A Multiview Approach to Tracking People in Crowded Scenes using a Planar Homography Constraint

Saad M. Khan and Mubarak Shah

University of Central Florida, USA

Abstract. Occlusion and lack of visibility in dense crowded scenes make it very difficult to track individual people correctly and consistently. This problem is particularly hard to tackle in single camera systems. We present a multi-view approach to tracking people in crowded scenes where people may be partially or completely occluding each other. Our approach is to use multiple views in synergy so that information from all views is combined to detect objects. To achieve this we present a novel planar homography constraint to resolve occlusions and robustly determine locations on the ground plane corresponding to the feet of the people. To find tracks we obtain feet regions over a window of frames and stack them creating a space time volume. Feet regions belonging to the same person form contiguous spatio-temporal regions that are clustered using a graph cuts segmentation approach. Each cluster is the track of a person and a slice in time of this cluster gives the tracked location. Experimental results are shown in scenes of dense crowds where severe occlusions are quite common. The algorithm is able to accurately track people in all views maintaining correct correspondences across views. Our algorithm is ideally suited for conditions when occlusions between people would seriously hamper tracking performance or if there simply are not enough features to distinguish between different people.

1 Introduction

Tracking multiple people accurately in dense crowded scenes is a challenging task primarily due to occlusion between people. If a person is visually isolated (i.e. neither occluded nor occluding another person in the scene) it is much simpler to perform the tasks of detection and tracking. This is because the physical attributes of the person's foreground blob like color distribution, shape and orientation remain largely unchanged as he/she moves. With increasing density of objects in the scene inter object occlusions increase. A foreground blob is no longer guaranteed to belong to a single person and may in fact belong to several people in the scene. Even worse, a person might be completely occluded by other people. Under such conditions of limited visibility and clutter it might be impossible to detect and track multiple people using only a single view. The logical step is to try and use multiple views of the same scene in an effort to recover information that might be missing in a particular view. In this paper

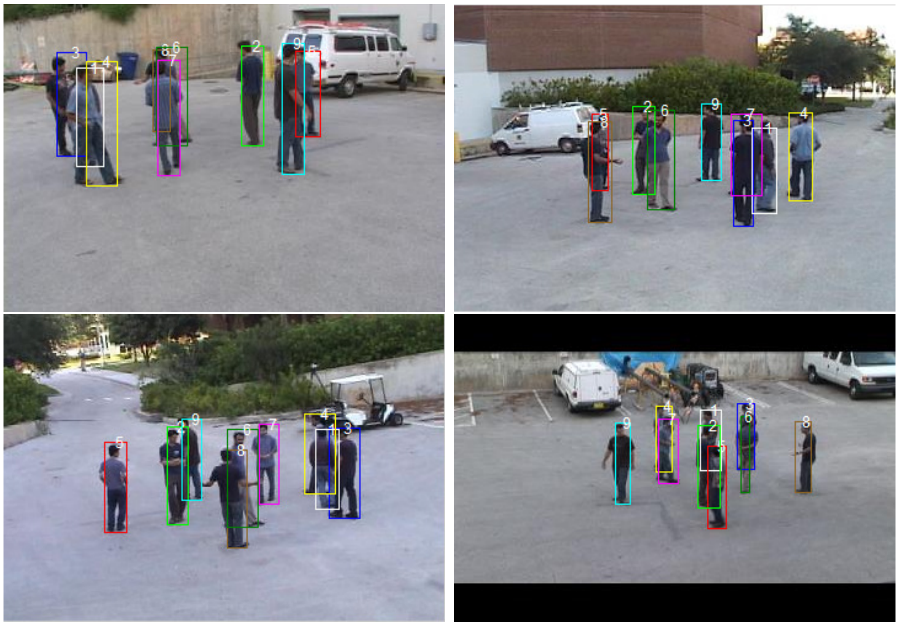


Fig. 1. Four views of a scene containing a crowd of nine people. The ground plane is clear and visible from each view. Notice the occlusions. The scene is so crowded that no person is visually isolated in every view. In fact most people are either occluded or occluding other people in every view. There are also cases of near total occlusion in views on the top row.

we propose a multi-view approach to detecting and tracking multiple people in crowded scenes. We are interested in situations where the crowds are sufficiently dense that partial or total occlusions are very common and it can not be guaranteed that any of the people will be visually isolated. Figure 1 shows four views of a crowded scene from one of our experiments that will be used to illustrate our method. Notice that no single person is viewed in isolation in all four images and there are cases of near total occlusion.

In our approach we do not use color models or shape cues of individual people. Our method of detection and occlusion resolution is based on geometrical constructs and only requires the distinction of foreground from background. At the core of our method is a novel planar homography constraint that combines foreground likelihood information (probability of a pixel in the image belonging to the foreground) from different views to resolve occlusions and determine ground plane locations of people. The homography constraint implies that only pixels corresponding to the ground plane locations of people (i.e, the feet) will consistently warp (under homographies of the ground plane), to foreground regions in every view. The reason we use foreground likelihood maps instead of binary foreground images is to delay the thresholding step to the last possible stage. Warping foreground likelihood maps from all views onto a reference view and multiplying them out, the pixels pertaining to feet of the people are segmented out. To track these regions we obtain feet blobs over a window of frames and stack them together creating a space time volume. Feet regions belonging to the same person form contiguous spatio-temporal regions that are clustered using a graph cuts segmentation approach. Each cluster is the track of a person and a slice in time of this cluster gives the tracked location.

It should be noted that we neither detect nor track objects from any single camera, or camera pair; rather evidence is gathered from all the cameras into a synergistic framework and detection and tracking results are propagated back to each view. We assume the ground plane homography between cameras is available which requires that the ground plane is visible in each view. This is a reasonable assumption in typical surveillance installations monitoring people in busy crowded places. Usually the ground plane occupies a large enough image region to be automatically detected and aligned using robust methods of locking onto the dominant planar motion (e.g via one of the 2D parametric estimation techniques such as [1, 2]). We do not assume that the camera calibration information is known.

The rest of the paper is structured as follows. In Section 2 we discuss related work. Section 3 details the observation and theory behind the homography constraint. In section 4 we present our algorithm that uses the homography constraint to segment out pixels representing ground locations of people in the scene. Section 5 describes our tracking methodology. Section 6 details our experiments and results providing insight into the utility and efficiency of our method. We conclude this paper in section 7.

2 Related Work

There is extensive literature on single-camera detection and tracking algorithms, almost all of which suffer from the difficulties of tracking multiple objects under occlusions. Zhao and Nevatia [3] presented a method for tracking multiple people in a single camera. They used 3D shape models of people that were projected back in image space to aid in segmentation and resolving occlusions. Each human hypothesis was then tracked in 3D with a Kalman filter using the objects appearance constrained by its shape. Okuma et al. [4] propose an interesting combination of Adaboost for object detection and particle filters for multiple-object tracking. The combination of the two approaches leads to fewer failures than either one on its own, as well as addressing both detection and consistent track formation in the same framework. Leibe et al. [5] present a pedestrian detection algorithm for crowded scenes. Their method operates in a top-down fashion, iteratively aggregating local and global patterns for better segmentation. These and other similar algorithms [6, 7, 8] are challenged by occluding and partially occluding objects, as well as appearance changes. Connected foreground regions may not necessarily correspond to one object, but might have parts from several of them.

Some researchers have developed multi-camera detection and tracking algorithms in order to overcome these limitations. Orwell et al. [9] present a tracking algorithm to track multiple objects in multiple views using ‘color’ tracking. They model the connected blobs obtained from background subtraction using color histogram techniques and use them to match and track objects. Cai and Aggarwal [10] extend a single-camera tracking system by starting with tracking in a single camera view and switching to another camera when the system predicts that the current camera will no longer have a good view of the subject. Krumm et al. [11] use stereo cameras and combine information from multiple stereo cameras in 3D space. They perform background subtraction and then detect human-shaped blobs in 3D space. Color histograms are created for each person and are used to identify and track people. Mittal et al. [12] use a similar method to combine information in pairs of stereo cameras. Regions in different views are compared with each other and back-projection in 3D space is done in a manner that yields 3D points guaranteed to lie inside the objects.

Even though these methods attempt to resolve occlusions, the underlying problem of using features that might be corrupted due to occlusions remains. The scene shown in figure 1 would be difficult to resolve for any of these methods. Not only are there cases of near total occlusion, the people are dressed in very similar colors. Using blob shapes or color distributions for region matching across cameras would lead to incorrect segmentations and detections.

The homography constraint we present in this paper and its application to localize people on a ground plane can also be interpreted as a visual hull intersection process. The difference is that unlike traditional visual hull intersection algorithms [13, 14, 15], our method uses only 2D constructs and dose not require camera calibration. This is because the homography constraint effectively performs visual hull intersection on a plane.

3 Homography Constraint

We begin with the basic notions of planar homographies. Let $p = (x, y, 1)$ denote the image location (in homogeneous coordinates) of a 3D scene point in one view and let $p' = (x', y', 1)$ be its coordinates in another view. Let H denote the homography of the plane Π between the two views and H_3 be the third row of H. When the first image is warped toward the second image using the homography H, then the point p will move to p_w in the *warped image*:[1]

$$p_w = (x_w, y_w, 1) = \frac{Hp}{H_3 p}.$$

For 3D points on the plane Π, $p_w = p'$. For 3D points off Π, $p_w \neq p'$. The misalignment $p_w - p'$ is called the plane parallax. Geometrically speaking warping pixel p from the first image to the second using the homography H amounts to projecting a ray from the camera center through pixel p and extending it till it

[1] For the remainder of this paper we will use only Hp to denote this operation

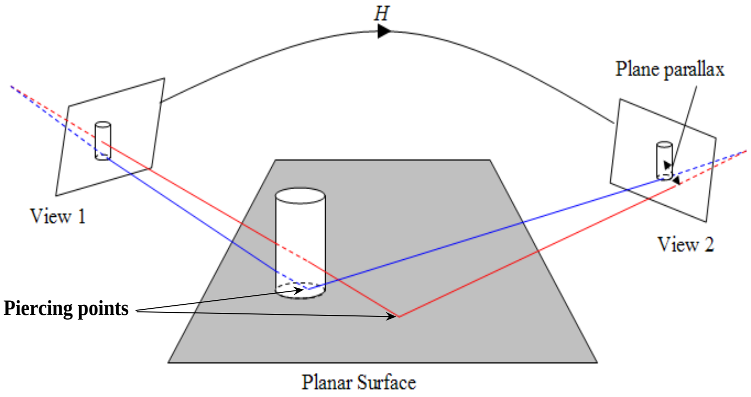


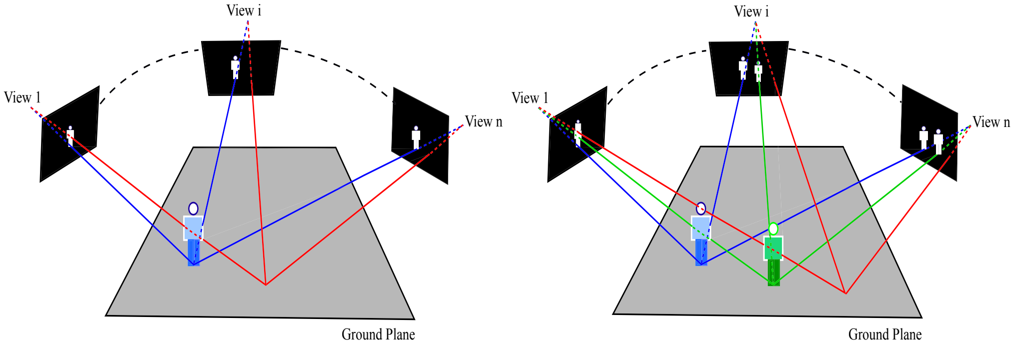
Fig. 2. The figure shows a cylinderical object standing on top a planar surface. The scene is being viewed by two cameras. H is the homography of the planar surface from view 1 to view 2. Warping a pixel from view 1 with H amounts to projecting a ray on to the plane at the piercing point and extending it to the second camera. Pixels that are image locations of scene points off the plane have plane parallax when warped. This can be observed for the red ray in the figure.

intersects the plane Π at the point often referred to as the 'piercing point' of pixel p with respect to plane Π. The ray is then projected from the piercing point onto the second camera. The point in the image plane of the second camera that the ray intersects is p_w. In effect p_w is where the image of the piercing point is formed in the second camera. As can be seen in figure 2, 3D points on the plane Π have no plane-parallax while those off the plane have considerable plane-parallax.

Suppose a scene containing a ground plane is being viewed by a set of wide-baseline stationary cameras. The background models in each view are available and when an object appears in the scene it can be detected as foreground in each view using background difference. Any 3D point lying inside the foreground object in the scene will be projected to a foreground pixel in every view. The same is the case for 3D points inside the object that lie on the ground plane, except however that the projected image locations in each view will be related by homographies of the ground plane. Now we can state the following proposition:

Proposition 1 If $\exists P \in \mathbf{R}^3$ such that it lies on plane Π and is inside the volume of a foreground object then, the image projections of the scene point P given by $p_1, p_2, \ldots, p_n$ in any n views satisfy both of the following:

- $\forall_i$, if Ψ_i is the foreground region in view i then, $p_i \in \Psi_i$,
- $\forall_{i,j} p_i = H_{i,j} p_j$, where $H_{i,j}$ is the homography of plane Π from view j to view i.



(a) (b)

Fig. 3. The figure shows people viewed by a set of cameras. The views show the foreground detected in each view. For figure (a) the blue ray shows how the pixels that satisfy the homography constraint warp correctly to foreground in each view, while others have plane parallax and warp to background. Figure (b) demonstrates how occlusion is resolved in view 1. Foreground pixels that belong to the blue person but are occluding the feet region of the green person satisfy the homography constraint (the green ray). This creates seemingly a see through effect where the feet of the occluded person can be detected.

As discussed earlier warping a pixel from one image to another using a homography of the ground plane amounts to projecting a ray through the pixel onto the piercing point and then projecting it to the second camera center. If the ray projected through a pixel in a view intersects the ground plane inside a foreground object in the scene, it follows from proposition 1 that the pixel will warp to foreground regions in all views. This can be formally stated as follows:

Proposition 2 Let Φ be the set of all pixels in a reference view and let H_i be the homography of plane Π in the scene from the reference view to view i. If $\exists p \in \Phi$ such that the piercing point of p with respect to Π lies inside the volume of a foreground object in the scene then $\forall_i p'_i \in \Psi_i$, where $p'_i = H_i p$ and Ψ_i is the foreground region in view i.

We call proposition 2 the *homography constraint*. The homography constraint has the dual action of segmenting out pixels that correspond to ground plane positions of people in the scene as well as resolving occlusion. To see this consider figure 3. Figure 3a shows a scene containing a person viewed by a set of cameras. The foreground regions in each view are shown as white on black background. A pixel that is the image of the feet of the person will have a piercing point on the ground plane that is inside the volume of the person. According to the homography constraint such a pixel will be warped to foreground regions in all views. This can be seen for the pixel in view 1 of figure 3a that has a blue ray projected through it. Foreground pixels that do not satisfy the homography constraint are images of points off the ground plane. Due to plane parallax they are warped to background regions in other views. This can be seen for the pixel with the red ray projected through it. Figure 3b shows how the homography constraint would resolve occlusions. The blue person is occluding the green person in view 1. This is apparent by the merging of their foreground blobs. In such a case there will be two sets of pixels in view 1 that satisfy the homoraphy constraint. The first set will contain pixels that are image locations of blue person's feet (same as in figure 3a). The other set of pixels are those that correspond to the blue person's torso region but are occluding the feet of the green person. Even though these pixels are image locations of points off the ground plane, they have piercing points inside a foreground object which in this case happens to be the green person. This process creates a seemingly see thorough effect detecting feet regions even if they are completely occluded by other people. It is obvious that having more people between the blue and the green person will not affect the detection of the green person.

It should be noted that the homography constraint is not limited to the ground plane and depending on the application any plane in the scene could be used. In the context of localizing people the ground plane is used and finding pixels in all views that satisfy the homography constraint will give us the locations of people's feet (location on ground). In the next section we develop an operator that does exactly this.

4 Using the Homography Constraint to Locate People

Let $\Phi_1, \Phi_2, \ldots, \Phi_n$ be the images of the scene obtained from n uncalibrated cameras. Let Φ_1 be a reference image. H_i is homography of the ground plane between the reference view Φ_1 and any other view i. Using homography H_i, a pixel p in the reference image is warped to pixel p'_i in image Φ_i. Let $x_1, x_2 \ldots, x_n$ be the observations in images $\Phi_1, \Phi_2, \ldots \Phi_n$ at locations $p'_1, p'_2 \ldots, p'_n$ respectively i.e $x_i = \Phi_i(p'_i)$. Let X be the event that pixel p has a piercing point inside a foreground object (i.e. p represents the ground location of a foreground object in the scene). Given $x_1, x_2 \ldots, x_n$, we are interested in finding the probability of event X happening, i.e $P(X \mid x_1, x_2 \ldots, x_n)$.
Using Bayes law:

$$P(X \mid x_1, x_2 \ldots, x_n) \propto P(x_1, x_2 \ldots, x_n \mid X) P(X). \tag{1}$$

The first term on the right hand side of equation 1 is the likelihood of making observation $x_1, x_2 \ldots, x_n$ given event X happens. By conditional independence we can write this term as:

$$P(x_1, x_2 \ldots, x_n \mid X) = P(x_1 \mid X) \times P(x_2 \mid X) \times \ldots \times P(x_n \mid X). \tag{2}$$

Now the homography constraint states that if a pixel has a piercing point inside a foreground object then it will warp to foreground regions in every view. Therefore it follows that:

$$P(x_i \mid X) \propto L(x_i), \tag{3}$$

where $L(x_i)$ is the likelihood of observation x_i belonging to the foreground. Plugging (3) into (2) and back into (1) we get:

$$P(X \mid x_1, x_2 \ldots, x_n) \propto \prod_{i=1}^{n} L(x_i). \tag{4}$$

Pixel p is classified as image of ground location of an object to be tracked if $P(X \mid x_1, x_2 \ldots, x_n)$ given by equation 4 is above a threshold. In the case foreground objects are people, pixel p will correspond to the feet of a person in the scene. Since pixel p and its warped locations in other views $p'_1, p'_2 \ldots, p'_n$ all have the same piercing point, they all correspond to the same location on the ground plane. Therefore by finding pixel p in the reference view that satisfies the homography constraint, we have in fact, determined the image locations in all views of a particular person's feet i.e $p'_1, p'_2 \ldots, p'_n$. This strategy also implicitly resolves the issue of correspondences across views. Note that it is irrelevant which view is chosen as the reference view. The results will be equivalent if some view other than $\varPhi_1$ was chosen as the reference. In the following subsection we outline our algorithm for finding the feet locations of people in the scene.

4.1 Algorithm

Our algorithm for locating people is quite straight forward. First we obtain the foreground likelihood maps in each view. This is done by modelling the background using a mixture of gaussians [16] and finding the probability for each pixel belonging to the foreground. In the second step instead of warping every pixel in the reference image to every other view we perform the equivalent step of warping the foreground likelihood maps from all the other views on to the reference view. These warped foreground likelihood maps are then multiplied according to equation 4 to produce what we call a 'synergy map'. A threshold is then applied to the synergy map to obtain pixels in the reference view that represent ground plane locations of people in the scene. This image is warped back from the reference view to every other view to obtain ground locations of people in each view. Following are the steps in our algorithm:

1. Obtain the foreground likelihood maps $\varPsi_1, \varPsi_2 \ldots, \varPsi_n$.
2. Warp likelihood maps to a reference view using homographies of the ground plane. Warped likelihood maps are $\varPsi'_1, \varPsi'_2 \ldots, \varPsi'_n$.
3. Multiply the warped likelihood maps to obtain the synergy map: $\theta_{synergy} = \prod_i \varPsi'_i$
4. Threshold the synergy map. For all pixels p in $\theta_{synergy}$
   - if $\theta_{synergy}(p) > T$ then return 1
   - else return 0
5. Warp thresholded image to every other view.

Figure 4 shows the algorithm applied to the scene shown in figure 1. The foreground likelihood maps are warped into view 4 (bottom right image of figure 1)

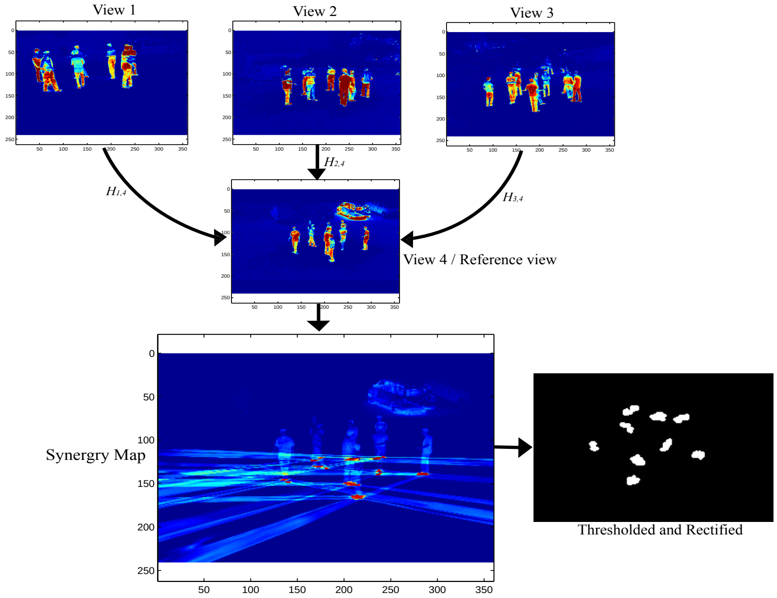


Fig. 4. The four smaller images are foreground likelihood maps obtained from the background model (mixture of gaussians) on the images shown in figure 1. In all images in the figure the colormap used assigns a hotter palette to higher values. View 4 was chosen as the reference view. The image on the bottom is the synergy map obtained by warping views 1, 2, and 3 onto view 4 and multiplying them together. The pixels representing the ground locations of the people are segmented out by applying an appropriate threshold. The binary image shown is the result of applying the threshold and rectifying with the ground plane (the white regions corresponding to the feet).

which was chosen as the reference view. Multiplying the warped views together we obtain the synergy map which clearly highlights the feet regions of all the people in the scene. The threshold T does not need to be precise as the values at the correct locations in the synergy map are typically several magnitudes higher than the rest. This is a natural consequence of the multiplication in step 3 and can be seen in figure 5. Notice how occlusions are resolved and the ground locations of people are detected. For the purpose of tracking the binary image obtained after thresholding is rectified with the ground plane. The rectified image is an accurate picture of the relative ground locations of the people in the scene.

5 Tracking

Instead of tracking in each view separately we track feet blobs only in the reference view and propagate the results to other views. This is simply because feet blobs in other views are obtained by warping feet blobs in the reference view

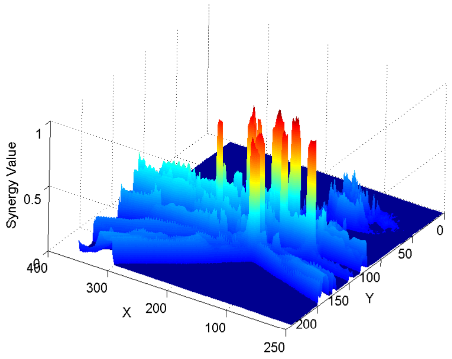
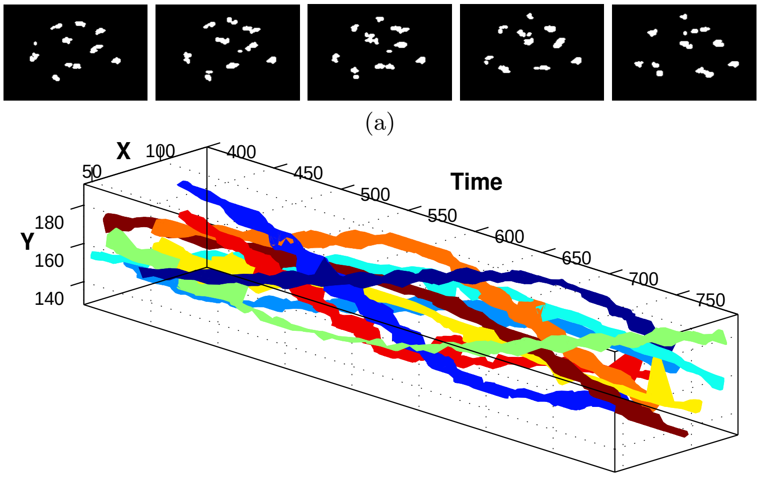


Fig. 5. A surface plot of the synergy map shown in figure 4. The peaks are the ground plane locations of the people in the scene.

and consequently the tracks can be obtained by warping as well. The only information available about the blobs is their relative ground plane positions (after rectifying with the ground). No other distinguishing feature is available. In fact a features like color could be misleading in the case of occlusions as already discussed in previous sections. Obtaining accurate tracks from these blobs is not a trivial task. The blobs represent feet of the people on the ground and the feet of a single person come close and move away every walk cycle. The result is one person's feet blob splitting and merging with itself. In fact one person's blob might temporarily merge with another person's if they come too close to each other.

Our tracking methodology is based on the observation that the feet of the same person are spatially coherent in time. That is to say that even though a person's feet might move away from each other, (as the person makes a forward step) over time they remain closer to each other than feet of other people. We therefore propose a look-ahead technique to solve the tracking problem using a sliding window over multiple frames. This information gathering over time for systems simulating the cognitive processes is supported by many researchers in both vision and psychology (e.g., [17], [18], [19]). Neisser [19] proposed a model according to which the perceptual processes continually interact with the incoming information to verify hypotheses formed on the basis of available information up to a given time instant. Marrs principle of least commitment [18] states that any inference in a cognitive process must be delayed as much as possible. Many existing algorithms use similar look-ahead strategies or information gathering over longer intervals of time (for example, by backtracking) [20, 21].

For a window of size w using the algorithm described in the previous section we obtain the blobs for all frames in the window. Stacking them together in a space time volume, blobs belonging to the same person will form spatially coherent clusters that appear like 'worms'. Figure 6b shows an example of worms formed from the sequence shown in figure 1. Each worm is in fact the track of a person's feet as he moves in time. To segment out worms belonging to different



(a)

(b)

Fig. 6. Figure (a) shows a sequence of frames with feet blobs obtained using our algorithm. Stacking them in time the feet blobs form spatially coherent 'worms' that can be seen in figure (b). Different worms clustered out are colored differently to help in visualization. The spirelling pattern of the worms is only a coincidence. This resulted because the people were walking in circles in this particular sequence.

people from this space time volume we use graph cuts to obtain the tightest clusters in this space time volume. Each blob pixel in the space time volume forms a node of a completely connected graph. Edge weights are assigned using the image distance (euclidean) between pixels connected by the edge. Using normalized cuts [22] on this graph we obtain the optimum clustering of blob pixels in the volume into worms. A slice in time of the worms give the ground plane locations of each person in the scene at that particular time. These are warped back to each view to obtain the image locations of each person's feet in different views.

6 Results and Discussions

To evaluate our approach we conducted several experiments with increasing number of people and varying the number of active cameras. The attempt was to increase the density of people till the algorithm broke down and to study the breakdown thresholds and other characteristics. Each sequence was roughly 750 frames long. The people were constrained to move in an area of approximately 5 meters by 5 meters to maintain the density of the crowd.

To negate a spot on the ground plane as a candidate location for a person, it must be visible as part of the background in atleast one view. If the spot is not occupied by a person but is occluded in each view (by the people in the scene), our algorithm will trigger a *false positive* at that spot. Note that this is not a

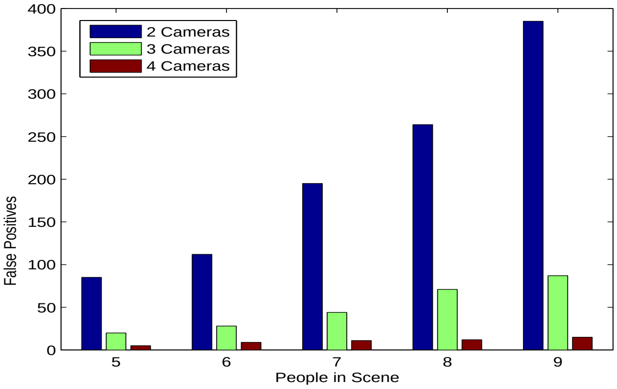


Fig. 7. A plot of false positives reported in our experiments. Each sequence was 500 frames long and contained between 5 and 9 people that were constrained to move in an area of 5x5 meters to simulate different densities of crowds. We varied the number of active cameras in different runs of the experiments to assess the effect of increasing and decreasing view points.

limitation of the homography constraint, which states that the region should project to foreground in *all* (every possible) view. Therefore by increasing the number of views of the scene we can effectively lower false positives. This trend can be observed in figure 7, that summarizes the performance of our algorithm for crowds of various densities with increasing number of cameras.

In figure 8 we show track results from the densest sequence we tested our algorithm on. Note that the tracking windows do not imply that all pixels of the people were segmented (only the pixels representing feet blobs are known). The purpose of the track windows is to aid in visualization. The width of each track window is set as the horizontal spread of the feet blobs. The height is calculated by starting from the feet pixels and moving up the connected foreground region before background is encountered. The sequence contained nine people and was captured from four different view points encircling the scene. Due to the density of the crowd occlusions were quite severe and abundant. An interesting thing to notice is the color similarity of the people in the scene. Naturally a method that uses color matching across views would perform poorly in such a situation whereas our method performs quite well.

One of the limitations of our method is its susceptibility to shadows. Currently the scheme incorporated in our method to handle shadows is to use HSV rather than RGB color space in background subtraction. This is sufficient for scenes like the one in figure 8 where the shadows are small and diffused. But with hard shadows our current implementation has increased false detections. We are working on several strategies to tackle this problem. One of the directions is to use an imaginary plane parallel to but higher than the ground plane in the homography constraint. This will cause foreground due to shadows to have plane parallax thus filtering them out.

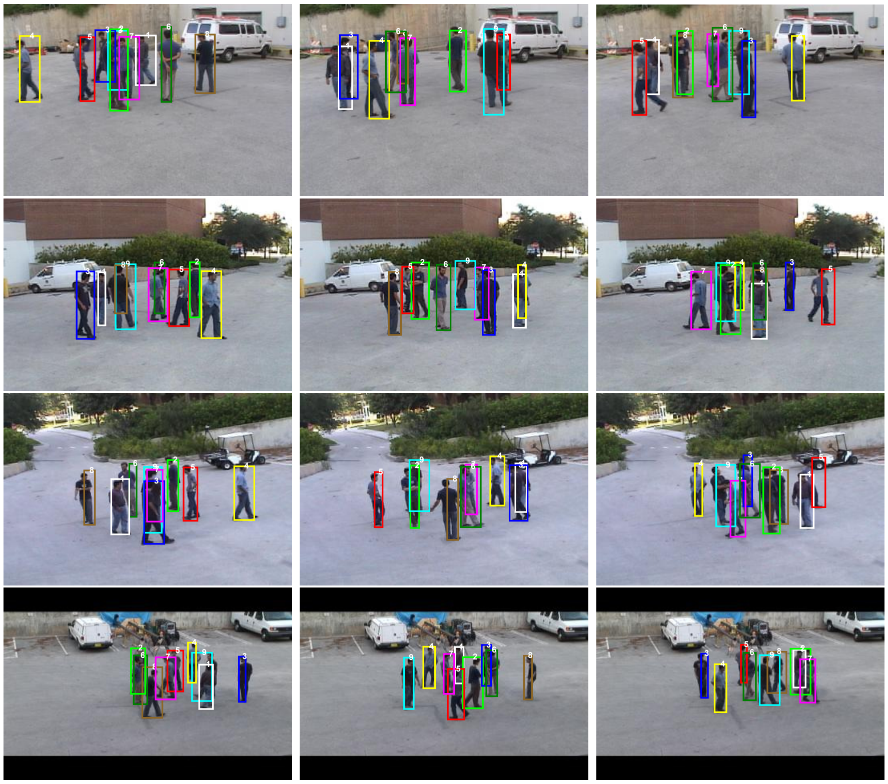


Fig. 8. Tracking results for a sequence containing 9 people captured from 4 view points. Top to bottom the rows correspond to views 1, 2, 3 and 4. Left to right the columns correspond to frames 100, 300 and 500 in the respective views. Track windows are color coded and numbered to show the correspondences that our algorithm accurately maintains across views.

7 Conclusions

In this paper we have presented a novel approach to tracking people in crowded scenes using multiple cameras. The major contribution of our work is the detection of ground plane locations of people and the resolution of occlusion using a planar homography constraint. Combining foreground likelihoods from all views into a reference view and using the homography constraint we segment out the blobs that represent the feet of the people in the scene. The feet are tracked by clustering them over time into spatially coherent worms. In the future we plan to investigate the use of multiple planes to handle shadows as well as complete segmentation of the people.

Acknowledgment: This material is based upon the work funded in part by the U.S. Government. Any opinions, findings and conclusions or recommenda-

tions expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

References

1. Irani, M., Rousso, B. and Peleg, S. 1994. Computing Occluding and Transparent Motions. IJCV, Vol. 12, No. 1.
2. Gurdjos, P. and Sturm, P. Methods and Geometry for Plane-Based Self-Calibration. CVPR, 2003.
3. Zhao, T. and Nevatia, T. 2004. Tracking Multiple Humans in Complex Situations, IEEE PAMI, 2004.
4. Okuma, K., Taleghani, A., Freitas, N., Little, J.J., and Lowe, D.G. 2004. A Boosted Particle Filter: Multitarget Detection and Tracking., ECCV 2004.
5. Leibe, B., Seemann, E., and Schiele, B. 2005. Pedestrian Detection in Crowded Scenes, CVPR 2005.
6. McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A. and Wechsler, H. 2000. Tracking Groups of People, CVIU 2000.
7. Rosales, R., and Sclaroff, S. 1999. 3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions, CVPR 1999.
8. Sidenbladh, H., Black, M.J., Fleet, D.J. 2000. Stochastic Tracking of 3D Human Figures Using 2D Image Motion, ECCV 2000.
9. Orwell, J., Massey, S., Remagnino, P., Greenhill, D., and Jones, G.A. 1999. A Multi-agent framework for visual surveillance, ICIP 1999.
10. Cai, Q. and Aggarwal, J.K. 1998. Automatic tracking of human motion in indoor scenes across multiple synchronized video streams, ICCV 1998.
11. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. 2000. Multi-camera multi-person tracking for easy living, IEEE International Workshop on Visual Surveillance.
12. Mittal, A., Larry, S.D. 2002. M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. IJCV, 2002.
13. Laurentini, A., 1994. The Visual Hull Concept for Silhouette Based Image Understanding, IEEE PAMI 1994.
14. Franco, J., Boyer, E., 2005. Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid, ICCV 2005.
15. Cheung, K.M., Kanade, T., Bouguet, J.-Y. and Holler, M., 2000. A real time system for robust 3d voxel reconstruction of human motions, CVPR 2000.
16. Stauffer, C., Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking, CVPR 1999.
17. Gibson, J.J. The Ecological Approach to Visual Perception. Boston: Houghton Mifflen 1979.
18. Marr, D. 1982. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. New York: W.H. Freeman.
19. Neisser, U. 1976. Cognition and Reality: Principles and Implications of Cognitive Psychology. San Francisco: W.H. Freeman.
20. Poore, A.B. 1995 Multidimensional Assignments and Multitarget Tracking. Proc. Partitioning Data Sets; DIMACS Workshop 1995.
21. Reid, D.B. 1979. An Algorithm for Tracking Multiple Targets. IEEE Trans. Automatic Control 1979.
22. Shi, J., Malik, J. 2000. Normalized Cuts and Image Segmentation, IEEE PAMI 2000.