Multicamera Pedestrian Detection Using Logic Minimization

Yuyao Yan[a,b], Ming Xu[a,*], Jeremy S. Smith[b], Mo Shen[c], Jin Xi[d]

[a] *Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China*
[b] *Department of Electrical Engineering and Electronics, University of Liverpool, L69 3BX, Liverpool, UK*
[c] *Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria, 3010, Australia*
[d] *Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, U.S.A*

Abstract

In this paper an algorithm for multicamera pedestrian detection is proposed. The first stage of this work is based on the probabilistic occupancy map framework, in which 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. In the second stage, where we borrowed the idea from the Quine-McCluskey method for logic function minimization, 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. Experiments on benchmark video datasets have demonstrated the improved performance of this algorithm in comparison with some benchmark non-deep or deep multicamera/monocular algorithms for pedestrian detection.

Keywords: pedestrian detection, multicamera, homography, logic minimization, video surveillance

*Corresponding Author: Ming Xu
Email address: ming.xu@xjtlu.edu.cn (Ming Xu)

1. Introduction

Pedestrian detection is an active research area in computer vision and pattern recognition. It has a variety of applications in video surveillance, autonomous driving, sport video analysis, etc. It is also the foundation of many video processing tasks such as pedestrian tracking, re-identification and behaviour recognition. Since pedestrians may be occluded by each other in a camera view, multiple cameras can be deployed to provide complementary information about the moving targets, which makes detection more robust and accurate.

When working with multiple camera views, 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 [1] [2]. For example, Hu et al. [2] 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. A good solution to this problem is that the individual cameras no longer extract features but transmit foreground bitmaps to the fusion centre. There are two benchmark works in this trend. Khan and Shah [3] 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, which is referred to as a bottom-up approach. Fleuret et al. [4] discretized the ground plane into a grid and modelled each pedestrian as a rectangle of the average size of pedestrians standing at a location. Then a Probabilistic Occupancy Map (POM) is calculated by seeking evidence from the foreground silhouettes in all camera views, which is referred to as a top-down approach.

Although both methods add robustness to pedestrian detection, many false positives may be generated. In Khan and Shah's method [3], the foreground projections of different pedestrians, each from a different camera view, may falsely intersect in the reference view, which gives rise to phantoms. In the POM method [4], the locations which are close to pedestrians may have high occupancy probabilities, even if they do not contain a pedestrian. Therefore, developing techniques to eliminate phantoms has become a challenging task.

In this paper an algorithm is proposed for multiview pedestrian detection, which is based on the POM framework [4]. In this algorithm, the ground plane is discretized into a grid and the joint occupancy likelihood at each location is calculated by taking into account a template matching response and the head/foot observability. The pedestrian candidates with low likelihoods are filtered out. At the second stage, each foreground region is decomposed into sub-regions according to the overlapping relationship of the surviving candidate boxes associated with that foreground region. Then a prime candidate chart is developed to select the essential candidates, each of which covers at least a foreground sub-region that is not covered by the other candidates. These essential candidates are identified as pedestrians. Afterwards non-essential candidates are selected to cover the remaining foreground sub-regions by following a repeated process, which alternates between merging redundant candidates and finding emerging "essential" candidates.

The contributions of this paper are threefold: (1) this paper reports the first work of using logic minimization method in multi-camera pedestrian detection. It greatly reduces the search space for an optimized solution and avoids the iterative POM computation, at each frame, as in [4], which is an advantage for real-time video surveillance applications; (2) In the calculation of the POM, not only the contribution of foreground pixels but also that of background pixels are considered, which improves the localization of pedestrian candidates in a crowd; (3) The head and foot likelihoods are involved in the POM, which can discriminate occluded pedestrians from phantoms more robustly.

The rest of this paper is organized as follows: In Section 2, the related work is reviewed. In Section 3, the estimation of the homographies for parallel planes is introduced. In Section 4, the occupancy likelihood map over the grid is described. The joint occupancy likelihood at each location is defined in Section 5. The global optimization of multiview pedestrian detection is described in Section 6. Section 7 details our experiment results. Finally conclusions are presented in Section 8.

2. Related work

Significant research has been undertaken to prune phantoms in multicamera pedestrian detection. Existing methods usually resort to temporal coherence, geometric constraints and/or colour cues. The temporal approach copes with phantoms in the tracking process. Since it is noted that phantoms appear from nowhere and are often unsteadily detected, the temporal coherence of each foreground intersection region is checked over some time [5] [6]. If a candidate cannot survive over that time period, then it is classified as a phantom. Liem and Gavrila [5] proposed that an object can only enter a scene from the border of the FOV; those initially detected in the middle of the FOV are phantoms. Similar tracking processes were carried out in [4] [3].

The geometric approach is based on the different heights and sizes of foreground intersection regions for phantoms and people. Khan and Shah [3] extended their early work by projecting the foreground likelihoods to a reference view with the homographies of a set of parallel planes and calculating across-plane foreground intersections. This approach can reduce the number of phantoms. Liu et al. [7] proposed an accelerated implementation of Khan's method by discretizing the ground plane and foreground regions. Eshel and Moses [8] used cameras looking downwards and found that when the viewing rays of two cameras intersect behind a pedestrian, the phantoms are lower than the pedestrian in 3D space. By limiting pedestrians' heights within a 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 [8] applied the pixelwise intensity correlation between aligned frames in a reference view to remove phantoms. However, this method is vulnerable to the occlusion between pedestrians.

Multiview pedestrian detection is sometimes thought of as an optimization problem. Fleuret et al. [4] 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. Then an iterative algorithm is utilized to find the optimal rectangles which cover more foreground pixels and less background pixels in both camera views. Ge et al. [9] proposed a generative sampling-based approach that models each pedestrian as an upright cylinder. Iterative Gibbs sampling is used to estimate the number and the locations of pedestrians in a crowd. Similar to [9], Utasi and Benedek [10] extended the classical Bayesian Marked Point Process (MPP) model to a 3DMPP model which utilizes the pixel-level features from pedestrians' heads and feet, instead of the whole silhouettes, to reduce the number of phantoms. Alahi et al. [11] modelled pedestrian detection as a linear inverse problem which is regularized by using a sparse binary occupancy vector. The occupancy vector is generated by the presence of pedestrians on each location of the grid . Then an iterative process was undertaken to find the optimal occupancy vector which contains the minimum number of non-zero elements and fits the multiview silhouettes. Peng et al. [12] proposed Multiview Bayesian Networks (MvBN) to prune phantoms in the frameowrk of the probabilistic occupancy map. They analyzed the occlusion relationship among rectangle models to identify phantoms using a Bayesian network. Yan et al. [13] started with a bottom-up approach to find pedestrian candidates and then used the Quine-McCluskey method based on occupancy likelihoods to identify pedestrians.

In recent years, with the emerging deep learning techniques, pedestrian detection is sometimes thought of as a recognition problem. In this approach, very accurate pedestrian models are trained over large-scale annotated datasets, and deep convolutional neural networks (CNN) have been found to be well suited to monocular pedestrian detection [14] [15] [16] and models of body parts are used in handling occlusions [17]. However, this approach is not so robust in the detection of occluded pedestrians and cannot provide the correct foot locations of partly occluded pedestrians. On the other hand, when such an approach is applied in multi-camera scenarios, there has been limited success to train a complete multiview processing model. Instead, most of existing applications depend on monocular CNN pedestrian detectors and then integrate the estimates from multiple camera views. For example, Xu et al. [18] used Faster RCNN to detect pedestrians in each view and inferred the ground-plane locations by clustering the 2D-location projections; Baque et al. [19] used a CNN to extract a body-part feature map from each view and inferred the 3D locations by minimizing the difference between a generative model and these feature maps; Zhang et al. [20] developed a

DeepPlayer model to identify players and their jersey IDs in each view, and then used the POM method to determine the location and ID of each player. This may be caused by the lack of large-scale multi-camera datasets which are annotated [21]. Therefore, the great potential of the deep leaning technique has not been fully revealed in multi-view pedestrian detection.

This paper is a significantly extended and improved work based on our previous work [13] in the following five aspects: (1) the bottom-up approach to finding pedestrian candidates is replaced by a top-down approach, which brings about robustness in coping with crowded scenarios and convenience in using more than two cameras; (2) the region-based POM calculation is replaced by a pixel-based method, which is more robust in coping with broken foregrounds; (3) the foreground ratios in the POM are replaced by template matching responses, which improves the localization of pedestrian candidates; (4) the summary of the optimization algorithm is replaced by a full algorithm; (5) an extensive performance evaluation and comparisons were carried out.

3. Homography Estimation

Planar homography is the relationship between a pair of captured images of the same plane, from two camera views. Let $\mathbf{p}$ and $\mathbf{p}'$ be the image coordinates of a point on such a plane in the two views. They are associated by the 3×3 homography matrix $\mathbf{H}$ as $\widetilde{\mathbf{p}}' \cong \mathbf{H}\widetilde{\mathbf{p}}$, where $\cong$ denotes the equivalence defined up to scale and the vectors with a tilde represent their homogeneous coordinates.

For a calibrated camera, a 3×4 projection matrix can be calculated using the intrinsic and extrinsic parameters: $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4]$. Then the homography matrix, between camera view c and the top view, for the ground plane is as follows:

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

The homography, between camera view c and the top view, for the plane parallel to the ground plane and at a height of h is as follows:

$$\mathbf{H}_h^{t,c} = [\mathbf{m}_1, \mathbf{m}_2, h\mathbf{m}_3 + \mathbf{m}_4] = \mathbf{H}_0^{t,c} + [\mathbf{0}|h\mathbf{m}_3] \ , \tag{2}$$

where $[\mathbf{0}]$ is a 3×2 zero matrix [22].

4. Occupancy Likelihood Maps

In this research background subtraction is used for the foreground extraction in each camera view. After connected component analysis, the foreground pixels are transformed into a foreground region map $F^c \in \{0,1\}^{W\times H}$ for camera view c, where $W \times H$ is the image resolution, C is the number of cameras and $c \in [1, C]$.

The goal of this approach is to detect *a priori* unknown number of pedestrians, from the *binary* foreground silhouettes in multiple camera views, at a *single* frame. The ground plane is discretized into a grid and each discrete location 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 generate foreground silhouettes in multiple camera views. The resolution of the grid is selected by a tradeoff between the accuracy and computational cost.

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 $\mathbf{H}_0^{t,c}$. Fig. 1 shows the grid of locations in two camera views and the top view. 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 pedestrian standing at the i-th location of camera view c can be obtained. Then the foreground silhouette of any pedestrian appearing at the i-th location of camera view c can be approximated by a filled rectangle of the average height H_i^c and width W_i^c at that location [4], as shown in Fig. 1. The rectangle pedestrian model is similar to the stixels [23] which are rectangular sticks standing vertically on the ground. However, the stixels are a medium-level representation between pixels and objects, while our rectangle model is at the object level.

Suppose $R_i^c \in \{0,1\}^{W\times H}$ is the synthetic binary image obtained by putting a filled rectangle r_i^c, of height H_i^c and width W_i^c, at the i-th location of an empty background image for camera view c. That is, $R_i^c(u,v) = 1$ if $u \in [u_i^c - W_i^c/2, u_i^c + W_i^c/2]$ and $v \in [v_i^c, v_i^c + H_i^c - 1]$; $R_i^c(u,v) = 0$ otherwise. The foreground pixels contained in R_i^c

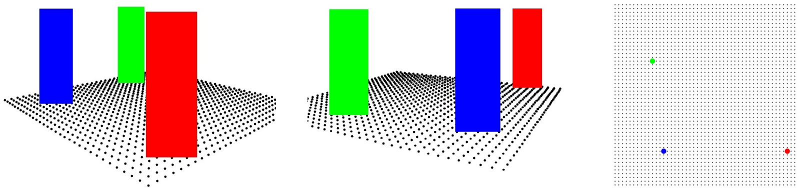


Figure 1: The discretized ground plane: a group of rectangles, which are of the average size of the pedestrians standing there, in two camera views and a top view.

are represented by $A_i^c \in \{0,1\}^{W \times H}$ defined as follows:

$$A_i^c = F^c \otimes R_i^c, \tag{3}$$

where $\otimes$ denotes pixelwise multiplication, as shown in Fig. 2(a).

Let $L_i \in \{0,1\}$ be the event that a pedestrian is located at the i-th location. Given foreground observations $A_i^1, A_i^2, \ldots, A_i^C$ at the i-th location of multiple camera views, we are interested in finding the posterior probability of event L_i occurring. The only observations used in this work are the foregrounds extracted from the multiple camera views. Colour appearance models or motion models for encoding temporal coherence are not used.

Using Bayes law,

$$P(L_i | A_i^1, A_i^2, \ldots, A_i^C) \propto P(A_i^1, A_i^2, \ldots, A_i^C | L_i) P(L_i) \ . \tag{4}$$

By conditional independence, we can rewrite the likelihood of making observations $A_i^1, A_i^2, \ldots, A_i^C$, given event L_i occurring as:

$$P(A_i^1, A_i^2, \ldots, A_i^C | L_i) = P(A_i^1 | L_i) P(A_i^2 | L_i) \ldots P(A_i^C | L_i) \ . \tag{5}$$

At a single frame, since there is no prior knowledge about the pedestrian presence, a uniform distribution is assumed for $P(L_i)$ over the G locations. Therefore, we have:

$$P(L_i | A_i^1, A_i^2, \ldots, A_i^C) \propto \prod\nolimits_{c=1}^{C} P(A_i^c | L_i) \ . \tag{6}$$

5. Joint Occupancy Likelihoods

At the i-th location of each camera view (say camera view c), three independent observations are derived, from the foreground pixels A_i^c within the rectangle 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.

By considering the conditional independence between the three measurements on the foregrounds, we have:

$$P(A_i^c|L_i) = P(t_i^c, d_i^c, h_i^c|L_i) = P(t_i^c|L_i)P(f_i^c|L_i)P(h_i^c|L_i)\ . \tag{7}$$

5.1. Template Matching Responses

The most intuitive way to select pedestrian candidates is based on the foreground ratio within each rectangle r_i^c, which seems consistent with the research objective of using the minimum number of pedestrian models (rectangles) to interpret all the observed foreground silhouettes [4] [13]. However, such a method may output a high foreground ratio in the rectangles which are lodged between side-by-side or crowded pedestrians, as it ignores the foreground pixel distribution within each rectangle and is equivalent to using a all-one template for matching.

Since the foreground pixels of a pedestrian are often distributed along the vertical central axis of the rectangle [12] and they are further surrounded by background pixels on the two sides of the rectangle, a ridge-like template is designed for pedestrian matching, in which not only the foreground pixels close to the vertical central axis but also the background pixels far from the central axis are rewarded.

The ridge-like template, for the i-th location of camera view c, is of the same size of the rectangle r_i^c. It does not vary with vertical coordinates but has a ridge-like profile in horizontal direction as shown in Fig. 2(b) and as defined as:

$$T_i^c(u') = 1 - 3|u'|/W_i^c\ , \tag{8}$$

where $u' \in [-W_i^c/2, W_i^c/2]$. The positive part of the template is used to reward the foreground pixels close to the central axis of the template, while the negative part is used to reward the background pixels at the borders. There are two zero-crossings at $u' = \pm W_i^c/3 (= \pm W_i^c/2 \times 2/3)$, which are the expected borders between the foreground and background pixels of a pedestrian.

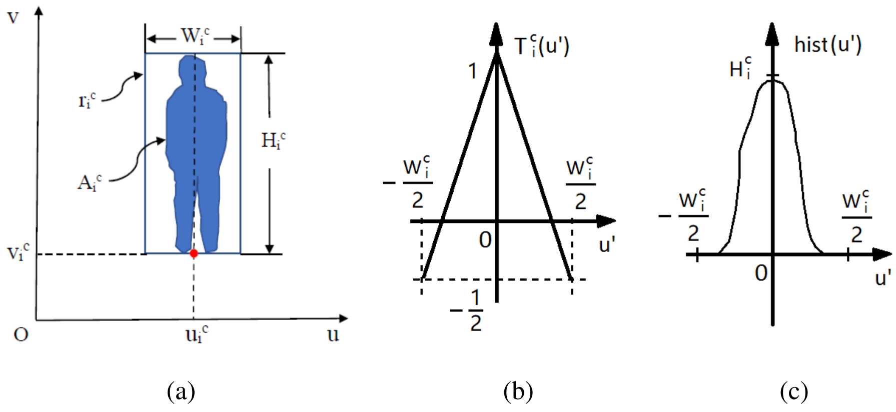


Figure 2: Template matching: (a) foreground pixels within a candidate box, (b) 1D template, and (c) vertical projection histogram of foreground pixels.

Since the 2D template varies with horizontal coordinates only, a fast implementation of the template matching is used by calculating the vertical projection histogram of the foreground pixels within rectangle r_i^c and then multiplying the histogram with the 1D template $T_i^c(u')$. The vertical projection histogram is defined as:

$$hist(u') = \sum\nolimits_{v=0}^{H_i^c-1} A_i^c(u_i^c + u', v) \ , \tag{9}$$

where $u' \in [-W_i^c/2, W_i^c/2]$, as shown in Fig. 2(c). Since $A_i^c(u,v) \in \{0,1\}$, one has $hist(u') \in [0, H_i^c]$. To avoid repeated calculation and to further accelerate this process, the histogram is calculated from a precomputed integral image [24] based on foreground region map F^c.

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

$$t_i^c = \sum\nolimits_{u'=-W_i^c/2}^{W_i^c/2} [hist(u') - H_i^c/2] \times T_i^c(u') \ , \tag{10}$$

where the subtraction within the square brackets shifts the $hist(u')$ values from $[0, H_i^c]$ to $[-H_i^c/2, H_i^c/2]$ so that the locations dominated by background pixels have negative values. The template matching has the maximum response when $A_i^c(u,v) = 1$ for $|u - u_i^c| \in [0, W_i^c/3]$, $v - v_i^c \in [0, H_i^c - 1]$ and $A_i^c(u,v) = 0$ otherwise (see Fig. 2(a)). 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:

$$t^c_{i,max} = \sum\nolimits_{u'=-W^c_i/2}^{W^c_i/2} (H^c_i/2) \times |T^c_i(u')| \ . \tag{11}$$

The template matching has the minimum response when $A^c_i(u,v) = 1$ for $|u - u^c_i| \in [W^c_i/3, W^c_i/2]$, $v - v^c_i \in [0, H^c_i - 1]$ and $A^c_i(u,v) = 0$ otherwise (Fig. 2(a)). 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:

$$t^c_{i,min} = -t^c_{i,max} \ . \tag{12}$$

Therefore, the likelihood for the template matching response is defined as the normalized response:

$$P(t^c_i|L_i) = \frac{t^c_i - t^c_{i,min}}{t^c_{i,max} - t^c_{i,min}} = \frac{t^c_i + t^c_{i,max}}{2t^c_{i,max}} \ . \tag{13}$$

Since $t^c_i \in [-t^c_{i,max}, t^c_{i,max}]$, we have $P(t^c_i|L_i) \in [0,1]$.

Fig. 3 shows a comparison between the foreground ratios and the template matching responses, which are obtained by shifting a human sized rectangle from left to right, on the given foreground silhouettes. The local maxima of the foreground ratios are located between two pedestrians (Fig. 3(c)(d)), while those of the template matching responses are aligned with the heads of these pedestrians. If such a rectangle only contains background pixels, the template matching response is not at its minimum value. The minimum value occurs at the two sides of each group of pedestrians, where background pixels are in the centre of the rectangle and foreground pixels are at the borders.

5.2. Foot and Head Positions

If a pedestrian is located at the i-th location, his/her foreground silhouette should be enclosed well in the rectangle r^c_i in camera view c: the feet are expected to be at the bottom of the rectangle and the top of head is expected to be at the top of the rectangle. However, 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,

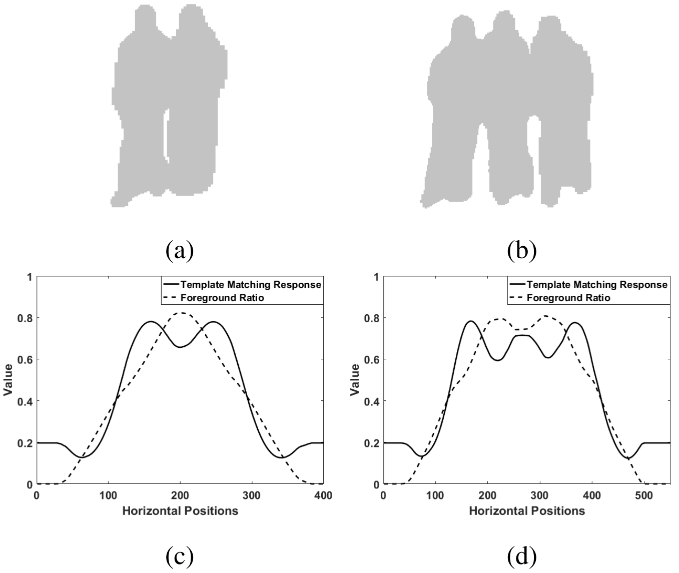

(a) (b) (c) (d)

Figure 3: A comparison of foreground ratios and template matching responses: (a)(b) foreground silhouettes and (c)(d) the comparison.

within the rectangle r_i^c, are Gaussian distributed:

$$\begin{aligned} f_i^c &\sim N(v_i^c, (\beta_f H_i^c)^2) \qquad (14) \\ h_i^c &\sim N(v_i^c + H_i^c, (\beta_h H_i^c)^2) \ , \qquad (15) \end{aligned}$$

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 $\beta_f, \beta_h \in (0, 1)$. The head and feet correspond to the highest and lowest foregrounds in rectangle r_i^c, respectively, as shown in Fig. 4(a)(b). Their vertical positions are extracted by using horizontal projection histograms in the rectangle and a width filter.

Such extracted feet are actually the ones 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 hidden behind the others (see Fig. 4(c)). Therefore, suppose the tail probability on the Gaussian distribution is denoted by $Q_G(x) = \int_x^\infty p_G(t)dt$, where $p_G(t)$ is the probability density function for

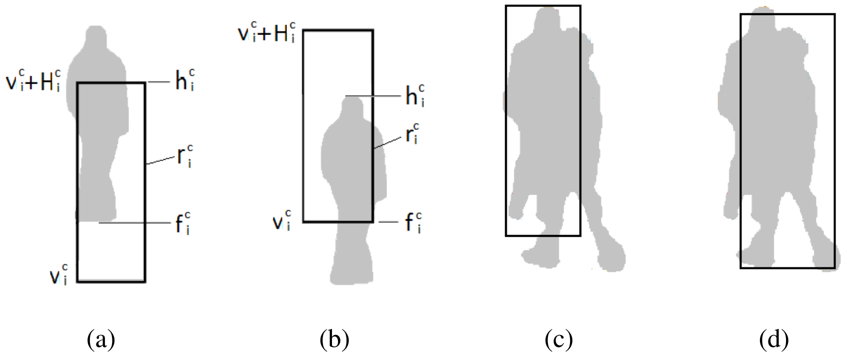


Figure 4: (a)(b) Schematic diagrams of the foot measurement f_i^c and head measurement h_i^c within a candidate box, (c) f_i^c from someone in front of candidate i, and (d) h_i^c from someone behind candidate i.

$N(0,1)$, the likelihood for such a foot observation can be expressed as:

$$P(f_i^c|L_i) = Q_G\left(\frac{f_i^c - v_i^c}{\beta_f H_i^c}\right) . \tag{16}$$

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 at the i-th location but is in front of the others (see Fig. 4(d)). The likelihood for the head observation is expressed as:

$$P(h_i^c|L_i) = Q_G\left(\frac{v_i^c + H_i^c - h_i^c}{\beta_h H_i^c}\right) . \tag{17}$$

In this implementation, the foot and head likelihoods are doubled so as to change their ranges from $[0, 0.5]$ to $[0, 1]$; the value of function Q_G is interpolated from a look-up table (LUT).

5.3. *Information Fusion across Multiple Views*

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. If $V_i^c = 0$, then $P(A_i^c|L_i)$ is set to 1. The number of the cameras, which cover the i-th location, is $C_i^V = \sum_{c=1}^{C} V_i^c$. The joint occupancy likelihood over C camera views is calculated as follows:

$$P(L_i|A_i^1, A_i^2, \ldots, A_i^C) \propto \left(\prod\nolimits_{c=1}^{C} P(A_i^c|L_i)\right)^{1/C_i^V} . \tag{18}$$

A Repulsive Spatial Sparsity (RSS) constraint [11] is applied to suppress those locations which are not the global maximum in a local area of radius ϵ.

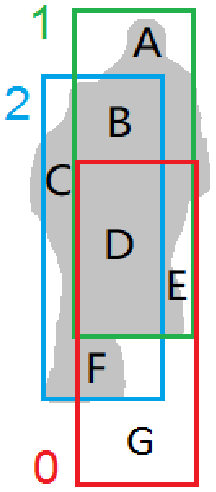
6. A Logic Minimization Approach

6.1. Prime Candidate Charts

The joint occupancy likelihood is derived separately for each pedestrian candidate. To encode the interactivity such as occlusion and grouping between pedestrians, global optimization is carried out for the multiview pedestrian detection [13]. We borrowed the idea from the Quine-McCluskey (QM) method [25] which has been used for the minimization of Boolean functions. The tabular form of this method makes it readily implemented by a computer programme.

To facilitate the use of the Quine-McCluskey (QM) method, each foreground region is decomposed into sub-regions according to the overlapping relationship of all the candidate boxes associated with that foreground region [13]. 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 (see an example in Fig. 5(a)).

Suppose there are N pedestrian candidates surviving in the occupancy likelihood filtering. The filled rectangles of these candidates are summed up in an image $B^c \in [0, 2^N - 1]^{W \times H}$, with weights in powers of 2, in each camera view (say camera c): $B^c = \sum_{i=0}^{N-1}(2^i \times R^c_{(i)})$, where the subscript (i) in parentheses is used to differentiate it from the original index of the G locations and represents the index of the N candidate boxes. 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-i indicates this sub-region is covered by candidate box i, where $i \in [0, N-1]$; otherwise, bit-i is zero (see Fig. 5(b)). 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



(a)

SUB-REGION	A	B	C	D	E	F	G
CANDIDATES	1	1,2	2	0,1,2	0,1	0,2	0
BINARY CODE	010	110	100	111	011	101	001
DECIMAL CODE	2	6	4	7	3	5	1

(b)

CANDIDATE		A	B	C	D	E	F
0		+	+	+	X	X	X
1	o	X	X	+	X	X	+
2	o	+	X	X	X	+	X

(c)

CANDIDATE		A	B	C	D	E	F
0		+	+	+	+	+	+
1	o	+	+	+	+	+	+
2	o	+	+	+	+	+	+

(d)

Figure 5: (a) Decomposition of a foreground region into sub-regions, (b) the information of the sub-regions, (c) the prime candidate chart with two essential candidates identified, and (d) the chart after the essential candidates are removed. An X indicates that the sub-region is covered by a candidate; otherwise, a plus sign is placed. The identified pedestrians are labelled with circles.

in a sub-region, respectively. Then the sub-regions, which are too small or have very low foreground ratios, e.g. sub-region G, are filtered out, .

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. In the prime candidate chart (see Fig. 5(c)), 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 pedestrian candidate box covers a given sub-region, an X is placed at the intersection of the corresponding row and column; otherwise, a plus sign is placed at the intersection. It is noted that each column exactly matches the binary code of the corresponding sub-region, if that column is read from bottom to top, X's are replaced by 1 and plus signs are replaced by 0.

6.2. Updating of A Prime Candidate Chart

The prime candidate chart is updated by using the QM Algorithm, where its functions are defined in Algorithm 1 and its main body is described in Algorithm 2. The inputs to 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 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 correspond to the sub-regions covered by this candidate are removed. 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, which may happen after the X's removal, the one with a lower joint occupancy likelihood is removed.

The updating procedure of a prime candidate chart is divided into three steps, as shown in Algorithm 2. Although this algorithm seems somewhat lengthy, it usually terminates after step 1. The remaining steps are designed to cope with the more complicated scenarios which rarely occur. Step 1 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 2 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

Algorithm 1: Function definition of the QM alogorithm

```
Function FindEssential(Q,STATUS)
% Q: a prime candidate chart; STATUS: assigned role
for each column (sub-region) in Q do
    if it contains only one X then
        The candidate is labelled as STATUS;
        The X's in this row are removed;
        The X's in the columns covered by this candidate are removed;
return Q
Function Merge(Q,P)
% Q: a prime candidate chart;
% P: a list of joint occupancy likelihoods
Flag=FALSE
for each row (candidate) in Q do
    if its X's are the same as another row then
        The X's for the candidate with a lower P value are removed;
        Flag=TRUE;
    else if its X's are contained by another row then
        The X's in this row are removed; Flag=TRUE;
return [Q,Flag]
```

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 3 is used to find alternative solutions on a trial basis.

In step 3, a column with the least number (non-zero) 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 1 and 2. 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'.

An example of the use of the QM algorithm is shown in Fig. 5(c). There is a single X in columns A and C. Candidate 1, the candidate for cell 1A (row 1 and column A), is labelled with a circle for 'PEDESTRIAN' in the STATUS column. Row 1 is then removed by replacing all the X's with plus signs. The columns covered by candidate 1, that is columns A, B, D and E, are also removed. Similarly, candidate 2 is labelled as 'PEDESTRIAN' due to cell 2C. Row 2, columns B, C, D and F are then removed. These lead to the chart as shown in Fig. 5(d), in which no X's are left and the algorithm terminates.

The second example is more challenging and goes through all three steps of the QM Algorithm, as shown in Fig. 6. The original prime candidate chart is shown in Fig. 6(a), in which candidate 0 is identified as an essential candidate and labelled with 'PEDESTRIAN' due to cell 0A. Fig. 6(b) is the chart after the removal of row 0 and columns A and B. There are no redundant candidates at this stage. The remaining X's are in cyclic form and each of the remaining columns contains two X's. The first remaining 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 Fig. 6(b), candidate 1 is considered as a trial row and labelled with an asterisk for 'TRIAL'. Then row 1 and

Algorithm 2: The update of a prime candidate chart

```
Input  : A prime candidate chart Q, a list of joint occupancy likelihoods P
Output: The prime candidate chart Q with assigned status for each candidate
Q=FindEssential(Q,‘PEDESTRIAN’);          // 1 essentializing
if no X’s are left in Q then
    return Q
repeat
    [Q,Flag]=Merge(Q,P);                               // 2 merging
    Q=FindEssential(Q,‘PEDESTRIAN’);
until Flag==FALSE
while there are X’s in Q do
    for all the columns that still contain X’s do
        Find a column with the minimum number of X’s;     // 3 grouping
    for each of the X’s in the selected column do
        Q’=Q;
        The other X’s in the same column in Q’ is removed;
        Q’=FindEssential(Q’,‘TRIAL’);
        repeat
            [Q’,Flag]=Merge(Q’,P);
            Q’=FindEssential(Q’,‘TRIAL’);
        until Flag==FALSE
        Backup Q’;
        Multiply the P values for all ‘TRIAL’ candidates;
    for all the X’s in the selected column do
        Select the X with the maximum product of P values;
        Q=Q’; Replace ‘TRIAL’ with ‘PEDESTRIAN’;
return Q;
```

columns C and F, covered by candidate 1, are removed, which leads to Fig. 6(c). Since candidates 2 and 4 are contained by candidate 3, the MERGE function gives rise to Fig. 6(d). As there is a single X's in columns D and E, candidate 3 becomes an essential candidate labelled with 'TRIAL'. In the second cloned chart as shown in Fig. 6(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 Fig. 6(f). Since candidates 1 and 3 are contained by candidate 2, the MERGE function gives rise to Fig. 6(g). As there is a single X's 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, then candidates 1 and 3 are accepted and the first cloned chart is used to update the prime candidate chart, as shown in Fig. 6(h).

7. Experimental Results

7.1. Experiment Setup

To evaluate the proposed algorithm, a number of experiments were performed on three benchmark datasets, which are the PETS2009 City Centre (CC) dataset [26], the S2L1 dataset [26] and the EPFL Terrace dataset [27].

The PETS2009 City Center (CC) dataset was captured in an outdoor environment and contains 8 camera views. Each camera view has 795 frames. An area-of-interest (AOI) of size 12.2 m×14.9 m was used in the experiments. The challenge of this dataset is the static occlusions in C1 and the inaccurate calibration of C5-C8. The PETS2009 S2L1 dataset comes from the same cameras as those in the CC dataset except C2.

The EPFL Terrace dataset is a challenging benchmark dataset which contains 4 eye-level camera views in a small space on a terrace. The video sequence has 5000 frames. An AOI was defined as a 5.3 m×5.0 m rectangle. The challenge of this dataset is the heavy occlusion between pedestrians and the poor foreground detection due to the automatic white balance of the cameras.

CANDIDATE		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		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	X	+	+	X
2		+	+	+	+	X	X
3		+	+	+	X	X	+
4		+	+	X	X	+	+

(b)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	+	+	+	+
2		+	+	+	+	X	+
3		+	+	+	X	X	+
4		+	+	+	X	+	+

(c)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	*	+	+	+	+	+	+
2		+	+	+	+	+	+
3	*	+	+	+	X	X	+
4		+	+	+	+	+	+

(d)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	X	+	+	X
2		+	+	+	+	X	X
3		+	+	+	X	X	+
4	*	+	+	X	X	+	+

(e)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	+	+	+	X
2		+	+	+	+	X	X
3		+	+	+	+	X	+
4	*	+	+	+	+	+	+

(f)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1		+	+	+	+	+	+
2	*	+	+	+	+	X	X
3		+	+	+	+	+	+
4	*	+	+	+	+	+	+

(g)

CANDIDATE		A	B	C	D	E	F
0	o	+	+	+	+	+	+
1	o	+	+	+	+	+	+
2		+	+	+	+	+	+
3	o	+	+	+	+	+	+
4		+	+	+	+	+	+

(h)

Figure 6: The updating of a prime candidate chart: (a) the original chart with essential candidate 0 being identified, (b) cloned chart 1 with candidate 1 being selected as a trial row, (c) cloned chart 1 with candidates 2 and 4 being contained by candidate 3, (d) cloned chart 1 with candidate 3 becoming an essential candidate, (e) cloned chart 2 with candidate 4 being selected as a trial row, (f) cloned chart 2 with candidates 1 and 3 being contained by candidate 2, (g) cloned chart 2 with candidate 2 becoming an essential candidate, and (h) cloned chart 1 is used to update the prime candidate chart due to its higher joint occupancy likelihood.

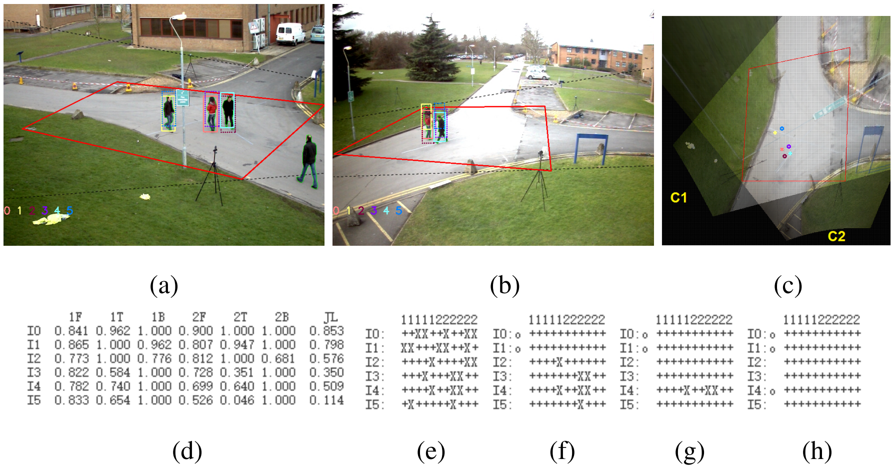
In the experiments on the PETS2009 CC and S2L1 datasets, the Gaussian mixture model was used for the foreground extraction in each camera view. The first 395 frames were used to generate the background model, and the remaining 400 frames were used to evaluate the performance of the proposed algorithm. The same range of frames were used for the test in [10] [12] due to the content discontinuity at frame 395. The parameters were set as follows: $h_a = 170$ cm, $h_w = 120$ cm, $\alpha = 0.40$, $\beta_f = 0.25$, $\beta_h = 0.25$ and $\epsilon = 40$ cm.

In the experiments on the EPFL Terrace dataset, due to the automatic white balance of the cameras, pedestrians close to a camera can significantly change the lightness of the camera view. To cope with this problem, SuBSENSE [28] was used to extract foregrounds, which is a pixelwise segmentation method based on spatiotemporal binary features as well as colours. The parameters were set as follows: $h_a = 200$ cm, $h_w = 120$ cm, $\alpha = 0.35$, $\beta_f = 0.10$, $\beta_h = 0.40$ and $\epsilon = 24$ cm. h_a was set at a larger value, since the pedestrians in this dataset are obviously higher than those in the CC dataset.

7.2. Qualitative Performance Evaluation

Fig. 7 shows the detection results at frame 465 of the PETS2009 CC dataset, where Figs. 7(a) and 7(b) are the two camera views and Fig. 7(c) is a synthetic top view. This frame was selected as 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 candidate boxes with their bottoms outside the overlapping FOVs were excluded in the detection. The area of interest (AOI) is represented by a red quadrangle. Only the candidates with their bottoms within the AOI were involved in a quantitative performance evaluation. The camera positions labelled in the top view are approximated ones, since they may go beyond the top view image. The contour of each foreground region is shown in green. Each candidate box, along with its ID number, is represented in a distinguished colour. The colour code is defined at the bottom of the two camera views. In the two camera views, an identified pedestrian is enclosed by a solid box, while a phantom is enclosed by a box of dashed lines. In the top view, an identified pedestrian is labeled with a disk, while a phantom is labeled with a circle.

Fig. 7(d) shows the joint occupancy likelihoods for the pedestrian candidates at



(a) (b) (c)

```
      1F    1T    1B    2F    2T    2B     JL
I0  0.841 0.962 1.000 0.900 1.000 1.000  0.853
I1  0.865 1.000 0.962 0.807 0.947 1.000  0.798
I2  0.773 1.000 0.776 0.812 1.000 0.681  0.576
I3  0.822 0.584 1.000 0.728 0.351 1.000  0.350
I4  0.782 0.740 1.000 0.699 0.640 1.000  0.509
I5  0.833 0.654 1.000 0.526 0.046 1.000  0.114
```

(d)

```
     11111222222        11111222222        11111222222        11111222222
I0:  ++XX++X++XX   I0:o +++++++++++   I0:o +++++++++++   I0:o +++++++++++
I1:  XX+++XX++X+   I1:o +++++++++++   I1:o +++++++++++   I1:o +++++++++++
I2:  ++++X++++XX   I2:  ++++X++++++   I2:  +++++++++++   I2:  +++++++++++
I3:  +++X+++XX++   I3:  +++++++XX++   I3:  +++++++++++   I3:  +++++++++++
I4:  ++++X++XX++   I4:  ++++X++XX++   I4:  ++++X++XX++   I4:o +++++++++++
I5:  +X+++++X+++   I5:  +++++++X+++   I5:  +++++++++++   I5:  +++++++++++
```

(e) (f) (g) (h)

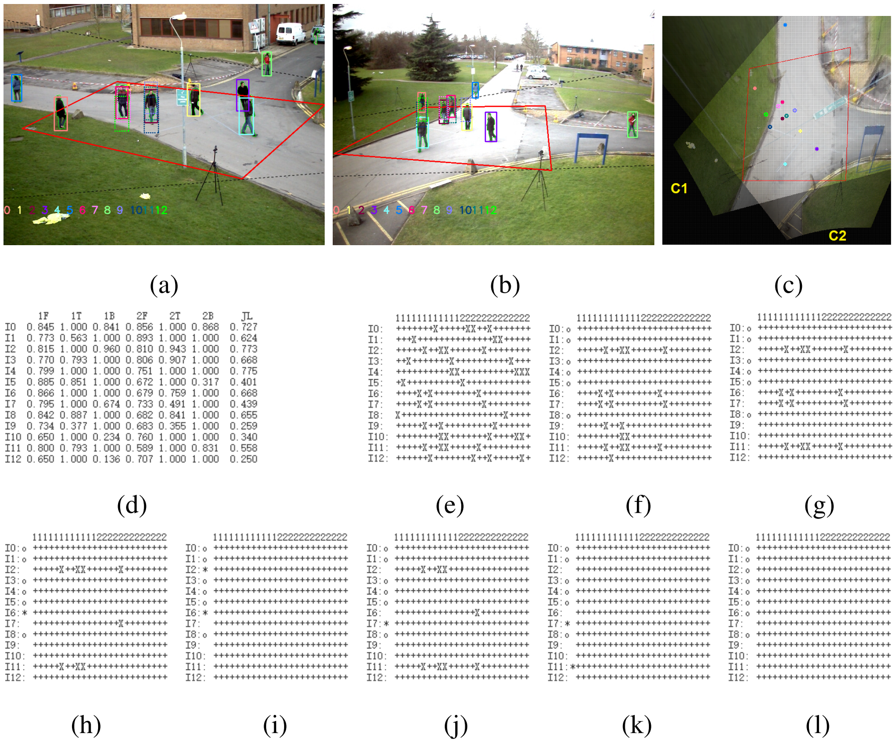
Figure 7: The detection results at frame 465 of the PETS2009 CC dataset: (a)(b) views C1 and C2, in which a pedestrian is in a solid box and a phantom is in a dashed-line box, (c) a synthetic top view, in which a pedestrian is labeled with a disk and a phantom is labeled with a circle, (d) the joint occupancy likelihoods, (e) the original prime candidate chart, (f) the chart after step 1, (g) the chart after step 2, and (h) the final results.

frame 465. Down the left-hand side are the pedestrian candidates. 1F and 2F are the likelihoods $P(t_i|L_i)$ in C1 and C2, 1T and 2T are the likelihoods $P(h_i|L_i)$, 1B and 2B are the likelihood $P(f_i|L_i)$, and 'JL' is the joint occupancy likelihood. In Fig. 7(a), pedestrian candidate 2, which is represented in brown, has a bottom lower than the bottom of its associated foreground region in both camera views and is therefore penalised by the low 1B (0.776) and 2B (0.681) values; Candidate 5, which is represented in blue, has a predicted top much higher than the top of its associated foregrounds in both camera views and is therefore penalised by very low 1T (0.654) and 2T (0.046) values - if this happens, it must correspond to a very short person; Candidate 1, represented in yellow, fits well to its associated foregrounds in both camera views and therefore has a higher joint occupancy likelihood.

Figs. 7(e)-(h) show the prime candidate charts at frame 465. Down the left-hand side of the charts is the list of pedestrian candidates (I0-I5). If a candidate is identified as a pedestrian, then it is labeled with a circle. At the top of each chart are the camera indices. Each column corresponds to a sub-region. The corresponding area for each sub-region can be found, in Figs. 7(a) and 7(b), by overlapping the candidate boxes which have an X at the intersection with that sub-region in Fig. 7(e). Fig. 7(e) is the original chart. Fig. 7(f) is the chart after step 1 when essential candidates are identified and removed. Fig. 7(g) is the chart after step 2 by merging redundant candidates 2, 3 and 5 into candidate 4, which leaves candidate 4 as an emerging essential candidate. In Fig. 7(h), candidates 0, 1 and 4 are correctly identified as pedestrians without resorting to step 3. Although I4 has a lower occupancy likelihood than I2 (phantom), it is still identified as a pedestrian; otherwise, the bottom of the foreground region associated with I4 in C2 cannot be interpreted.

Fig. 8 shows the detection results at frame 739 of the PETS2009 CC dataset. This frame was selected because the proposed algorithm had to go through all three steps in the prime candidate chart. Fig. 8(e) is the original prime candidate chart. Fig. 8(f) is the chart after step 1 by removing essential candidates 0, 1, 3, 4, 5 and 8. Fig. 8(g) is the chart after step 2 by removing redundant candidates 9, 10 and 12. The remaining X's in the chart are in cyclic form. Therefore, the first remaining column is selected for trial. Two cloned charts are made. In the first cloned chart, shown in Fig. 8(h), candidate 6 is selected as a trial row. The relevant row I6 and three columns are removed. The two contained candidates 7 and 11 are merged into candidate 2. This leaves candidate 2 as an emerging essential candidate, as shown in Fig. 8(i). In the second cloned chart as shown in Fig. 8(j), candidate 7 is selected as a trial row. The two contained candidates 2 and 6 are merged into candidate 11. This leaves candidate 11 as an emerging essential candidate. as shown in Fig. 8(k). As the product of the joint occupancy likelihoods for candidates 2 and 6 is higher than that for candidates 7 and 11, the first cloned chart is accepted to update the prime candidate chart, as shown in Fig. 8(l).

Fig. 9(a) shows the detection results at frame 719 of the PETS2009 CC dataset with three camera views. This frame was selected because an eye-level camera view (C5) was added. In camera view C5, due to the poor calibration, the pedestrian associated with candidate 1 has a top of head well above the candidate box. However, there is no additional candidate box to enclose the foregrounds for his head, because such a candidate, if it exists, corresponds to someone standing behind him but there is no foreground evidence for such a pedestrian in the other camera views. Fig. 9(b) is the detection results at frame 706 on the PETS2009 S2L1 dataset with four camera views.



(a) (b) (c)

	1F	1T	1B	2F	2T	2B	JL
I0	0.845	1.000	0.841	0.856	1.000	0.868	0.727
I1	0.773	0.563	1.000	0.893	1.000	1.000	0.624
I2	0.815	1.000	0.960	0.810	0.943	1.000	0.773
I3	0.770	0.793	1.000	0.806	0.907	1.000	0.668
I4	0.799	1.000	1.000	0.751	1.000	1.000	0.775
I5	0.885	0.851	1.000	0.672	1.000	0.317	0.401
I6	0.866	1.000	1.000	0.679	0.759	1.000	0.668
I7	0.795	1.000	0.674	0.733	0.491	1.000	0.439
I8	0.842	0.887	1.000	0.682	0.841	1.000	0.655
I9	0.734	0.377	1.000	0.683	0.355	1.000	0.259
I10	0.650	1.000	0.234	0.760	1.000	1.000	0.340
I11	0.800	0.793	1.000	0.589	1.000	0.831	0.558
I12	0.650	1.000	0.136	0.707	1.000	1.000	0.250

(d)

```
      1111111111112222222222222
I0:   +++++++X+++++XX++X+++++++
I1:   +++X++++++++++++++XX+++++
I2:   +++++X++XX++++++X++++++++
I3:   ++X+++++++X++++++++++X+++
I4:   ++++++++++XX++++++++++XXX
I5:   +X++++++++++X++++++++++++
I6:   ++++X+X++++++++X+++++++++
I7:   ++++X+X+++++++++X++++++++
I8:   X+++++++++++++++++++X++++
I9:   +++++X++X+++++++++X++++++
I10:  ++++++++XX+++++++X++++XX+
I11:  +++++X++XX+++++X+++++++++
I12:  ++++++X+++++++X++X+++++X+
```

(e)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:   +++++X++XX++++++X++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:   ++++X+X++++++++X+++++++++
I7:   ++++X+X+++++++++X++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++X++X++++++++++++++++
I10:  ++++++++XX+++++++++++++++
I11:  +++++X++XX+++++X+++++++++
I12:  ++++++X++++++++++++++++++
```

(f)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:   +++++X++XX++++++X++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:   ++++X+X++++++++X+++++++++
I7:   ++++X+X+++++++++X++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:  +++++X++XX+++++X+++++++++
I12:  +++++++++++++++++++++++++
```

(g)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:   +++++X++XX++++++X++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:*  +++++++++++++++++++++++++
I7:   ++++++++++++++++X++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:  +++++X++XX+++++++++++++++
I12:  +++++++++++++++++++++++++
```

(h)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:*  +++++++++++++++++++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:*  +++++++++++++++++++++++++
I7:   +++++++++++++++++++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:  +++++++++++++++++++++++++
I12:  +++++++++++++++++++++++++
```

(i)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:   +++++X++XX+++++++++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:   +++++++++++++++X+++++++++
I7:*  +++++++++++++++++++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:  +++++X++XX+++++X+++++++++
I12:  +++++++++++++++++++++++++
```

(j)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:   +++++++++++++++++++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:   +++++++++++++++++++++++++
I7:*  +++++++++++++++++++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:*+++++++++++++++++++++++++
I12:  +++++++++++++++++++++++++
```

(k)

```
      1111111111112222222222222
I0:o  +++++++++++++++++++++++++
I1:o  +++++++++++++++++++++++++
I2:o  +++++++++++++++++++++++++
I3:o  +++++++++++++++++++++++++
I4:o  +++++++++++++++++++++++++
I5:o  +++++++++++++++++++++++++
I6:o  +++++++++++++++++++++++++
I7:   +++++++++++++++++++++++++
I8:o  +++++++++++++++++++++++++
I9:   +++++++++++++++++++++++++
I10:  +++++++++++++++++++++++++
I11:  +++++++++++++++++++++++++
I12:  +++++++++++++++++++++++++
```

(l)

Figure 8: The detection results at frame 739 of the PETS2009 CC dataset: (a)(b) views C1 and C2, (c) the synthetic top view, (d) the joint occupancy likelihoods, (e) the original prime candidate chart, (f) after step 1, (g) after step 2, (h) cloned chart 1 when I6 is a trial row, (i) cloned chart 1 when I2 becomes essential, (j) cloned chart 2 when I7 is a trial row, (k) cloned chart 2 when I11 becomes essential, and (l) the final result.

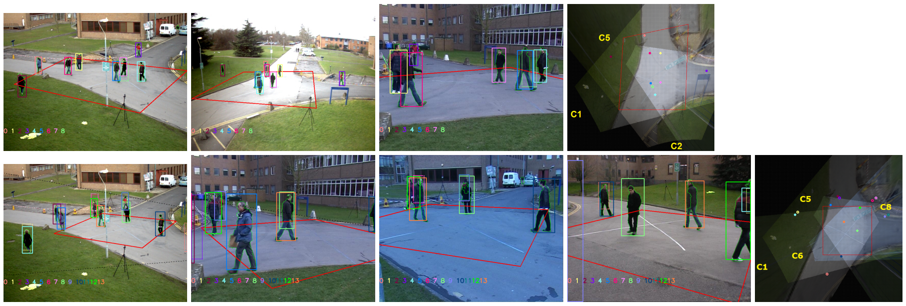


Figure 9: (a) (top) The results at frame 719 of the PETS2009 CC dataset: views C1, C2, C5 and a top view. (b) (bottom) The results at frame 706 on the PETS2009 S2L1 dataset: views C1, C5, C6, C8 and a top view.

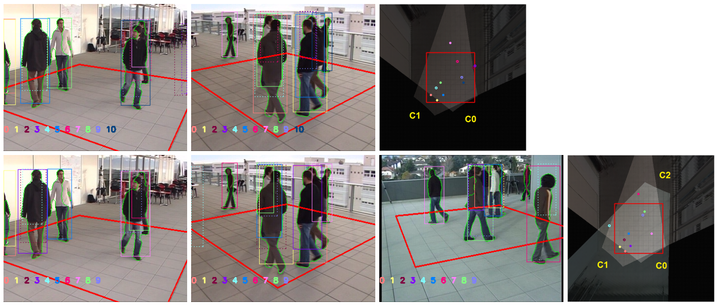


Figure 10: (a) (top) The results at frame 2350 of the EPFL Terrace dataset: views C0, C1 and a top view. (b) (bottom) The results at the same frame: views C0, C1, C2 and a top view.

Fig. 10(a) shows the detection results at frame 2350 of the EPFL Terrace dataset with two camera views. This frame was selected because there is a missed detection due to insufficient observations. Candidate 6, represented in magenta, is a real pedestrian but was recognised as a phantom. This pedestrian is hidden behind others and is in the same line of sight with another two pedestrians in both camera views. It is rather difficult to identity him by human observation. Candidate 6 has a high occupancy likelihood but is not identified as a pedestrian, because it does not uniquely cover any part of foregrounds. When more cameras were used, the problem of insufficient observations can be solved. Fig. 10(b) shows the detection results at the same frame with three camera views, in which this pedestrian (candidate 8 in light green) is correctly detected. He is completely observed by camera C2 and uniquely covers a part of the foreground.

Fig. 11(a) shows the detection results at frame 1475 of the EPFL Terrace dataset with 4 camera views. 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. Candidate 9, which is a pedestrian, is merged with other pedestrians in all the camera views but is correctly detected. Fig. 11(b) shows the detection results at frame 1925 with 4 camera views. The foreground detection is very poor in three of the four camera views, due to the colour similarity between the foregrounds and backgrounds. However, all the pedestrians are still detected, given the redundancy of the foreground observations from multiple views.

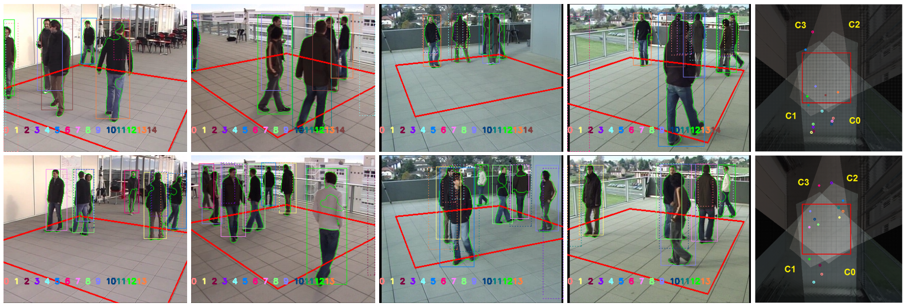


Figure 11: (a) (top) The results at frame 1475 of the EPFL Terrace dataset: views C0, C1, C2, C3 and a top view. (b) (bottom) The results at frame 1925 of the same dataset: views C0, C1, C2, C3 and a top view.

7.3. Quantitative Performance Evaluation

For a performance evaluation and a comparison with benchmark multicamera detection algorithms, five metrics were used: MDR (missed detection rate), FDR (false detection rate), TER (total error rate), PRECISION and RECALL. Suppose GT, TP, FP and FN are the numbers of ground-truth pedestrians, true positives, false positives and false negatives, respectively, where $GT = TP+FN$. We have $MDR = FN/GT$, $FDR = FP/GT$, $TER = MDR + FDR$, $PRECISION = TP/(TP + FP)$, and $RECALL = TP/GT$. A lower value in MDR, FDR and TER, or a larger value in PRECISION and RECALL, indicates better performance. MDR, PRECISION and RECALL values are less than or equal to 1, but FDR and TER may exceed 1 in case of many false positives. The evaluation and comparison were based on the PETS2009 CC dataset, PETS2009 S2L1 dataset and EPFL Terrace dataset. These video datasets, as well as the five performance metrics, were selected because they are widely used in the evaluation of existing algorithms for multiview pedestrian detection [10] [12].

Since the ground truth data of the PETS2009 datasets are not publicly available, they were created by ourselves and are available at [29]. Those of the EPFL Terrace dataset were obtained from [27]. Both record the pedestrians' locations on the ground plane. When the proposed algorithm was compared with other multicamera algorithms, the ground plane distance $r = 0.5m$ was used as the threshold for each detection and its matched ground-truth pedestrian. When the proposed algorithm was compared with deep-learning monocular algorithms, the locations of all the detections and ground-truth pedestrians were warped to each camera view by putting rectangles at the corresponding locations. The height and width of each rectnangle are based on the average size of the pedestrians standing at that location. The rectangle overlap ratio $IoU = 0.5$ [30] was used as the threshold for each detection and its matched ground-truth pedestrian.

Our performance evaluation results were compared with those of some benchmark non-deep multicamera algorithms POM [4], 3DMPP [10], MvBN [12], Khan's [3] and Ge and Collins's [9]. The tracking components in these algorithms were removed. 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 the other algorithms, since the implementation code is not available, we used their own evaluation results based on the same camera views. These algorithms were compared in the five performance metrics, as shown in Table 1, where C is the number of the camera views used and 'Eval.' indicates who made the evaluation. For those which were evaluated in PRECISION and RECALL only, such as MvBN [12] and the deep learning ones later, their MDR and FDR data were retrieved from their PRECISION and RECALL values. For the 3DMPP method [10], when more than one detection were matched to the same ground truth pedestrian, one detection was thought of as a TP and the others were thought of as FPs. The data in bold are the best results in the same comparison. A down (up) arrow indicates that a lower (higher) value corresponds to a better performance. As shown in Table 1, our algorithm outperforms the other algorithms in terms of TER, MDR and RECALL on these datasets; It competes with the MvBN and 3DMPP methods for the best performer in FDR and PRECISION. It is noted that the TER of our algorithm, on the EPFL Terrace dataset, tends to decrease whenever an additional camera view is added. However, this is not true for the PETS2009 dataset due to the poor calibration of cameras C5-C8.

The proposed algorithm was also compared with some deep multicamera detection algorithms such as RCNN-2D/3D [18], POM-CNN [19] and Deep Occlusion [19]. In addition to the five performance metrics, MODA and MODP [30] were added into the performance evaluation. To give a fair comparison and expose the efficiency of the QM algorithm, the background subtraction for foreground detection in the QM algorithm

Table 1: Performance comparison with non-deep multicamera detection algorithms ($r = 0.5m$)

PETS2009 CC dataset							
C	Method	Eval.	MDR↓	FDR↓	TER↓	Precision↑	Recall↑
2	Khan	Ours	0.091	0.286	0.377	0.761	0.909
	POM	[10]	N/A	N/A	0.267	N/A	N/A
	3DMPP	[10]	N/A	N/A	0.309	N/A	N/A
	MvBN	[12]	0.10	0.03	0.13	0.97	0.90
	QM	Ours	**0.045**	**0.027**	**0.072**	**0.973**	**0.955**
3	POM	[10]	0.073	0.179	0.252	0.837	0.927
	3DMPP	[10]	0.096	0.026	0.122	0.972	0.904
	QM	Ours	**0.030**	**0.022**	**0.052**	**0.978**	**0.970**
PETS2009 S2L1 dataset							
4	POM	[12]	0.30	0.07	0.37	0.91	0.70
	Ge [9]	[21]	0.11	0.16	0.27	0.85	0.89
	MvBN	[12]	0.05	0.06	0.11	0.94	0.95
	QM	Ours	**0.042**	**0.013**	**0.055**	**0.987**	**0.958**
EPFL Terrace dataset							
2	POM	[10]	N/A	N/A	0.845	N/A	N/A
	3DMPP	[10]	N/A	N/A	0.370	N/A	N/A
	MvBN	[12]	0.19	**0.05**	0.24	**0.94**	0.81
	QM	Ours	**0.120**	0.098	**0.218**	0.900	**0.880**
3	POM	[10]	0.331	0.355	0.686	0.653	0.669
	3DMPP	[10]	0.083	**0.048**	0.131	**0.950**	0.917
	QM	Ours	**0.037**	0.062	**0.099**	0.935	**0.939**
4	QM	Ours	0.034	0.037	0.071	0.964	0.966

was replaced by deep-learning based DeepLab algorithm [31], which is called QM + DeepLab. The performance evaluation is based on ground-plane distance threshold $r = 0.5m$. The comparison results are shown in Table 2. QM + DeepLab and QM algorithms are obviously much better than the three deep multicamera algorithms in all the seven performance metrics. QM + Deeplab algorithm further outperforms the QM due to the improved quality of the foreground detection by using DeepLab.

To illustrate the benefits of using multiple cameras, the proposed algorithm was further compared with some state-of-the-art, deep-learning based algorithms for monocu-

Table 2: Performance comparison with deep multicamera detection algorithms ($r = 0.5m$)

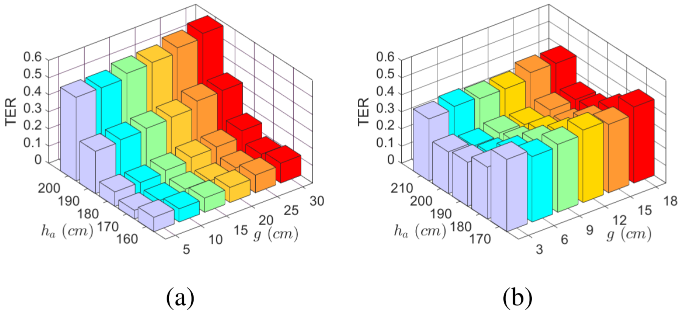
EPFL Terrace dataset with 4 cameras								
Method	Eval.	MDR↓	FDR↓	TER↓	Precision↑	Recall↑	MODA↑	MODP↑
RCNN-2D/3D	[19]	0.50	0.61	1.11	0.39	0.50	-0.11	0.28
POM-CNN	[19]	0.22	0.20	0.42	0.80	0.78	0.58	0.46
Deep Occlusion	[19]	0.18	0.11	0.29	0.88	0.82	0.71	0.48
QM	ours	0.034	0.037	0.071	0.96	0.97	0.93	**0.79**
QM+DeepLab	ours	**0.020**	**0.023**	**0.043**	**0.98**	**0.98**	**0.96**	**0.79**

Table 3: Performance comparision with deep monocular detection algorithms ($IoU = 0.5$)

EPFL Terrace dataset with 4 cameras								
Method	Eval.	MDR↓	FDR↓	TER↓	Precision↑	Recall↑	MODA↑	MODP↑
Faster RCNN	ours	0.30	**0.03**	0.33	0.95	0.70	0.67	0.74
Mask RCNN	ours	0.19	0.04	0.23	0.95	0.81	0.77	0.73
YOLOv3	ours	0.20	0.04	0.24	**0.96**	0.80	0.77	0.72
QM	ours	0.043	0.062	0.105	0.94	0.96	0.90	**0.80**
QM+DeepLab	ours	**0.023**	0.062	**0.085**	0.94	**0.98**	**0.92**	**0.80**

lar pedestrian detection, such as Faster RCNN [14], Mask RCNN [15] and YOLOv3 [32]. The performance evaluation is based on the threshold $IoU = 0.5$ in each camera view and then taking the average across all camera views. The pedestrians who are outside the AOI region or the FOV of a camera view were excluded in the performance evaluation in that camera view. The comparison results are shown in Table 3. The QM + DeepLab and QM algorithms are much better than the three deep monocular algorithms in terms of MDR, TER, RECALL, MODA and MODP. The missed detections in the monocular algorithms are usually caused by partial occlusion. The proposed algorithms also have similar performance with the deep monocular algorithms in FDR and PRECISION.

In the proposed algorithm, there are two crucial parameters. One is the average height of pedestrians h_a. The other is the grid resolution g. These two parameters were validated by using the PETS2009 CC dataset and Terrace dataset with two camera views, as shown in Fig. 12. In the PETS2009 CC dataset, when $h_a = 160$ cm to 180



(a) (b)

Figure 12: Parameter validation on the average height h_a and grid resolution g: (a) PETS2009 CC dataset and (b) EPFL Terrace dataset.

cm and $g = 5$ cm to 20 cm, the TER is lower than 0.115 and its minimum value 0.072 is detected at $h_a = 170$ cm and $g = 5$ cm. In the Terrace dataset, when $h_a = 190$ cm to 200 cm and $g = 3$ cm to 15 cm, the TER is lower than 0.261 and its minimum value 0.218 is detected at $h_a = 200$ cm and $g = 6$ cm.

To investigate the efficiency of the proposed algorithm, the number of steps in the prime candidate chart at each frame was counted. In the PETS2009 CC dataset with two views, the algorithm terminated after steps 1, 2 and 3 in 89%, 7% and 4% of the frames, respectively. In the EPFL Terrace dataset with four views, these are 61%, 39% and 0% of the frames. The speed of the proposed algorithm was also tested by using a PC with an Intel i5 4-core CPU running at 3.20 GHz. The results are shown in Table 4. The execution time for running the proposed algorithm is made up of three main parts: foreground extraction (GMM/SuBSENCE/DeepLab), probabilistic occupancy maps (POM) and the QM algorithm. Only the POM part is influenced by the grid resolution and its computation is very efficient by using the integral images [24]. The time for the Terrace dataset is much longer than that for the CC dataset, since the foreground regions in the Terrace dataset are much larger. The time for the QM algorithm is neglectable. In our experiments, all the deep learning methods were implemented on GPUs. We have tested the speed of DeepLab using a NVIDIA RTX 2080S GPU. The foreground extraction time for the Terrace dataset with 2 camera views is 244 ms per frame, which makes the QM + DeepLab algorithm running at a frame rate of 4 fps.

Table 4: Speed evaluation of the proposed algorithms.

PETS2009 CC dataset (two camera views)						
Grid Resolution (cm)	5	10	15	20	25	30
(1) GMM (ms)	35	35	36	35	34	35
(2) POM (ms)	37	12	6	5	5	4
(3) QM (ms)	5	6	5	5	6	5
Total Time/Frame (ms)	77	53	47	45	45	44
FPS	13.0	18.9	21.3	22.2	22.2	22.7
EPFL Terrace dataset (two camera views)						
Grid Resolution (cm)	3	6	9	12	15	18
(1) SuBSENCE (ms)	95	94	95	93	93	94
(2) POM (ms)	166	43	22	14	10	8
(3) QM (ms)	1	2	1	2	1	1
Total Time/Frame (ms)	262	139	118	109	104	103
FPS	3.8	7.2	8.5	9.2	9.6	9.7

8. Conclusions

We have proposed the use of the QM algorithm for multiview pedestrian detection. Its improved performance has been demonstrated in comparison with benchmark non-deep or deep multicamera/monocular algorithms in this field. This algorithm is iteratively switched between finding essential candidates and finding redundant candidates, in which the match score (the joint occupancy likelihood) does not play a central role. This is in contrast to traditional data association schemes too reliant on match scores and thus greatly reduces the search space for an optimized solution. It is worth mentioning that this algorithm only starts with the binary foreground silhouettes at a single frame. If temporal and colour information is incorporated, its performance can be further improved. Future work includes the use of the QM algorithm for multicamera object tracking and deep multicamera pedestrian detectors.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (NSFC) under Grant 60975082 and Xi'an Jiaotong-Liverpool University under Grant RDF-17-01-33.

References

[1] M. Xu, J. Orwell, L. Lowey, D. Thirde, Architecture and algorithms for tracking football players with multiple cameras, IEE Proc. Vision, Image and Signal Processing 152 (2) (2005) 232–241.

[2] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, S. Maybank, Principal axis-based correspondence between multiple cameras for people tracking, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 663–671.

[3] S. M. Khan, M. Shah, Tracking multiple occluding people by localizing on multiple scene planes, IEEE Trans. Pattern Anal. Mach. Intell. 31 (3) (2009) 505–519.

[4] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 267–282.

[5] M. Liem, D. M. Gavrila, Multi-person tracking with overlapping cameras in complex, dynamic environments, in: British Machine Vision Conf., 2009, pp. 199–218.

[6] D. Arsic, E. Hristov, N. Lehment, B. Hornler, B. Schuller, G. Rigoll, Applying multi layer homography for multi camera person tracking, in: ACM/IEEE Int. Conf. on Distributed Smart Cameras, 2008, pp. 1–9.

[7] C. W. Liu, H. T. Chen, K. H. Lo, C. J. Wang, J. H. Chuang, Accelerating vanishing point-based line sampling scheme for real-time people localization, IEEE Trans. Circuits Syst. Video Techn. 27 (3) (2017) 409–420.

[8] R. Eshel, Y. Moses, Tracking in a dense crowd using multiple cameras, Int. J. of Computer Vision 88 (1) (2010) 129–143.

[9] W. Ge, R. T. Collins, Crowd detection with a multiview sampler, in: European Conf. on Computer Vision, 2010, pp. 324–337.

[10] Á. Utasi, C. Benedek, A bayesian approach on people localization in multicamera systems, IEEE Trans. Circuits Syst. Video Techn. 23 (1) (2013) 105–115.

[11] A. Alahi, L. Jacques, Y. Boursier, P. Vandergheynst, Sparsity driven people localization with a heterogeneous network of cameras, J. of Math. Imaging and Vision 41 (1) (2011) 39–58.

[12] P. Peng, Y. Tian, Y. Wang, J. Li, T. Huang, Robust multiple cameras pedestrian detection with multi-view bayesian network, Pattern Recognition 48 (5) (2015) 1760–1772.

[13] Y. Yan, M. Xu, J. S. Smith, Multiview pedestrian localisation via a prime candidate chart based on occupancy likelihoods, in: IEEE Int. Conf. on Image Processing, 2017, pp. 2334–2338.

[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real time object detection with region proposal networks, IEEE Trans. on Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.

[15] K. He, G. Gkioxari, P. Dollr, R. Girshick, Mask R-CNN, IEEE Trans. on Pattern Anal. Mach. Intell. 42 (2) (2020) 386–397.

[16] X. Wei, H. Zhang, S. Liu, Y. Lu, Pedestrian detection in underground mines via parallel feature transfer network, Pattern Recognition 103 (2020) 107195.

[17] C. Zhou, J. Yuan, Multi-label learning of part detectors for occluded pedestrian detection, Pattern Recognition 86 (2019) 99–111.

[18] Y. Xu, X. Liu, Y. Liu, S. Zhu, Multi-view people tracking via hierarchical trajectory composition, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 4256–4265.

[19] P. Baqué, F. Fleuret, P. Fua, Deep occlusion reasoning for multi-camera multi-target detection, in: IEEE Int. Conf. on Computer Vision, 2017, pp. 271–279.

[20] R. Zhang, L. Wu, Y. Yang, Y. Chen, M. Xu, Multi-camera multi-player tracking with deep player identification in sports video, Pattern Recognition 102 (2020) 107260.

[21] T. Chavdarova, F. Fleuret, Deep multi-camera people detection, in: IEEE Int. Conf. on Machine Learning and Appl., 2017, pp. 848–853.

[22] J. Ren, M. Xu, J. S. Smith, S. Cheng, Multi-view and multi-plane data fusion for effective pedestrian detection in intelligent visual surveillance, Multidimensional Systems and Signal Processing 27 (4) (2016) 1007–1029.

[23] M. Cordts, T. Rehfeld, L. Schneider, D. Pfeiffer, M. Enzweiler, S. Roth, M. Pollefeys, U. Franke, The stixel world: A medium-level representation of traffic scenes, Image and Vision Computing 68 (2017) 40–52.

[24] F. C. Crow, Summed-area table for texture mapping, Computer Graphics 18 (3) (1984) 207–212.

[25] E. J. McCluskey, Minimization of Boolean functions, Bell System Technical Journal 35 (6) (1956) 1417–1444.

[26] PETS2009 Dataset, `http://www.cvg.reading.ac.uk/PETS2009`.

[27] EPFL Terrace Dataset, `https://cvlab.epfl.ch/data/pom`.

[28] P.-L. St-Charles, G.-A. Bilodeau, R. Bergevin, SuBSENSE: A universal change detection method with local adaptive sensitivity., IEEE Trans. on Image Processing 24 (1) (2015) 359–373.

[29] XJTLU CVLab, `https://github.com/xjtlu-cvlab/QM.git`.

[30] R. Kasturi, D. Goldgof, P. Soundararajan, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 319–336.

[31] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: European Conf. on Computer Vision, 2018, pp. 833–851.

[32] J. Redman, A. Farhadi, YOLOv3: An incremental improvement, arXiv :1804.02767 (2018).