A Fast Method for Tracking People with Multiple Cameras

Alparslan Yildiz and Yusuf Sinan Akgul
{yildiz, akgul}@bilmuh.gyte.edu.tr

Vision Lab. Gebze Institute of Technology
http://vision.gyte.edu.tr

Abstract. We propose a multi-camera method to track several persons using constraints from the epipolar and projective geometries. The method is very accurate, fast, and simple.
We first compute accumulator images for each time frame that shows the probability of object positions on the ground. We developed a voting based method that allows employment of the integral images to make the accumulator computation very fast. Next, we perform two-pass 3D tracking on the volume generated by stacking these accumulator images. Our main contributions are the fast computation of the accumulator images and application of fast 3D tracking methods like the Kalman Smoother instead of the computationally expensive methods like the Viterbi algorithm.
The proposed tracking method is evaluated on people videos captured using four synchronized cameras.

1 Introduction

Tracking people in crowded scenes is a challenging task in computer vision and related areas. If a person is clearly visible in a camera view, it is relatively easy to track that person. However, occlusions occur when there are several moving persons in the scene and the tracking problem becomes non-trivial. Although there are monocular solutions [12], this problem is usually addressed using multiple cameras viewing the same scene. By registering the position information from different camera views, one can generate accurate estimates for people's locations. Planar world assumption is common in multi camera object tracking systems [1], [2], [3], [4]. These systems constrain the problem space by assuming that objects mostly move on the ground plane. This constraint allows systems to use planar homographies that can conveniently map object positions between views. Planar world assumption is violated for objects that float above the ground plane, which is not common in everyday life. Another constraint that can considerably reduce the problem space would be to use epipolar geometry in object tracking, which surprisingly is not properly used in multi camera object tracking systems. Although the epipolar constraint cannot be directly used for object position mapping between views, it can be very effective if it is used with other projective geometry concepts such as vanishing points. Furthermore, epipolar geometry is always valid for all kinds of objects including the floating objects. Our system employs both the planar world assumption and the epipolar constraints to produce tracking results that are more accurate and faster to compute compared to the state-of-art tracking methods. The underlying algorithm is also a lot simpler than other methods.

The ground plane assumption has some disadvantages in addition to floating objects. When we consider people walking on the ground plane, the most significant information comes from the feet which is very unstable compared to the other human parts. Chang and Gong [5] has partially addressed this problem by tracking people on virtual planes above the ground plane. They use tracked points to find virtual planes at the head level for each person and continue to track the people on their own virtual planes. They found this method as more accurate than using the ground plane because the heads of people move more smoothly than their feet.

Khan and Shah [6] extended the single virtual plane idea to multiple parallel planes above the ground plane. For each virtual plane they compute view-to-view homographies. It is actually sufficient to find a homography between ground planes of the two views because all the other virtual plane homographies between these views are calculated depending on the ground homography. Instead of using only a single reference plane, many virtual planes increase the accuracy of tracking because the position estimates must satisfy the data from each virtual plane. Clearly, the higher the number of planes, the better the tracking results. However, using higher number of planes increases the running times dramatically.

Fleuret et al. [7], proposed a probabilistic method for tracking people from multiple views without using any virtual planes but the ground plane. However, they incorporate object location information above the ground plane using occupancy models of objects in each view. The occupancy model answers the question: "How would an object (as a silhouette) at a position be seen from a view?". They use this observation model and find an occupancy map at each time step using an Expectation-Maximization algorithm which is inherently slow due to its iterative nature. The occupancy maps have notable peaks for the most probable object locations. These maps are later given to a Hidden Markov Model (HMM) as the observations. In order to obtain acceptable tracking times, the number of possible object positions need to be kept under some threshold which lowers the resolution of object positions. This leads to a trade-off between position precision and tracking speed. Finally, the requirement of full camera calibration is another drawback of [7].

We argue that the main issue of multi-view tracking systems are their speed versus accuracy trade-off. In this paper, we propose to use a voting based method to accumulate the evidence data of the possible object positions. Voting methods usually bring advantages such as being fast, parallizable, and non-iterative. By using efficient integral image processing approaches, we increase the efficiency of voting based evidence accumulation even further, which results in considerable speed gains in object tracking. Voting based methods also allow convenient integration of constraints. We use this advantage by including epipolar and

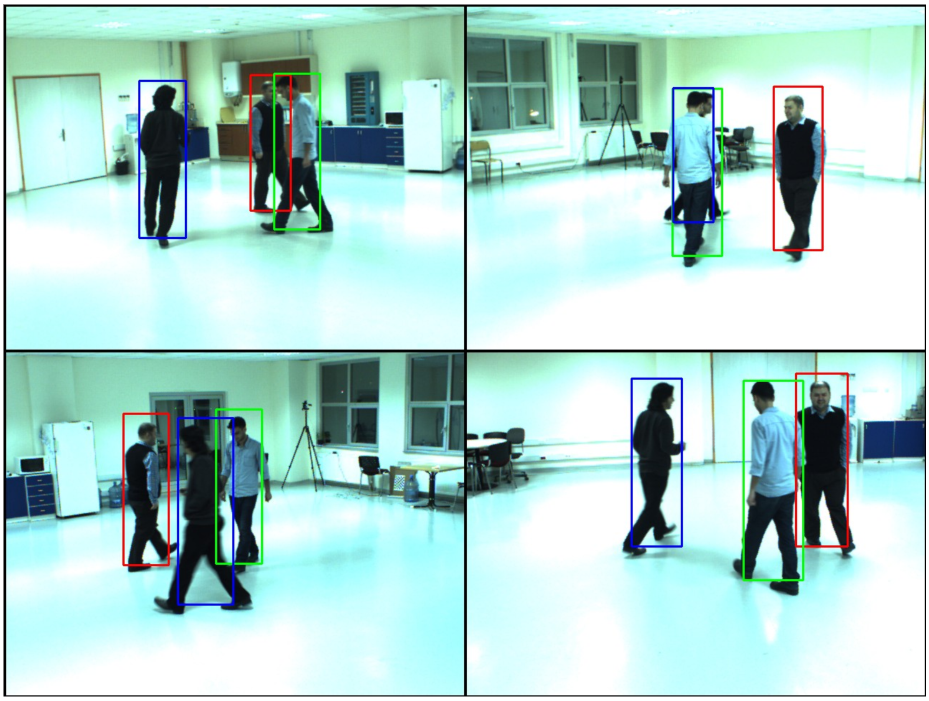


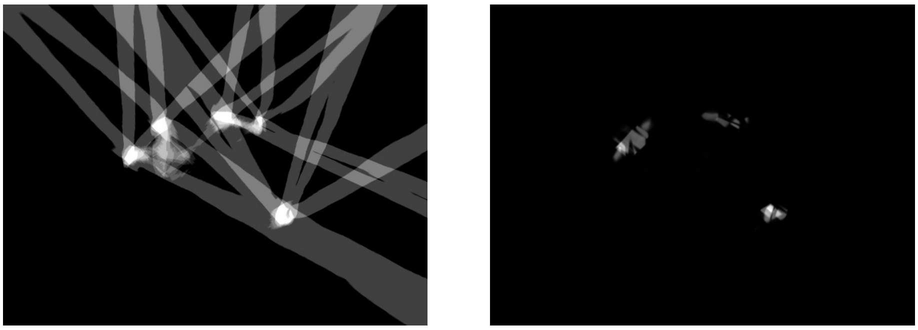
Fig. 1. Sample tracking results. The same time instant is shown from 4 differente views. A different box color is used for each person.

perspective geometry based restrictions into the voting process to increase the tracking accuracy. As a result, the overall tracking process becomes faster and more accurate.

The reminder of the paper is organized as follows; we present our multi-camera people tracking method in Section 2. We present the experimental results of our method in Section 4. Finally, we provide the concluding remarks in Section 5.

2 Multi-Camera People Tracking

Multi-camera people tracking is usually done using the ground plane assumption. According to this assumption, people in the scene move on the ground plane while touching the ground plane at a single location. This assumption allows methods to use 2D planar homographies that can conveniently map any ground plane position on a view to any other view. The homographies for the ground plane in different views are extracted easily with the help of image features [9]. Rather than tracking people in each view separately, the common practice is to transform the ground plane points in each view to a common rectified plane. This transformation is again possible with 2D homographies. Tracking on the



(a) Occupancy map using only the ground plane.

(b) Occupancy map using 7 virtual planes.

Fig. 2. Occupancy maps for object positions

rectified plane is done by summing all the information in all views together on the rectified plane. With the ground plane assumption, positions of people that move on the ground plane will theoretically correspond to the same rectified plane position. Detailed information can be found in [1], [2], and [6].

We consider the accumulation of all the information in all views into the rectified plane as a voting process on an accumulator, where foreground pixels in each view vote for a ground plane position. Object positions would correspond to peaks on this accumulator. Since the detection of binary foreground/background pixels is hard to achieve accurately, foreground likelihoods in each view are used for voting. Fig. 2(a) shows the accumulator image for the ground plane. Foreground log-likelihoods are transformed using 2D homographies onto the accumulator image and summed for all views. As Fig. 2(a) shows, peaks on the accumulator are not clearly detectable. To overcome this problem, [6] offered using additional planes above the ground plane to incorporate more information. In this method, the foreground likelihood maps for each view are transformed onto the same accumulator image using the homographies for each virtual plane. Fig. 2(b) shows the accumulator image formed using 7 virtual planes. Using more virtual planes clearly improve the results as the peaks in Fig. 2(b) are more convenient to work on than the peaks in Fig. 2(a). However, the computational cost of the voting increases linearly with the number of virtual planes used.

In the following sub sections, we give our fast voting method for accumulator computation, our object top point computation method and our localization method in detail.

2.1 Fast voting

We propose a novel and very efficient method for the formation of the accumulation image. The computational cost of our method is independent of the number of planes used and linear in the number of pixels and the number of views.

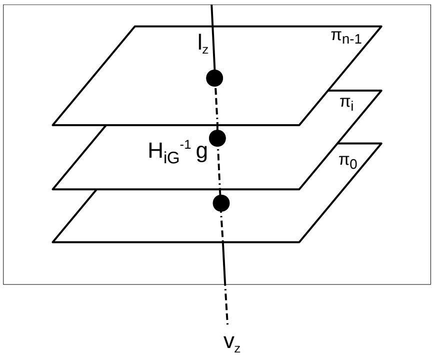


Fig. 3. Virtual planes shown for a view.

Let us visualize the voting process for multiple virtual planes as in Fig. 3. Here, n virtual planes are used and labeled as π_0, ..., π_{n-1}, where π_0 corresponds to the ground plane. $\mathbf{v}_z$ is the vanishing point of the z-direction of the ground plane.

If we consider voting only from a single view, we can write the vote for a location $\mathbf{g}$ on the accumulator G as

$$G(\mathbf{g}) = \sum_i f(\mathbf{H}_{iG}^{-1}\mathbf{g})\,, \tag{1}$$

where i iterates for all the virtual planes used, f is the foreground log-likelihood map, and $\mathbf{H}_{iG}$ is the 2D homography from the virtual plane i to the accumulator image. Such an accumulator image would correspond to the occupancy map of [6], as shown in Fig. 2(b). The locations on the foreground likelihood map, from which votes are sampled, form a straight line ($\mathbf{l}_z$ in Fig. 3) which is the projection of a 3D line L_z perpendicular to the ground plane. For the location $\mathbf{g}$ on accumulator G, sampling foreground likelihoods will come from this line for any virtual plane. Since using a higher number of virtual planes increases the accuracy, we can consider using an infinite number of virtual planes. Some of the intersections of the virtual planes and the line L_z correspond to pixel centers on views. If we call these intersections as $\mathbf{p}' \in \mathbf{l}_z$, we can rewrite Eq. 1 as

$$G(\mathbf{g}) = \sum_{\mathbf{p}' \in l_z} f(\mathbf{p}')\,, \tag{2}$$

where $\mathbf{l}_z = \mathbf{v}_z \times (\mathbf{H}_{0G}^{-1}\,\mathbf{g})$ and $\mathbf{H}_{0G}$ is the 2D homography from the ground plane to the accumulator image. We can compute Eq. 2 for one location $\mathbf{g}$ in constant time using integral images [8], if we rectify f such that $\mathbf{l}_z$ is axis-aligned with the images. This is possible with a 2D perspective transform $\mathbf{H}_z$ which sends $\mathbf{v}_z$ to the ideal point $[0\,1\,0]^T$ [11]. Applying $\mathbf{H}_z$ on the foreground likelihood maps will ensure that $\mathbf{l}_z$ is axis-aligned. The above method of forming the accumulator image uses every possible pixel value on $\mathbf{l}_z$. Thus, the method extracts information

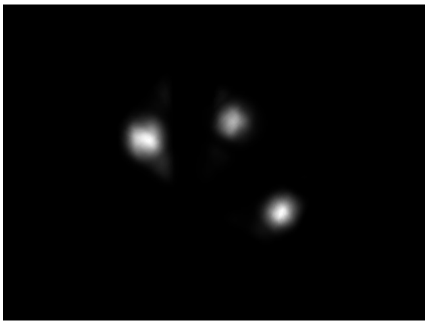


Fig. 4. Accumulator image calculated using Eq. 3.

from the image data by using a maximal number of applicable virtual planes in optimal time. The accumulator images produced by Eq. 2 would correspond to occupancy maps of [6] using an infinite number of virtual planes. However, such an occupancy map could not be computed in practical time with the method of [6].

We further constrain Eq. 2 into an interval on $\mathbf{l}_z$ that correspond to the pixels possibly belonging to an object. Pixels outside of this interval do not belong to the object, hence they should not corrupt the accumulator space. Clearly, one end of this interval is $\mathbf{H}_{0G}^{-1}\,\mathbf{g}$ which is the lowest possible object pixel on an image. The other end of the interval is the top point of an object which is directly related to the height of an object in the image. Our method for the computation of object top points for multiple views is discussed in the following subsection.

Finally, we normalize the summation in Eq. 2 as

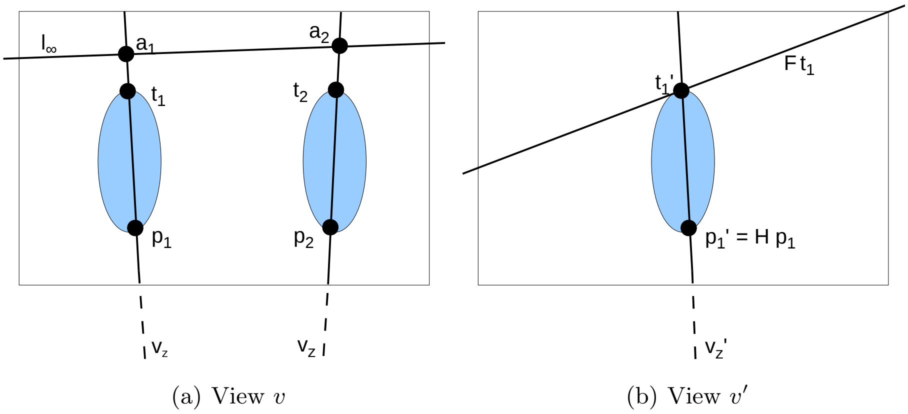
$$G(\mathbf{g}) = \frac{1}{\|\mathbf{H}_{0G}^{-1}\,\mathbf{g} - \mathbf{t}\|} \sum_{\mathbf{p}' \in [\mathbf{l}_z]} f(\mathbf{p}')\,, \tag{3}$$

where $\mathbf{t}$ is the expected object top point and $[\mathbf{l}_z]$ represents the interval on $\mathbf{l}_z$. Note that, this normalization, which also removes the perspective effects introduced by $\mathbf{H}_z$, is necessary since an object's viewed height changes with its position.

Rather than using lines, a slightly better (smoother) voting can be performed using object boxes, whose vertical symmetry axes are aligned to $\mathbf{l}_z$. Note that, it is still constant time to vote for a point $\mathbf{g}$ on G, using integral images. Fig. 4 shows the accumulator image formed using this method, for comparison with the occupancy maps in Fig. 2.

2.2 Computing object top points

Assuming an object's 3D height does not change dramatically through the video, we can find the object's 2D height in the image for a position using the projective



(a) View v

(b) View v'

Fig. 5. Computing object top points (and heights) in different views.

invariant cross-ratio. The cross-ratio is a scalar value obtained from 4 colinear points and is invariant under any projective transformation of the points [11]. As suggested by [10], the object bottom $\mathbf{p}$, the object top $\mathbf{t}$, the vanishing point $\mathbf{v}_z$ in the z-direction of the ground plane, and a point $\mathbf{a}$ (colinear with the 3 points) on the vanishing line $\mathbf{l}_\infty$ of the ground plane form a cross-ratio (see Fig. 5(a)). The vanishing point $\mathbf{v}_z$ is fixed for a given view. The point $\mathbf{a}$ for an object can be computed as $\mathbf{a} = (\mathbf{v}_z \times \mathbf{p}) \times \mathbf{l}_\infty$. So we can say that the cross-ratio of an object for a view, is defined by the bottom and top points of the object on that view. Here, the bottom point of an object corresponds to the object position.

Given a cross-ratio and 3 of the 4 points defining it, we can unambigiously compute the last point. In our case, for a given cross-ratio and an object position, we can compute the object's top point on the image, which directly gives the object height in the same view. Also, the width of an object's bounding box is simply a multiple of the object height. This multiplier is found empirically in our experiments.

It should be noted that we use a constant cross-ratio for a specific view. In our experiments, we observed that a single cross-ratio for heights of multiple objects is sufficient since the 2D height of a person does not need to be calculated pixel-precise. Accumulated probabilites for close points around the actual object position will result in a smooth peak. However, the same cross-ratio cannot be used for different views, despite the projective invariancy of cross-ratios. The reason for this is that we have all the point correspondences for the 4 points defining the cross-ratio but one, which is the point $\mathbf{a}$. We can find the same vanishing line $\mathbf{l}_\infty$ on any view, but the cross-ratio will not be the same because $\mathbf{l}_\infty$ in each view does not correspond to the same real-world height unless all cameras are at the same height from the ground plane. To overcome this problem, we use a constant cross-ratio in a reference view to compute object top points which are transformed to other views using the epipolar geometry as described below.

Given an object's top point in a source view, we use a novel method for finding the object's top point in any destination view. Let us denote the source view as v which includes the bottom and top points $\mathbf{p}$ and $\mathbf{t}$. The destination view is denoted by v' which includes the bottom and top points $\mathbf{p}'$ and $\mathbf{t}'$. Finally, the vanishing points in the z-direction of the ground plane are $\mathbf{v}_z$ and $\mathbf{v}'_z$, respectively, for views v and v'. Clearly, $\mathbf{p}' = \mathbf{Hp}$, where $\mathbf{H}$ is the ground plane homography from v to v'. As any point correspondence between two views must satisfy the epipolar constraint, it can be shown that

$$\mathbf{t}' = (\mathbf{Ft}) \times (\mathbf{v}'_z \times \mathbf{Hp}), \tag{4}$$

where $\mathbf{F}$ is the fundamental matrix from v to v', see Fig. 5.

The above method is an elegant way of finding object heights in different views without assuming explicit camera calibration. It also does not assume any specific camera configuration. Note that, Eq. 4 is our main tool to include the epipolar constraints into the voting process. Its main task is to disregard points on $\mathbf{l}_z$ that are irrelevant to object positions.

2.3 Localizing objects

A 3D volume of accumulator space is constructed by stacking accumulator images of each time frame. The peaks in each accumulator image form paths in this accumulator volume which will correspond to object trajectories (see Fig. 6). In order to find such paths, we use dynamic programming which is optimal and efficient for finding curves in 3D space. However, even dynamic programming becomes intractable for large accumulator spaces. Fleuret et al. [7] have addressed this problem using batch employment of dynamic programming (Viterbi algorithm) for 100-frame batches. We have observed in our experiments that, when

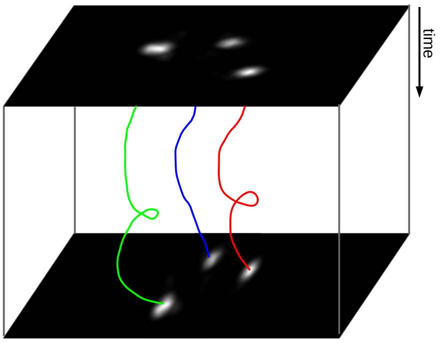


Fig. 6. Object trajectories are shown as paths in the 3D accumulator space.

objects are clearly visible in most of the views, the peaks on the accumulator spaces are also clearly visible and detectable. As a result, for time steps where objects are mostly unoccluded, we detect object positions locally on corresponding accumulator images. The single time step detection is applied to all time steps and only ambigious intervals are extracted to be solved using dynamic programming. The ambigious intervals usually consist of 8-10 time steps.

For a single time step, we have the accumulator image where peaks corresponding to object locations may or may not be clearly visible. We detect the peaks on the accumulator image using 2D k-means clustering algorithm for the object positions. For time steps where objects are unoccluded, the result of the k-means clustering has very low in-cluster-variance compared to the time steps where objects are severely occluded. Simple thresholding is found to be sufficient for determining ambigious time steps. The value of the threshold is not a major factor since we can always increase the threshold to mark more time steps as ambigious, and they would be solved using dynamic programming.

Simultaneous optimization for multiple objects is intractable even for small time intervals using dynamic programming. As the common practice in literature, we find paths for each object seperately starting with the most clearly visible object. We use the peak positions from the surrounding unambiguous time steps as the seed positions for dynamic programming. This approach would result in a continuous transition between the ambiguous time intervals and unambiguous time steps.

The data cost of the dynamic programming formulation is the accumulator image value and the smoothness cost is the Euclidian distance between the positions on consecutive time steps. If we formuate the same problem as an HMM, the data and the smoothness costs are identical to the observation and state transition probabilities, respectively.

As a final polishing, we employ a two-pass Kalman smoother to the object trajectories. This step sometimes has very good effects because some positions in the object trajectory are chosen locally on a single time step. The Kalman smoother enforces smoothness on this kind of positions.

3 Experiments

We have validated our methods with several experiments. In all our experiments we used 4 cameras, which capture 640×480 resolution images at 15 fps. We calculate the cross-ratio for the reference view based on the average height of the object blobs generated from foreground likelihood maps. For each view, we compute $\mathbf{H}_z$ (see Section 2.1) directly from the vanishing point $\mathbf{v}_z$, which is computed by detecting and intersecting vertical line segments in the images using the RANSAC method. We compute ground plane homographies and fundamental matrices using the correspondences of SIFT features. Finally, we compute the rectifying homography for the reference view $\mathbf{H}_{0G}$ by manually providing 4 scene points that roughly correspond to a square on the ground plane.

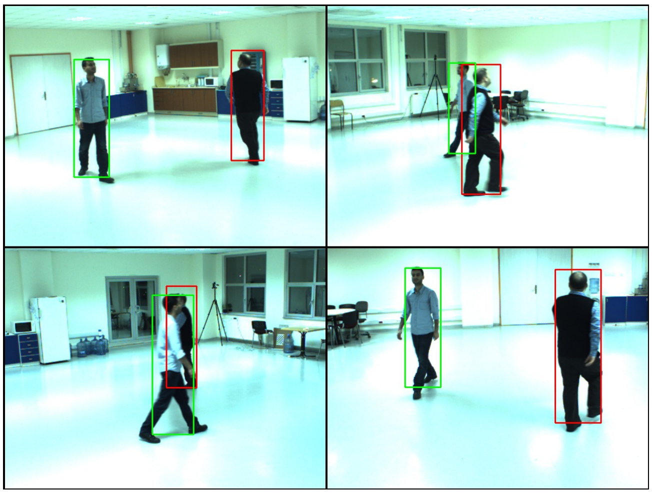


Fig. 7. Sample tracking results, see text for details.

In Figs. 1 and 7, we present sample ouputs of our tracking method. A different box color is used for each person. Experimental results show that our method succesfully recovers object positions, even under severe occlusions.

We use a very simple background subtraction to obtain the foreground likelihood maps. However, we observed that simple background segmentation is adequate for our tracking method. Using a more complicated foreground segmentation method would be a trivial extension to improve the visual results only slightly, and this would increase the overall computation time for a single time step.

Implementation of our tracking method on a 2.4Ghz Quad desktop PC spends 120msec for a time step on average, which is roughly 8 fps. This includes the time required for trajectory extraction and the smoothing steps for four cameras. Fleuret et al. [7] report a processing rate of 2 fps for similar experiments, which is significantly slower than our results.

4 Conclusions and Discussion

We presented a very effective multi-camera people tracking method using purely geometric constraints. Our tracking method accurately recovers object positions, even for the time steps where some objects are occluded in more than one view.

Major contributions of our tracking method include the employment of voting based accumulation using integral images for a very fast computation of possible object locations. Effective application of dynamic programming for position estimation for only ambigious time intervals is the other contribution of our work. We introduced a new method for calculating object heights using the epipolar geometry constraints. We do not require explicit camera calibration.

One of the drawbacks of our tracking method is that, it cannot give accurate results when ambigious time intervals are too long. This drawback is also present in the previous work on multi-camera people tracking. In such situations, there are multiple solutions to the tracking formulation and the solution the system produces may not be the correct one.

One of the main advantages of our method is being highly parallelizable because it is voting based and non-iterative unlike previously mentioned methods. Our future work includes implementation of our method on a GPU and verification of the results using a database with a ground-truth for human motion.

References

1. Kim K. and Davis L., "Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering," *In Proc. ECCV,* 2006
2. S. M. Khan and M. Shah. "A multiview approach to tracking people in crowded scenes using a planar homography constraint," *In Proc. ECCV,* 2006
3. Lee L., Romano R. and Stein G., "Monitoring activities from multiple video streams: establishing a common coordinate frame," *PAMI, IEEE Transactions on*, vol.22, no.8, pp.758-767, Aug 2000
4. Black J., Ellis T., Rosin P., "Multi view image surveillance and tracking," *Motion and Video Computing, 2002. Proceedings. Workshop on* , vol., no., pp. 169-174, 5-6 Dec. 2002
5. Chang T., Gong S., "Tracking Multiple People with a Multi-Camera System," *IEEE Workshop on Multi-Object Tracking (WOMOT'01).*
6. Khan S. Shah M., "Tracking Multiple Occluding People by Localizing on Multiple Scene Planes," *PAMI*, 2009.
7. Fleuret F., Berclaz J., Lengagne R., Fua P., "Multicamera People Tracking with a Probabilistic Occupancy Map," *PAMI*, 2008.
8. Viola P., Jones M., "Rapid Object Detection using a Boosted Cascade of Simple Features," *CVPR*, 2001.
9. Lowe D., "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
10. Crimisini A., Reid I., Zisserman A., "Single View Metrology," *ICCV*, 1999.
11. Hartley R., Zisserman A., "Multiple View Geometry in Computer Vision", Cambridge Univ. Press, 2002.
12. Andriluka M., Roth S., Schiele B., "Monocular 3D Pose Estimation and Tracking by Detection", IEEE Conference on Computer Vision and Pattern Recognition *(CVPR)*, 2010.