	0.740
I8	0.760	1.000	1.000	0.708	0.797	1.000	0.851	1.000	1.000	1.000	1.000	1.000	0.715
I9	1.000	1.000	1.000	0.704	1.000	0.921	0.804	0.975	1.000	0.802	0.980	1.000	0.737
I10	1.000	1.000	1.000	0.654	0.990	1.000	0.770	0.976	1.000	0.848	0.981	1.000	0.739
I11	0.751	0.689	1.000	0.751	0.803	1.000	0.738	1.000	1.000	1.000	1.000	1.000	0.613
I12	0.819	0.942	1.000	0.511	0.910	1.000	0.802	0.971	1.000	0.852	0.980	1.000	0.694

Figure 5-41 The joint occupancy likelihoods for the pedestrian candidates at frame 2025.

```
     111111111112222222222223333333333333333344444444
I0:  ++++++++++++++++++++++++++X+++++++++++++++++X++
I1:  ++X++++++++++++XXX++++++++++++++XXX++++XXX++++
I2:  ++++++XXXXXX++++++++++++++XXX++++++X++++++++++
I3:  ++++XXXX++XX++++++++++++++XXX++++++++++++++++++
I4:  +++++XX+++++X+X+++XX++++++++++XXXXX+XX++++++++
I5:  +++X++++++++++X+XXXX++X+++++++XXXX++X+++++++XX
I6:  ++++XXX+++X++X+++++++++XXX++++++++++++++++++X
I7:  XXX++++++++++++++++X+++XX++++++++++++++++X++++
I8:  ++++++XXX+++X+X++++X+++++++++XXXXXX+XXX+++++++
I9:  ++++++++++++++++++++XX++++X++++++++++++++++XX++
I10: ++++++++++++++++++++XX++++XX++++++++++++++++XX++
I11: +++++XX+++++++X++XXX++++++++++XXXX+XXX+++++++
I12: +XX++++++++++++XXXXX++X+++++++XXX+++++++XX++++
```

(a)

```
P = (7)(7+12)(1+7+12)(5)(3+6)
    (3+4+6+11)(2+3+4+6+8+11)(2+3+8)(2+8)(2)
    (2+3+6)(2+3)(4+8)(6)(4+5+8+11)
    (1+12)(1+5+12)(1+5+11+12)(4+5+11+12)(4+5+8+11+12)
    (7+9+10)(9+10)(5+12)(6)(6+7)
    (0+6+7)(2+3+9+10)(2+3+10)(2+3)(8)
    (4+5+8+12)(4+5+8+11+12)(1+4+5+8+11+12)(1+4+5+8+11)(1+4+8+11)
    (2)(4+5+8+11)(4+8+11)(8+11)(1)
    (1+12)(1+7+12)(9+10)(0+9+10)(5)
    (5+6)
```

(b)

```
First step — find essential candidates:

P = (7)(5)(2)(6)(8)(1)
    (9+10)(9+10)(0+9+10)

Second step — find groups:

P = (7)(5)(2)(6)(8)(1)
    (9+10)
```

(c)

```
Third step — multiply out:

Fourth step — compare likelihood:

9 Joint Likelihood:0.737
10 Joint Likelihood:0.739

P = (7)(5)(2)(6)(8)(1)(10)
```

(d)

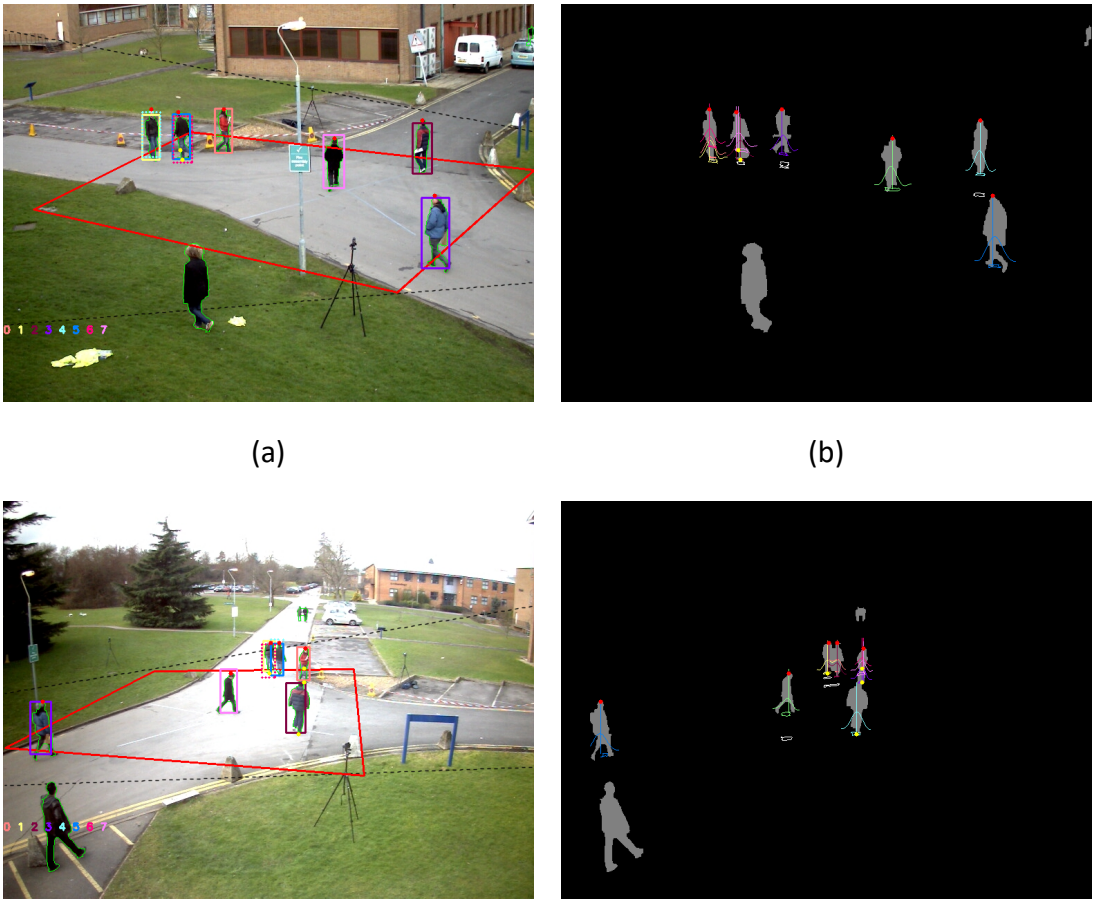
Figure 5-42 The results of the Petrick's method at frame 2025: (a) the prime candidate chart, (b) the Petrick function, (c) the simplification of the Petrick function and (d) the comparison of the joint occupancy likelihoods of two solutions.

Results on the PETS2009 CC Dataset

In the experiment on the PETS2009 CC dataset, two far-field views (C1 and C2) are selected. Compared with the EPFL Terrace Dataset, the PETS2009 CC dataset has better foreground detection but the far-field views are a challenge for the top of head detection. In the joint likelihoods and prime candidate chart, the camera indices 1 and 2 are corresponding to C1 and C2 in the dataset.

Figure 5-43 shows the detection results at frame 666 on the PETS2009 CC dataset with two camera views, where Figure 5-43 (a) and (c) are the two camera views, Figure 5-43 (b) and (d) are the foreground region maps of C1 and C2, and Figure 5-43 (e) is a synthetic top view. The borderlines of the overlapping fields of view are shown as black dashed lines in camera views. This frame was selected because it is a simple example but there is a competition between two sets of candidates, which is similar to the example in Figure 5-7. In C2, the top of head of candidate 2 in dark red is not detected, because the foreground of this candidate is merged with that of candidate 0.

Figure 5-44 shows the joint occupancy likelihoods for the pedestrian candidates at frame 666. Figure 5-45 shows the results of the Petrick's method at this frame. Figure 5-45 (a) shows the prime candidate chart. Figure 5-45 (b) is the corresponding Petrick function. Figure 5-45 (c) details the simplification process and the solution of the Petrick function. After step 1, candidates 0, 2, 3 and 7 are identified as essential. At this stage, five sum terms along with the four essential candidates in the Petrick function are remaining. Then, this function is further simplified by using Eq. (4.9), (4.10) and (4.11) iteratively. As shown in the result, there are two sets of solutions for this function. Candidates {1, 5} and {4, 6} cover the same sub-regions. Therefore, the joint occupancy likelihoods of these two sets of solutions are compared, as shown in Figure 5-45 (d). Finally, candidates 0, 1, 2, 3, 5 and 7 are correctly detected as pedestrians.



(a) (b)

(c) (d)

(e)

Figure 5-43 The detection results at frame 666 on the EPFL Terrace dataset with two camera views: (a) (c) camera views C1 and C2, (b) (d) foreground region maps of camera views C1 and C2, and (e) a synthetic top view.

	1F	1T	1B	2F	2T	2B	JL
I0	0.791	0.951	0.951	0.887	0.871	1.000	0.743
I1	0.875	1.000	1.000	0.767	0.791	1.000	0.729
I2	0.879	1.000	0.914	0.763	1.000	1.000	0.783
I3	0.824	1.000	1.000	0.747	0.959	1.000	0.768
I4	0.892	1.000	1.000	0.682	0.723	1.000	0.663
I5	0.824	1.000	0.952	0.722	1.000	1.000	0.752
I6	0.798	1.000	0.766	0.722	1.000	0.619	0.523
I7	0.821	0.798	1.000	0.664	0.708	0.950	0.541

Figure 5-44 The joint occupancy likelihoods for the pedestrian candidates at frame 666.

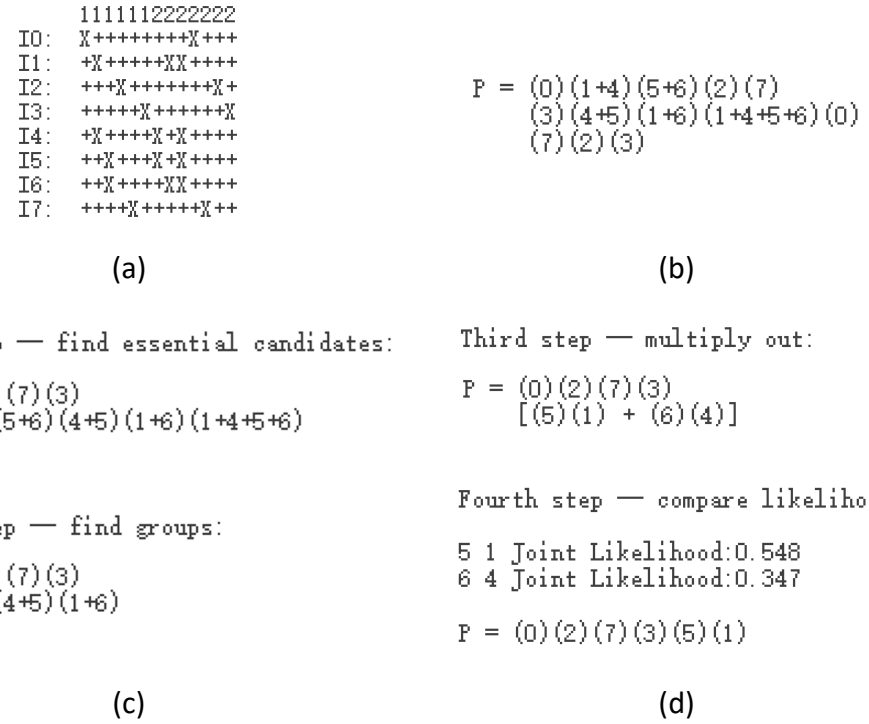
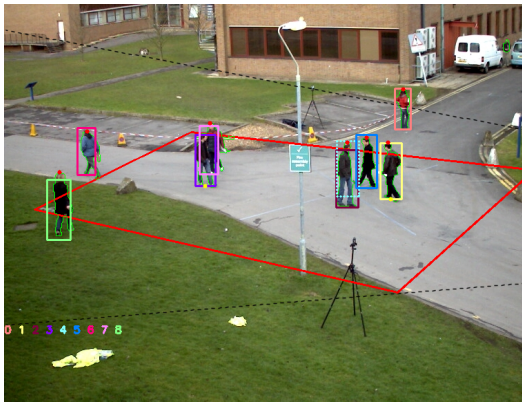
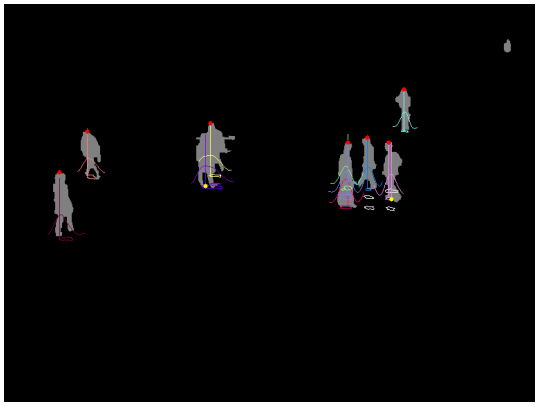


Figure 5-45 The results of the Petrick's method at frame 666: (a) the prime candidate chart, (b) the Petrick function, (c) the simplification of the Petrick function and (d) the comparison of the joint occupancy likelihoods of two solutions.

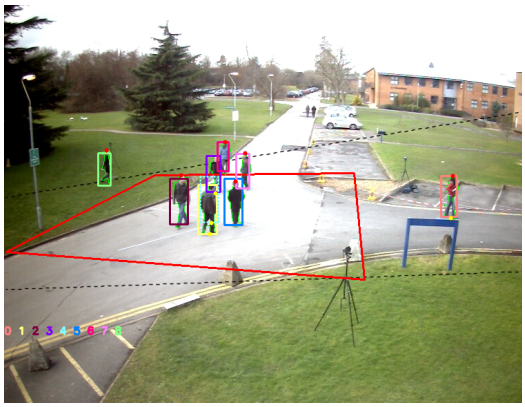
Figure 5-46 shows the detection results at frame 723 on the PETS2009 CC dataset with two camera views. Figure 5-46 (a) and (b) are the two camera views, Figure 5-46 (b) and (d) are the foreground region maps of C1 and C2, and Figure 5-46 (e) is a synthetic top view. This frame was selected because it demonstrates the value of simultaneously using both the template matching and top-of-head detection. An example is shown in C1. Candidate 7 in pink is on the upper right corner of candidate 3 in purple and they are overlapped. In this case, the top of head of candidate 3 cannot be detected and the observation of this candidate comes from the local maximum of the template matching response. The local maximum of the template matching response of candidate 7 is ignored because it is very close to the top of head. Therefore, the template matching response and the top-of-head detection are complementary. In addition, the top-of-head points of candidates 3 and 9 in C2 are also not detected, but the template matching can provide good observations. Figure 5-47 and Figure 5-48 show the joint occupancy likelihoods and the results of the Petrick's method at frame 723. Candidates 0, 1, 2, 3, 5, 7 and 8 are correctly detected as pedestrians.



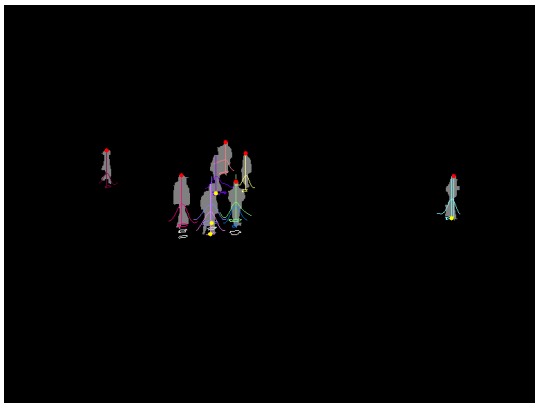
(a)



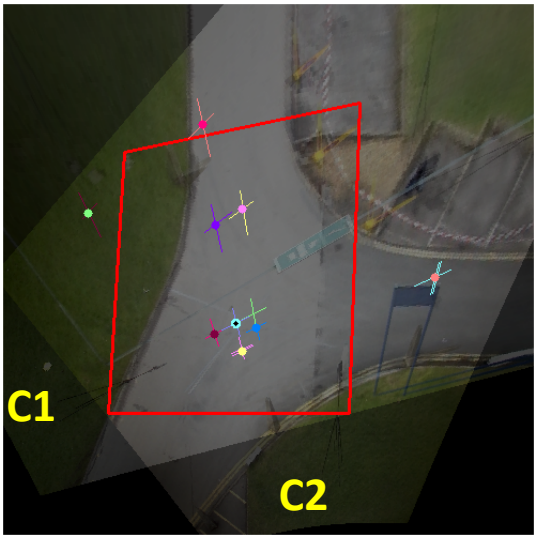
(b)



(c)



(d)



(e)

Figure 5-46 The detection results at frame 723 on the EPFL Terrace dataset with two camera views: (a) (c) camera views C1 and C2, (b) (d) foreground region maps of camera views C1 and C2, and (e) a synthetic top view.

	1F	1T	1B	2F	2T	2B	JL
I0	0.878	0.947	1.000	0.856	1.000	1.000	0.844
I1	0.830	1.000	1.000	0.837	1.000	1.000	0.834
I2	0.830	1.000	1.000	0.814	1.000	1.000	0.822
I3	0.766	1.000	1.000	0.819	1.000	1.000	0.792
I4	0.800	0.921	1.000	0.777	1.000	1.000	0.757
I5	0.793	0.837	1.000	0.779	0.815	1.000	0.649
I6	0.739	0.773	1.000	0.778	1.000	1.000	0.666
I7	0.775	1.000	1.000	0.727	1.000	1.000	0.751
I8	0.765	1.000	0.786	0.713	1.000	0.803	0.587

Figure 5-47 The joint occupancy likelihoods for the pedestrian candidates at frame 723.

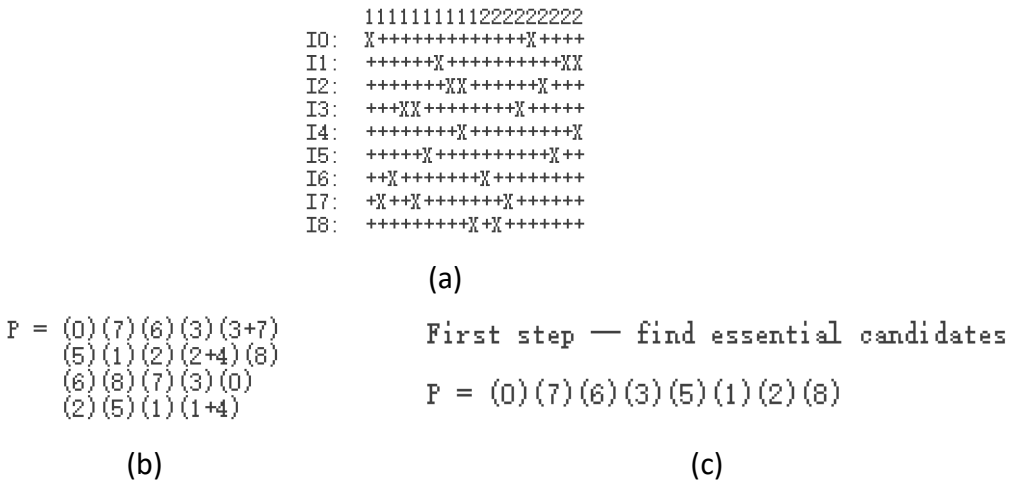


Figure 5-48 The results of the Petrick's method at frame 732: (a) the prime candidate chart, (b) the Petrick function, and (c) the simplification of the Petrick function.

5.3 Quantitative Results

For performance evaluation of the proposed algorithm, the ground truths of the PETS2009 City Centre dataset, PETS2009 S2L1 dataset and EPFL Terrace dataset were used and a program was developed to automatically compare the locations of detected pedestrians with those of the ground-truth pedestrians.

The ground truths of the PETS2009 City Centre and S2L1 dataset are recorded as the manually annotated rectangle for each pedestrian in camera view C1, which were created by Anton Milan [96]. At each frame, the ground truths of the pedestrians in camera view C1 are recorded as a set of rectangles represented by the coordinates of their top-left corners and sizes. Such ground truths are converted to a set $G^1 = \{G_1^1, \dots, G_{N_G}^1\}$ of N_G rectangles, in which G_j^1 is a synthetic image by putting a filled rectangle on an empty background and $G_j^1 \in \{0,1\}^{W \times H}$, where the value for background pixels is set to zero and that in the rectangle is set to one. Since the rectangles were manually annotated, the height and width of each rectangle fit the corresponding pedestrian very well.

The ground truths of the EPFL Terrace dataset are the locations for the pedestrians in a top view, which were created by EPFL [97]. It is available in one of every 25 frames. The location of each pedestrian was then warped to each camera view by putting a filled rectangle, at the corresponding location in camera c, on an empty background. Therefore, the ground truth for camera view c is a set $G^c = \{G_1^c, \dots, G_{N_G}^c\}$ of $\mathrm{N_G}$ rectangles. Since the rectangles were generated by the program, the height and width of each pedestrian in the ground truth are the average height and width of the pedestrians standing at a specific location in camera view c.

The set of the detected pedestrians in camera view c is denoted by a set $D^c = \{D_1^c, \dots, D_{N_D}^c\}$ of $\mathrm{N_D}$ rectangles, in which D_k^c is a filled rectangle put on an empty background and $D_k^c \in \{0,1\}^{W \times H}$. Since the rectangles were generated by the program, the height and width of each pedestrian are the average height and width of the pedestrians at that location.

In the evaluation, the true positives (TP) were obtained by finding all the correctly detected pedestrians in the set of ground truth (GT). The false positive (FP) is the number of the detected pedestrians which cannot match any ground-truth pedestrian. The false negative (FN) is the number of the ground-truth pedestrians which cannot match any detected pedestrian. It can be found that GT = TP + FN. The method to calculate TP, FP and FN are introduced in the following section.

5.3.1 Evaluation Methodology

Given the ground truth pedestrians G^c and the detected pedestrians D^c, the overlap information between them can be used to evaluate the performance of the detection algorithm. The overlap ratio between a ground-truth pedestrian and a detected pedestrian is defined as the intersection area of the two rectangles divided by the union area of them, which is shown as follows:

$$\rho_{j,k}^{c} = \frac{\#(G_j^c \cap D_k^c)}{\#(G_j^c \cup D_k^c)} \tag{5.1}$$

Figure 5-49 shows an example of the overlap ratio. The rectangle with dotted lines represents a ground-truth pedestrian; the rectangle with dashed lines represents a detected pedestrian; the rectangle with solid lines represents the intersection area and the regions in grey are the union area. When the overlap ratio between a ground-truth pedestrian and a detected pedestrian in camera c is equal to or greater than 30%, they are thought of being matched in this camera view, which is expressed as:

$$M_{j,k}^{c} = \begin{cases} 1, & if\ \rho_{j,k}^{c} \geq 0.3 \\ 0, & otherwise \end{cases} \tag{5.2}$$

The example in Figure 5-49 shows the case that the overlap ratio between the ground-truth pedestrian and the detected pedestrian is 30%. Suppose the rectangle of the ground-truth pedestrian and that of the detected pedestrian are of the same size, a 30% overlap ratio means the area of the intersection is $6/13$ of the ground-truth rectangle area.

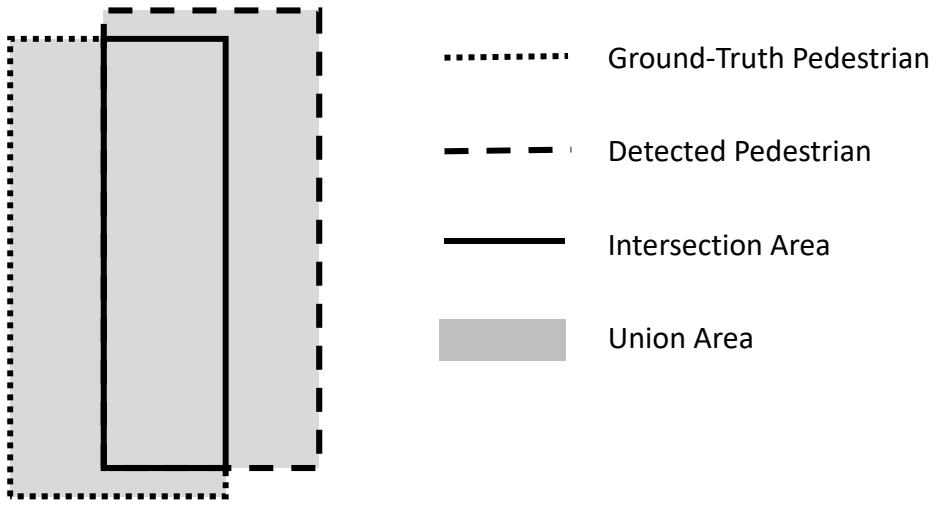


Figure 5-49 An example of the overlap ratio.

If the ground truth is available in all camera views, a pedestrian is thought of being correctly detected only when the ground-truth pedestrian and the detected pedestrian are matched in all the camera views, which is expressed as:

$$M_{j,k} = \begin{cases} 1, & if \prod_{c=1}^{C} M_{j,k}^{c} = 1 \\ 0, & otherwise \end{cases} \tag{5.3}$$

Therefore, the false negative (FN) is the number of the ground-truth pedestrians which cannot match any detected pedestrian.

$$\mathrm{FN} = \left\{ j \middle| \sum_{k=1}^{n} M_{j,k} = 0 \right\} \tag{5.4}$$

Then, the true positive number (TP) is calculated as follows:

$$\mathrm{TP} = \mathrm{GT} - \mathrm{FN} \tag{5.5}$$

To avoid multiple detections being matched to one ground truth, the false positives (FP) are calculated as the difference between the number of the detected pedestrians and that of the true positives:

$$\mathrm{FP} = N_D - \mathrm{TP} \tag{5.6}$$

In the evaluation, it is difficult to decide whether a person near the borders of the AOI is inside or outside the AOI. To cope with this problem, a buffer zone around the AOI was used. It was set to $\pm$15 cm in the EPFL Terrace dataset and $\pm$25 cm in

the PETS2009 CC dataset and PETS2009 S2L1 dataset. When a false detection in the buffer zone is matched to a ground truth out of the AOI, this false detection is ignored. All the false negatives in the buffer zone are neglected, too.

For a performance comparison with other state-of-the-art algorithms, five metrics were used: MDR (missed detection rate), FDR (false detection rate), TER (total error rate), $\mathrm{PRECISION}$ and RECALL. The definitions of these metrics are as follows:

$$\begin{aligned} \mathrm{MDR} &= \mathrm{FN}/\mathrm{GT} \\ \mathrm{FDR} &= \mathrm{FP}/\mathrm{GT} \\ \mathrm{TER} &= \mathrm{MDR} + \mathrm{FDR} \\ \mathrm{PRECISION} &= \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \\ \mathrm{RECALL} &= \mathrm{TP}/\mathrm{GT} \end{aligned} \tag{5.7}$$

A lower value in MDR, FDR and TER, or a larger value in $\mathrm{PRECISION}$ and RECALL indicates better performance. Note that MDR, $\mathrm{PRECISION}$ and RECALL are less than or equal to 1; FDR and TER may exceed 1 in case of many false alarms. Since in some compared methods, only the $\mathrm{PRECISION}$ and RECALL results are available, then MDR and FDR are retrieved from $\mathrm{PRECISION}$ and RECALL. Since $\mathrm{PRECISION} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and $\mathrm{RECALL} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, we have:

$$\begin{aligned} \mathrm{FP}/\mathrm{TP} &= 1/\mathrm{PRECISION} - 1 \\ \mathrm{FN}/\mathrm{TP} &= 1/\mathrm{RECALL} - 1 \end{aligned} \tag{5.8}$$

Then,

$$\mathrm{MDR} = \mathrm{FN}/(\mathrm{TP} + \mathrm{FN}) = 1 - \mathrm{RECALL} \tag{5.9}$$

$$\begin{aligned} \mathrm{FDR} &= \mathrm{FP}/(\mathrm{TP} + \mathrm{FN}) \\ &= (\mathrm{FP}/\mathrm{TP})/[1 + (\mathrm{FN}/\mathrm{TP})] \\ &= (1/\mathrm{PRECISION} - 1)\mathrm{RECALL} \end{aligned} \tag{5.10}$$

5.3.2 Validation of Parameters

This experiment aims at validating key parameters of the proposed algorithms. In the top-down approach, there are two key parameters. The first parameter is the average height of pedestrians, which affects the height and width of candidate box r_i^c and the standard deviations of the Gaussian distributions for foot and head likelihoods. The second parameter is the grid resolution, which is defined as the distance between two adjacent discrete locations. In the bottom-up approach, only the average height of pedestrians is validated. Furthermore, the speeds of the proposed algorithms were also tested.

The PETS2009 CC dataset and EPFL Terrace dataset were used to validate these parameters. In this experiment, two camera views were used. In the PETS2009 CC dataset, the range of the average height was set from 160 cm to 200 cm; in the EPFL Terrace dataset, it was set from 170 cm to 210 cm. In the top-down approach, the grid resolution was set as the multiples of 5 cm for the PETS2009CC dataset and it was set as the multiples of 3 cm in the EPFL Terrace dataset. The grid resolution in the PETS2009 dataset is lower than that of the EPFL Terrace dataset because the AOI in the PETS2009 dataset is larger than that in the EPFL Terrace dataset.

Top-Down Approach with Quine-McCluskey Method

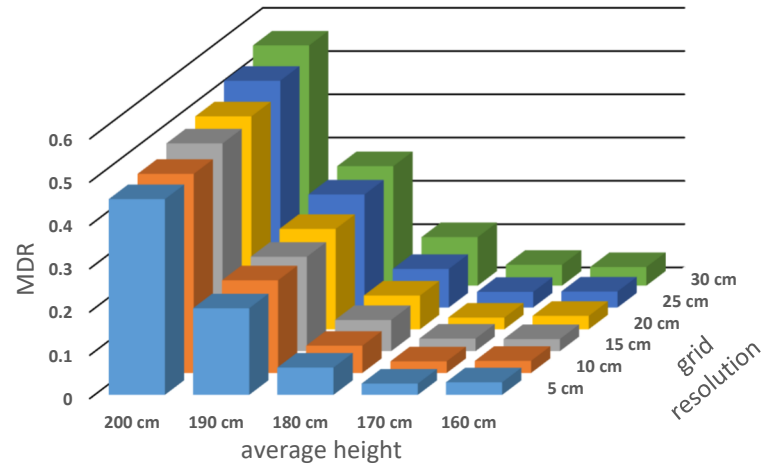
Table 5.3 shows the MDR, FDR and TER in variation with the average height of pedestrians and the grid resolution in the top-down approach on the PETS2009 CC dataset. Figure 5-50 is the visualisation of the result for each metric. The average height of pedestrians varies from 160 cm to 200 cm with an increment of 10 cm, and the grid resolution varies from 5 cm to 35 cm with an increment of 5 cm. The missed detections in this experiment are mainly due to foreground detection failures and the static occlusion by a road sign. The false detections often occur in accompany with missed detections. When a pedestrian is missed in detection, its foreground region may become the essential part of a phantom. It is observed that the MDR increases when the grid resolution decreases and the FDR is not so

dependent on the grid resolution. The MDR notably increases when the average height increases from 180 cm to 190 cm. This corresponds to the scenario where the top of the candidate box is much higher than the pedestrian's head, the head likelihood is greatly punished, and the candidate will be filtered out due to its very low joint likelihood. The minimum value TER = 0.051 is identified at the average height 170 cm and the grid resolution 5 cm.

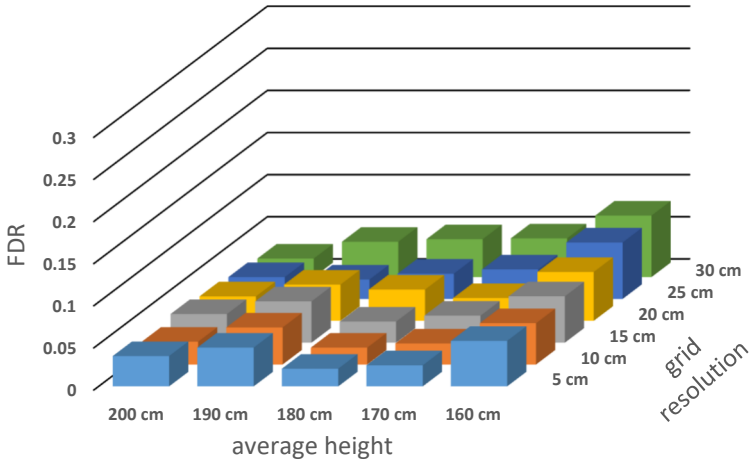
Table 5.4 shows the MDR, FDR and TER in variation with the average height of pedestrians and the grid resolution on the EPFL Terrace dataset. Figure 5-51 is the visualisation of the result for each metric. In the experiments, the average height of pedestrians varies from 170 cm to 210 cm with an increment of 10 cm, and the grid resolution varies from 3 cm to 21 cm with an increment of 6 cm. Since the camera views in this dataset are at eye level. The size of pedestrians in these camera views is notably larger than those in the PETS2009 CC dataset. Since the occlusion between pedestrians is significantly heavier, the TER is greater than that of the PETS2009 CC dataset. The missed detections in this experiment are mainly due to foreground detection failures and the lack of the observations of some pedestrians (e.g. Figure 5-19). The false detections often occur in accompany with missed detections. It can be observed that the MDR is not significantly changed except the grid is too sparse (21 cm) or the average height is too high (210 cm). The FDR increases when the average height of pedestrians decreases. This corresponds to the scenario where a candidate box is obviously lower than the corresponding pedestrian and the foregrounds of the pedestrian's head and shoulder may become the essential part of a phantom. The minimum value TER = 0.222 is identified at the average height 200 cm and the grid resolution 6 cm.

Table 5.3 Validation of the average height and grid resolution on the PETS2009 CC dataset with two camera views.

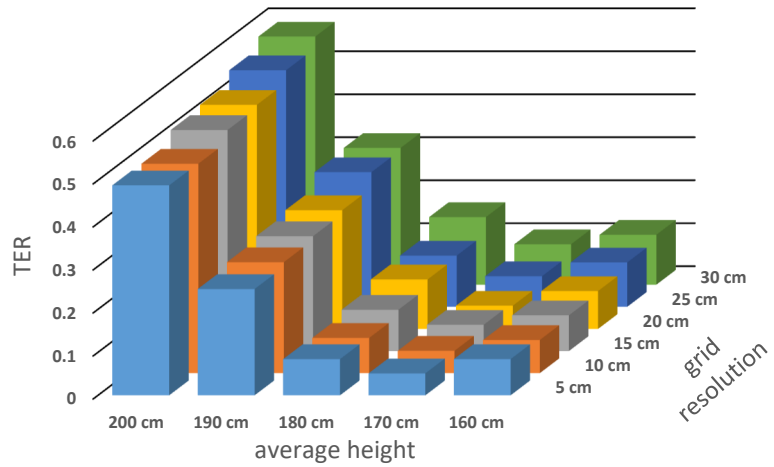
Average Height		Grid resolution					
		5 cm	10 cm	15 cm	20 cm	25 cm	30 cm
160 cm	MDR	0.029	0.028	0.028	0.031	0.037	0.043
	FDR	0.054	0.049	0.055	0.058	0.067	0.073
	TER	0.084	0.077	0.083	0.088	0.103	0.116
170 cm	MDR	0.026	0.027	0.029	0.027	0.036	0.048
	FDR	0.025	0.025	0.032	0.027	0.035	0.046
	TER	0.051	0.052	0.061	0.054	0.071	0.094
180 cm	MDR	0.063	0.063	0.072	0.078	0.089	0.112
	FDR	0.021	0.020	0.025	0.037	0.030	0.045
	TER	0.084	0.082	0.096	0.115	0.119	0.157
190 cm	MDR	0.200	0.214	0.218	0.232	0.261	0.276
	FDR	0.046	0.044	0.049	0.043	0.023	0.042
	TER	0.247	0.258	0.267	0.276	0.313	0.318
200 cm	MDR	0.452	0.460	0.480	0.492	0.524	0.555
	FDR	0.036	0.027	0.034	0.029	0.026	0.022
	TER	0.488	0.487	0.514	0.521	0.550	0.577



(a)



(b)

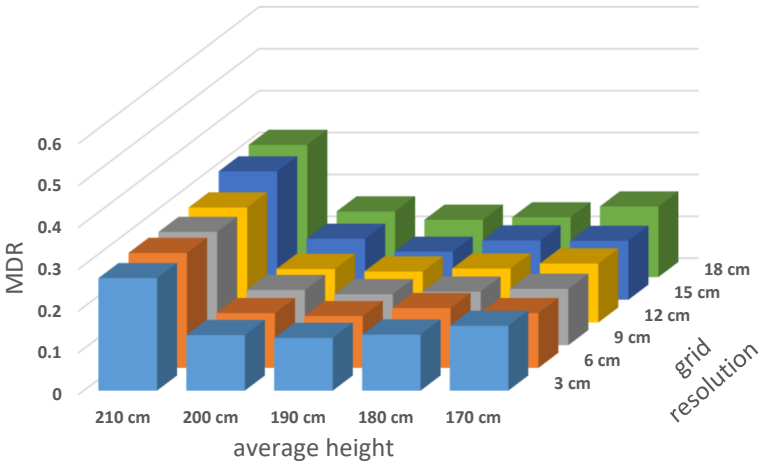


(c)

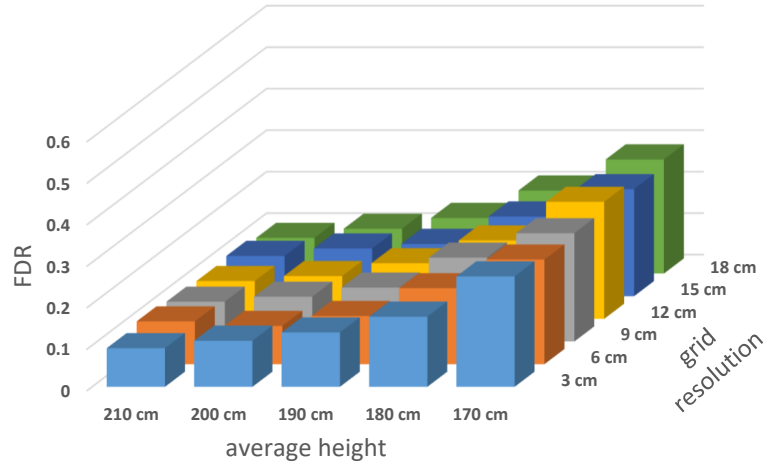
Figure 5-50 Visualisation of the validation results on the average height and grid resolution, in terms of MDR, FDR and TER.

Table 5.4 Validation of the average height and grid resolution on the EPFL Terrace dataset with two camera views.

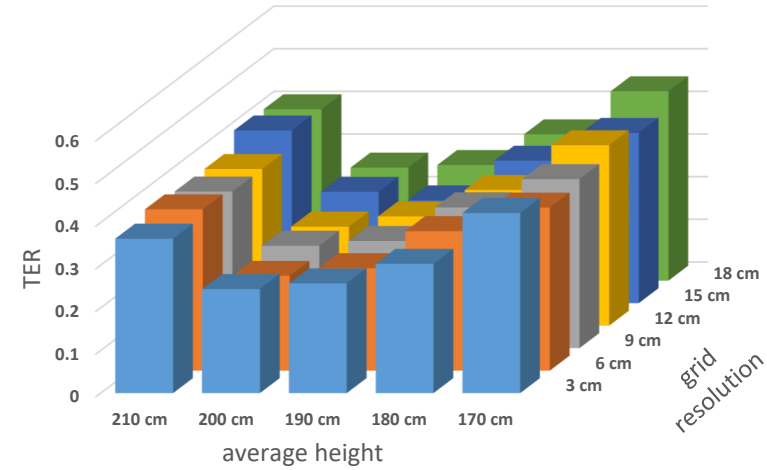
Average Height		Grid Resolution					
		3 cm	6 cm	9 cm	12 cm	15 cm	18 cm
170 cm	MDR	0.155	0.131	0.135	0.141	0.141	0.169
	FDR	0.266	0.252	0.261	0.282	0.258	0.275
	TER	0.422	0.382	0.396	0.423	0.399	0.444
180 cm	MDR	0.134	0.143	0.128	0.129	0.142	0.143
	FDR	0.169	0.183	0.202	0.189	0.192	0.200
	TER	0.303	0.327	0.330	0.318	0.334	0.343
190 cm	MDR	0.126	0.124	0.122	0.122	0.115	0.137
	FDR	0.131	0.116	0.13	0.134	0.126	0.134
	TER	0.257	0.240	0.251	0.256	0.241	0.271
200 cm	MDR	0.133	0.131	0.132	0.128	0.146	0.157
	FDR	0.111	0.092	0.108	0.103	0.115	0.108
	TER	0.244	0.222	0.240	0.232	0.261	0.265
210 cm	MDR	0.269	0.275	0.271	0.275	0.307	0.316
	FDR	0.093	0.103	0.096	0.091	0.097	0.086
	TER	0.362	0.378	0.367	0.367	0.405	0.402



(a)



(b)



(c)

Figure 5-51 Visualisation of the validation results on the average height and grid resolution, in terms of MDR, FDR and TER.

Bottom-Up Approach with Petrick's Method

Table 5.5 and Table 5.6 show the MDR, FDR and TER in variation with the average height of pedestrians in the bottom-up approach on the PETS2009 CC dataset and the EPFL Terrace dataset. Figure 5-52 is the visualisation of the result for each metric. The average height of pedestrians varies from 160 cm to 200 cm for the PETS2009 CC dataset and from 170 cm to 210 cm for the EPFL Terrace dataset. For both the datasets, the increment of the average height of pedestrians is 10 cm.

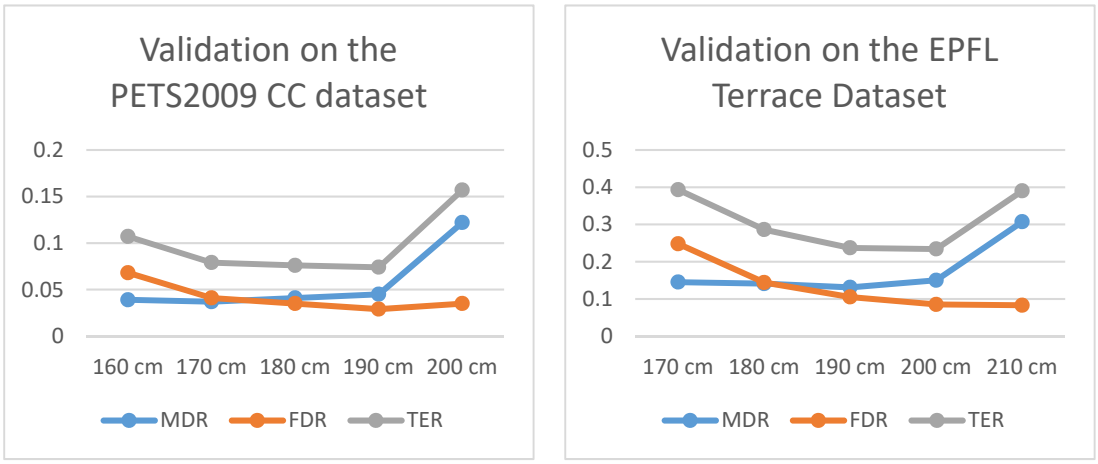
On both the datasets, the FDR increases when the average height of pedestrians decreases, which is similar to that in the top-down approach. The MDR is not significantly changed except the average height of pedestrians is set too high (200 cm in the PETS2009 CC dataset and 210 cm in the EPFL Terrace dataset). On the PETS2009 CC dataset, it can be seen that the MDR is significantly increased at an average height of 200 cm in the bottom-up approach and at an average height of 190 cm in the top-down approach. The overall MDR of the top-down approach is lower than that of the bottom-up approach. The missed detections in the bottom-up approach are mostly caused by the broken foreground regions, which affects the generation of the foreground intersection regions and the detection of the pedestrians' heads and feet.

Table 5.5 Validation of the average height of pedestrians on the PETS2009 CC dataset with two camera views.

Average Height	160 cm	170 cm	180 cm	190 cm	200 cm
MDR	0.039	0.037	0.041	0.045	0.122
FDR	0.068	0.041	0.035	0.029	0.035
TER	0.107	0.079	0.076	0.074	0.157

Table 5.6 Validation of the average height of pedestrians on the EPFL Terrace dataset with two camera views.

Average Height	170 cm	180 cm	190 cm	200 cm	210 cm
MDR	0.145	0.141	0.131	0.150	0.307
FDR	0.248	0.144	0.105	0.085	0.083
TER	0.393	0.286	0.237	0.234	0.390



(a) (b)

Figure 5-52 Visualisation of the validation results on the average height in terms of MDR, FDR and TER: (a) validation on the PETS2009 CC dataset and (b) validation on the EPFL Terrace Dataset.

Speed Test

The speeds of the proposed algorithms were also tested by using a PC with an Intel i5 4-core CPU running at 3.20 GHz. Execution times for running the algorithm with different grid resolutions were tested on the PETS2009 CC dataset and the EPFL Terrace dataset with two camera views. The time spent for processing each frame was obtained by taking the average. For the PETS2009 CC dataset, the GMM algorithm was used for the foreground detection. To cope with the automatic white balance in the EPFL Terrace dataset, SuBSENSE was used to extract foregrounds, which is more time-consuming. If the GMM algorithm or the SuBSENSE algorithm for foreground segmentation are replaced by a faster but less adaptive method, the

processing speed can be greatly increased. The results are shown in Table 5.7 and Table 5.8.

With the increase of the grid resolution in the top-down approach, it will take more time to generate the candidate boxes. The integral image [98] was used to speed up the candidate generation for the top-down approach. The value of each point in the integral image is the sum of all the pixels above and to the left of that point. Therefore, the sum of foreground pixels within any rectangular area only requires four points in the integral image. The estimation of the template matching response as well as foot and head likelihoods will be more efficient. In this step, the time cost of the bottom-up approach is higher than that of the top-down approach on the PETS2009 CC Dataset and most cases on the EPFL Terrace Dataset. The time for the QM method and the Petrick's method is neglectable. The time for the Terrace dataset is much longer than that for the PETS2009 CC dataset, since the foreground regions in the former dataset are much larger than those in the latter.

Table 5.7 Execution times for running the proposed algorithm on the PETS2009 CC dataset with two camera views.

Approach	Top Down	Top Down	Top Down	Top Down	Top Down	Top Down	Bottom Up
Grid Resolution (cm)	5	10	15	20	25	30	NA
Time/Frame (ms)	77	53	47	45	45	44	93
GMM (ms)	35	35	36	35	34	35	31
Candidate Generation (ms)	37	12	6	5	5	4	56
QM Method (ms)	5	6	5	5	6	5	NA
Petrick's Method (ms)	NA	NA	NA	NA	NA	NA	6
FPS	13.0	18.9	21.3	22.2	22.2	22.7	10.8

Table 5.8 Execution times for running the proposed algorithm on the EPFL Terrace dataset with two camera views.

Approach	Top Down	Top Down	Top Down	Top Down	Top Down	Top Down	Bottom Up
Grid Resolution (cm)	3	6	9	12	15	18	NA
Time/Frame (ms)	262	139	118	109	104	103	169
SuBSENCE (ms)	95	94	95	93	93	94	95
Candidate Generation (ms)	116	43	22	14	10	8	71
QM Method (ms)	1	2	1	2	1	1	NA
Petrick's Method (ms)	NA	NA	NA	NA	NA	NA	3
FPS	3.8	7.2	8.5	9.2	9.6	9.7	5.9

5.3.3 Quantitative Evaluation

In this section, the two proposed methods are compared with other state-of-the-art algorithms in the five metrics: MDR, FDR, TER, PRECISION and RECALL. Three datasets were used in this evaluation. These datasets, as well as the five performance metrics, were selected because they are widely used in the evaluation of the existing algorithms for multiview pedestrian detection.

Our performance evaluation results were compared with those of some state-of-the-art algorithms POM [5], 3DMPP [8], MvBN [11], Khan's [67], Ge and Collins's [69] and three methods based on deep learning, such as CNN/CRF [75], POM+CNN [75] and IDIAP [76]. In the proposed methods, the parameters were set to minimize TER, while the corresponding FDR, MDR, PRECISION and RECALL values are also listed. The evaluation results of POM method on the PETS2009 S2L1 dataset

are obtained from [11] and those on the PETS2009 CC dataset and the EPFL Terrace dataset are obtained from [8]. The evaluation results of Ge and Collines's are obtained from [76]. In our evaluation of Khan's method, five parallel planes evenly distributed across the average height of pedestrians were used and the threshold for multi-layered foreground intersections was set to four layers. For 3DMPP and MvBN, we used their own evaluation results and the same performance metrics in the belief that the parameters of an algorithm can be best adjusted by its authors. The POM and 3DMPP were evaluated only in TER on the PETS2009 CC datasets with two camera views and the EPFL Terrace datasets with two camera views. The MvBN was evaluated in TER, PRECISION and RECALL in [11]. Its MDR and FDR data were retrieved from its PRECISION and RECALL values, as shown in Eq. (5.9) and (5.10).

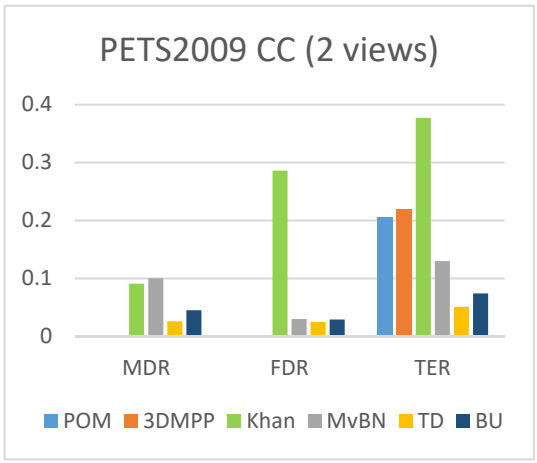
In Table 5.9, the proposed top-down approach (TD) significantly outperforms the other algorithms in terms of TER, MDR and RECALL on the PETS2009 CC dataset and PETS2009 S2L1 dataset. The MDR of the proposed algorithm is less than half of the MDR values of the 3DMPP and MvBN methods. The proposed bottom-up approach (BU) is compared with other algorithms in two camera views. Compared with the top-down approach, the bottom-up approach has the same FDR but has a higher MDR. The visualisation of the evaluation results on MDR, FDR and TER is shown in Figure 5-53.

Table 5.10 shows the comparison on the EPFL Terrace dataset. In the results by using two camera views, the MDRs of the two proposed methods are lower than that of the MvBN method, but the FDRs of the two proposed methods are higher than that of the MvBN method. Overall, the TERs of the two proposed methods are lower than that of MvBN method. In the results by using two camera views, the TERs of two proposed methods are lower than that of 3DMPP. However, when the third camera is involved, the TER of 3DMPP is lower than that of the proposed bottom-up approach but is still higher than that of the top-down approach. Overall, the proposed algorithm significantly outperforms the other algorithms in terms of TER, MDR and RECALL on the EPFL Terrace dataset. A comparison on MDR, FDR and TER is shown in Figure 5-54.

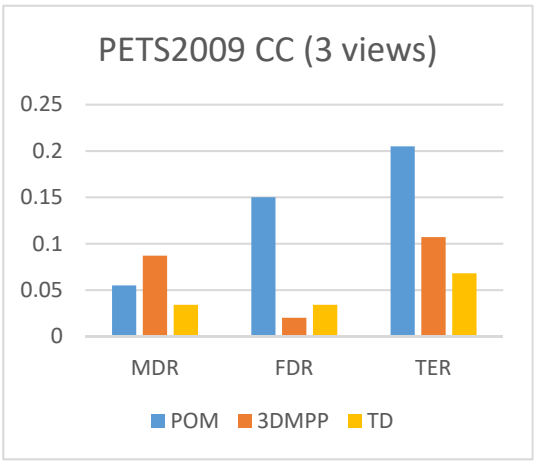
Figure 5-55 shows a performance comparison of the proposed algorithms with increasing cameras. In the top-down approach, MDR and FDR decrease significantly when the number of cameras increases. In the bottom-up approach, when the number of cameras increases, FDR decrease significantly, but the MDR decreases slowly. MDR in the cases of three cameras and four cameras are the same. That may be caused by the additional opposite camera view, which brings negative effects in the bottom-up approach.

Table 5.9 Evaluation results on the PETS2009 CC and S2L1 datasets with different camera views.

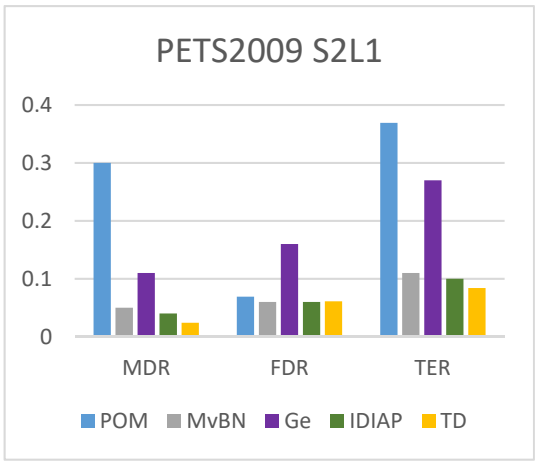
Dataset	Views	Method	MDR	FDR	TER	PRECISION	RECALL
PETS2009 CC	2	Khan	0.091	0.286	0.377	0.761	0.909
PETS2009 CC	2	POM	N/A	N/A	0.206	N/A	N/A
PETS2009 CC	2	3DMPP	N/A	N/A	0.22	N/A	N/A
PETS2009 CC	2	MvBN	0.10	0.03	0.13	0.97	0.90
PETS2009 CC	2	TD	0.026	0.025	0.051	0.975	0.974
PETS2009 CC	2	BU	0.045	0.029	0.074	0.971	0.955
PETS2009 CC	3	POM	0.055	0.150	0.205	0.863	0.945
PETS2009 CC	3	3DMPP	0.087	0.020	0.107	0.979	0.913
PETS2009 CC	3	TD	0.034	0.034	0.068	0.966	0.966
PETS2009 S2L1	4	POM	0.30	0.07	0.37	0.91	0.70
PETS2009 S2L1	4	Ge	0.11	0.16	0.27	0.85	0.89
PETS2009 S2L1	4	MvBN	0.05	0.06	0.11	0.94	0.95
PETS2009 S2L1	4	IDIAP	0.04	0.06	0.10	0.094	0.096
PETS2009 S2L1	4	TD	0.024	0.061	0.084	0.941	0.976



(a)



(b)

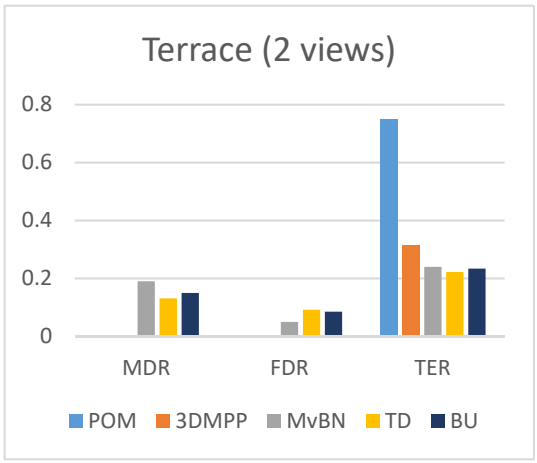


(c)

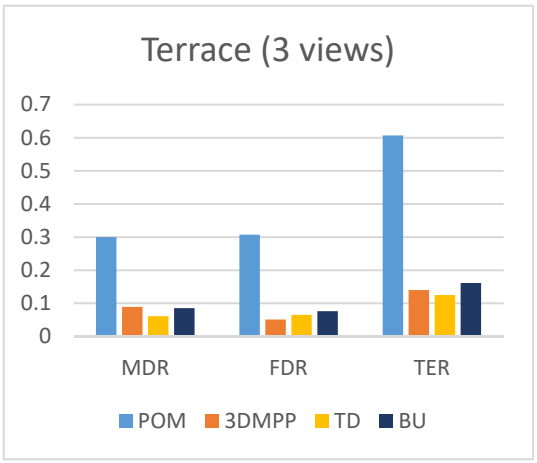
Figure 5-53 A comparison of the MDR, FDR and TER on the PETS2009 CC dataset: (a) two camera views, (b) three camera views, and (c) on the PETS2009 S2L1 with four camera views.

Table 5.10 Evaluation results on the EPFL Terrace dataset with different camera views.

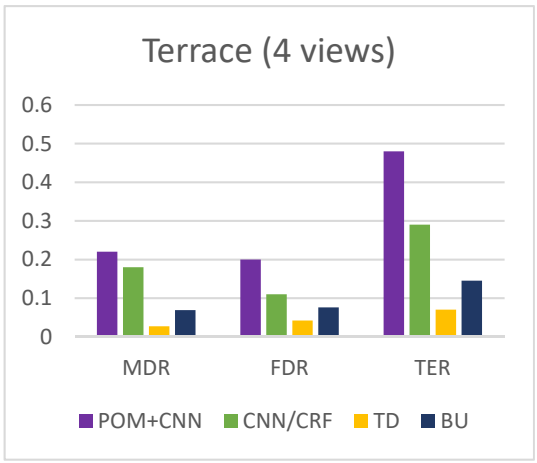
Dataset	Views	Method	MDR	FDR	TER	PRECISION	RECALL
Terrace	2	POM	N/A	N/A	0.749	N/A	N/A
Terrace	2	3DMPP	N/A	N/A	0.316	N/A	N/A
Terrace	2	MvBN	0.19	0.05	0.24	0.94	0.81
Terrace	2	TD	0.131	0.092	0.222	0.904	0.869
Terrace	2	BU	0.150	0.085	0.234	0.909	0.850
Terrace	3	POM	0.300	0.307	0.607	0.695	0.700
Terrace	3	3DMPP	0.089	0.051	0.140	0.947	0.911
Terrace	3	TD	0.061	0.065	0.125	0.935	0.939
Terrace	3	BU	0.085	0.076	0.161	0.923	0.915
Terrace	4	POM+CNN	0.22	0.20	0.42	0.80	0.78
Terrace	4	CNN/CRF	0.18	0.11	0.29	0.88	0.82
Terrace	4	TD	0.027	0.042	0.070	0.959	0.973
Terrace	4	BU	0.069	0.076	0.145	0.925	0.931



(a)

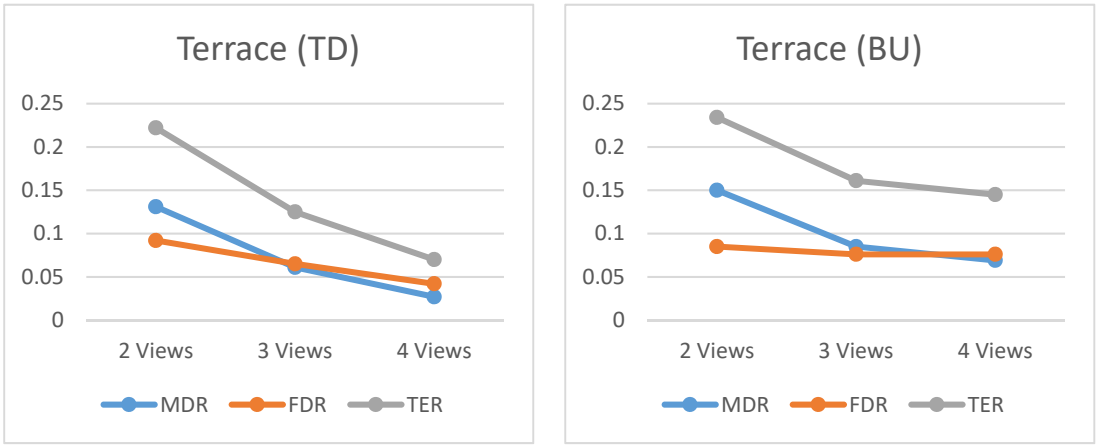


(b)



(c)

Figure 5-54 A comparison of the MDR, FDR and TER on the EPFL Terrace dataset: (a) two camera views, (b) three camera views and (c) four camera views.



(a) (b)

Figure 5-55 A comparison of the MDR, FDR and TER on the EPFL Terrace dataset with different camera views: (a) top-down approach and (b) bottom-up approach.

Chapter 6

Conclusions and Future Work

In this thesis, an algorithm for multi-camera pedestrian detection is proposed, which can be divided into two stages. In the first stage, two efficient approaches are introduced to propose the pedestrian candidates which include real pedestrians and phantoms. In the second stage, the pedestrians are further identified from the candidates by using a logic minimisation approach.

In the first stage, the pedestrian candidates are proposed by fusing the foreground silhouettes in each camera view. Two approaches are proposed for this stage. The first one is based on the POM framework and the second one is based on the Khan and Shah's method. Both the approaches use a template to model both the foreground pixels and background pixels of the pedestrians based on the foreground silhouettes. The joint occupancy likelihood of each pedestrian candidate is calculated by taking into account the template matching response and the head/foot observability.

In the second stage, the pedestrians are further identified from the candidates by using the logic minimisation approach. The Quine-McCluskey (QM) method, which is used for logic function minimisation, is borrowed to identify pedestrians and phantoms from the candidates. A prime candidate chart is developed by decomposing the foreground regions into sub-regions and used to reduce the search space for an optimised solution for pedestrian detection. Furthermore, an alternative approach, the Petrick's method, is used for finding the minimum set of pedestrian candidates to cover all the foreground sub-regions of interest.

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

- This thesis proposes two pedestrian detection algorithms using multiple cameras. A top-down approach and a bottom-up approach are used,

respectively, to fuse the foreground silhouettes from all camera views. Experimental results on benchmark video datasets have shown the proposed methods have good performances in comparison with the state-of-the-art algorithms in terms of total error rates.

- The proposed method transforms the optimisation problem into a logic minimisation problem, which makes the pedestrian detection algorithm no longer rely on an iterative method to estimate the locations of pedestrians. For example, the POM and MvBN methods need 50 and 15 iterations, respectively, to reach their conditions of convergence.
- The top-down approach is a pixel-wise approach which is insensitive to the broken foreground detection and has a lower total error rate but is time-consuming in the calculation of the occupancy likelihoods, even if the occupancy likelihood only needs to be estimated once at each frame and computational reduction is carried out. This cannot be avoided in similar approaches such as POM and MvBN.
- The bottom-up approach has a higher total error rate but is faster. Compared with the top-down approach, the bottom-up approach, which is a region-based approach, is sensitive to broken foreground regions and is not good at pedestrian detection in two opposite camera views. Because the vertical lines passing the tops of heads or the local maxima of the template matching response are parallel, when they are projected from a pair of opposite cameras in the top view, the foreground intersection points cannot be estimated accurately.

Although the proposed methods achieve promising results, some aspects need to be further investigated:

- The proposed methods are based on the average height and width of pedestrians. Short pedestrians or crouching pedestrians will be punished by low head likelihoods, which may lead to missed detections. A tall pedestrian's head will go beyond the candidate box, which may lead to a false positive in the detection when a phantom covers the sub-region of

this pedestrian's head. Leaping pedestrians are difficult to be detected because they are punished by low foot likelihoods. Therefore, to generate height adaptive candidate boxes is a part of future work.

- In the proposed algorithm, the template matching and the head and foot likelihoods are calculated separately. Thefuture work may focus on an efficient technique to estimate the occupancy likelihoods. For example, combine the estimation of the head and foot likelihoods into the template matching or convert the foreground silhouettes to a feature map and then the occupancy likelihood may be extracted more efficiently.
- Although the proposed algorithm has better performance than deep-learning based algorithms, deep learning is still a good direction for pedestrian detection. It has already obtained good results on single-camera pedestrian detection, though there are still challenges on the occlusion of pedestrians. Furthermore, the lack of large-scale multi-camera datasets which are annotated is also a limitation for applying deep learning method in multiview pedestrian detection. Therefore, future work based on the deep learning methods can focus on developing an end-to-end multiview pedestrian detector which can utilize the information from multiple camera views to cope with the occlusion. The lack of training samples may be overcome by using the simulated datasets generated by 3D engines which can emulate the positions and motion of 3D objects by computers.

Appendix

Publication List

Zezhi Chen, Yuyao Yan and Tim Ellis, “Lane detection by trajectory clustering in urban environments”, in Proceedings of the IEEE International Conference on Intelligent Transportation Systems, pp. 3076-3081, 2014.

Yuyao Yan, Ming Xu, and Jeremy S. Smith, “Multiview pedestrian localisation via a prime candidate chart based on occupancy likelihoods”, in Proceedings of the IEEE International Conference on Image Processing, pp. 2334–2338, 2017.

Yuyao Yan, Ming Xu, and Jeremy S. Smith, “Generalized vertical projection histograms using multi-plane homology”, IET Electronics Letters, Vol. 55, DOI 10.1049/el.2018.6516, 2019

Yuyao Yan, Ming Xu, Jeremy S. Smith, Mo Shen and Jin Xi, “Optimization of Multiview Pedestrian Localization Using Logic Minimization”, submitted to IEEE Transactions on Multimedia, 2018

Bibliography

[1] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 22, no. 8, pp. 745-746, 2000.

[2] M. Xu, J. Orwell, L. Lowey, and D. Thirde, "Architecture and algorithms for tracking football players with multiple cameras," *IEE Proceedings-Vision, Image Signal Processing,* vol. 152, no. 2, pp. 232-241, 2005.

[3] 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 Machine Intelligence,* vol. 28, no. 4, pp. 663-671, 2006.

[4] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*, pp. 133-146, 2006.

[5] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 30, no. 2, pp. 267-282, 2008.

[6] R. Eshel and Y. Moses, "Tracking in a dense crowd using multiple cameras," *International Journal of Computer Vision,* vol. 88, no. 1, pp. 129-143, 2010.

[7] A. Utasi and C. Benedek, "A 3-D marked point process model for multi-view people detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3385-3392, 2011.

[8] Á. Utasi and C. Benedek, "A Bayesian Approach on People Localization in Multicamera Systems," *IEEE Transactions on Circuits Systems for Video Technology,* vol. 23, no. 1, pp. 105-115, 2013.

[9] C.-W. Liu, H.-T. Chen, K.-H. Lo, C.-J. Wang, and J.-H. Chuang, "Accelerating Vanishing Point-Based Line Sampling Scheme for Real-Time People Localization," *IEEE Transactions on Circuits Systems for Video Technology,* vol.

27, no. 3, pp. 409-420, 2017.

[10] 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, no. 1-2, pp. 39-58, 2011.

[11] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang, "Robust multiple cameras pedestrian detection with multi-view Bayesian network," *Pattern Recognition,* vol. 48, no. 5, pp. 1760-1772, 2015.

[12] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, Cybernetics, Part C,* vol. 34, no. 3, pp. 334-352, 2004.

[13] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision Image Understanding,* vol. 104, no. 2-3, pp. 90-126, 2006.

[14] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision Image Understanding,* vol. 81, no. 3, pp. 231-268, 2001.

[15] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters,* vol. 34, no. 1, pp. 3-19, 2013.

[16] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proceedings-Vision, Image Signal Processing,* vol. 152, no. 2, pp. 192-204, 2005.

[17] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 22, no. 8, pp. 809-830, 2000.

[18] P. Remagnino *et al.*, "An Integrated Traffic and Pedestrian Model-Based Vision System," in *British Machine Vision Conference*, 1997.

[19] F. Bremond and M. Thonnat, "Tracking multiple nonrigid objects in video sequences," *IEEE Transactions on Circuits Systems for Video Technology,* vol. 8, no. 5, pp. 585-591, 1998.

[20] S. L. Dockstader and A. M. Tekalp, "Tracking multiple objects in the presence of

articulated and occluded motion," in *Proceedings of the Workshop on Human Motion*, pp. 88-95, 2000.

[21] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *European Conference on Computer Vision*, pp. 343-357, 2002.

[22] R. Rosales and S. Sclaroff, "Improved tracking of multiple humans with trajectory prediction and occlusion modeling," Boston University Computer Science Department, Thesis, 1998.

[23] S. S. Intille, J. W. Davis, and A. F. Bobick, "Real-time closed-world tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697-703, 1997.

[24] M. Xu, T. Ellis, S. J. Godsill, and G. A. Jones, "Visual tracking of partially observable targets with suboptimal filtering," *IET computer vision,* vol. 5, no. 1, pp. 1-13, 2011.

[25] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *International Conference on Image Processing*, pp. I-I, 2002.

[26] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision,* vol. 57, no. 2, pp. 137-154, 2004.

[27] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 90-97, 2005.

[28] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[30] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1491-1498, 2006.

[31] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," *British Machine Vision Conference,* pp. 1-11, 2009.

[32] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers,* vol. 100, no. 1, pp. 67-92, 1973.

[33] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[34] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, 2010.

[35] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," *International Journal of Computer Vision,* vol. 110, no. 1, pp. 58-69, 2014.

[36] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision,* vol. 104, no. 2, pp. 154-171, 2013.

[37] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, pp. 379-387, 2016.

[38] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3286-3293, 2014.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 38, no. 1, pp. 142-158, 2016.

[41] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015.

[42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 39, no. 6, pp. 1137-1149, 2017.

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[44] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517-6525, 2017.

[45] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, pp. 21-37, 2016.

[46] J. Black and T. Ellis, "Multi camera image tracking," *Image and Vision Computing,* vol. 24, no. 11, pp. 1256-1267, 2006.

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

[48] 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*, pp. 113-118, 2000.

[49] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 253-259, 1999.

[50] 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, no. 1, Article No. 092827, 2007.

[51] V. Kettnaker and R. Zabih, "Counting people from multiple cameras," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp. 267-271, 1999.

[52] 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*, pp. 267-272, 2003.

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

[54] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for Easyliving," in *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, pp. 3-10, 2000.

[55] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics Automation,* vol. 3, no. 4, pp. 323-344, 1987.

[56] 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*, pp. 521-527, 1999.

[57] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 25, no. 10, pp. 1355-1360, 2003.

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

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

[60] 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*, pp. 3-10, 2001.

[61] 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 *European Conference on Computer Vision*, pp. 18-36, 2002.

[62] D. Delannay, N. Danhier, and C. De Vleeschouwer, "Detection and recognition of sports (wo) men from multiple views," in *Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1-7, 2009.

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

[64] M. Morbee, L. Tessens, H. Aghajan, and W. Philips, "Dempster-Shafer based multi-view occupancy maps," *Electronics Letters,* vol. 46, no. 5, pp. 341-343, 2010.

[65] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer,* no. 6, pp. 46-57, 1989.

[66] K. Otsuka and N. Mukawa, "Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 90-97, 2004.

[67] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 31, no. 3, pp. 505-519, 2009.

[68] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

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

[70] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2913-2920, 2009.

[71] K.-H. Lo and J.-H. Chuang, "Vanishing point-based line sampling for real-time people localization," *IEEE Transactions on Circuits Systems for Video Technology,* vol. 23, no. 7, pp. 1209-1223, 2013.

[72] J. Berclaz, F. Fleuret, and P. Fua, "Principled detection-by-classification from multiple views," in *Proceedings of the International Conference on Computer*

Vision Theory and Applications, pp. 375-382, 2008.

[73] M. Golbabaee, A. Alahi, and P. Vandergheynst, "Scoop: A real-time sparsity driven people localization algorithm," *Journal of Mathematical Imaging and Vision,* vol. 48, no. 1, pp. 160-175, 2014.

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

[75] P. Baqué, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 271 - 279, 2017.

[76] T. Chavdarova and F. Fleuret, "Deep multi-camera people detection," in *Proceedings of the IEEE International Conference on Machine Learning and Applications*, pp. 848-853, 2017.

[77] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4256-4265, 2016.

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

[79] M. Liem and D. M. Gavrila, "Multi-person tracking with overlapping cameras in complex, dynamic environments," in *British Machine Vision Conference*, pp. 199-218, 2000.

[80] 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 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1-9, 2008.

[81] 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*, pp. 756-761, 2012.

[82] J. Ren, M. Xu, J. S. Smith, H. Zhao, and R. Zhang, "Multi-view visual surveillance

and phantom removal for effective pedestrian detection," *Multimedia Tools Applications,* vol. 77, pp. 18801-18826, 2018.

[83] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision,* vol. 12, no. 1, 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 IEEE International Conference on Computer Vision*, pp. 78-81, 1997.

[85] 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*, pp. 135-140, 1996.

[86] 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*, pp. 8-14, 1998.

[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, no. 7, pp. 1151-1163, 2002.

[88] 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*, pp. 2246-2252, 1999.

[89] *PETS2009 Dataset*. Available: http://www.cvg.reading.ac.uk/PETS2009/a.html

[90] R. Hartley and A. Zisserman, *"Multiple view geometry in computer vision"*. Cambridge university press, 2003.

[91] W. V. Quine, "The problem of simplifying truth functions," *The American mathematical monthly,* vol. 59, no. 8, pp. 521-531, 1952.

[92] W. V. Quine, "A way to simplify truth functions," *The American mathematical monthly,* vol. 62, no. 9, pp. 627-631, 1955.

[93] S. R. Petrick, "A direct determination of the irredundant forms of a Boolean function from the set of prime implicants," *Air Force Cambridge Res. Center Tech. Report,* pp. 56-110, 1956.

[94] M. Karnaugh, "The map method for synthesis of combinational logic circuits," *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics,* vol. 72, no. 5, pp. 593-599, 1953.

[95] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing,* vol. 24, no. 1, pp. 359-373, 2015.

[96] *Anton Milan Dataset*. Available: http://www.milanton.de/

[97] *EPFL Dataset*. Available: https://cvlab.epfl.ch/data/pom

[98] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8-14: IEEE, 2001.