People detection under occlusion in multiple camera views

Thiago T. Santos and Carlos H. Morimoto
Institute of Mathematics and Statistics
University of São Paulo
Rua do Matão, 1010 - 05508-090 São Paulo, Brazil
{thsant, hitoshi}@ime.usp.br

Abstract

This paper proposes a method to locate people on a reference plane using multiple cameras. Previous works rely on people trajectories and color models to solve occlusion. This new approach solves people detection under occlusion by accumulating evidence from multiple views instantaneously and does not rely on previous segmentation of individuals in foreground data or any tracking information.

First, foreground data from one view, segmented using background subtraction, is projected onto the ground plane or reference image. The projected foreground of a second view overlaps the first projected foreground only on the points where the foreground intersects the ground plane. Thus, by accumulating the evidence from multiple views, people can be located by detecting local maxima on the accumulated reference image. Experimental results using publicly available data from PETS'06 [9] show that the method robustly locates people in very challenging situations with occlusion in most of the views. The locations on the ground plane can further be used for segmentation and tracking on each camera view under severe occlusion.

1. Introduction

In recent years, the use of multiple cameras for people segmentation and tracking have gained more attention. Multiple cameras are useful to recover 3D space information from the scene and solve occlusion in crowded environments. One of the key aspects of multiple camera surveillance is how to define the correspondence between objects found in each camera view. To match people across multiple cameras, Hu *et al.* [5] represents each person by her principal axis. Their system relies on the fact that the intersection of the principal axis of a person in a view and transformed principal axis of this person in another view corresponds to the "ground-point" of this person in the first view. The likelihood between two axes from different views is computed comparing their intersection with a predicted ground-point. 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-objects occlusion can degrade the axis location performance.

Kim and Davis [6] combined the principal-axes crossing idea to a particle filtering framework for people tracking. First, a set of particles (ground points) is draw from filter dynamics. Then these points are integrated to appearance models [7] of each individual to perform people segmentation in each camera view. Once foreground pixels are segmented and classified (each person is a class), the principal-axes are computed and, using correspondence derived from classification, new ground points are estimated by axes intersection, refining localization. The algorithm follows updating the samples set, according to an observation equation. The drawback is that their system requires that individuals appear initially as isolated foreground blobs to proper modeling.

Chang and Gong [2] defined a Bayesian belief network to match subjects between consecutive frames (tracking) and between multiple camera views at the same time. This system needs to perform segmentation of heads to define the network probabilities. Thus occlusion is still an issue.

Previous methods for segmentation of groups of people in individuals are affected by two main problems. First, partial and total occlusion are common. In places such as airport halls or train stations, people frequently walk together 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 [5].

The present method does not rely on single view segmentation of the subjects, neither on tracking or color models. Multiple view perspective geometry is applied to collect evidence of people presence on the ground plane. Multiple camera foreground information is then integrated, lo-

1530-1834/08 $25.00 © 2008 IEEE
DOI 10.1109/SIBGRAPI.2008.25

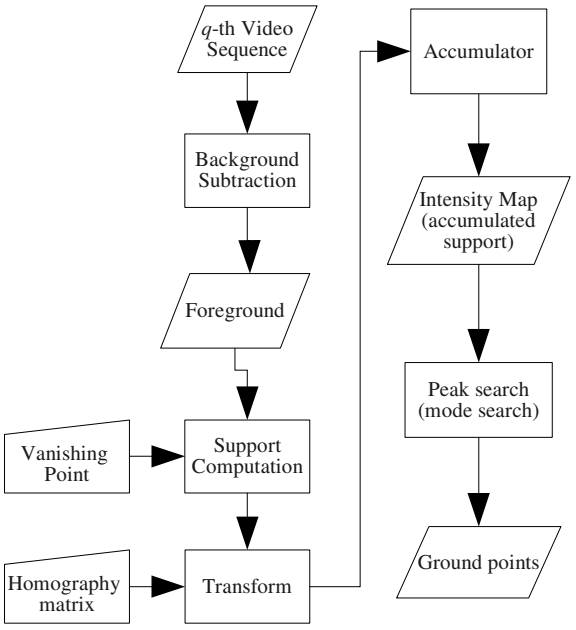


Figure 1. Overview of the proposed system.

cating the people on the ground plane, solving occlusion and defining people correspondence between camera views. Thus people position and segmentation are obtained simultaneously on a reference image, where occlusion is naturally resolved and many camera views can be easily integrated.

The rest of this paper is organized as follows. The method is described in Section 2. Some results obtained using the PETS'06 video sequences are shown in Section 3. Finally, Section 4 presents some conclusions and points for further work.

2. The method

Figure 1 gives an overview of the method. Each static camera feeds a background subtraction module. A mixture of Gaussians models the background color distribution for each pixel. The segmented foreground is used to compute evidence of people presence for each pixel on the reference image π (floor plane). A linear algorithm computes the amount of "foreground mass" relying in each position in the floor plane. Perspective is carefully considered to accurately detect objects near and far away from the cameras. Homographies transform each camera view to the reference plane, where multi-view evidence is integrated. Finally, ground points (people location on the ground plane) are computed on regions that obey some size (height and area) constraints.

2.1. Background subtraction

The color distribution for each background pixel in time is modeled as a mixture of Gaussian distributions [8]. This Gaussians mixture approach is able to deal with multiple modes on the background color distribution probability.

A pixel $\mathbf{x}$ presents color $f(\mathbf{x})$, 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 [10].

A pixel's distribution is modeled by K Gaussians. The k-th Gaussian presents mean vector $\mu_k = \langle \mu_k^r, \mu_k^g, \mu_k^I \rangle$, covariance matrix Σ_k and a *weight* w_k, the probability this pixel has subclass k. An expectation-maximization (EM) algorithm combined to an agglomerative clustering strategy [1] is applied to estimate K and the mixture parameters for each pixel. Because the training set is not free of moving objects, the *background distribution* is represented by the Gaussians whose weight w_k is greater than a threshold T_w.

Each pixel $\mathbf{x}_i$ is compared against all subclasses in the background mixture model. The probability of $\mathbf{x}_i$ be in the k-th subclass is

$$p_k(\mathbf{x}_i) = \frac{1}{(2\pi)^{3/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(f(\mathbf{x}_i)-\mu_k)^t \Sigma_k^{-1}(f(\mathbf{x}_i)-\mu_k)}.$$

If there is a subclass k where $p_k(\mathbf{x}_i) \geq 0.5$, $\mathbf{x}_i$ is classified as background.

Shadows are a common source of background misdetection. We use an additional test, based on Wang and Suter [10] work, to perform some shadow removal. Let $f^I(\cdot)$ denotes the pixel's intensity component in f. If $\mathbf{x}_i$ chromaticity fits r and g models and

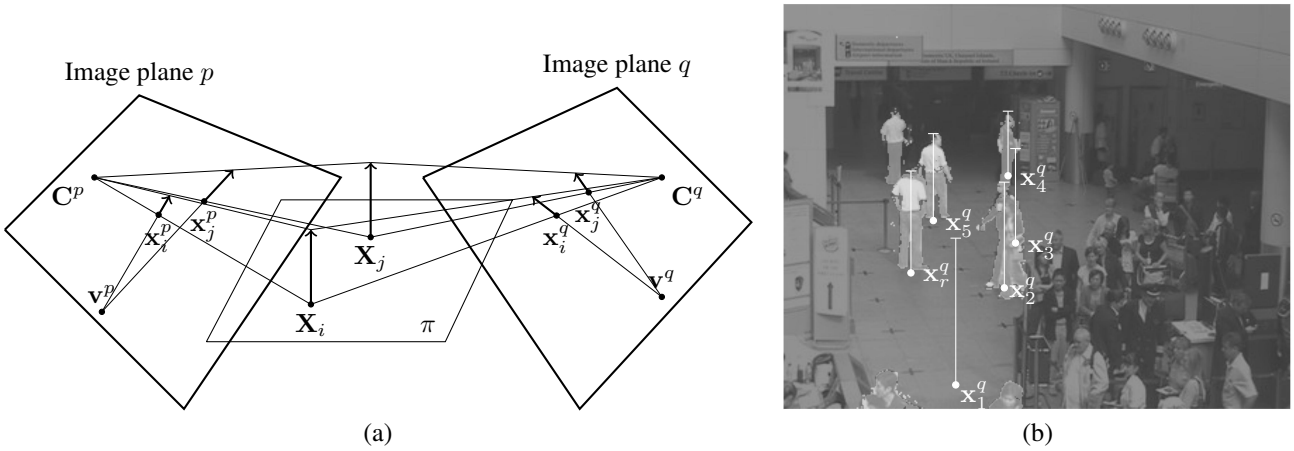
$$T_{\text{shadow}} \leq \frac{f_I(\mathbf{x}_i)}{\mu_k^I} \leq 1.0,$$

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

2.2. Support computation

Let $\mathbf{x}^q = (x, y)$ be a pixel on the image plane of camera q and consider that $\mathbf{x}^q$ is a *ground point*, *i.e.*, a point on the ground plane where an object is located. The *support* of $\mathbf{x}^q$ is given by the amount of foreground pixels standing above it, off the ground plane.

Under perspective projection, vertical parallel lines in space map onto a pencil of lines intersecting at a common point in the image plane, the *vanishing point* $\mathbf{v}^q$ [3, 4], as shown in Figure 2 (a). The pixels on the line defined by $\mathbf{x}^q$ and $\mathbf{v}^q$ correspond to the set of pixels that stand over $\mathbf{x}^q$.



(a) (b)

Figure 2. (a) Perspective transform for two cameras p and q with projection centers $\mathbf{C}^p$ and $\mathbf{C}^q$ and vanishing points $\mathbf{v}^p$ and $\mathbf{v}^q$. (b) Perspective correction and height filtering. The bright areas correspond to foreground. Bars correspond to the height of the person standing on $\mathbf{x}_r^q$.

The support is determined by the number of foreground pixels on this line.

Simple pixel counting to determine support is not accurate due to perspective effects. Figure 2 (b) shows six bars of different pixel lengths. All of them correspond to the *same* height h of the person standing at $\mathbf{x}_r^q$ but at different locations $\mathbf{x}_i^q$. Using h_r as reference height and using some affine points [3], it is possible to define, for each pixel $\mathbf{x}_i^q$, the length in pixels that corresponds to h_r, assuming that $\mathbf{x}_i^q$ is a point on the ground plane. This length will be used as a normalization factor for support.

In the image plane of camera q, consider $\mathbf{x}_r^q$, the ground point of a reference object with height h_r. Let $\hat{\mathbf{x}}_r^q$ be the projection of $\mathbf{x}_r^q$ on a parallel plane h_r units far from the floor plane, as shown in Figure 3. Let d denote the distance in pixels between two points and assume we know $d(\mathbf{x}_r^q, \hat{\mathbf{x}}_r^q)$. We are looking for the distance $d(\mathbf{x}_i^q, \hat{\mathbf{x}}_i^q)$, where the reference object is placed on a different location $\mathbf{x}_i^q$.

Criminisi *et al.* [3] applied the cross-ratio to find the relation

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

between the reference height h_r and the camera height h_q, the distance from the camera center to the floor plane. The points $\mathbf{c}_r^q$ and $\mathbf{c}_i^q$ are the projections of $\mathbf{x}_r^q$ and $\mathbf{x}_i^q$ onto ground plane vanishing line $\mathbf{l}$ (see Figure 3).

The same ratio can be found using

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

We are interested in the distance $\eta(\mathbf{x}_i^q) = d(\mathbf{x}_i^q, \hat{\mathbf{x}}_i^q)$, but $\hat{\mathbf{x}}_i^q$ is unknown. Let $\alpha(\mathbf{x}_i^q) = d(\mathbf{x}_i^q, \mathbf{v}^q)$ and $\beta(\mathbf{x}_i^q) = d(\mathbf{x}_i^q, \mathbf{c}_i^q)$. The terms on $\hat{\mathbf{x}}_i^q$ can be rewritten as

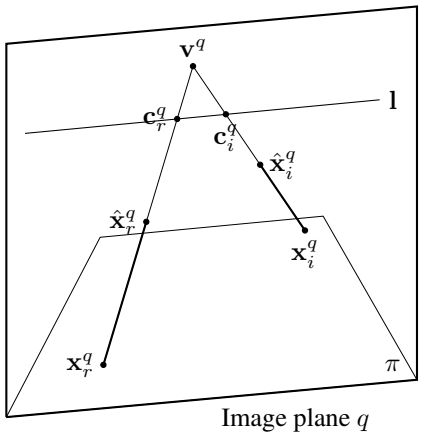


Figure 3. Single view geometry: $\mathbf{l}$ is the ground plane vanishing line and $\mathbf{v}^q$ the vertical vanishing point. It is possible to predict the objects's length in pixels when placed in any position $\mathbf{x}_i^q$ in π [3, 4].

$$d(\hat{\mathbf{x}}_i^q, \mathbf{v}^q) = \alpha(\mathbf{x}_i^q) - \eta(\mathbf{x}_i^q) \tag{3}$$

$$d(\hat{\mathbf{x}}_i^q, \mathbf{c}_i^q) = \beta(\mathbf{x}_i^q) - \eta(\mathbf{x}_i^q). \tag{4}$$

Let

$$\gamma = \frac{d(\hat{\mathbf{x}}_r^q, \mathbf{c}_r^q)\, d(\mathbf{x}_r^q, \mathbf{v}^q)}{d(\mathbf{x}_r^q, \mathbf{c}_r^q)\, d(\hat{\mathbf{x}}_r^q, \mathbf{v}^q)}. \tag{5}$$

Using the equality between Equations 1 and 2 we have, af-

ter simple algebraic manipulation:

$$\eta(\mathbf{x}_i^q) = \frac{\alpha(\mathbf{x}_i^q)\beta(\mathbf{x}_i^q)(1-\gamma)}{\alpha(\mathbf{x}_i^q) - \beta(\mathbf{x}_i^q)\gamma}. \quad (6)$$

The value of $\eta(\mathbf{x}_i^q)$ can be pre-computed for each $\mathbf{x}_i^q$ and used as a normalization factor in further support computation.

We will filter objects by their height. As the system is looking for people, an appropriated range $[h_{\min}, h_{\max}]$ is adopted (picking the height of a reference individual, for example). A person cannot present support below the minimum height $h_{\min}$ or beyond a maximum $h_{\max}$. Figure 2 (b) illustrates the idea. Bright areas mark the foreground segmented from image q. The bar directions are defined by the ground points $\mathbf{x}_i^q$ and the vanishing point $\mathbf{v}^q$. The bar lengths in pixels correspond to $h_{\max}$. The support of $\mathbf{x}_i^q$ is the amount of foreground pixels along its corresponding bar. Observe that the point $\mathbf{x}_1^q$ does not present any support and that $\mathbf{x}_2^q$, $\mathbf{x}_3^q$, $\mathbf{x}_4^q$ and $\mathbf{x}_5^q$ present similar support. This illustrates another useful feature in adopting $h_{\max}$: an arbitrary large height value would make $\mathbf{x}_2^q$ present much more support than the other points. That is not the case, the scene does not contain very tall single objects but a line of 3 people under occlusion. Many points will present high support values because a single foreground image cannot present accurate locations for objects under occlusion ($\mathbf{x}_3^q$ is not an object ground point, for example). In the next step, support from multiple camera views are combined to compute people locations.

The support $S_q(\mathbf{x}_i^q)$ can be computed efficiently for all pixels $\mathbf{x}_i^q$ in a line passing through $\mathbf{v}^q$ (i.e., a line orthogonal to the ground plane π). Let $\mathbf{s} = \langle \mathbf{x}_1^q, ..., \mathbf{x}_n^q \rangle$ be the line segment obtained by constraining the line by the image frame, as seen in Figure 4. Algorithm 1 computes the support by counting the number of foreground pixels projecting onto $\mathbf{x}_i^q$ and using η to get the support value in reference units. The maximum support is constrained avoiding an objects extending beyond $h_{\max}$.

Suppose we are processing $\mathbf{x}_{280}^q$ and found 240 foreground pixels counted at this point ($F[280] = 240$ – see Figure 4). Using $\eta(\mathbf{x}_{280}^q)$ and $h_{\max}$, we verify that, if $\mathbf{x}_{280}^q$ was a ground point, the tallest object would cover no more than 120 pixels, reaching pixel $\mathbf{x}_{160}^q$ at most (Line 9 of the algorithm). Inspecting $\mathbf{x}_{160}^q$, we found 140 foreground pixels observed at that point ($F[160] = 140$). So, there are 100 foreground pixels between $\mathbf{x}_{160}^q$ and $\mathbf{x}_{280}^q$. These foreground pixels are the evidence about object presence at $\mathbf{x}_{280}^q$. As η is a function of $\mathbf{x}_i^q$, j is different for each point, what justifies the procedure in Lines 9–14.

Background classification error obviously influence the correct computation of an object's support. For example, when people are dressed using color matching 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.

Algorithm 1 Support algorithm. It computes the support $S_q(\mathbf{x}_i^q)$ for all points $\mathbf{x}_i^q$ in segment $\mathbf{s}$.

```
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 then
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 then
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
```

Figure 5 shows support results for three different cameras. The figure shows support peaks near people's feet, as expected. But in the first row, false foreground detection caused by shadow produces high support values in regions of the ground plane presenting no objects. Although shadow misdetection can become an issue in single view processing, multiple view integration is able to minimize this problem.

2.3. Multiple camera views integration

To discriminate each person location, the support from other cameras have to be integrated. Only true ground points will present high support in multiple camera views. For example, in Figure 2 (b), a false ground point $\mathbf{x}_3^k$ has high support but it is unlikely that the same occurs in another camera. In fact, this will only happen when the point belongs to the baseline of two cameras.

The homography matrix $\mathtt{H}_q$ maps ground points $\mathbf{x}_i^q$ in image plane q to ground points $\mathbf{x}_i$ on the ground plane π to:

$$\mathbf{x}_i = \mathtt{H}_q \mathbf{x}_i^q. \quad (7)$$

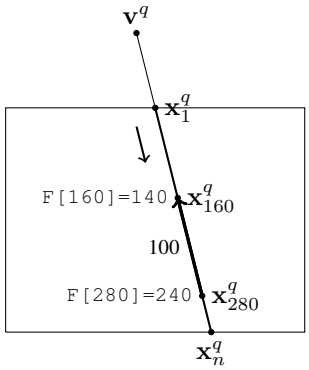


Figure 4. An example of iteration for Algorithm 1 ($i = 280$). Line 9 make us to check pixel $\mathbf{x}_{160}$ – unoccluded even for the tallest object. There are 100 pixels between $\mathbf{x}_{160}$ and $\mathbf{x}_{280}$. Refer to text for discussion.

Using a set of points on the image plane and a set of corresponding points in π, $\mathtt{H}_q$ can be estimated by the direct linear transformation algorithm (DLT) [4].

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

$$A(\mathbf{x}_i) = \sum_{q=1}^{Q} S_q(\mathtt{H}_q^{-1}\mathbf{x}_i). \quad (8)$$

where A is the *accumulator* image (Figure 6). Objects can be located by finding regions on A presenting high support.

A threshold T_S is used to select points $\mathbf{x}_i \in \pi$ presenting the minimum accepted amount of support. To set this threshold, we use the h_{min} and the number of cameras able to cover the point.

A smoothed version of A is used for thresholding. Smoothing will integrate the support information in a neighborhood of $\mathbf{x}_i$. After thresholding, we get a set of *blobs*, sets of connected points presenting high support. Morphological filters (closing and opening) are used to remove small blobs. The kernel used has size w, forcing objects to cover a minimum area on the plane for detection. To represent the object position as a single point, the local maxima on the blob is selected.

3. Results

The system was tested using the S7 dataset from the PETS 2006 Benchmark Data [9]. This dataset presents video recorded at Victoria Station in London, UK. Video from three cameras was used, showing that few cameras are enough to produce good location results.

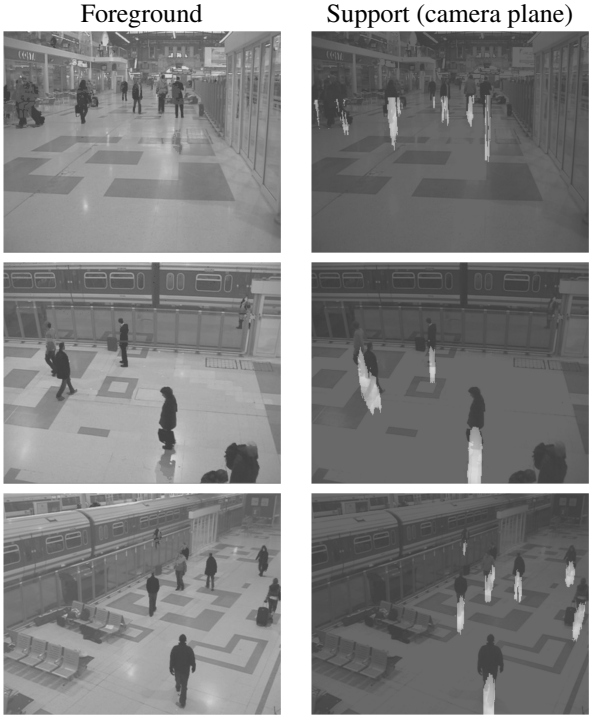


Figure 5. Support computation: the foreground images (a) are input for the support algorithm. The support (b) peaks near people feet.

Manual calibration was used to compute the vanishing points of each camera and the appropriate homography matrix to the ground plane π. The height of a selected individual was used to define the reference height unit. The allowed height range was set to $[0.7, 1.0]$ units (that is 70% to 100% of the reference man's height). The morphological kernel used in the experiments has size $w = 7$, ensuring that people are about 40 cm apart from each other.

Figures 7 and 8 shows the results for three situations presenting occlusion cases. The first row displays the floor plane schema and the detected objects positions. These points are classified as people's ground points and are shown as white 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 Figure 7. Frame 3300 in Figure 8 shows an example of occlusion under three views. The proposed system is able to detect each individual successfully.

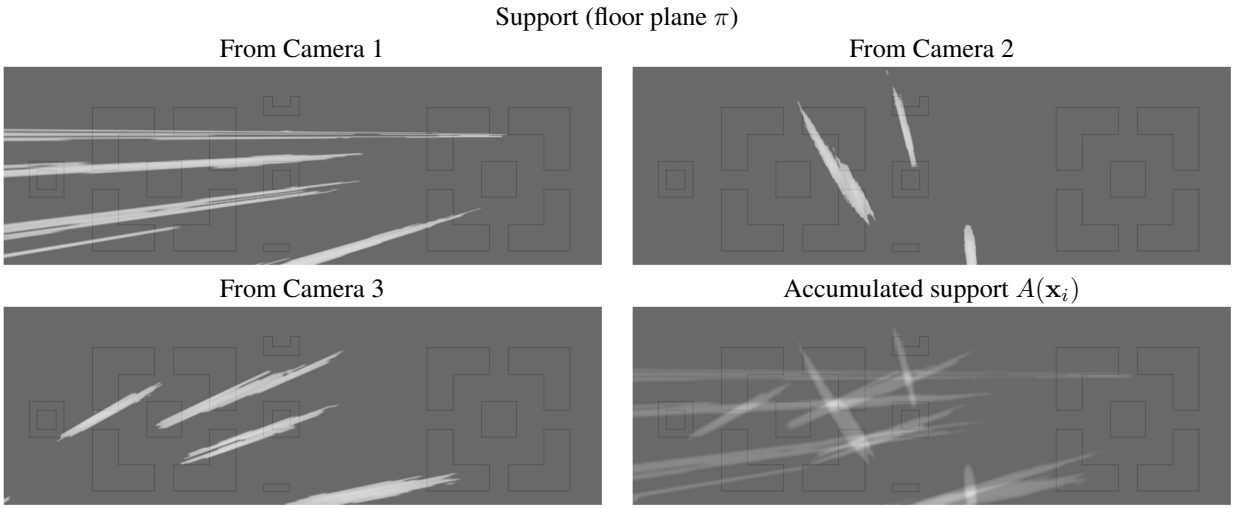


Figure 6. Multi-view integration. Homographies are used to warp support from the original camera view to the floor plane π. On true ground points, objects locations on π, the accumulated support $A(\mathbf{x}_i)$ peaks.

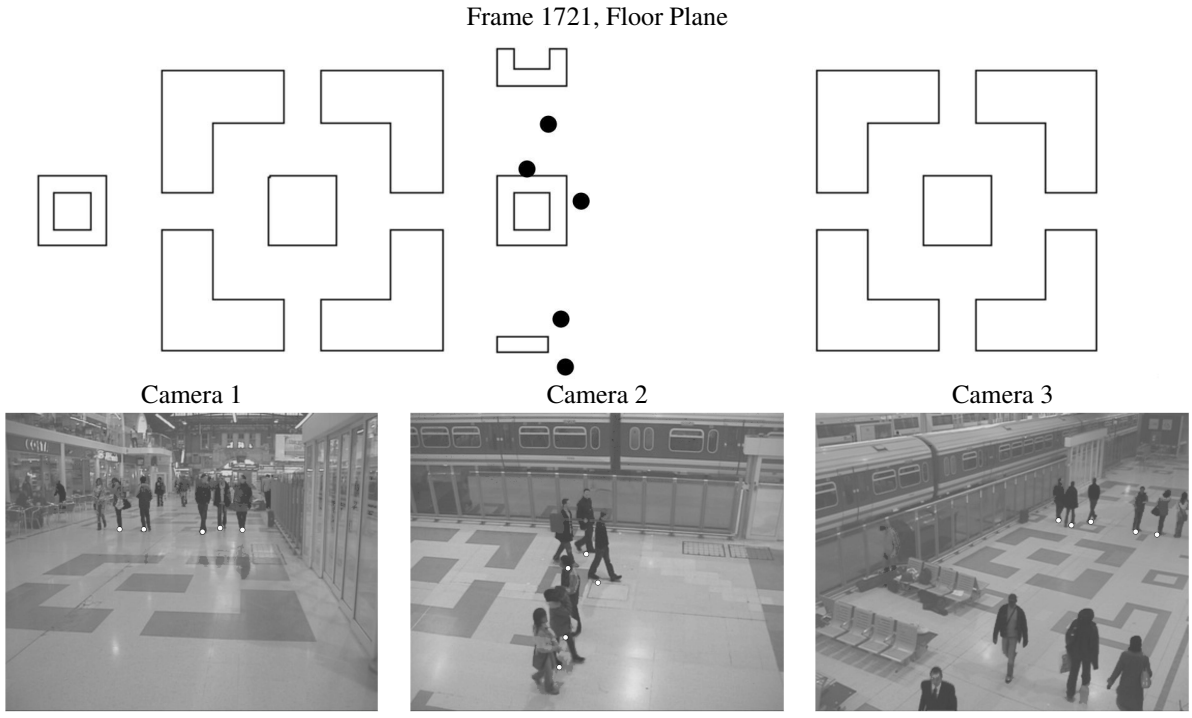


Figure 7. Local maxima corresponds to people's location on the reference ground plane (marked with white dots). The homographies $\mathtt{H}_q$ are used to map the people's ground points back to each camera view.

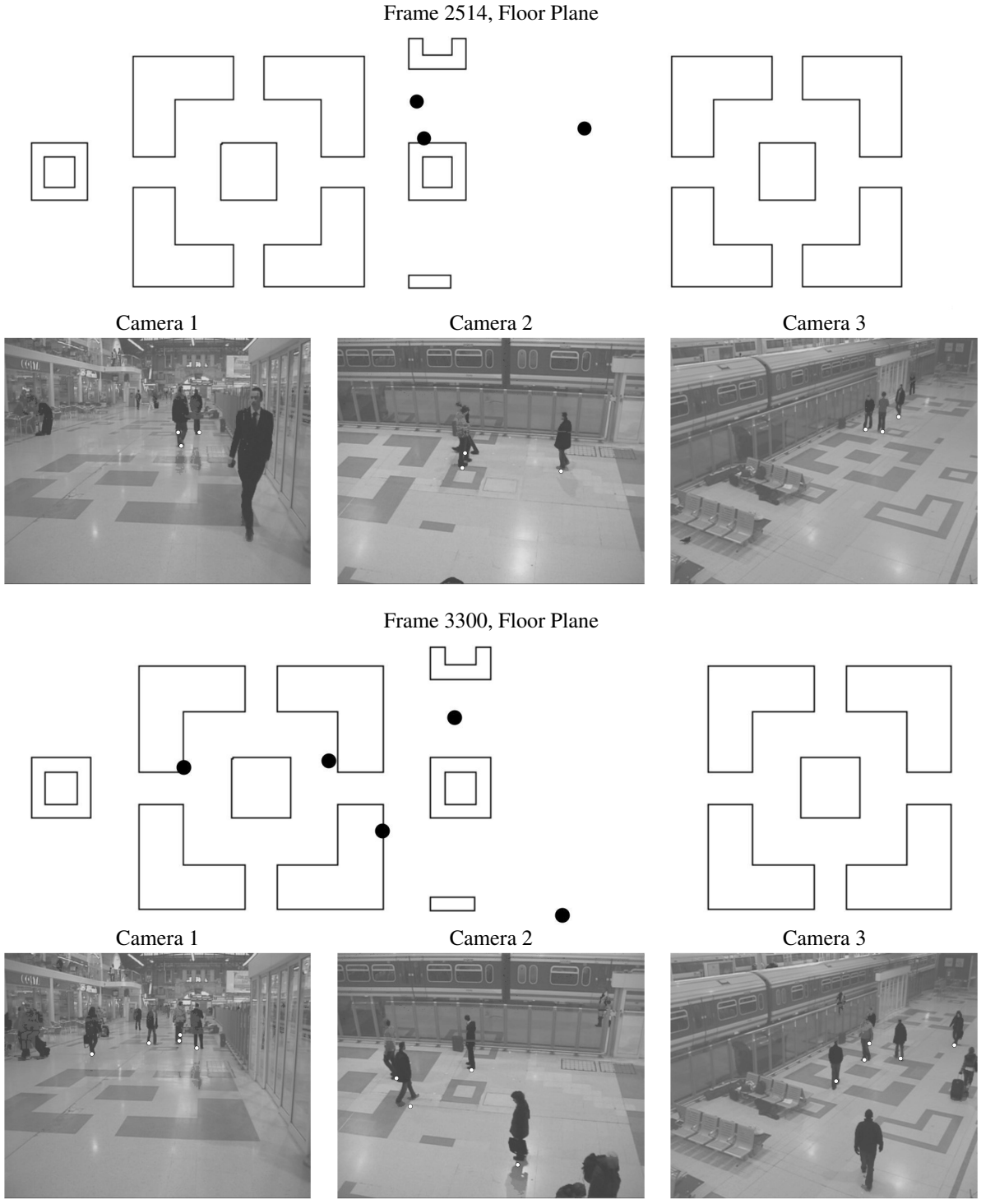


Figure 8. Another two examples from PETS'06 dataset. Note frame 3300 presents occlusion in all camera views but the system could accurately find the right people location.

4. Conclusions

This work presented a new method to reliably locate people on the ground plane using multiple camera views, solving hard occlusions cases. The method does not require initial people segmentation or tracking, relying only on multi camera geometric properties. For each view, the evidence of a person on a certain location is accumulated on the ground plane using a homography. A reference height is used to represent support in the same unit for all ground pixels, making direct comparison and global thresholding possible. Experimental results using challenging data from PETS 2006 shows the robustness of the method. The system is also robust to large background subtraction errors, although severe global illumination changes and camouflage against background can degenerate the results.

Further work will use the ground points as observations in a tracking module, starting the targets positions, updating the movement models and computing individual trajectories. The ground points will be also used to refine background subtraction and to help people segmentation in each view.

Acknowledgements

T. T. Santos acknowledges support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES – grant BEX 2686/06). T. T. Santos and C. H. Morimoto acknowledge financial support from Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP). The authors would like to thank the reviewers for their valuable suggestions and William Schwartz and Celina M. Takemura for the careful reading of the manuscript.

References

[1] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/~bouman, April 1997.

[2] T.-H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *Proceedings of 2001 IEEE Workshop on Multi-Object Tracking*, pages 19–26, July 2001.

[3] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004.

[5] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):663–671, 2006.

[6] K. Kim and L. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *Proceedings of 9th European Conference on Computer Vision (ECCV'06)*, volume 3953, pages 98–109, Graz, Austria, 2006.

[7] A. Senior, A. Hampapur, Y.-L. Tian, L. Browna, S. Pankantia, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24(11):1233–1243, November 2006.

[8] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of 1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, volume 2, pages 246–252, Los Alamitos, CA, USA, 1999.

[9] D. Thirde, L. Li, and J. Ferryman. Overview of the PETS2006 challenge. In *Proceedings of 9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)*, pages 47–50, New York, USA, June 2006.

[10] H. Wang and D. Suter. A re-evaluation of mixture of gaussian background modeling. In *Proceedings of 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, volume 2, pages 1017–1020, 2005.