Pattern Recognition Letters 32 (2011) 47–55

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multiple camera people detection and tracking using support integration

Thiago T. Santos *, Carlos H. Morimoto

Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010, 05508-090 São Paulo, Brazil

ARTICLE INFO

Article history:
Available online 19 May 2010

Keywords:
People tracking
Multiple view integration
Video surveillance and monitoring
Homography constraint

ABSTRACT

This paper proposes a method to locate and track people by combining evidence from multiple cameras using the homography constraint. The proposed method use foreground pixels from simple background subtraction to compute evidence of the location of people on a reference ground plane. The algorithm computes the amount of support that basically corresponds to the "foreground mass" above each pixel. Therefore, pixels that correspond to ground points have more support. The support is normalized to compensate for perspective effects and accumulated on the reference plane for all camera views. The detection of people on the reference plane becomes a search for regions of local maxima in the accumulator. Many false positives are filtered by checking the visibility consistency of the detected candidates against all camera views. The remaining candidates are tracked using Kalman filters and appearance models. Experimental results using challenging data from PETS'06 show good performance of the method in the presence of severe occlusion. Ground truth data also confirms the robustness of the method.

1. Introduction

This paper presents a multiple camera solution to the problem of tracking a group of people. Multiple camera views can be used to recover 3D structure information and solve occlusion in environments with several individuals. Recently, several works have suggested a simpler approach that can be used with a network of sparse uncalibrated cameras based on the homography constraint (Eshel and Moses, 2008; Fleuret et al., 2008; Hu et al., 2006; Kim and Davis, 2006; Santos and Morimoto, 2008). The homography constraint establishes that multiple projections of the principal axis of an elongated object using a homography from each camera view q to the ground or reference plane Π intersect at the position of the object in the reference plane ("ground position" of the object).

Kim and Davis (2006) use the homography constraint within a particle filtering framework for people tracking. First, a set of particles that correspond to ground positions is draw from the filter dynamics. Each particle is associated with an appearance model (Senior et al., 2006) to perform people segmentation in each camera view. Once foreground pixels are segmented and classified into single objects (persons), the principal axis of each person is computed and the homography constraint is used to compute their locations. The main drawback of the system is its requirement that individuals must initially appear as isolated foreground blobs to proper modeling.

To detect multiple people using multiple camera views Hu et al. (2006) use the homography constraint for pairs of cameras. By projecting the principal axis of a person from camera view q to p, the likelihood between two axes from these different views is computed comparing their intersection with a predicted ground position. To compute this point the authors combine single view foreground segmentation with Kalman filter based tracking. The likelihood is used to drive the axis correspondence process. The system relies on individual segmentation, so inter-object occlusion can degrade the axis location performance.

Eshel and Moses (2008) use the homography constraint in several planes parallel to the ground plane, searching for heads in the higher planes. All camera views are mapped using homographies to a reference plane and intensity correlation is used to detected candidate heads. A nearest neighbor approach is applied to find correspondences along time, producing tracks. In a further step, tracks are combined in individual trajectories by the use of six different measurements to evaluate track overlap, distance, and direction. According to authors, people dressing in similar colors are a main source of false positives, a natural drawback from the correlation approach. The cameras are placed at high elevations and the authors report that the performance of the system deteriorates considerably when less than five cameras are used.

Fleuret et al. (2008) use a probabilistic framework to perform simultaneous detection and tracking. Their model is a combination of a simple motion model with an appearance model. The appearance model is composed of an RGB color density and a ground plane occupancy map. In the occupancy map, the ground plane is partitioned into a regular grid and the probability of occupancy

* Corresponding author.
E-mail address: thiago@acm.org (T.T. Santos).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.05.016

of each grid cell is estimated using results from background subtraction. This occupancy model is a conditional distribution between the foreground and the occupied cells configuration. The Viterbi algorithm is used to find the most likely trajectory for each individual and a greedy heuristic is applied to optimize one trajectory after other. For reliable detection and location, each person must be seen as an individual blob in at least one view.

Previous methods for single person segmentation are affected by two main problems. First, partial and total occlusion are common in crowed scenes such as the one in Fig. 3(b). In places such as airport halls or train stations, people frequently walk in small groups most of the time, causing occlusion in all camera views. Second, when color models are used for segmentation, people dressed with similar colors become another source of problems (Hu et al., 2006).

The main contribution of this paper is the definition of a novel algorithm based on the homography constraint that does not rely on single view segmentation of the subjects or previous tracking information. Instead of a *segment-then-locate* approach, we propose a *locate-then-segment* approach, integrating available information of all cameras before any detection decision. This paper extends our previous work presented in Santos and Morimoto (2008) in several ways. First the people detection method was made more robust to false positives with the introduction of a new filtering algorithm. This paper also introduces a multiple person tracking algorithm based on Kalman filters and appearance models, and more extensive experimental results are presented using ground truth tracking data.

Because the system does not require previous object segmentation for people detection, our work has some similarities with the very recent work of Khan and Shah (2009). Their work use the homography constraint to fuse foreground likelihood information from multiple views to resolve occlusions and localize people on a reference scene plane. Similar to Eshel and Moses (2008), Khan and Shah (2009) also rely on multiple planes parallel to the ground to improve the robustness of the method. Detection and tracking are performed simultaneously by graph cuts segmentation of tracks in the space–time occupancy likelihood data.

In our method, multiple view perspective geometry and the homography constraint are applied to collect evidence of people presence from each camera view. Our method elegantly integrates the information of all parallel planes by projecting the foreground directly on the reference plane and accumulating the evidence from multiple cameras. Occlusion and people detection are solved simultaneously at each time using the accumulated evidence from all cameras. We have tested the method using very challenging data from PETS'06 with good results. The next section describes the method in detail. Experimental results are presented in Section 3. Section 4 concludes the paper.

2. Multiple person detection and tracking

Fig. 1 shows a block diagram of our proposed multiple person detection and tracking system. Each static camera q feeds a background subtraction module. The background color distribution for each pixel is modeled using mixture of Gaussians. The segmented foreground is used to compute evidence of people presence for each pixel on the reference image Π (floor plane). Our algorithm computes the amount of support that basically corresponds to the "foreground mass" above each pixel. Therefore, pixels that correspond to ground points have more support. Perspective is carefully considered to accurately detect objects near and far away from the cameras. The support computed from each camera view is transformed to the ground plane using the appropriate homography. The ground plane accumulates the evidence from all views. People detection is performed by locating regions of local maxima in the ground plane accumulator. Once people candidates are detected, appearance models are computed for each candidate. We have developed an efficient algorithm to match the detected candidates with tracked objects. Each tracked object is represented by its appearance model and an associated Kalman filter. Trackers that are assigned to candidates during the matching process are updated. Observations that do not match any tracker are potential new targets, and trackers that do not receive a match are considered lost.

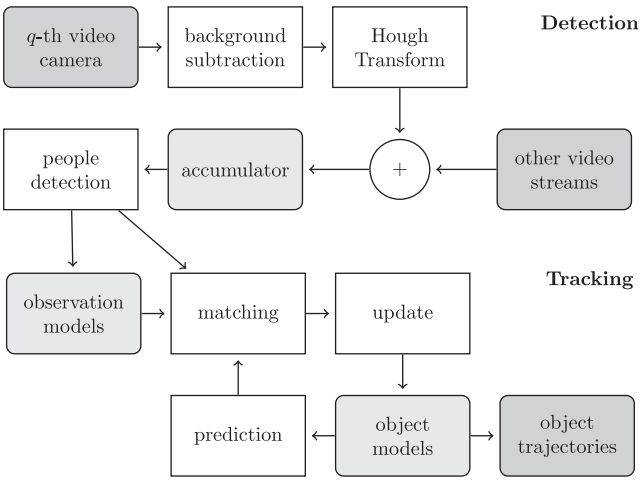

Fig. 1. Block diagram of the multiple person detection and tracking system.

2.1. Background subtraction

The color distribution for each background pixel in time is modeled as a mixture of Gaussian distributions (Stauffer and Grimson, 1999). This Gaussians mixture approach is able to deal with multiple modes on the background color distribution probability.

A pixel $\boldsymbol{x}$ presents color $f(\boldsymbol{x})$, represented in *rgI* space (normalized red, normalized green and light intensity). Normalized color is less sensitive (compared to RGB space) to small changes in illumination caused by shadows (Wang and Suter, 2005).

The color distribution of a pixel is modeled by K Gaussians. The k-th Gaussian presents mean vector $\mu_k = \langle \mu_k^r, \mu_k^g, \mu_k^I \rangle$, a diagonal covariance matrix Σ_k and a *weight* w_k, that correspond to the probability that the pixel has a subclass k. An expectation–maximization (EM) algorithm combined to an agglomerative clustering strategy (Bouman, 1997) is applied to estimate K and the mixture parameters of each color distribution. Because the training set is not free from moving objects, the *background distribution* is represented by the Gaussians whose weight w_k is greater than a threshold T_w.

Each pixel $\boldsymbol{x}_i$ is compared against all subclasses in the background mixture model. The pixel is classified as foreground if:

$$|f_c(\boldsymbol{x}_i) - \mu_k^c| > T_b \cdot \sigma_k^c \tag{1}$$

for all channels $c = r, g, I$, where T_b is a decision boundary threshold and σ_k^c is the variance in channel c found in the diagonal matrix Σ_k – independence between channels is assumed for simplicity.

Shadows are a common source of artifacts. We use an additional test, based on Wang and Suter (2005) work, to perform shadow removal. Let $f^I(\cdot)$ denotes the intensity of a pixel in f. If $\boldsymbol{x}_i$ chromaticity fits the pixel r and g models and

$$T_{\text{shadow}} \leqslant \frac{f_I(\boldsymbol{x}_i)}{\mu_k^I} \leqslant 1.0,$$

where T_{shadow} is a threshold, then $\boldsymbol{x}_i$ will be classified as background. The idea is that a background pixel will present just a fraction of its expected intensity value within shadow regions.

2.2. Support computation

Let Π be the ground or reference plane, $\boldsymbol{x}^q$ be a foreground pixel of camera q corresponding to the projection of the point $\boldsymbol{X} \in \Pi$, and let the pixel relation $\boldsymbol{y}$ `above` $\boldsymbol{x}$ be true iff the foreground pixel $\boldsymbol{y}$ lies on the half line defined by the ray $\boldsymbol{x} + \vec{\text{up}}$, and false otherwise, where $\vec{\text{up}}$ is a unit vector pointing to the up direction.

Just for illustration purposes, consider a single person scenario represented by a line segment L, shown in Fig. 2. Let $\boldsymbol{X}_i \in \Pi$ be the bottom end of L, $\boldsymbol{l}^q$ the projection of L for camera q, and $\boldsymbol{x}_i^q$ the projection of $\boldsymbol{X}_i \in \boldsymbol{l}^q$. Then all pixels $\boldsymbol{x}_j^q \in \boldsymbol{l}^q$ such that $i \neq j$, are `above` $\boldsymbol{x}_i^q$. We define support $S(\boldsymbol{x}_i^q)$ as the number of foreground pixels `above` $\boldsymbol{x}_i^q$.

Notice that $S(\boldsymbol{x}_i^q)$ can be computed for any $\boldsymbol{x}_i^q$ regardless of a true correspondence between $\boldsymbol{x}_i^q$ and a ground point in Π because only the `above` relation is used. The vanishing point in the vertical direction can be used to compute the true $\vec{up}$ direction for every pixel $\boldsymbol{x}^q$. For a blob corresponding to the segmentation of a person using the background subtraction algorithm, the support of every pixel $\boldsymbol{x}_i^q$ within the blob can be computed and back-projected onto the ground plane. Regions on the ground plane with large local support values are good candidates for the location of a person.

2.2.1. Perspective normalization

Due to perspective, simple pixel counting to compute $S(\boldsymbol{x}_i^q)$ is not accurate. Fig. 3(b) shows six vertical bars of different lengths. All of them correspond to the *same* height h of the person standing at $\boldsymbol{x}_r^q$ but at different locations $\boldsymbol{x}_i^q$. Therefore, in order to use support to compute object locations, the support values must be normalized to compensate for perspective effects. Using an object of known height h_r as reference, seen by every camera q at $\boldsymbol{x}_r^q$, we pre-compute a normalization factor $\eta(\boldsymbol{x}_i^q)$, for all $\boldsymbol{x}_i^q$, that corresponds to the inverse of the height h_r when the reference object is placed at the ground position corresponding to $\boldsymbol{x}_i^q$.

For any camera q, let $\boldsymbol{x}_r^q$ be the position of the reference object with height h_r. Let $\hat{\boldsymbol{x}}_r^q$ be the projection of $\boldsymbol{x}_r^q$ onto a parallel plane h_r units far from Π, as shown in Fig. 4. Let $d(i,j)$ denote the distance in pixels between any two points (i,j) and assume that $d(\boldsymbol{x}_r^q, \hat{\boldsymbol{x}}_r^q)$ is known (the reference height). Then the height $d(\boldsymbol{x}_i^q, \hat{\boldsymbol{x}}_i^q)$ of the object when placed at $\boldsymbol{x}_i^q$ can be estimated using the cross-ratio invariance property of projective geometry (Criminisi et al., 2000).

Criminisi et al. (2000) applied the cross-ratio to find the relation:

$$\frac{h_r}{h_q} = 1 - \frac{d(\hat{\boldsymbol{x}}_r^q, \boldsymbol{c}_r^q)\, d(\boldsymbol{x}_r^q, \boldsymbol{v}^q)}{d(\boldsymbol{x}_r^q, \boldsymbol{c}_r^q)\, d(\hat{\boldsymbol{x}}_r^q, \boldsymbol{v}^q)}, \tag{2}$$

between the reference height h_r and the camera height h_q (the distance from the camera center to the reference plane Π) when the reference object is located at $\boldsymbol{x}_r^q$. The points $\boldsymbol{c}_r^q$ and $\boldsymbol{c}_i^q$ are the projec-

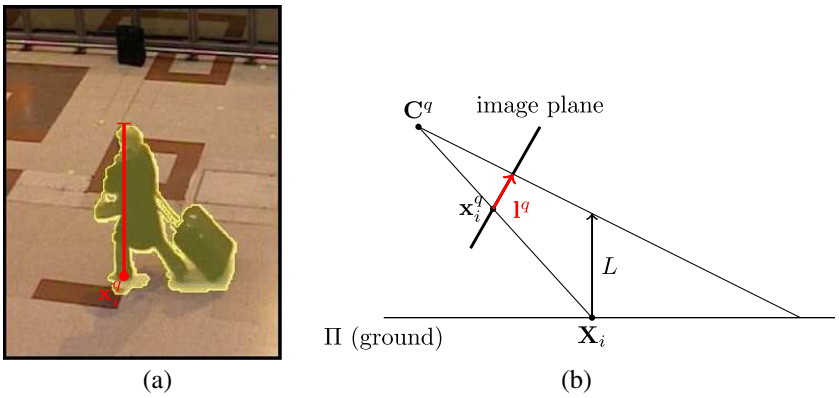

Fig. 2. (a) The support of a point $\boldsymbol{x}_i^q$ in the image plane is the amount of pixels of the object seen above it (highlighted in yellow). (b) Projection on camera q – $\boldsymbol{C}^q$ is the camera center. The support is a speculation about the amount of "mass" of an object that may be relying on the position $\boldsymbol{X}_i$ in the ground plane. See the text for details. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

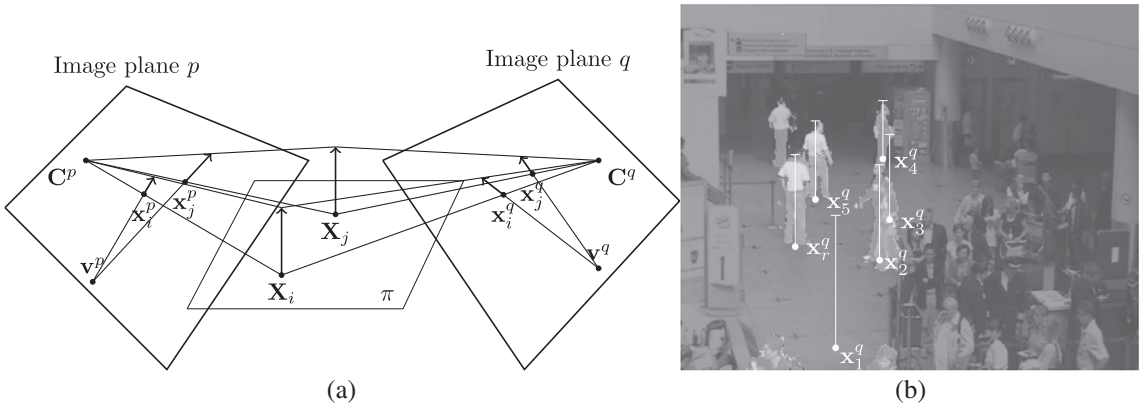

Fig. 3. (a) Perspective transformation for two cameras p and q with projection centers $\boldsymbol{C}^p$ and $\boldsymbol{C}^q$ and vanishing points $\boldsymbol{v}^p$ and $\boldsymbol{v}^q$. (b) Perspective correction and height filtering. The bright areas correspond to segmented foreground. The vertical bars correspond to the height of the person standing at $\boldsymbol{x}_r^q$ seen at different locations $\boldsymbol{x}_i^q$.

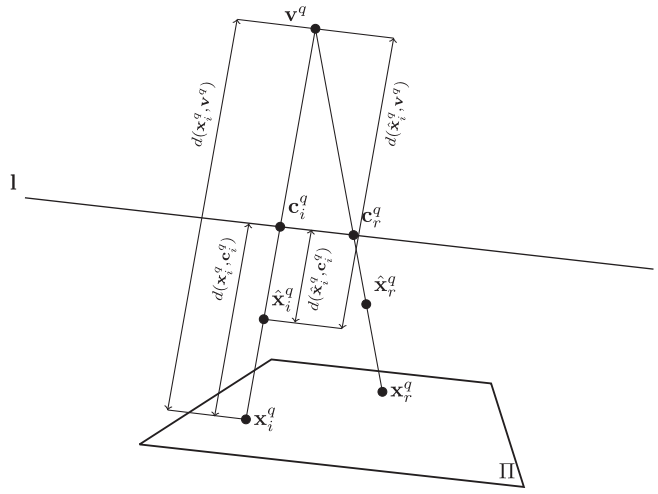

Fig. 4. Distances for the computation of the perspective normalization factor for the reference position $\boldsymbol{x}_r^q$ and an arbitrary position $\boldsymbol{x}_i^q$. $\boldsymbol{l}$ is the ground plane vanishing line (horizon seen by camera q) and $\boldsymbol{v}^q$ is the vertical vanishing point.

tions of $\boldsymbol{x}_r^q$ and $\boldsymbol{x}_i^q$ onto the ground plane vanishing line $\boldsymbol{l}$, as seen in Fig. 4.

A similar equation can be computed when the reference object is placed at $\boldsymbol{x}_i^q$:

$$\frac{h_r}{h_q} = 1 - \frac{d(\hat{\boldsymbol{x}}_i^q, \boldsymbol{c}_i^q)\, d(\boldsymbol{x}_i^q, \boldsymbol{v}^q)}{d(\boldsymbol{x}_i^q, \boldsymbol{c}_i^q)\, d(\hat{\boldsymbol{x}}_i^q, \boldsymbol{v}^q)}. \tag{3}$$

Now consider $\alpha(\boldsymbol{x}_i^q) = d(\boldsymbol{x}_i^q, \boldsymbol{v}^q)$ and $\beta(\boldsymbol{x}_i^q) = d(\boldsymbol{x}_i^q, \boldsymbol{c}_i^q)$. Then terms on $\hat{\boldsymbol{x}}_i^q$ can be rewritten as

$$d(\hat{\boldsymbol{x}}_i^q, \boldsymbol{v}^q) = \alpha(\boldsymbol{x}_i^q) - \eta(\boldsymbol{x}_i^q), \tag{4}$$

$$d(\hat{\boldsymbol{x}}_i^q, \boldsymbol{c}_i^q) = \beta(\boldsymbol{x}_i^q) - \eta(\boldsymbol{x}_i^q). \tag{5}$$

Defining:

$$\gamma = \frac{d(\hat{\boldsymbol{x}}_r^q, \boldsymbol{c}_r^q)\, d(\boldsymbol{x}_r^q, \boldsymbol{v}^q)}{d(\boldsymbol{x}_r^q, \boldsymbol{c}_r^q)\, d(\hat{\boldsymbol{x}}_r^q, \boldsymbol{v}^q)}, \tag{6}$$

and using the equality between (2) and (3), it results that:

$$\eta(\boldsymbol{x}_i^q) = \frac{\alpha(\boldsymbol{x}_i^q)\beta(\boldsymbol{x}_i^q)(1-\gamma)}{\alpha(\boldsymbol{x}_i^q) - \beta(\boldsymbol{x}_i^q)\gamma}. \tag{7}$$

The value of $\eta(\boldsymbol{x}_i^q)$ is pre-computed for each $\boldsymbol{x}_i^q$ and used as a perspective normalization factor for the computation of support.

2.2.2. Bounded support computation

Because objects occlude each other, blobs segmented using background subtraction might be composed of several objects. Large elongated blobs produce large number of false positives due to false high support values. By limiting object heights within an appropriated range ($h_{\min}$,$h_{\max}$), the maximum normalized support value is also bounded and many of the false positives candidates are filtered out. Small objects with low support values can also be filtered using $h_{\min}$.

Thus a candidate object for tracking cannot present support below the minimum height $h_{\min}$ or above a maximum $h_{\max}$. Fig. 3(b) illustrates the idea. Bright areas mark the foreground segmented from camera q. The vertical bar directions are defined by the ground points $\boldsymbol{x}_i^q$ and the vanishing point $\boldsymbol{v}^q$. The bar lengths in pixels correspond to $h_{\max}$. The support of $\boldsymbol{x}_i^q$ is the amount of foreground pixels along its corresponding bar. Observe that the point $\boldsymbol{x}_1^q$ does not present any support and that $\boldsymbol{x}_2^q$, $\boldsymbol{x}_3^q$, $\boldsymbol{x}_4^q$ and $\boldsymbol{x}_5^q$ present similar support values. Observe that the line of three people under occlusion would cause unrealistically high support values in a large region.

The bounded normalized support $S_q(\boldsymbol{x}_i^q)$ can be computed efficiently for all pixels of a line defined by $\boldsymbol{x}_i^q$ and $\boldsymbol{v}^q$ (i.e., a line orthogonal to the ground plane Π) as follows.

Let $\boldsymbol{s} = \langle \boldsymbol{x}_1^q, \ldots, \boldsymbol{x}_n^q \rangle$ be the line segment obtained by constraining the line by the image frame, as seen in Fig. 5. Algorithm 1 computes the support by counting the number of foreground pixels projecting onto $\boldsymbol{x}_i^q$ and using the perspective normalization factor $\eta(\boldsymbol{x}_i^q)$ to get the support value in reference units. The maximum support is constrained to filter out objects extending beyond $h_{\max}$.

As an example to better understand the algorithm, consider that at location $\boldsymbol{x}_{280}^q$ there are 240 foreground pixels `above`, i.e., $F[280] = 240$, as seen in Fig. 5. According to the pre-computed values of $\eta(\boldsymbol{x}_{280}^q)$ and $h_{\max}$, the tallest allowed object at location $\boldsymbol{x}_{280}^q$ would cover up to 120 pixels and reach pixel $\boldsymbol{x}_{160}^q$ (see line 1 of the algorithm). Since $F[160] = 140$ (there are 140 foreground pixels above $\boldsymbol{x}_{160}^q$), there are 100 foreground pixels between $\boldsymbol{x}_{280}^q$ and $\boldsymbol{x}_{160}^q$. This number, normalized by $\eta(\boldsymbol{x}_{280}^q)$ and bounded, is the support due to the evidence at $\boldsymbol{x}_{280}^q$.

Background segmentation errors affect the correct computation of an object's support. For example, when people are dressed using colors similar to the background color distribution, parts of their bodies are misdetected. The foreground pixel counting used in Lines 4–8 address this issue and does not constrain support computation to perfect background classification.

Fig. 6 shows support results for three different cameras. The figure shows support peaks near people's feet, as expected. Some false foreground detection seen in the top row images are caused by shadows, that produce high support values in regions of the ground plane. Although shadow artifacts can become an issue in single view processing, multiple view integration is able to reduce this problem. If a region with no people presents some support from shadow in a camera view, it is unlikely the same will happens in the other views.

Algorithm 1: Algorithm to compute the support $S_q(\boldsymbol{x}_i^q)$ for all points $\boldsymbol{x}_i^q$ in segment $\boldsymbol{s}$.

```
1: procedure SUPPORT (s = ⟨x_1^q, ..., x_n^q⟩, h_min, h_max, η)
2:     F[0] ← 0
3:     for i ← 1,n do
4:         if x_i^q is FOREGROUND then
5:             F[i] ← F[i − 1] + 1
6:         else
7:             F[i] ← F[i-1]
8:         end if
9:         j ← i − h_max · η[x_i^q]
10:        if j > 0
11:            h ← (F[i] − F[j])/η[x_i^q]
12:        else
13:            h ← F[i]/η[x_i^q]
14:        end if
15:        if h ⩾ h_min
16:            S_q(x_i^q) ← h
17:        else
18:            S_q(x_i^q) ← 0
19:        end if
20:    end for
21:    return S_q
22: end procedure
```

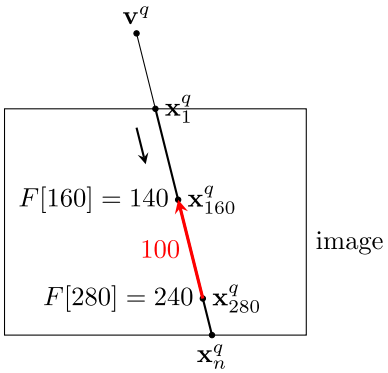

Fig. 5. An iteration of Algorithm 1 for ($i = 280$). Line 9 inspects the pixel $\boldsymbol{x}_{160}$, which corresponds to the height of the tallest expected object. Since the value of $F[140] = 140$, there must be 100 foreground pixels between $\boldsymbol{x}_{160}$ and $\boldsymbol{x}_{280}$.

2.3. Integration of multiple camera views

In the absence of occlusions, the support information computed from a single camera provides sufficient evidence to locate people on the ground plane, though a certain number of false detections and misses might occur. The detection algorithm can be made a lot more robust by combining the evidence from all cameras that see a particular ground region.

For example, in Fig. 3(b), a false ground point $\boldsymbol{x}_3^q$ has high support but it is unlikely that the same occurs in another camera. In fact, a pair of occluding objects seen in camera q might show as occluding objects for a different camera p iff the objects are along the baseline of the two cameras.

The homography matrix H_q maps ground points $\boldsymbol{x}_i^q$ in image plane q to ground points $\boldsymbol{X}_i$ of the ground plane Π according to:

$$\boldsymbol{X}_i = \mathrm{H}_q \boldsymbol{x}_i^q. \tag{8}$$

Using a set of points on the image plane and a set of corresponding points in Π, H_q can be estimated by a direct linear transformation algorithm (Hartley and Zisserman, 2004).

Let $S_q(\boldsymbol{x}_i^q)$ be the support computed at point $\boldsymbol{x}_i^q$ for camera q. All support data from Q cameras can be integrated on Π by

$$A(\boldsymbol{X}_i) = \sum_{q=1}^{Q} S_q\left(\mathrm{H}_q^{-1}\boldsymbol{x}_i\right), \tag{9}$$

where A is the *accumulator* image (Fig. 7). Objects can be located by segmenting regions of A that present large support values.

A threshold T_S is used to select points $\boldsymbol{X}_i \in \Pi$ presenting good support values. The threshold parameter at $\boldsymbol{X}_i \in \Pi$ takes into consideration $h_{\min}$ and the number of cameras able to see that location. Points of local maxima are computed by a mean-shift procedure. Mean-shift blurring process (Cheng et al., 1995) moves data points in the gradient direction of a smoothed version of the original function. Applied to A, the process integrates the support information within a neighborhood of $\boldsymbol{X}_i$.

Let G be the set of found local maxima points. Points $\boldsymbol{X}_i \in G$ correspond to real people locations and some false positives. Main sources of false positives are severe occlusion in all views and people aligned in the baseline of a pair of cameras. The idea to filter the false positives is to select a subset of G that, under total occlusion relations, is able to "explain" the occurrence of the remaining points.

Points in G are labeled UNSELECTED and inserted in a priority queue ordered by $A(\boldsymbol{X}_i)$. We pop the queue, marking the current point $\boldsymbol{X}_i$ as SELECTED. Then we visit all the points $\boldsymbol{X}_j$ that are occluded by $\boldsymbol{X}_i$. If $\boldsymbol{X}_j$ is UNSELECTED and it is occluded by a SELECTED point in all views, it will be labeled COVERED and removed from the queue.

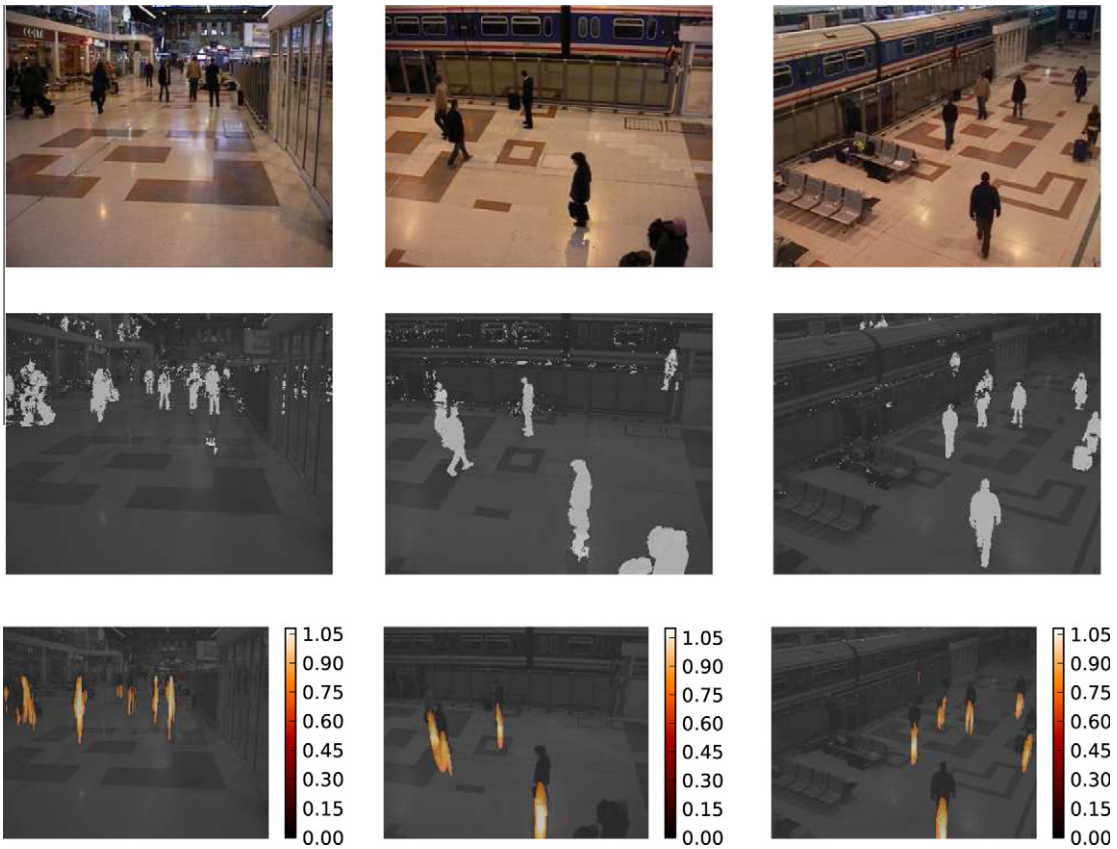

Fig. 6. (a) Input images for the support algorithm. (b) Observe that the support peaks at the ground positions of each person.

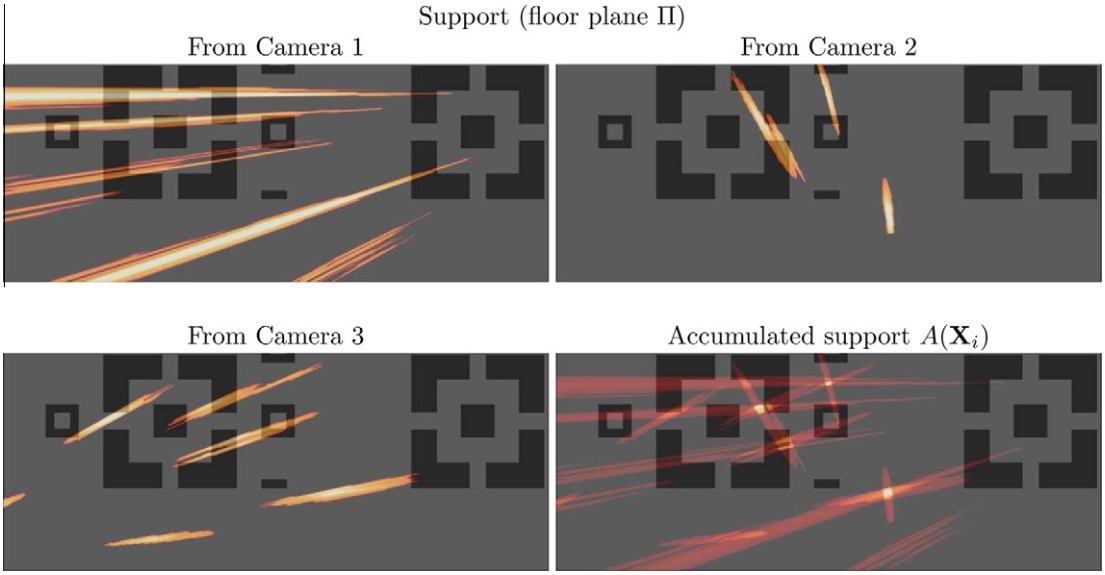

Fig. 7. Multi-view integration for 3 cameras. Homographies are used to warp support from the original camera view to the floor plane Π. The accumulated support $A(\boldsymbol{X}_i)$ peaks on true object positions.

We repeat this procedure until no more UNSELECTED points are available. SELECTED points are returned as people location candidates and will be further used as measurements by the tracking module. This procedure ensures that the removed false positives are fully justified as spurious interactions from evidences of people in other locations.

2.4. Object tracking

Our system tracks multiple objects simultaneously using one Kalman filter per object. A tracked object (person) is represented by a multi-view appearance model. The model consists of two RGB color histograms for each camera view, corresponding to the top and bottom parts of the object (shirt and pants). Each model also keeps a foreground and occlusion mask for each camera. The color histograms, foreground, and occlusion masks are updated at every frame.

Before updating the tracker at every new frame t, appearance models for the detected target candidates (called the observation appearance models) are build using the list of candidate positions computed as described in previously. A bounding box for each camera view is computed from the position and estimated height (support) of the candidate object. The RGB color histograms, foreground, and occlusion masks are computed using such bounding boxes.

To efficiently determine the assignment of observations to targets all possible assignments we have developed the following greedy algorithm.

First candidate positions z_i are paired with all trackers T_j that expects the tracked object to be at a vicinity of z_i. All such pairs are inserted in a priority queue according to the probability $p(z_i|x_j, \sigma_j)$, where z_i is the observation position on the ground plane and x_j and σ_j are respectively the state and covariance matrix of the Kalman filter T_j.

Next the first pair of the queue is popped and their appearance models are used to test if the observation actually matches the tracked object. An observation matches an object iff there is good similarity between their color models. Color similarity is computed using histogram intersection. In case the tracker is updated using the matched observation, the object appearance model is also updated using the observation appearance model and a learning factor alpha as follows. Let $H_{q,\,t}[b]$ be the histogram value for bin b in the a color model of camera q at frame t and let H^o be the corresponding observation model. Then

$$H_{q,t+1}[b] = (1-\alpha)H_{q,t}[b] + \alpha H^o_{q,t+1}[b], \tag{10}$$

Observation z_i that are matched are marked as USED, so no other tracker will be updated using z_i. The process continues until the queue is empty. The greedy algorithm might not assign all observations to all trackers. Observations that are not assigned to a tracker correspond to potential new objects so a new tracker is created. Each tracker T_j keeps a counter to register the number of successful assignments, and a flag. Upon creation new trackers receive a NEW flag and their appearance models initialized to the observation appearance models.

After the counter registers a large enough number of assignments, the tracker flag is updated to ON. At this moment, the tracker is assumed to be following a real subject. If a tracker is not assigned to any observation, its flags is updated to LOST. A LOST tracker is updated using the Kalman prediction and its covariance matrix is increased to enhance the chances of the tracker to find a match in the next frame. A tracker that keeps a LOST flag for a long time is finished and removed from the list of trackers. Trackers presenting the ON flag have priority on the assignment queue and LOST trackers have priority over NEW ones.

3. Results

The system was tested using the S7 dataset from the PETS 2006 Benchmark Data (Thirde et al., 2006). This dataset presents video recorded at Victoria Station in London, UK. Video from three cameras was used, demonstrating that just a few cameras are enough to produce good detection and tracking results. We used half of S7 frame sequence in our tests (the last 1500 frames of the original 3000 sequence – about 1 min of video). The sequence presents 22 individuals walking in a hall. About 1/3 of the hall area is covered by three cameras. The baseline of the two cameras that cover the remaining area crosses the entire hall, creating severe occlusion situations.

Image points were manually selected to compute the vanishing points of each camera and the appropriate homography matrix to

the ground plane Π. The height of a person was used to define the reference height unit. Parameters were setted manually to obtain satisfactory results in a 10 s long sample of the video in question. The allowed height range was set to [0.6, 1.1] units (that is 60–110% of the reference man's height). An unit flat kernel of width 19 pixels was applied in the mean-shift local maxima detection procedure (1 pixel ∼2 cm in the reference ground plane image). Trajectories from the tracking module shorter than 50 frames (about 2 s) are considered false positives and removed.

3.1. Object detection

Figs. 8 and 9 show results for two situations presenting occlusion cases. The first row displays the floor plane square texture pattern and the detected object positions. These points are classified as people's ground points and are shown as red dots in the next row. Homographies are used to map the ground points back to each camera view.

The subjects of interest are the people visible on the floor plane diagram in the first row of Fig. 8. Frame 3300 in Fig. 9 shows an example of occlusion under three views. The proposed system is able to detect each individual successfully.

3.2. Tracking

Ground truth was manually created to evaluate tracking results. The position of each individual was manually annotated for 150 frames, 10 frames apart for the 1500 frames of the S7 PETS'06 sequence. Consistent labeling was associated to each person. Table 1 summarizes the results. All 22 subjects were successfully associated to one or more tracks produced by the system. Only one of the tracks does not match any subject. Ideally, one tracker should be associated to one person for the whole sequence. The proposed system produced an average of 1.32 tracks per trajectory, which corresponds to few errors during tracking. There was only one track exchange amongst all trackers for the whole sequence that took place between two near individuals, seen only by 2 cameras, in occlusion and aligned to the cameras baseline.

Fig. 10 shows the root mean square deviation between the estimated trajectories and the ground truth positions for each subject.

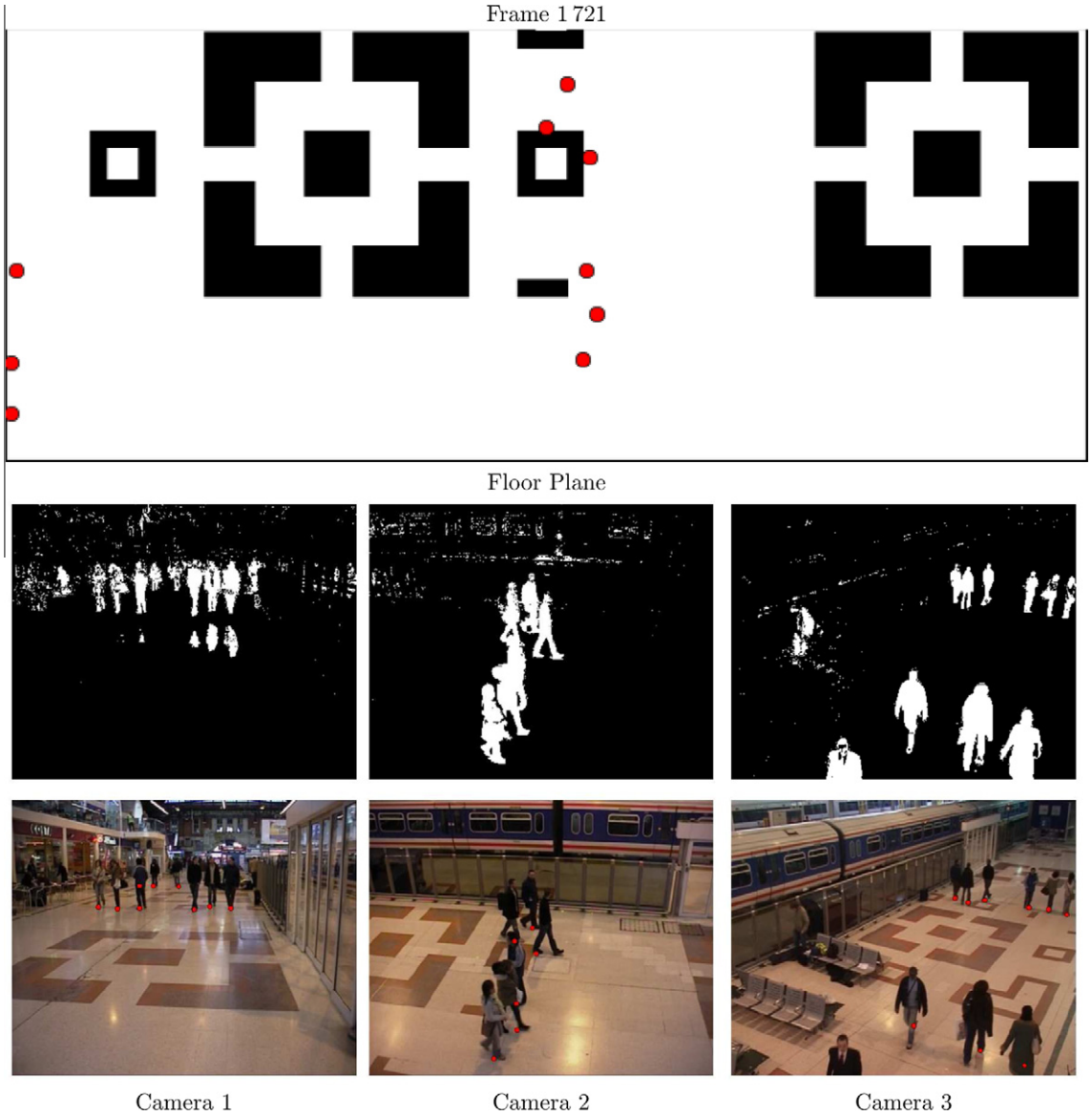

Fig. 8. Local maxima correspond to location of people on the reference ground plane (marked with dots). The homographies H_q are used to map the people's ground points back to each camera view.

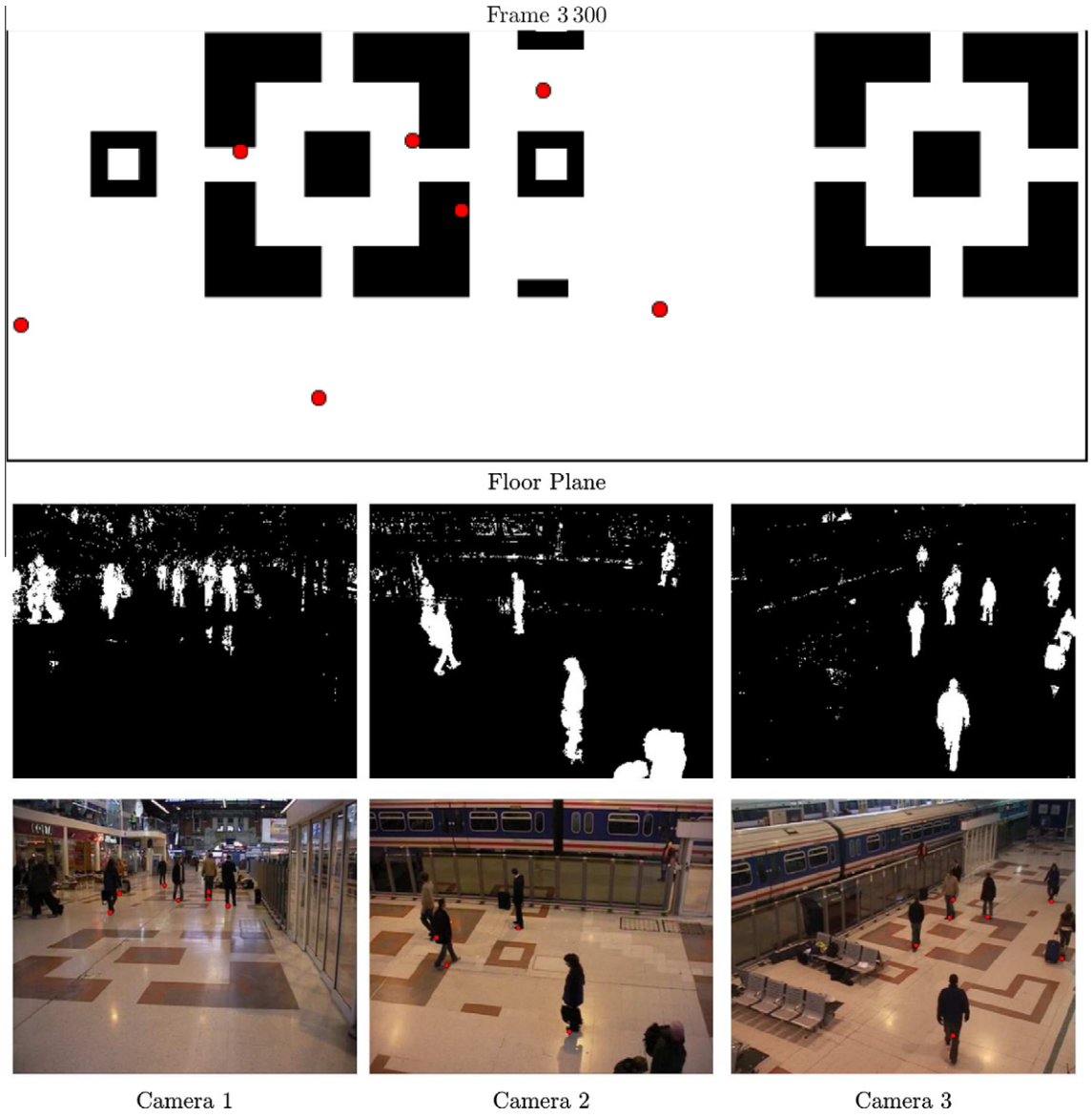

Fig. 9. Another example from PETS'06 dataset. Frame 3300 presents occlusion in all camera views but the system could accurately find the right people location.

Table 1
Tracks found by the tracking procedure compared to ground truth people trajectories.

	PETS 2006 S07
Number of trajectories	22
Found tracks	30
Trajectory recall	100.00%
Trajectory precision	96.67%
Tracks per trajectory	1.3182

The largest deviation was about 50 cm and its associated to a running man in the video sequence (subject 14). Fig. 11 displays the estimated and ground truth trajectory for subject 19. This subject crosses the entire hall and is occluded by other people several times.

3.3. Limitations

Although the good results, the proposed method presents limitations. First, the method assumes the ground is a flat surface, what could not be true in some challenging situations (a field or a

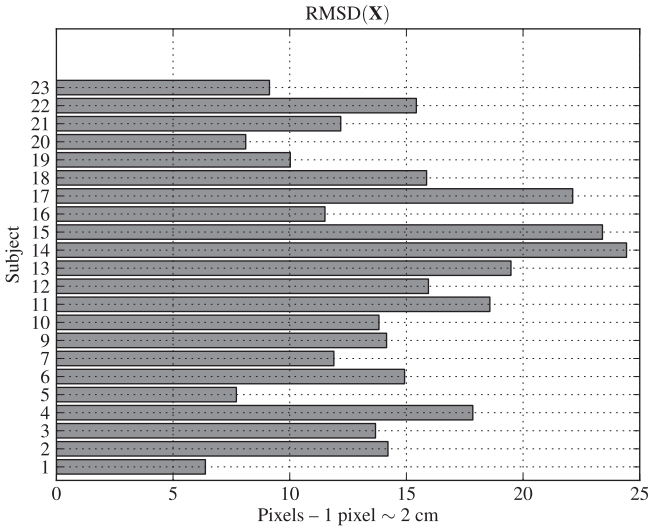

Fig. 10. Root mean square deviation for PETS 2006 S07 sequence.

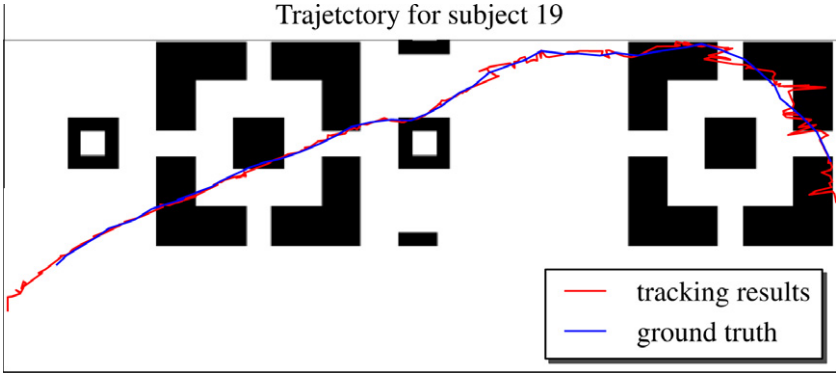

Fig. 11. Trajectory for subject 19. The subject was occluded several times along the trajectory. There are foreground misdetection at some points, caused by color similarity between his clothes and the background. The baseline between cameras 1 and 3 is marked as a dashed gray line.

meadow, for example). Moving cameras pose the hard issue of homography estimation for each new view, what could be impracticable in some situations. Finally, a dense crowd could produce a high number of false positives caused by the challenging occlusions in all camera views, although the use of a greater number of cameras presenting appropriated views of the environment could yet produce useful results.

4. Conclusions

A novel method to locate people on the ground plane using multiple camera views was presented. The main advantage of the method is that it does not require initial people segmentation or tracking. The robustness of the method is due to the accumulation of support from all cameras. The support of a candidate object location is defined as the amount of foreground pixels above that location. Therefore, pixels that correspond to ground points have more support. The support is normalized to compensate for perspective effects and accumulated on the reference plane for all camera views. The detection of people on the reference plane becomes a search for regions of local maxima in the accumulator. The paper also introduces a filtering algorithm that eliminates many false positives by checking the consistency of the location against the remaining objects for all camera views. The remaining candidates are tracked using Kalman filters and appearance models. Challenging sequences from PETS 2006 were used to test the system and show its robustness to severe occlusion situations using just 3 sparse cameras. Ground truth data also confirms the tracking accuracy of the method.

Future work includes further experimentation in crowded scenarios and trajectory analysis for event detection. Another topic for future investigation is the comparison to state-of-the-art methods and benchmarking that should be made by the use of new public data and annotation sets, as the PETS 2009 dataset.

Acknowledgments

Thiago T. Santos acknowledges support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES – Grant No. BEX 2686/06). Thiago T. Santos, Carlos H. Morimoto acknowledge financial support from Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

References

Bouman, C.A., 1997. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from: <http://www.ece.purdue.edu/~bouman> (April 1997).

Cheng, Y., 1995. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Machine Intell. 17 (8), 790–799. doi:10.1109/34.400568.

Criminisi, A., Reid, I.D., Zisserman, A., 2000. Single view metrology. Internat. J. Comput. Vision 40 (2), 123–148.

Eshel, R., Moses, Y., 2008. Homography based multiple camera detection and tracking of people in a dense crowd. In: Proc. 2008 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008), Los Alamitos, CA, USA. pp. 1–8. doi:10.1109/CVPR.2008.4587539.

Fleuret, F., Berclaz, J., Lengagne, R., Fua, P., 2008. Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. Pattern Anal. Machine Intell. 30 (2), 267–282. doi:10.1109/TPAMI.2007.117.

Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Cambridge University Press.

Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S., 2006. Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans. Pattern Anal. Machine Intell. 28 (4), 663–671. doi:10.1109/TPAMI.2006.8.

Khan, S., Shah, M., 2009. Tracking multiple occluding people by localizing on multiple scene planes. IEEE Trans. Pattern Anal. Machine Intell. 31 (3), 505–519. doi:10.1109/TPAMI.2008.102.

Kim, K., Davis, L., 2006. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: Proc. 9th Eur. Conf. on Computer Vision (ECCV'06), Graz, Austria, Vol. 3953. pp. 98–109. doi:10.1007/11744078_8.

Santos, T.T., Morimoto, C.H., 2008. People detection under occlusion in multiple camera views. In: Proc. XXI Braz. Symp. on Computer Graphics and Image Processing (SIBGRAPI'08). IEEE Computer Society, Los Alamitos, pp. 53–60. doi:10.1109/SIBGRAPI.2008.25.

Senior, A., Hampapur, A., Tian, Y.-L., Browna, L., Pankantia, S., Bolle, R., 2006. Appearance models for occlusion handling. Image Vision Comput. 24 (11), 1233–1243.

Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking. In: Proc. 1999 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99), Los Alamitos, CA, USA, Vol. 2. pp. 246–252.

Thirde, D., Li, L., Ferryman, J., 2006. Overview of the PETS2006 challenge. In: Proc. 9th IEEE Internat. Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006), New York, USA. pp. 47–50.

Wang, H., Suter, D., 2005. A re-evaluation of mixture of gaussian background modeling. In: Proc. 30th IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2005), Vol. 2. pp. 1017–1020.