M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo

Anurag Mittal and Larry S. Davis

Department of Computer Science
University of Maryland
College Park, MD 20742
{anurag,lsd}@umiacs.umd.edu

Abstract. We present a system that is capable of segmenting, detecting and tracking multiple people in a cluttered scene using multiple synchronized cameras located far from each other. The system improves upon existing systems in many ways including: (1) We do not assume that a foreground connected component belongs to only one object; rather, we segment the views taking into account color models for the objects and the background. This helps us to not only separate foreground regions belonging to different objects, but to also obtain better background regions than traditional background subtraction methods (as it uses foreground color models in the algorithm). (2) It is fully automatic and does not require any manual input or initializations of any kind. (3) Instead of taking decisions about object detection and tracking from a single view or camera pair, we collect evidences from each pair and combine the evidence to obtain a decision in the end. This helps us to obtain much better detection and tracking as opposed to traditional systems.

Several innovations help us tackle the problem. The first is the introduction of a region-based stereo algorithm that is capable of finding 3D points inside an object if we know the regions belonging to the object in two views. No exact point matching is required. This is especially useful in wide baseline camera systems where exact point matching is very difficult due to self-occlusion and a substantial change in viewpoint. The second contribution is the development of a scheme for setting priors for use in segmentation of a view using bayesian classification. The scheme, which assumes knowledge of approximate shape and location of objects, dynamically assigns priors for different objects at each pixel so that occlusion information is encoded in the priors. The third contribution is a scheme for combining evidences gathered from different camera pairs using occlusion analysis so as to obtain a globally optimum detection and tracking of objects.

The system has been tested using different density of people in the scene which helps us to determine the number of cameras required for a particular density of people.

Keywords: Multi-camera Tracking, Region-Based Stereo, Grouping and Segmentation

Fig. 1. Four images from a 6-perspective sequence at a particular time instant. The boxes show the positions found by the algorithm.

1 Introduction

In this paper we address the problem of segmenting, 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. This is normally the case in many surveillance applications. Figure 1 shows four images from a 6-perspective sequence that will be used to illustrate our algorithm. Notice that in all four images, there is substantial occlusion so that one cannot assume that we are seeing a person in isolation. We assume that our cameras are calibrated, and that people are moving on a calibrated ground plane. We also assume that the cameras are frame synchronized.

The paper develops several novel ideas in order to solve the problem. The first and most important is the introduction of a region-based stereo algorithm that is capable of finding 3D points inside an object if we know the regions belonging to the object in two views. No exact point matching is required. This is especially useful in wide baseline camera systems where exact matching is very difficult due to self-occlusion and a substantial change in viewpoint. The second contribution is the development of a scheme for setting priors for use in segmentation of a view using bayesian classification. The

scheme, which assumes knowledge of approximate shape and location of objects, dynamically assigns priors for different objects at each pixel so that occlusion information is encoded in the priors. These priors are used to obtain good segmentation even in the case of partial occlusions. The third contribution is a scheme for combining evidences gathered from different camera pairs using occlusion analysis so as to obtain a globally optimum detection and tracking of objects. Higher weight is given to those pairs which have a clear view of that location than those whose view is potentially obstructed by some objects. The weight is also determined dynamically and uses approximate shape features to give a probabilistic answer for the level of occlusion.

Our system takes a unified approach to segmentation, detection and tracking using multiple cameras. We neither detect nor track objects from a single camera or a camera pair; rather evidence is gathered from multiple camera pairs and the decisions of detection and tracking are taken at the end by combining the evidences in a robust manner taking occlusion into consideration. Also, we do not simply assume that a connected component of foreground pixels corresponds to a single object. Rather, we employ a segmentation algorithm to separate out regions belonging to different people. This helps us to handle the case of partial occlusion and allows us to track people and objects in a cluttered scene where no single person is isolated in any view.

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. [6] developed a single camera 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 [7], they incorporate stereo information into their system. Kettnaker and Zabih [11] developed a system for counting the number of people in a multi-camera environment where the cameras have a non-overlapping field of view. Darrell et. al. [3] developed a tracking algorithm that uses a stereo pair of cameras and integrates stereo, color and face pattern detection. Dense stereo processing is used to isolate people from other objects and people in the background, and faces and bodies of people are tracked. All of these methods use a single viewpoint (using one or two cameras) for a particular part of the scene and would have problems in the case of objects occluded from that viewpoint.

Orwell et. al. [16] 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 [17], Orwell et. al. present a multi-agent framework for determining whether different agents are assigned to the same object seen from different cameras. This method would have problems in the case of partial occlusions where a connected foreground region does not correspond to one object, but has parts from several of them.

Cai and Aggarwal [1] 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. Since in our algorithm, we collect evidences from different pairs and only take the decision at the end, we expect our algorithm to perform better than this approach.

Intille et. al. ([9] and [10]) present a system which is capable of tracking multiple non-rigid objects. The system uses a top-view camera to identify individual blobs and a "closed-world" assumption to adaptively select and weight image features used for matching these blobs. Putting a camera(s) on the top is certainly a good idea since it reduces occlusion, but is not possible in many situations. Also, the advantage of a camera on top is reduced as we move away from the camera, which might require a large number of cameras. Such a camera system would also not be able to identify people or determine other important statistics (like height or color distributions) and hence may not be very useful for many applications.

Krumm et. al. [13] present an algorithm that has goals 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 histograms are created for each person and are used to identify and track people over time. The method of using short-baseline stereo matching to back-project into 3D space and integrating information from different stereo pairs has also been used by Darrell et. al. [4]. In contrast to [13] and [4], our approach utilizes the wide-baseline camera arrangement that has the following advantages:

(1) It provides many more camera pairs that can be integrated ($C\,_2^n$ as compared to $n/2$ for short baseline stereo, using n cameras),

(2) It has higher accuracy in back-projection and lower sensitivity to calibration errors, and

(3) It provides more viewing angles with the same number of cameras so that occlusion can be handled better.

On the other hand, the short-baseline stereo pair camera arrangement used, e.g., in [4] has the advantages of

(1) more accurate correspondences due to small change in viewpoint, and

(2) better understood matching algorithms.

It is not evident which method is better and it appears that a combination of the two methods might yield the best results.

Our region-based stereo algorithm can be considered to lie between wide-baseline stereo algorithms, which try to match exact 3D points across the views, and volume intersection algorithms which 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 challenge of incorrect matches due to a substantial change in viewpoint. Although some work has been done to improve upon these methods(e.g. [18] and [8]), they are still not very robust due to this fundamental difficulty.

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 work recently (for e.g. [20]) 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 match points exactly across views; neither do we perform volume intersection without regard to the objects seen. Rather, determination of regions belonging to different objects is sufficient to yield 3D points guaranteed to lie inside the objects.

3 General Overview of the Algorithm

Our system models different characteristics of people by observing them over time. These models include color models at different heights of the person and "presence" probabilities along the horizontal direction at different heights. These models are used to segment images in each camera view. The regions thus formed are matched across views using our region-matching stereo algorithm which yields 3D points potentially lying inside objects. These points are projected onto the ground plane and ground points are used to form an object location likelihood map using Gaussian kernels for a single image pair. The likelihood maps are combined using occlusion analysis to obtain a single map, which is then used to obtain ground plane positions of objects in the scene. The algorithm is then iterated using these new ground plane positions and this process is repeated until the ground plane positions are stable. The final ground plane positions are then used to update the person models, and the whole process is repeated for the next time step.

4 Modeling People

We model the appearance and locations of the people in the scene. These models, which are developed by observing people over time (method explained in section 9), help us segment people in the camera views. These models are developed from the sequences automatically; no manual input is required.

4.1 *Color Models*

One of the attributes useful to model is the color distribution at different heights of the person. A single color model for the whole person would not be able to capture the vertical variation of the color. On the other hand, modeling the horizontal distribution of color is very difficult without full 3D surface reconstruction, which would be too time-consuming and hence not too interesting for tracking and surveillance type of applications. In order to model the color distribution at different heights, we use the well-known method of non-parametric Gaussian kernel estimation technique which is well suited to our system. (see [5] for more details). Since the intensity levels change across cameras due to aperture effects, and due to shadow and orientation effects in the same view, we only use the ratios $r/(r+g+b)$ and $g/(r+g+b)$ in the color models.

4.2 *"Presence" Probabilities*

For our segmentation algorithm, we want to determine the probability that a particular person is "present" (i.e. occupies space) along a particular line of sight. Towards that end, we define "Presence" Probability (denoted by $L(h, w)$) as the probability that a person is present(i.e. occupies space) at height h and distance w from the vertical line passing through the person's center. This probability is a function of both the distance w and height h since, e.g., the width of a person near the head is less than the width near the center. This probability function also varies from person to person. The method for estimating this probability by observation is described in section 9.

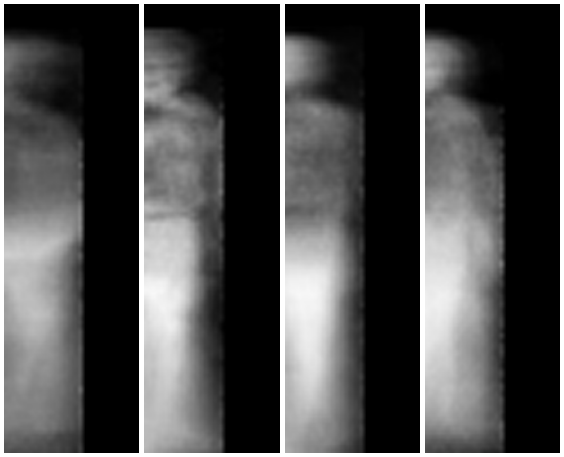

Fig. 2. Sample Presence Probabilities of people observed over time.

5 Pixel Classification in a Single View

We use Bayesian Classification to classify each pixel as belonging to a particular person, or the background. The *a posteriori* probability that an observed pixel $\mathbf{x}$ (containing both color and image position information) belongs to a person j (or the background) is

$$P_{posterior}(j/\mathbf{x}) \propto P_{prior}(j)P(\mathbf{x}/j) \tag{1}$$

The pixel is then classified as

$$\textit{Most likely class} = \max_{j}(P_{posterior}(j/\mathbf{x})) \tag{2}$$

$P(\mathbf{x}/j)$ is given by the color model of the person at height h. For the background, we use a background model of the scene using the method described in [14].

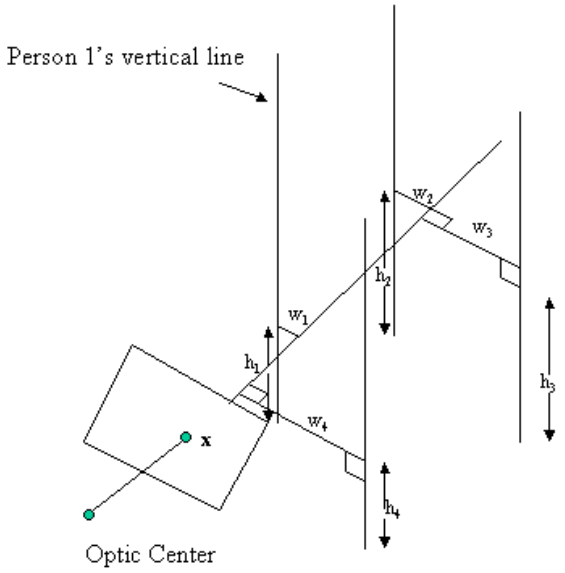

Fig. 3. Measuring distances (and heights) from the line of sight

We want the prior probabilities to include occlusion information so that the prior for a person in front is higher near his estimated position compared to the priors far away from him and compared to a person in rear. Doing this in a structured, consistent and logical manner is the challenge. We employ the following methodology. For each pixel **x**, we project a ray in space passing through the optical center of the camera (see Figure 3). We calculate the minimum distances w_j of this ray from the vertical lines passing through the currently estimated centers of the people. Also calculated are the heights h_j of the shortest line segments connecting these lines. Then, the prior probability that a pixel **x** is the image of person j is set as

$$P_{prior}(j) = L_j(h_j, w_j) \prod_{k \text{ occludes } j} (1 - L_k(h_k, w_k)) \tag{3}$$

$$P_{prior}(background) = \prod_{all\ j} (1 - L_j(h_j, w_j)) \tag{4}$$

where $L_j(h_j, w_j)$ is the "presence" probability described in section 4.2. A person "k occludes j" if the distance of k to the optical center of the camera is less than the distance of j to the center.

The motivation for the definition is that a particular pixel originates from a person if and only if (1) the person is present along that line of sight (Probability for this = L_j),

Fig. 4. The result of segmenting images shown in Figure 1

and (2) no other person in front of her is present along that line of sight (Probability = 1 - L_k). If no person is present along a particular line of sight, we see the background. The classification procedure enables us to incorporate both the color profile of the people, and the occlusion information available in a consistent and logical manner.

It is interesting to note that we expect to obtain better background subtraction using our segmentation procedure than using traditional background subtraction methods because we take into account models of the foreground objects in the scene in addition to information about the background that is the only input for traditional background subtraction methods. Indeed, this is what we observe during experiments.

We need a procedure to detect new people entering the scene and bootstraping the algorithm in order to make it fully automatic. Towards that end, we detect unclassified pixels as those for which $P_{prior} * P(c/j)$ is below a given threshold for all the person models and the background, i.e. none of the person models or the background can account for the pixel with a high enough probability. For these pixels, we use a simple color segmentation algorithm, which groups together pixels having similar color characteristics. This segmentation creates additional regions in the image and these regions are also matched across cameras as described in the next section.

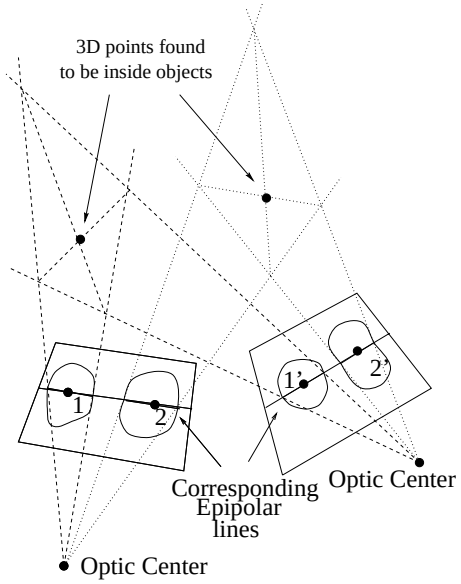

Fig. 5. The point of intersection of the diagonals of the quadrilateral formed by back-projecting the endpoints of the matched segments yields a 3D point lying inside an object. The matching segments are 1 and 1', and 2 and 2' respectively.

6 Region-Based Stereo

Along epipolar lines in pairs of views, we match regions from one camera view to the regions in the other. Segments belonging to the same person in different views (as determined by the classification algorithm) are matched to each other. Regions corresponding to unclassified pixels are matched to each other based on color characteristics. For each matched pair of segments, we project the end-points of the segments and form a quadrilateral in the plane of the corresponding epipolar lines. The point of intersection of the diagonals of this quadrilateral is taken to be belonging to the object (see Figure 5). This is because, for a convex object this is the only point that can be guaranteed to lie inside the object (see proof in Appendix). This is assuming that the complete object is visible and segmented completely as one region in each view. For any other 3D point in the plane of the epipolar lines, it is possible to construct a case in which this point will lie outside the object.

7 Producing Likelihood Estimates on the Ground Plane

Having obtained 3D points belonging to people, we want to detect and track people in a robust manner rejecting outliers. Assuming the people are standing upright or are otherwise extended primarily in the vertical direction, one natural way to do that would be to do the estimation on the ground plane after projecting the 3D points onto it. It is also possible to do clustering in 3D and this would be the method of choice for many

applications. However, for our application, estimation on the ground plane is better since we are dealing with only walking people. We define a "likelihood" measure which estimates whether a particular location on the ground plane is occupied by an object. We develop likelihood maps for each camera pair used and then combine these maps in a robust manner using the occlusion information available.

7.1 Likelihood from a Single Camera Pair

A simple way to develop likelihood maps using ground points is to use Gaussian kernels. The weight and standard deviation of the kernels 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). This gives higher weight to points originating from longer segments than from smaller ones. This is done for each pair of cameras for which the segmentation and matching is performed.

7.2 Combining Results from Many Camera Pairs Using Occlusion Analysis

Given the likelihood maps from matching across pairs of cameras, we describe a method for combining likelihood maps that makes use of occlusion information available from the approximate position of the people. For each of the cameras, we form a probability map that gives us the probability that a particular location **x** is visible from the camera. First of all, the camera center is projected onto the ground plane. Then, for each point **x** on the ground plane, we calculate the perpendicular distance w_j of each person j from the line joining the camera center and the point **x**. Then, defining "presence" probabilities $L_j()$ in a way similar to section 4.2, but taking only the width as parameter (by averaging over the height parameter), we find the probability that the point **x** is visible from the camera c as

$$P_c(\mathbf{x}) = \prod_{j \textit{ occludes } \mathbf{x}} (1 - L_j(w_j)) \tag{5}$$

where j occludes **x** if its distance from the camera is less than **x**. Now, for a particular camera pair $(c1, c2)$, the weight for the ground point **x** is calculated as

$$w_{(c1,c2)}(\mathbf{x}) = P_{c1}(\mathbf{x}) P_{c2}(\mathbf{x}) \tag{6}$$

The weight is essentially the probability that **x** is visible from both the cameras. The weighted likelihood value is then calculated as

$$Lk(\mathbf{x}) = \frac{\sum_{(c1,c2)} w_{(c1,c2)}(\mathbf{x}) Lk_{(c1,c2)}(\mathbf{x})}{\sum_{(c1,c2)} w_{(c1,c2)}(\mathbf{x})} \tag{7}$$

This definition helps us to dynamically weigh the different likelihood values such that the values with the highest confidence level (least occlusion) are weighted the most. Note that the normalization constant is different for each ground plane point and changes over time.

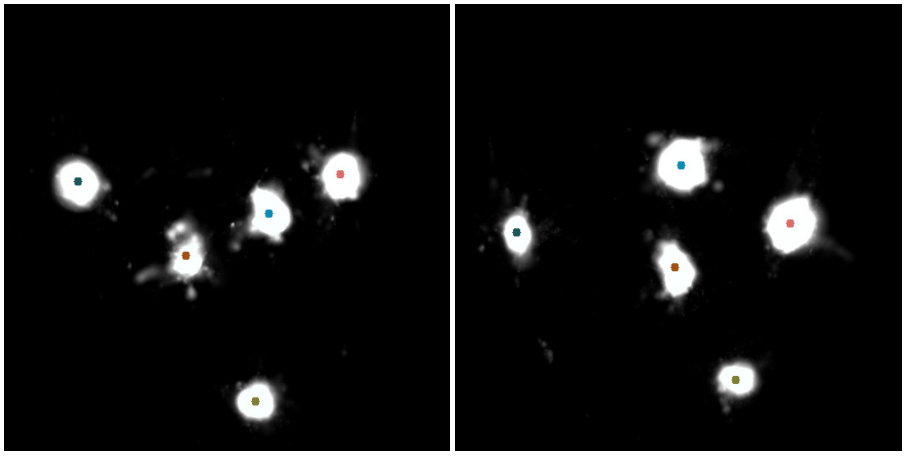

Fig. 6. (a) The likelihood map obtained for the image set shown in Figure 1 by applying the occlusion-analysis weighting scheme. The dots show the position state variable of the Kalman filter tracking the person. (b) The likelihood map for another time step from the same sequence.

8 Tracking on the Ground Plane

After obtaining the combined likelihood map, we identify objects by examining likelihood clusters and identifying regions where the sum of likelihoods exceeds a given threshold. The centroids of such likelihood "blobs" are obtained simply using

$$\mathbf{x}_{centroid} = \frac{\sum_{\mathbf{x},\mathbf{x}\in region} \mathbf{x} * Lk(\mathbf{x})}{\sum_{\mathbf{x}} Lk(\mathbf{x})} \tag{8}$$

where $Lk(\mathbf{x})$ is the likelihood at point $\mathbf{x}$. These object blobs are then tracked over time using a *Kalman filter*.

9 Updating Models of People

Observed images and information about the current position of the people are used to update models of people and create ones for the "new" people detected. For each pixel, we calculate the "presence" probabilities L_j for each person as described earlier. We determine if L_j is above a certain threshold for a particular person and below another (lower) threshold for all others. This helps us in ensuring that the pixel is viewing the particular person only and nothing else (except the background). In order to determine if the pixel belongs to the background or not, we use the background model to determine the probability that the pixel color originates from the background. If this probability is below a certain threshold, then we determine that the pixel belongs to the person; else it belongs to the background. If it belongs to the person, it is added as a kernel to the color model of the person at that height. We update the "presence" probability L_j for the person by incrementing the count for the total number of observations at height h

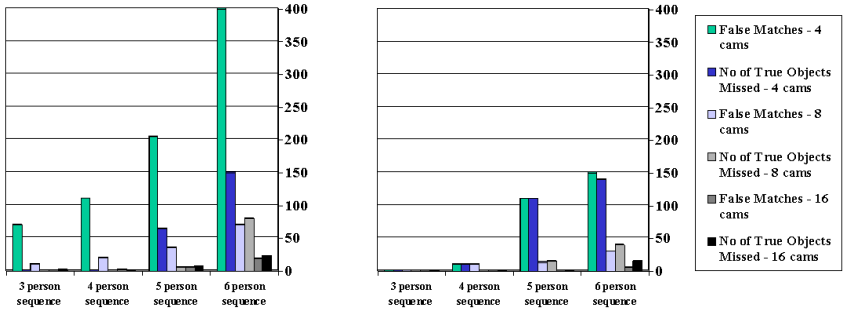

Fig. 7. Cumulative errors for four sequences of 200 time steps each by (a) averaging likelihoods and using no occlusion analysis, and (b) using occlusion analysis.

and width w for the person and incrementing the count for positive matches only if this pixel is determined to belong to the person (according to the above mentioned method). The "presence" probability at that height and width is then simply the second count divided by the first.

10 Implementation and Experiments

Image sequences are captured using up to 16 color CCD cameras. These cameras, which are attached to "Acquisition" PCs via frame grabbers, are capable of being externally triggered for synchronization purposes. Cameras are located at positions surrounding the lab so that they see the objects from different viewpoints. All of the cameras are calibrated using a global coordinate system and the ground plane is also determined. Frame synchronization across cameras is achieved using a TTL-level signal generated by a Data Translation DT340 card attached to a controller PC, and transmitted via coaxial cables to all the cameras. For video acquisition, the synchronization signal is used to simultaneously start all cameras. No timecode per frame is required.

In the distributed version of the algorithm where we use a Pentium II Xeon 450MHz PC for each of the cameras, the system currently takes about 2 seconds per iteration of the ground plane position finding loop. On the average, we need about 2 - 3 iterations per time step, so the running time of the algorithm is about 5 seconds per time step. We believe that by code optimizations and faster processors, we will be able to run the algorithm in real time.

In order to evaluate our algorithm, we conducted experiments on four sequences containing 3, 4, 5 and 6 people respectively. 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 consisted of 200 frames taken at the rate of 10 frames/second and people were constrained to move in a region approximately

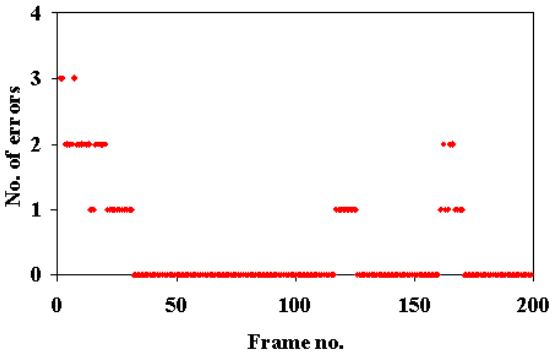

Fig. 8. Total errors as a function of time for the sequence with 5 people using 8 cameras. Note how the errors decrease with time as the models become more robust. Errors after the initial period occur mainly because of people coming too close to each other.

3.5mX3.5m in size. Matching was done for only adjacent pairs of cameras (n pairs)and not for all of the C_2^n pairs possible. This helps us control the time complexity of the algorithm, but reduces the quality of the results obtained.

For each of the sequences, we calculated the number of false objects found and the number of true objects missed by the algorithm. We calculated these metrics using 4, 8 and 16 cameras in order to study the effect of varying the number of cameras and to determine the breakdown characteristics, thus enabling us to determine the minimum number of cameras required to properly identify and track a certain number of objects. The cumulative errors over the 200 frames are shown in Figure 7(b). Also shown in Figure 7(a) are the error metrics obtained when the likelihood values obtained from different cameras are weighted equally and occlusion analysis is not used. This helps us observe the improvement obtained by using the occlusion analysis scheme. Most of the errors occur when the models for people are not very accurate, e.g., in the beginning and when a new person enters the scene. However, as models become better, it is able to correct itself after a few time steps only. The sequences containing 5 and 6 people have severe occlusion at many time steps such that a person is surrounded by the others in such a way that he is not visible from any of the cameras. This results in these people not being detected for those time steps.

11 Summary and Conclusions

In this paper, we have presented a system for segmenting, detecting and tracking multiple people using multiple synchronized cameras located far from each other. It is fully automatic and does not require any manual input or initialisations. It is able to handle occlusions and partial occlusions caused by the dense location of these objects and hence can be useful in many practical surveillance applications.

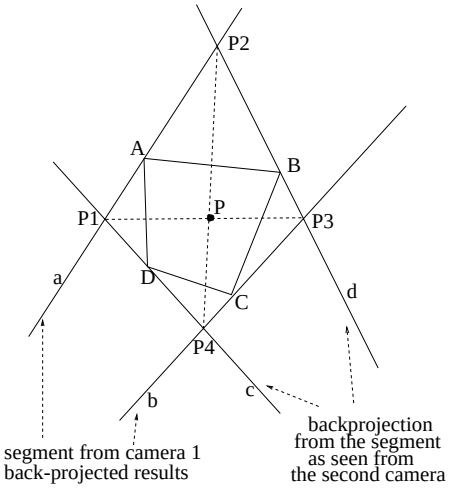

Fig. 9. Illustration for Appendix - shows that, for a convex object, the point of intersection of the diagonals of the quadrilateral formed by back-projecting the end-points of the matched segments is the only point guaranteed to lie inside the object

Acknowledgments

We would like to thank Ramani Duraiswami for reading the manuscript and suggesting improvements in it. This research has been funded by NSF grant no. EIA9901249.

Appendix

In this section, we prove that, in the case of a convex object O, the point of intersection of the diagonals of the quadrilateral formed by backprojecting the end-points of corresponding segments of that convex object is guaranteed to lie inside the object; and that no other point can be guaranteed thus.

We prove this with the help of an illustration showing the plane corresponding to the epipolar lines. (see Figure 9). 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 c and d be the corresponding rays from the second camera. 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. Let P be the point of intersection of the diagonals of $P_1P_2P_3P_4$. Since camera 1 sees some point on line a that belongs to O, and O is guaranteed to lie between rays c and d, we can conclude that there exists a point on the line segment P_1P_2 that lies on the boundary of O. Let this point be called A. Similarly, we can conclude the existence of points from O 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 O.

Now, consider the line segment AB. Omitting details, we can easily prove that the point P lies on the same side of AB as the quadrilateral $ABCD$. Similarly, we can prove that P lies on the same side of lines BC, CD and DA as the quadrilateral $ABCD$. But this means that P lies inside $ABCD$, hence inside O.

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.

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

References

1. Cai Q. and Aggarwal J.K. 1998. Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams. In *6th Internation Conference on Computer Vision*, Bombay, India, pp. 356-262.
2. Collins R.T., Lipton A.J., and Kanade T. 1999. A System for Video Surveillance and Monitoring. *American Nuclear Society Eighth International Topical Meeting on Robotics and Remote Systems*, Pittsburgh.
3. Darrell T., Gordon G., Harville M., and Woodfill J. 1998. Integrated Person Tracking Using Stereo, color, and Pattern Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 601-608.
4. Darrell T., Demirdjian D., Checka N., and Felzenszwalb P. 2001. Plan-View Trajectory Estimation with Dense Stereo Background Models. In *IEEE International Conference on Computer Vision.*, Vancouver, Canada.
5. Elgammal A., Duraiswami R. and Davis L.S. 2001. Efficient Non-parametric Adaptive Color Modeling Using Fast Gauss Transform. *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii.
6. Haritaoglu I., Harwood D. and Davis, L.S. 1998. W4:Who, When, Where, What: A Real Time System for Detecting and Tracking People. *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 222-227.
7. Haritaoglu I., Harwood D., and Davis L.S. 1998. W4S: A real-time system for detecting and tracking people in 2 1/2D. *5th European Conference on Computer Vision*, Freiburg, Germany.
8. Horaud R. and Skordas T. 1989. Stereo Correspondence through Feature Grouping and Maximal Cliques. *IEEE Journal on Pattern Analysis and Computer Vision*, vol 11(11):1168-1180.
9. Intille S. S. and Bobick A. F. 1995. Closed-World Tracking. *5TH International Conference on Computer Vision*, Cambridge, MA, pp. 672-678.
10. Intille S.S., Davis, J.W. and Bobick A.F. 1997. Real-Time Closed-World Tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 697-703.
11. Kettnaker V. and Zabih R. 1999. Counting People from Multiple Cameras. In *IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, pp. 267-271.
12. Sander P.T., Vinet L, Cohen L. and Gagalowicz A. 1989. Hierarchical Region Based Stereo Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego.
13. Krumm J., Harris S., Meyers B., Brumitt B., Hale M. and Shafer S. 2000. Multi-camera Multi-person Tracking for EasyLiving. *3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland.
14. Mittal A. and Huttenlocher D. 2000. Site Modeling for Wide Area Surveillance and Image Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina.

15. Mittal A. and Davis L.S. 2001. Unified Multi-Camera Detection and Tracking Using Region-Matching. In *IEEE Workshop on Multi-Object Tracking*, Vancouver, Canada.
16. Orwell J., Remagnino P. and Jones G.A. 1999. Multi-Camera Color Tracking. *Proceedings of the 2nd IEEE Workshop on Visual Surveillance*, Fort Collins, Colorado.
17. Orwell J., Massey S., Remagnino P., Greenhill D., and Jones G.A. 1999. A Multi-agent Framework for Visual Surveillance. *International Conference on Image Analysis and Processing*, Venice, Italy, pp 1104-1107.
18. Pritchett P., and Zisserman A. 1998. Wide Baseline Stereo Matching. In *Sixth International Conference on Computer Vision*, Bombay, India, pp. 754-760.
19. Rosales R. and Sclaroff S. 1999. 3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 117-123.
20. Snow D., Viola P., and Zabih R. 2000. Exact Voxel Occupancy Using Graph Cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina.
21. Wren C.R., Azarbayejani A., Darrell T. and Pentland A.P. 1997. Pfinder: Real-time Tracking of the Human Body. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol 19. 7.