Unified Multi-camera Detection and Tracking Using Region-Matching

Anurag Mittal*and Larry Davis
University of Maryland
College Park, MD 20742
{anurag, lsd}@umiacs.umd.edu

Abstract

We describe an algorithm for detecting and tracking multiple people in a cluttered scene using multiple synchronized cameras located far away from each other. This camera arrangement results in multiple wide-baseline camera systems. We segment each image, and then, for each pair we compare regions across the views along epipolar lines. The centers of the matching segments are then back-projected to identify 3D points in the scene potentially corresponding to people. These 3D points are then projected onto the ground plane. The results from these wide-baseline camera systems are then combined using a scheme that rejects outliers and gives very robust estimates of the 2D locations of the people. These estimates are then used to track people across time. We have found that the algorithm works quite well in practice in scenes containing multiple people even when they occlude each other in every camera view.

1. Introduction

In this paper we address the problem of detecting and tracking multiple people using a multi-perspective video approach. In particular, we are concerned with the situation when the scene being viewed is sufficiently "crowded" that one cannot assume that any or all of the people in the scene would be visually isolated from any vantage point. Figure 1 shows eight images from a 16-perspective sequence that will be used to illustrate our algorithm. Notice that in all eight images, no single person in viewed in isolation- i.e. neither occludes another person nor is occluded by another person. We assume that our cameras are calibrated, and that people are moving on a calibrated ground plane.

We present an algorithm that takes a unified approach to detection and tracking using multiple cameras. We neither detect nor track objects from a single camera; rather information is matched across many camera pairs and hypothesized object locations and tracks are combined in a robust manner in 3D. Although we use background subtraction, we do not assume that a connected component of foreground pixels corresponds to a single object. Rather, we employ a segmentation algorithm to separate out regions and then match them across the views. This helps us to handle the case of partial occlusion and allows us to track people and objects in a cluttered scene where we cannot see any single person separated out from the others in any view.

*Corresponding author

2. Related Work

There are numerous single-camera detection and tracking algorithms, all of which face the same difficulties of tracking 3D objects using only 2D information. These algorithms are challenged by occluding and partially-occluding objects, as well as appearance changes. Some researchers have developed multi-camera detection and tracking algorithms in order to overcome these limitations.

Haritaoglu et. al. [2] have developed a system which employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso etc.) and tracks them using appearance models. In [3], they incorporate stereo information into their system. Kettnaker and Zabih [4] have developed a system for counting the number of people in a multi-camera environment where the cameras have a non-overlapping field of view. By combining visual appearance matching with mutual content constraints between cameras, their system tries to identify which observations from different cameras show the same person.

Cai and Aggarwal [5] extend a single-camera tracking system by switching between cameras, trying to always track any given person from the best possible camera - e.g. a camera in which the person is unoccluded. Orwell et. al. [7] present a tracking algorithm to track multiple objects using multiple cameras using "color" tracking. They model the connected blobs obtained from background subtraction using color histogram techniques and use them to match and track objects. In [8], Orwell et. al. present a multi-agent framework for determining whether different agents are assigned to the same object seen from different cameras. The DARPA VSAM project at CMU has developed a system for video surveillance using multiple pan/tilt/zoom cameras which classifies and tracks multiple objects in outdoor environment [9]. All these systems use background subtraction techniques in order to separate out the foreground and iden-

tify objects, and would fail for cluttered scenes with more densely located objects and significant occlusion.

Darrell et. al. [6] developed a tracking algorithm which integrates stereo, color segmentation and face pattern detection. Currently, their system is limited to only one stereo rig, and is unable to track occluded objects.

Krumm et. al. [1] present an algorithm which has goals that are very similar to ours. They use stereo cameras and combine information from multiple stereo cameras (currently only 2) in 3D space. They perform background subtraction and then detect human-shaped blobs in 3D space. Color and other information is used to identify and track people over time.

Our method can be considered to be between wide-baseline stereo algorithms, which try to match exact 3D points across the views, and volume intersection algorithms which try to find the 3D shape of an object by intersection in 3D space without regard to the intensity values observed (except for background subtraction). Wide-baseline stereo algorithms have the problem of incorrect matches due to a substantial change in the viewpoint, thus rendering traditional methods like correlation and sum of squared difference inappropriate. Although some work has been done to improve upon these methods, they are still not very robust due to the fundamental difficulty of matching points seen from very different viewpoints.

On the other hand, volume intersection is very sensitive to background subtraction errors, so that errors in segmenting even one of the views can seriously degrade the recovered volume. Although there has been some work recently (for e.g. [14]) addressing some of these issues, these methods also have problems, especially in the case where the objects are occluded in some views by other objects. Back-projection in 3D space without regard to color also yields very poor results in cluttered scenes, where almost all of the camera view is occupied by the foreground.

In contrast, we do not try to match points exactly across views; neither do we perform volume intersection without regard to the objects seen. Rather, 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.

3 General Overview of the Algorithm

We first run a background subtraction algorithm on each of the camera views, and then, apply an image segmentation algorithm to the foreground regions. The segmentation algorithm differentiates between different objects even though they might occur in the same connected component as found by the background subtraction algorithm, but, of course oversegments the component into many pieces. We next match regions along epipolar lines in pairs of cameras

Figure 1: Eight images from a 16-perspective sequence at a particular time instant.

views. The mid-points of the matched segments along the epipolar lines of each stereo pair are back-projected to yield 3D points, which are then projected onto the ground plane. These ground points are then used to form an object location probability distribution map using gaussian kernels for a single image pair. The probability distribution maps are then combined using outlier-rejection techniques to yield a robust estimate of the 2D position of the objects, which is then used to track them. The following sections describe these steps in detail.

4 Background Subtraction and Segmentation in a Single View

We first separate out the foreground from the background. This is done using a robust technique which is capable of removing shadows. The method that we use is described in [13].

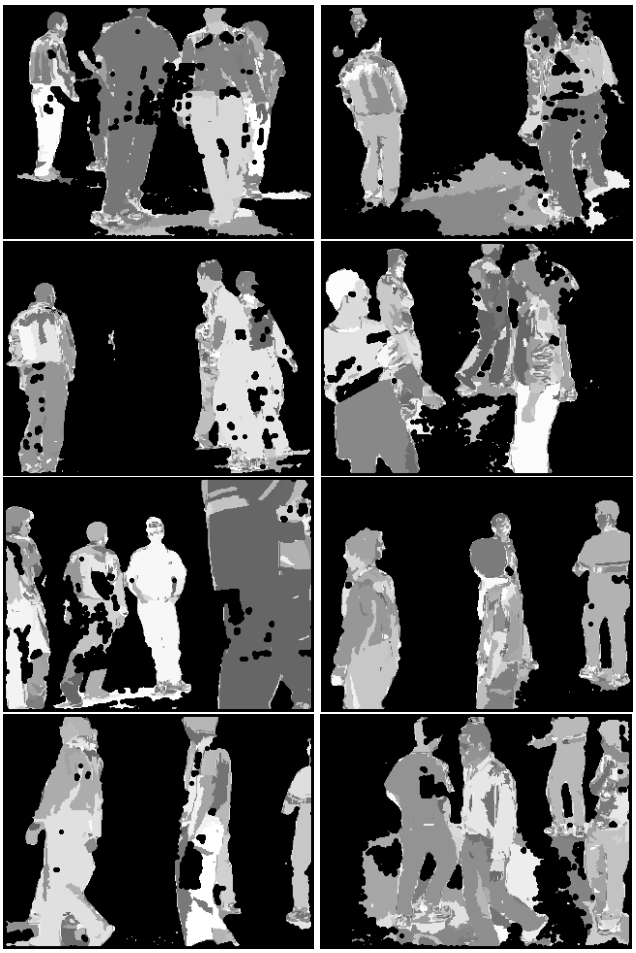

Figure 2: The images from Figure 1, background-subtracted and segmented. The segments are colored randomly and the background is black.

After obtaining the foreground, we apply a segmentation algorithm to it to separate out different regions. Currently, we have implemented a simple color segmentation algorithm, which, after smoothing the image, groups together pixels having similar color characteristics. We neglect the intensity of the pixels so that lighting and orientation of the surface have limited effect, creating regions having "constant" color. A more general segmentation algorithm should use texture segmentation. However, all of the more complex algorithms that we tried did not yield stable results across time and across the cameras. We are currently working on this aspect to make our program work on more general types of objects. The detection and tracking, however, remains the same and a suitable texture segmentation algorithm could be easily integrated into the system. Figure 2 shows the result of background-subtracting and segmenting the images from Figure 1.

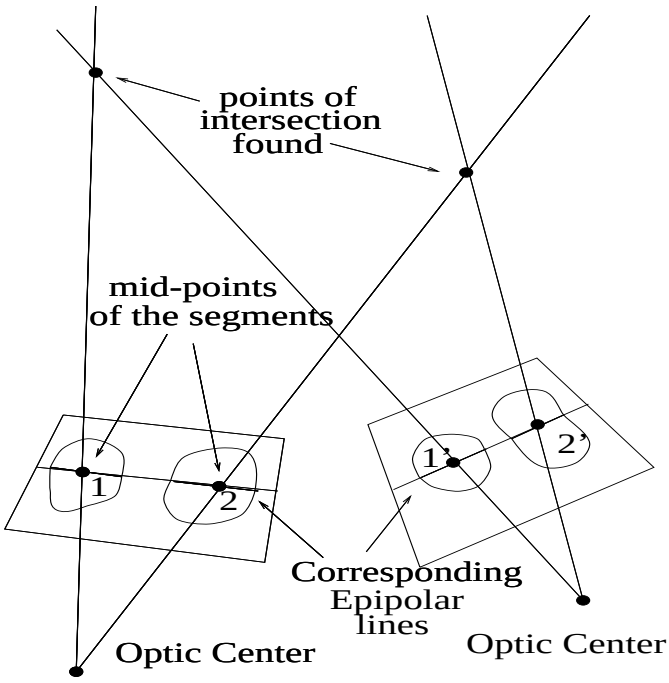

Figure 3: The mid-points of the matched segments are back-projected to obtain 3D points lying inside the objects. The matching segments are 1 and 1', and 2 and 2' respectively.

5 Matching Regions Across Views

After segmentation, we analyze epipolar lines across pairs of views. Along each epipolar line, we match all of the segments from one camera view to the segments in the other view based on color characteristics. Even if one segment matches to more than one segment in the other view, we do not select among these matches and consider all of the matched pairs as positive matches, hoping to reject the false one in later stages. Matching color across the views requires color calibration of the cameras with respect to each other, so that similar colors have similar values as seen from different views. We use the color ratio rather than the color values themselves to eliminate intensity variation across the views caused by different view angles and camera parameters. These ratios remain the same across the views so that matching yields good results.

For each matched pair of segments, we take the *mid-points* of the segments along the epipolar line and back-project them to obtain a 3D point. This 3D point is then projected onto the ground plane to obtain the 2D position of the point on the ground plane (see Figure 3). These points are then processed as described in the next section.

The motivation of taking the mid-point of the matched segments to obtain the 3D point is as follows: We prove (in Appendix A) that in the case of orthographic projection and

a convex object, the point of intersection of the rays through the mid-points of corresponding segments is the only point that can be guaranteed to lie inside the object. If we take any other pair of points on the segments, it is possible to construct a case in which the point of intersection of these rays will not lie inside the object. Although these results do not hold in the general case of a pin-hole camera projection (see Appendix B for a counterexample), in the case of objects being far from the cameras, we obtain a point which is inside or close to the the object in most cases.

An interesting observation here is that these segment midpoints do not generally correspond to a conjugate pair. Nevertheless, the back-projection of the mid-points of the segments does produce a 3D point that is guaranteed to lie inside the object even though the two mid-points typically are the images of two different 3D points.

Also interesting to note is that by matching along epipolar lines, we guarantee the two back-projected rays meet at a point on a plane passing though the optic centers of the cameras. An alternative scheme of trying to match the regions directly (without epipolar lines) and then employing some scheme to identify common points from the regions would be riddled with many problems and would probably yield matching points such that the back-projected rays do not intersect in 3D space at all.

6 Producing Probability Estimates from a Single Pair of Cameras

Once the matching is completed for a given pair of cameras, we calculate probability estimates for the presence of an object at ground plane points using a kernel estimation technique. For each of the 2D ground points obtained using segment matching, we add a gaussian kernel to the probability distribution in the 2D space of the ground plane. The standard deviation of the kernel is based on the minimum width of the segments that matched to give rise to that point, and the camera instantaneous fields of view (IFOV). The probability from all these gaussian kernels is then integrated to obtain a probability distribution map in the 2D space of the ground plane. This is done for each pair of cameras for which the segmentation and matching is performed. Thus, the probability associated with any point x on the 2D plane is given by

$$Prob(\mathbf{x}) = \Sigma_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-(\mathbf{x}_i - \mathbf{x})^2}{2\sigma_i^2}\right)$$

where $\mathbf{x_i} = (x_i, y_i)$ and σ_i are the 2D position and standard deviation of the i-th gaussian kernel.

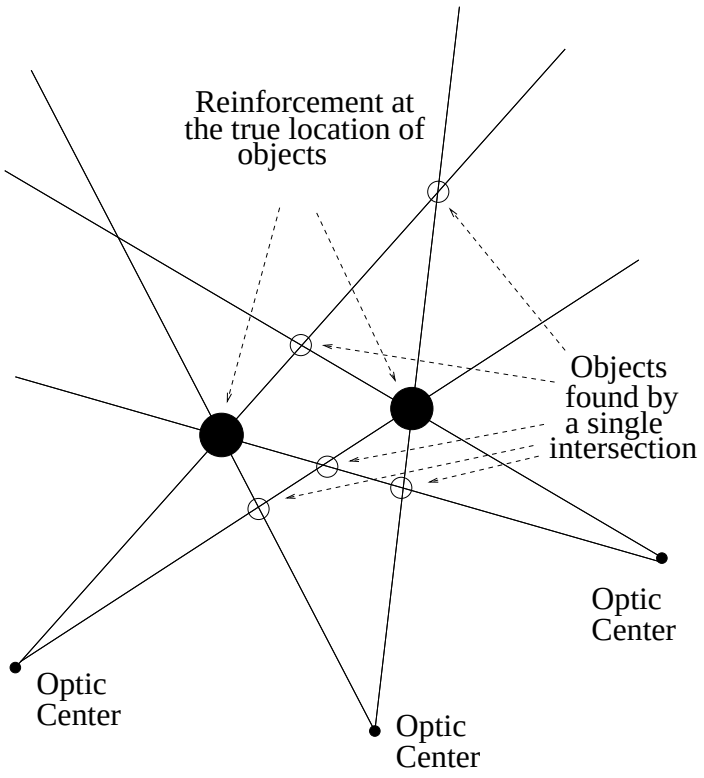

Figure 4: If segment matching fails to distinguish between two objects, the matches would reinforce each other only at the true location of the objects, and the false matches would get eliminated by the weighting scheme.

7 Combining Results from Many Camera Pairs

Given the probability distributions from matching across pairs of cameras, we describe a method for combining the results that rejects outliers and peaks detection results. The simplest method just adds all the probability values, and this method does yield good results because the probability values at the true locations of the objects reinforces each other, whereas the values at false locations are scattered about and occur at different 2D locations for different pairs of cameras. This can be seen in the example illustrated in Figure 4 where two objects of similar color cannot be distinguished while matching from any single pair of cameras. During the combination of results from multiple image pairs, the correct matches reinforce each other, but the false matches do not. Thus, combination of results results in concentration of probabilities only at the true locations. The probability map using simple addition for the image set in Figure 1 is shown in Figure 5.

Although simple addition of probability values yields good results, we can improve things significantly by elimi-

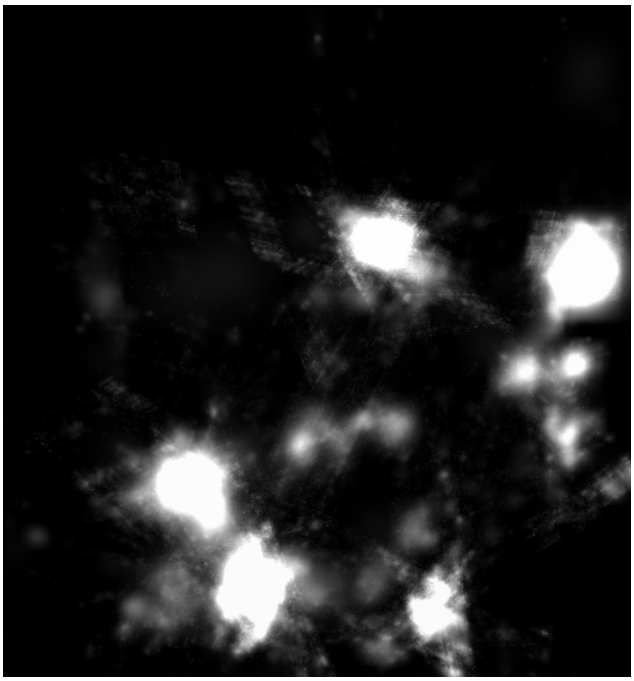

Figure 5: The probability map obtained for the image set shown in Figure 1 by simply adding the probability values from all camera pairs.

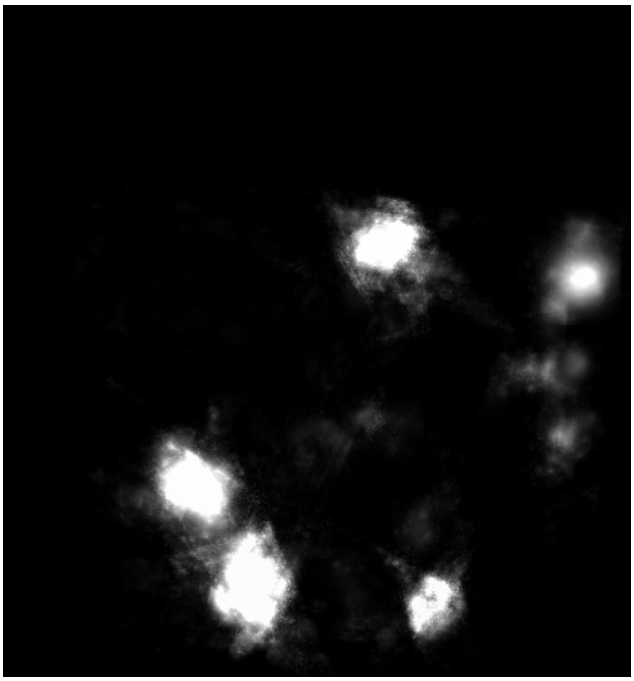

Figure 6: The probability map obtained for the image set shown in Figure 1 by applying the gaussian weighting scheme.

nation of false matches. This is motivated by the observation made above that there will be a large number of camera pairs predicting an object at the correct location, but only a few of them will predict an object at an incorrect one. Therefore, we apply the following weighting method to the probability values at each of the 2D positions to obtain a single probability distribution over the 2D space.

First, the probability values at each 2D location are sorted. Then, we apply a gaussian weighting function to calculate weights for each of the sorted values. Values at the extremes (i.e. the smallest and the largest) are least weighted and the values in the middle are weighted the most. This is a generalization of the method of taking the median to reject outliers. Instead of taking one value, however, we apply a gaussian weighting function centered at the median value. This method will subdue the effect due to matches that are found by only a few pairs of cameras. In practice, we have found this approach to work very well and we present results in section 9. The results of applying this method to Figure 5 is shown in Figure 6.

8 Tracking in the Ground Plane

After obtaining the probability distribution using all pairs, we identify objects by thresholding the probabilities and running a connected components algorithm on the thresholded values. Then, the centroids of the connected components are found. The centroids for the previous time steps matched to the object are fit to a curve assuming constant acceleration. This curve is simply a second-order polynomial:

$$\mathbf{x}(t) = \mathbf{a} + \mathbf{b}.t + \mathbf{c}.t^2$$

The curve is then used to predict the position of the object at the next time interval and a search is made in a neighborhood around this predicted point. The process is repeated for each time step to track objects over time. Even if an object is not found, the predicted position is retained for some time and the object is deleted only if it is not found over an extended period. We have found this simple algorithm to work quite well given the robust probability values obtained from the previous steps.

9 Implementation and Experiments

In our system, image sequences are captured using 16 color cameras synchronized for simultaneous capture. The cameras are located at positions surrounding the room so that they see the objects from different viewpoints. All of the cameras are calibrated using a global coordinate system.

In order to evaluate our algorithm, we conducted experiments on four sequences containing 3, 4, 5 and 6 people

	No. of cameras →	4	8	16
No. of people	No. of True Objects			
3	900	795	0	1
4	1200	1208	135	104
5	1500	209	1	0
6	1800	435	1	0

Table 1: No. of false matches integrated over all frames of the 300-frame sequences

	No. of cameras →	4	8	16
No. of people	No. of True Objects			
3	900	0	27	35
4	1200	0	6	27
5	1500	200	531	242
6	1800	100	457	312

Table 2: No. of true objects missed

respectively. Each sequence consists of 300 frames and people were constrained to move in a region approximately 3.5mX3.5m in size. For each of the sequences, we calculated the number of false objects found, number of true objects missed by the algorithm, and the average probability value at non-object locations. We calculated these metrics using 4, 8 and all 16 cameras in order to study the effect of varying the number of cameras, thus enabling us to determine the minimum number of cameras required to properly identify and track a certain number of objects. All of these are calculated using a threshold probability value which seems to give the best results for 16 cameras. The results obtained are shown in tables 1, 2 and 3.

As was expected, generally, the errors decrease both by increasing the number of cameras and decreasing the number of objects. Although the general trend is sometime broken in a single metric, when the three tables are viewed together, it is seen that the trend is mostly maintained. For example, although there is a large drop in the number of false matches while going from the sequence with 4 people to the sequence with 5 people, there is a corresponding large increase in the number of true objects not found. We attribute these changes to the fact that we have different sequences with different people, which makes it quite difficult to compare the results.

10. Summary and Conclusions

In this paper, we have presented a method for detecting and tracking densely located multiple objects using multiple synchronized cameras located far away from each other. The method matches regions along epipolar lines in camera pairs in order to obtain ground points guaranteed to lie inside objects. Results from camera pairs are integrated using an outlier-rejection scheme so as to obtain robust estimates of the 2D locations of objects, which are then used to track objects across time.

No. of cameras → No. of people	4	8	16
3	0.1281	0.848	0.1019
4	0.1890	0.1259	0.1255
5	0.1682	0.1525	0.1340
6	0.2134	0.1915	0.1584

Table 3: Average non-normalized probability value at Non-object locations. These values can be compared to the threshold value of 5.0 used to identify objects.

Future direction of work includes utilization of object properties like color and shape to improve tracking and improvement in matching by utilization of object properties and occlusion information from previous time frames.

Appendix A

In this section, we prove that, in the case of orthographic projection and a convex object, the intersection of the mid-points of corresponding segments is guaranteed to lie inside the object; and that no other point can be guaranteed thus.

We illustrate this with the help of an illustration showing the plane corresponding to the epipolar lines. (see Figure 7). Let a and b be the rays back-projected from the left and right ends of the segment as seen from the first camera. Let us assume that this segment covers the whole object. In the case of an orthographic projection, a and b will be parallel. Let c and d be the corresponding rays from the second camera. Also, let e and f represent rays back-projected from the mid-points of the segments in camera one and two respectively. Now, let P_1, P_2, P_3 and P_4 be the points of intersection of a, b, c and d as shown in the diagram. Since $a \| b$ and $c \| d$, $P_1P_2P_3P_4$ is a parallelogram. Since camera 1 sees a point on line a belonging to this object, and the object is guaranteed to lie between c and d, we can conclude that there exists a point on the line segment P_1P_2 that lies on the object. Let this point be called A. Similarly, we can conclude the existence of points on line segments P_2P_3, P_3P_4 and P_4P_1. Let these points be called B, C and D respectively. Since the object is convex, we can now conclude that all points lying inside the quadrilateral ABCD also lie within the object.

Now, consider the line passing through A and B. It divides the 2D plane in two parts, the quadrilateral ABCD lying on one side of it. The point P must lie on the side

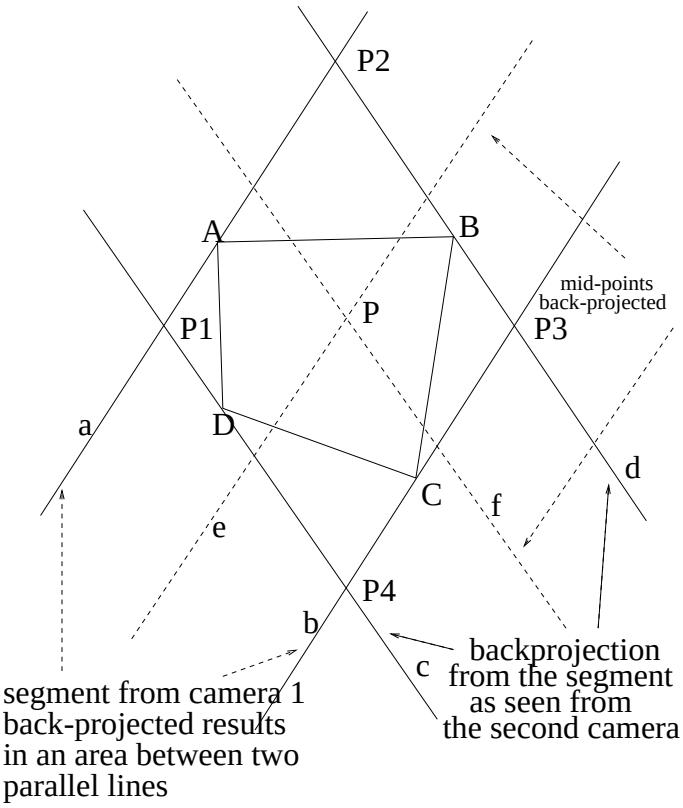

Figure 7: Illustration for appendix A - shows that, in the case of orthographic projection and a convex object, the point of intersection of the back-projected mid-points is the only point guaranteed to lie inside the object

including the quadrilateral ABCD. For, suppose it does not. Then, since $P_1P_2P_3P_4$ is a parallelogram, the point P is also the mid-point of the line segment P_1P_3. Therefore, since the point P lies on the side of line AB not containing ABCD, at least one of A and B must lie on the side of line P_1P_3 towards P_4, which is a contradiction since both A and B lie towards P_2. Therefore, we can conclude that the point P must lie on the side of line AB containing ABCD. Similarly, we can prove that the point P must lie on the side towards ABCD, as seen from BC, CD and AD. But this means that the point P lies inside ABCD, and hence the object.

For any point P' other than P, it is possible to place A, B, C and D such that the point P' lies outside the quadrilateral ABCD. For, it must lie on one side of at least one of the lines P_1P_3 and P_2P_4. If it lies on the side of P_1P_3 towards P_2, then we can place AB such that P' lies on the side of AB towards P_2, thus implying that it lies outside ABCD. We can similarly prove for the other cases.

Therefore, the point P is the only point guaranteed to lie inside the object.

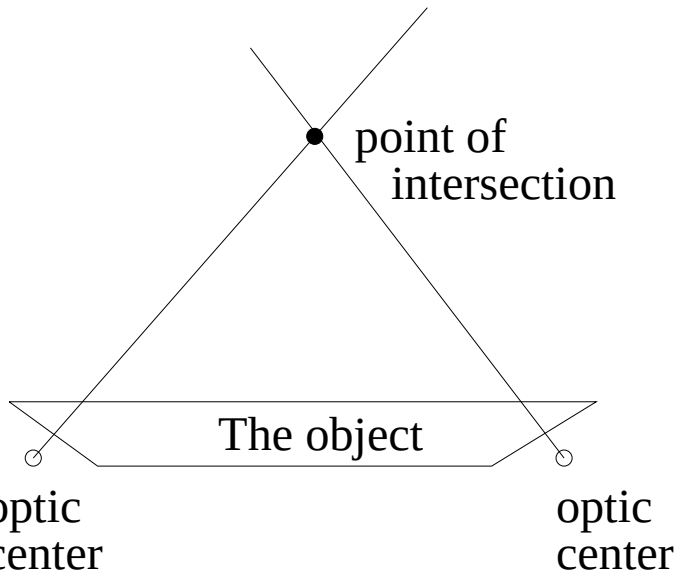

Figure 8: Illustration for appendix B - shows, for pin-hole camera projection model, a case where the back-projections of the mid-points do not intersect inside the object

Appendix B

In this section, we show an example where, for a pin-hole model of camera projection and a convex object, the intersection of the mid-points of the corresponding segments does not lie inside the object. See Figure 8.

References

[1] "Multi-camera multi-person tracking for EasyLiving," J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer. *3rd IEEE International Workshop on Visual Surveillance*, 2000.

[2] I. Haritaoglu, D. Harwood, L. Davis. "W4:Who, When, Where, What: A Real Time System for Detecting and Tracking People." *Third International Conference on Automatic Face and Gesture, Nara, April 1998.*

[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4S: A real-time system for detecting and tracking people in 2 1/2D," *5th European Conference on Computer Vision*, Freiburg, Germany, 1998.

[4] V. Kettnaker, Ramin Zabin. "Counting People from multiple cameras", in *IEEE ICMCS*, Florence, Italy, 1999. p. 267-271.

[5] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams," *6th International conference on Computer Vision*, Bombay, India, 1998, pp. 356-362.

[6] T. Darrel, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, color, and Pattern Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA 1998, pp. 601-608.

[7] J. Orwell, P. Remagnino and G.A. Jones. "Multi-Camera Color Tracking." *Proceedings of the 2nd IEEE Workshop on Visual Surveillance, 1998.*

[8] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G.A. Jones, "A Multi-agent Framework for Visual Surveillance," *International Conference on Image Analysis and Processing,* Venice, Italy, 1999, pp. 1104-1107.

[9] R.T. Collins, A. J. Lipton, and T. Kanade, "A System for Video Surveillance and Monitoring," *American Nuclear Society Eighth International Topical Meeting on Robotics and Remote Systems,* Pittsburgh, PA 1999.

[10] R. Rosales and S. Sclaroff, "3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions," *CVPR,* Fort Collins, CO, 1999, pp. 117-123.

[11] S. S. Intille and A.F. Bobick, "Closed-World Tracking," *5TH ICCV,* Cambridge, MA, 1995, pp. 672-678.

[12] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects," *7th ICCV,* Kerkyra, Greece, 1999, pp. 572-578.

[13] T. Horprasert, D. Harwood, and L.S. Davis. "A Robust Background Subtraction and Shadow Detection." *Proc. ACCV 2000,* Taipie, Taiwan, January 2000.

[14] D. Snow, P. Viola, and R. Zabih. "Exact Voxel Occupancy Using Graph Cuts." *Proc. CVPR,* Hilton Head, SC 2000.

[15] R. Want, A. Hopper, V. Falcao, and J. Gibbons. "The Active Badge Location System," *ACM Transactions on Information Systems,* vol. 10, pp 91-102, 1992.

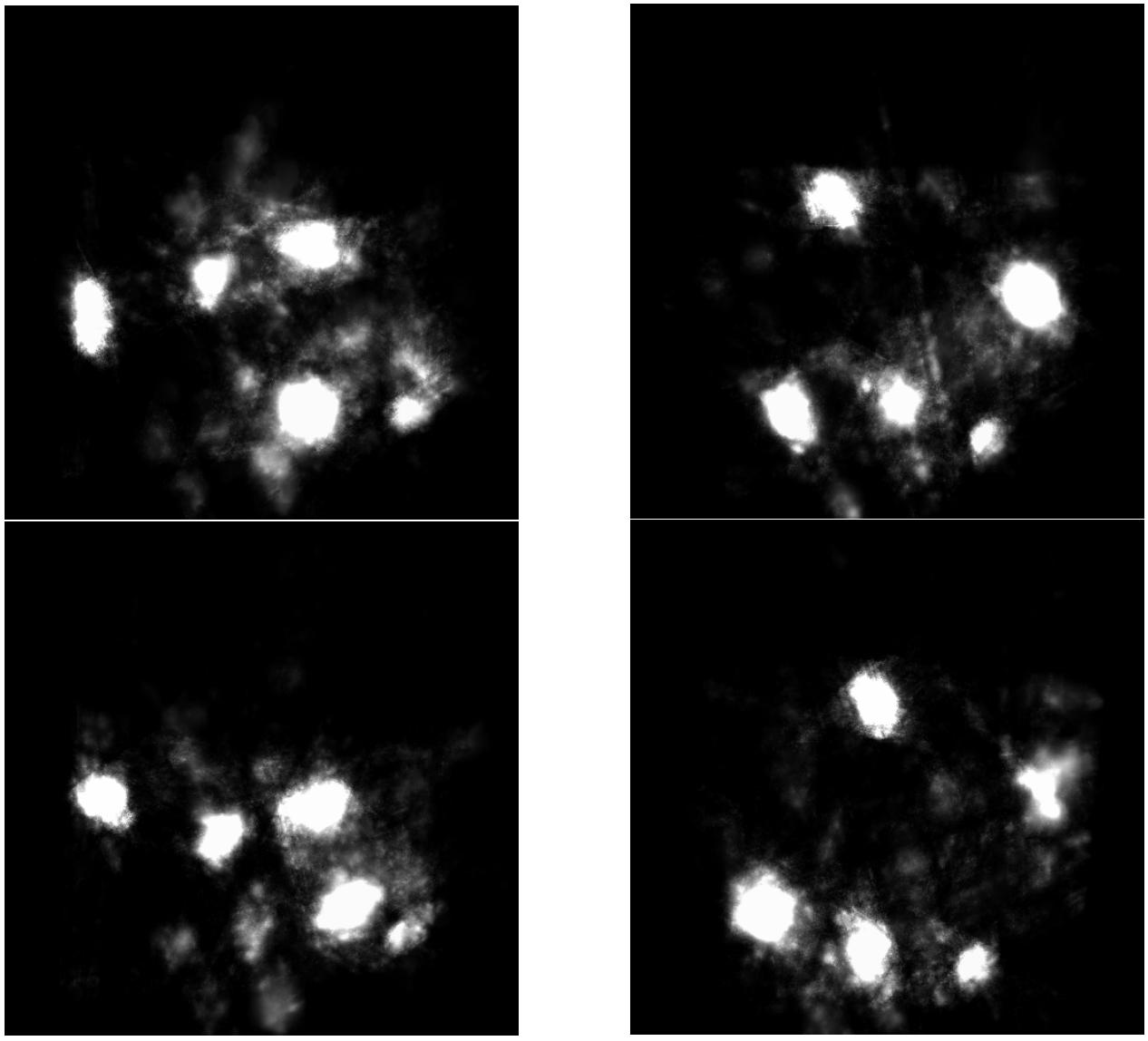

Figure 9: The probability maps obtained at four different time steps for a sequence from which the image set in Figure 1 was taken. All 16 cameras were used.