M_2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene

Anurag Mittal (anurag@cs.umd.edu) and Larry S. Davis
(lsd@cs.umd.edu)
Department of Computer Science
University of Maryland
College Park, MD 20742

Abstract.

When occlusion is minimal, a single camera is generally sufficient to detect and track objects. However, when the density of objects is high, the resulting occlusion and lack of visibility suggests the use of multiple cameras and collaboration between them so that an object is detected using information available from all the cameras in the scene.

In this paper, we present a system that is capable of segmenting, detecting and tracking multiple people in a cluttered scene using multiple synchronized surveillance cameras located far from each other. The system is fully automatic, and takes decisions about object detection and tracking using evidence collected from many pairs of cameras. Innovations that help us tackle the problem include a region-based stereo algorithm capable of finding 3D points inside an object knowing only the projections of the object (as a whole) in two views, a segmentation algorithm using bayesian classification and the use of occlusion analysis to combine evidences from different camera pairs.

The system has been tested using different densities of people in the scene. This helps us determine the number of cameras required for a particular density of people. Experiments have also been conducted to verify and quantify the efficacy of the occlusion analysis scheme.

Keywords: Multi-Camera Tracking, Region-Based Stereo, Grouping and Segmentation, Wide-Baseline Stereo

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. Figure 1 shows images from a 6-perspective sequence that will be used to illustrate our algorithm. Notice that in all four images, no single person is 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 also assume that the cameras are frame synchronized.

We present an algorithm that takes a unified approach to segmentation, detection and tracking using multiple cameras. We neither detect nor track objects from a single camera, or camera pair; rather evidence is gathered from

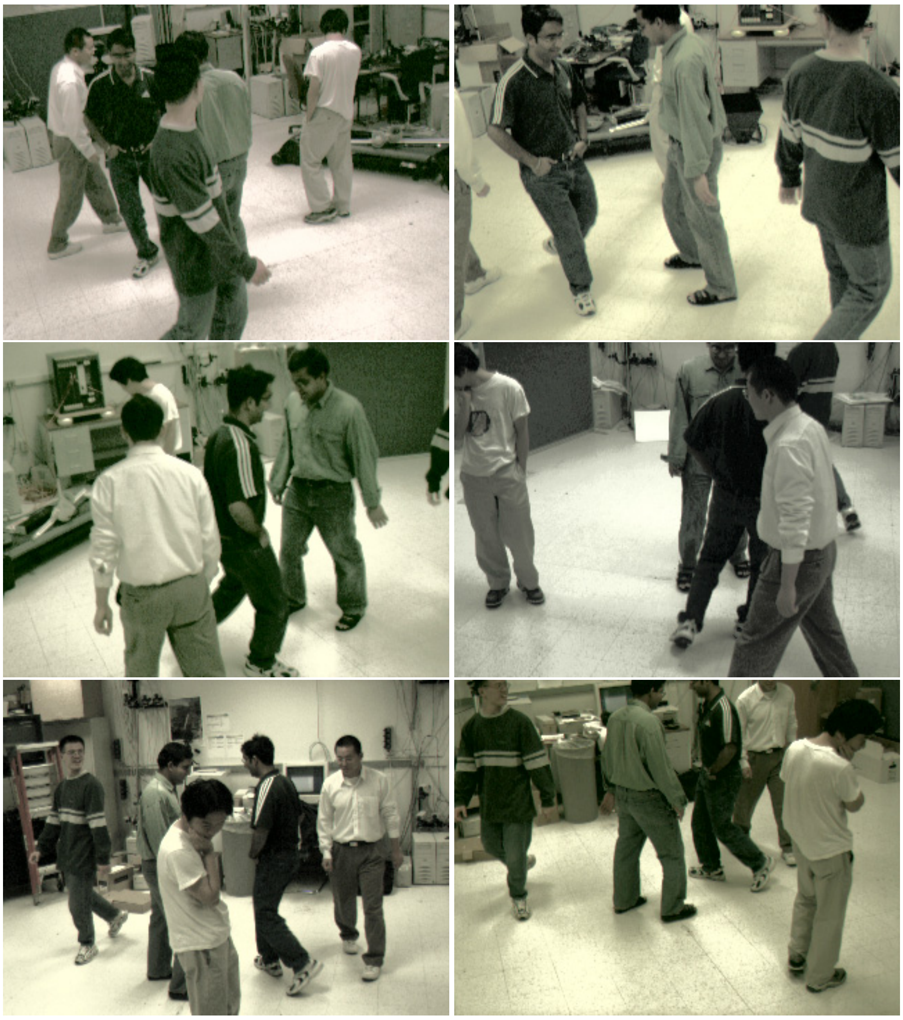

Figure 1. Images from a 6-perspective sequence at a particular time instant.

multiple camera pairs and decisions of detection and tracking are taken at the end by combining the evidence 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 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.

Good segmentation of people in a crowded scene is facilitated by models of the people being viewed. Unfortunately, the problem of detecting and finding the positions of the people requires accurate image segmentation in the face of occlusions. Therefore, we take a unified approach to the problem and solve both of them simultaneously. The algorithm uses segmentation results to find people's ground plane positions and then uses the ground plane positions thus obtained to obtain segmentations; the process is iterated until the results are stable. This helps us obtain both good segmentations and ground plane position estimates simultaneously.

The paper develops several novel ideas. 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 evidence 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 a 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.

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. (1998a) 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 (Haritaoglu et. al., 1998b), they incorporate stereo information into their system. Kettnaker and Zabih (1999a) 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.

Darrell et. al. (1998) developed a tracking algorithm which integrates stereo, color and face pattern detection. Dense stereo processing is used to isolate people from other objects and people in the background. Skin-hue classification is used to identify and classify body parts of a person and a face pattern detection algorithm is used to find the face of a person. Faces and bodies of people are then 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.

The DARPA VSAM project at CMU developed a system (Collins et. al., 1999) for video surveillance in an outdoor environment using multiple pan/tilt/zoom cameras. The system identifies blobs in the scene using motion detection and stores various information like blob size and color histogram for them. These blobs are then classified into different types of objects using neural networks.

Orwell et. al. (1999a) 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 (Orwell et. al., 1999b), Orwell et. al. present a multi-agent framework for determining whether different agents are assigned to the same object seen from different cameras.

Rosales and Sclaroff (1999) use a single camera tracking system that unifies object tracking, 3D trajectory estimation, and action recognition. An extended Kalman filter is used to compute trajectories, which are used to reason about occlusion.

These methods 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 (1998) 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. The nonrigidity of the human body is handled by matching points of the middle line of the human image using a Bayesian classification scheme. Features like location, intensity and geometric features are used for tracking.

Intille et. al. (Intille and Bobick, 1995; Intille et. al., 1997) 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 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 distribution) and hence may not be very useful for many applications. Therefore, we only consider cameras close to the height of the people.

Krumm et. al. (2000) 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 for 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. (2001). In contrast to (Krumm et. al., 2000) and (Darrell et. al., 2001), our approach utilizes the wide-baseline camera arrangement that has the following advantages:

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

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

(3) It provides many more camera pairs that can be integrated. Placing the cameras as far away from each other as possible and matching every one of them with every other using wide baseline stereo gives us C_2^n pairs. However, placing cameras in pairs of two for short baseline stereo yields only $n/2$ pairs for matching, using n cameras.

On the other hand, the short-baseline stereo pair camera arrangement used, e.g., in (Darrell et. al., 2001) has the advantages of

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

(2) better understood matching algorithms.

Our region matching 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, thus rendering traditional methods like correlation and sum of squared difference inappropriate. Although some work has been done to improve upon these methods(e.g. (Pritchett and Zisserman, 1998; Georgis et. al., 1995; Horaud and Skordas, 1989)), 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 work recently (for e.g. (Snow et. al., 2000)) addressing some of these issues, occlusion is still a major problem. Back-projection in 3D space without regard to color also yields very poor results in cluttered scenes, where almost all of the cam-

Algorithm 1 Algorithm Structure

(1) Initialize using previous ground plane positions. In the beginning, assume there are no estimates available and all people are new.
(2) Segment images using person models and ground plane position estimates.
(3) Estimate ground plane positions using region matching.
(4) Repeat steps 2 and 3 till ground plane position estimates are stable.
(5) Update person models using information from images and the ground plane positions found.
(6) Repeat steps 1-5 for next time step.

era view is occupied by the foreground. Some recent work (Kutulakos and Seitz, 2000; Faugeras and Keriven, 1998) has addressed some of these issues, but these methods fall in the category of full 3D surface reconstruction, which is very time-consuming and not possible for surveillance applications where computational time is a critical factor.

We do not 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. Clustering these points allows us to detect and track people.

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, "presence" probabilities along the horizontal direction at different heights, and the ground plane positions tracked using a Kalman filter. These models are used to segment images in each camera view. The regions thus formed are matched in pairs of camera views along epipolar lines. The matched segments are then used to yield 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 so that pairs which have a clear view of a particular location are given higher weight than those whose views are obstructed by some other person. The algorithm is then iterated using these new ground plane positions and this process is repeated until the ground plane positions are stable. These iterations help to improve the quality and stability of the results. 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 shape of the people in the scene. In order to simplify the problem, we assume that people are standing upright and that they do not squat or jump. These scenarios can also be included, but only at the cost of accuracy of the results since the models will have to be less discriminatory. These models are developed by observing people over time (method explained in section 11) and 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. The distance from the ground plane to the top of the person is divided into regions of equal length and a color model is developed for each region. A single color model for the whole person would not be able to capture the vertical variation in 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 for tracking and surveillance type of applications.

Pixels belonging to a particular person at a particular height are identified and the color model is developed using the well-known method of non-parametric kernel density estimation technique ((Elgammal and Davis, 2000)). Let $x_1, x_2, ..., x_n$ be n observations determined to be belonging to the color model. The probability density function can be non-parametrically estimated (Scott, 1992) using the kernel estimator K as

$$Pr(c) = \frac{1}{n}\sum_{i=0}^{n} K(c - c_i) \tag{1}$$

If we choose our kernel estimator function, K, to be the Normal function, $N(0, \Sigma)$, where Σ represents the kernel function bandwidth, then the density can be written as

$$Pr(c) = \frac{1}{n}\sum_{i=0}^{n} \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(c-c_i)^T \Sigma^{-1}(c-c_i)} \tag{2}$$

If we assume independence between the different channels with kernel bandwidth σ_j^2 for the jth channel, then

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & . & . & . \\ 0 & . & . & \sigma_d^2 \end{pmatrix} \tag{3}$$

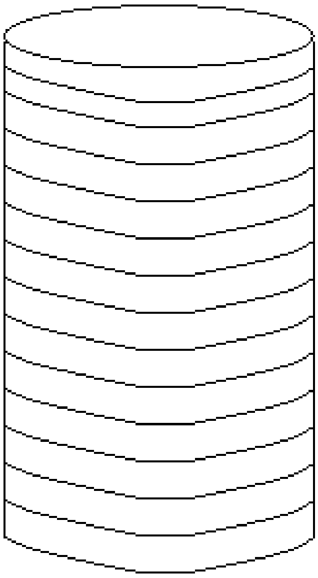

Figure 2. A color model is developed for each height slice of the person

Since the intensity levels change across cameras due to aperture effects, and due to shadow and orientation effects in the same view, we use normalized color (i.e. the ratios $r/(r+g+b)$ and $g/(r+g+b)$). In order to have some differentiation between colors that are very different from each other in intensity but similar in chromaticity (for e.g. white and black), we add intensity as the third variable with high variance.

Note that having this color model does not restrict us to detecting people wearing uniform color clothes. Non-uniform colors can be handled as long as there are kernels belonging to all the colors present at a particular height. This is possible for us to do since we are viewing the people from all sides and are therefore able to build a color model that captures the color profile from all sides and hence is not viewpoint dependent.

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 at height h and distance

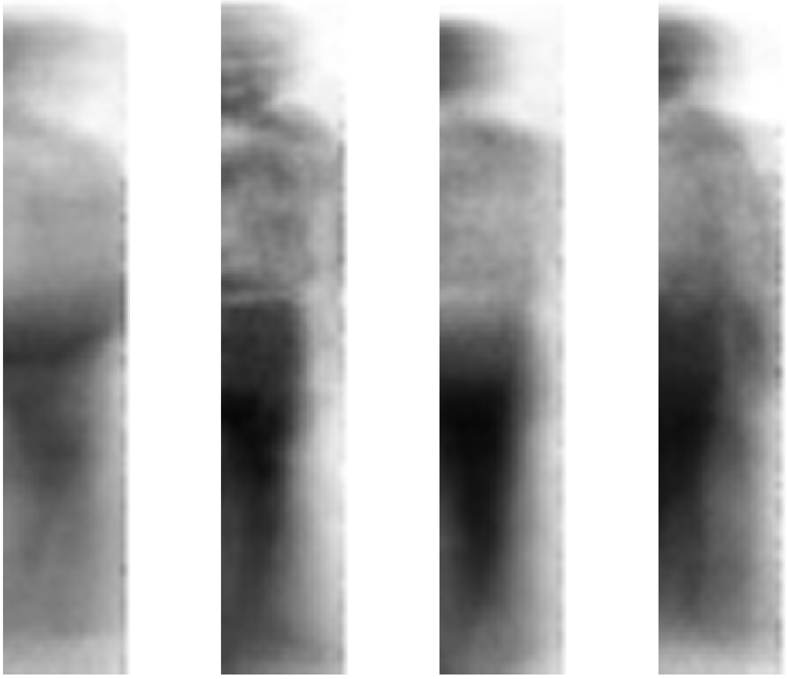

Figure 3. Sample Presence Probabilities of people observed over time. The x-axis is the width, y-axis is the height from the ground and the values shown are probabilities scaled by a factor of 256 for display purposes.

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.

We estimate this probabilty by observation over time using the method described in section 10 (see Figure 3 for some samples). However, since this method is reliable only after a certain number of time steps, we employ a heuristic to model the "presence" probabilities in cases where we do not yet have sufficient data available. We model the width W of the person at a particular height h as a random variable with Gaussian distribution and assume that the person occupies all space from the center up to width W. Then, the probability that a person j is present at distance w and height h is given by

$$L_j(h, w) = P_{j,h}(w < W) = 1 - CDFG_W(w)$$

where $CDFG_W$ is the cumulative density function for the Gaussian function for W. The mean and standard deviation for W is varied to accomodate the fact that people have different distributions at different heights.

The model for presence probability used here is very effective in detecting the torso of the person, but is not very effective in detecting hands and legs far away from the center of the person. However, since our goal here is to find the location of the person, it is more important to find the torso than the hands and legs. If we want to detect hands and legs far from the person, one would have to use some other model for the shape prior, or would have to modify our model so that hands and legs are handled appropriately.

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

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

The pixel is then classified as

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

$P(\mathbf{x}/j)$ 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 (Mittal and Huttenlocher, 2000).

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. We employ the following methodology. For each pixel $\mathbf{x}$, we project a ray in space passing through the optical center of the camera (see Figure 4). We find the perpendicular 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 these perpendicular lines(these line segments are horizontal). Then, the *a priori* probability that a pixel $\mathbf{x}$ is the image of person j is

$$P_{prior}(j) = L_j(h_j, w_j) \prod_{\textit{k occludes j}} (1 - L_k(h_k, w_k)) \qquad (4)$$

$$P_{prior}(background) = \prod_{\textit{all j}} (1 - L_j(h_j, w_j)) \qquad (5)$$

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.

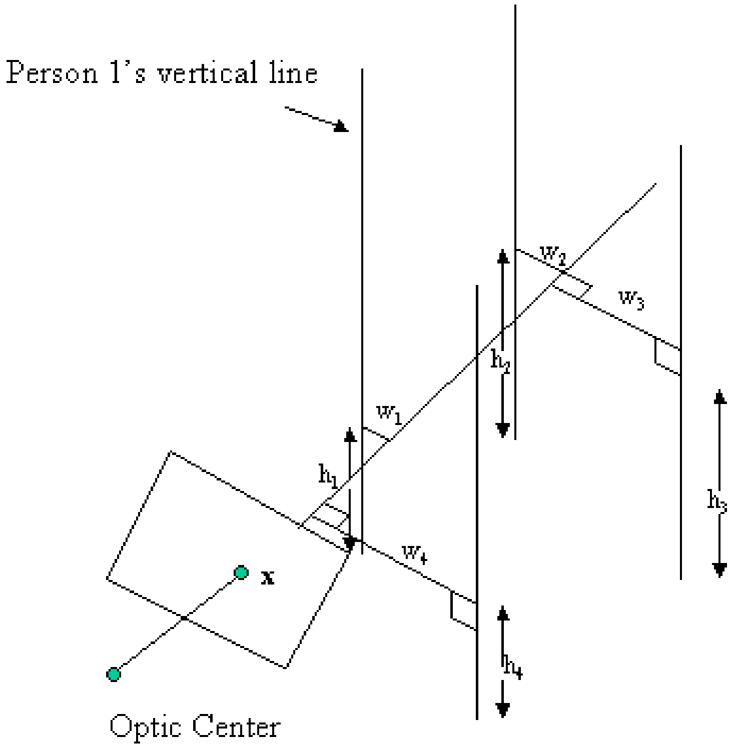

Figure 4. Measuring distances (and heights) from the line of sight

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), *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 discrimination from the background using our algorithm 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.

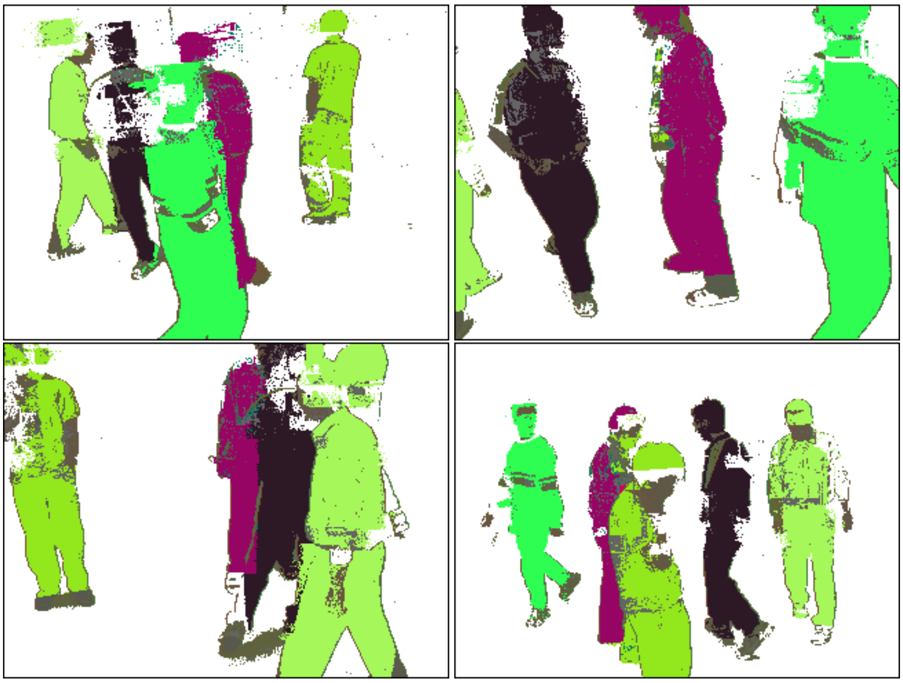

Figure 5. The result of segmenting four of the images shown in Figure 1

5.1. Detecting New People and Bootstrapping

The algorithm as described above assumes that we have information about the people visible in the scene and fails when there are people for whom we do not have any information or have inaccurate information. In order to detect "new" people in the scene and to correct inaccurate person models, 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. This method works not only for detecting "new" people entering the scene but also for bootstrapping the algorithm since in the beginning, all people are detected as "new".

6. Region-Based Stereo

After segmentation, we analyze epipolar lines across pairs of views. We first construct epipolar lines such that there are a fixed number of lines for two views. For instance, if the epipole lies inside a view, the lines are taken at an angular separation of $180/n$ where n is the number of lines desired. If the epipole lies outside the view, then the lines are constructed so that they have a fixed angular separation and cover the whole image. Along an epipolar line, we divide the line into "segments" belonging to different regions. If a line intersects a region twice, the two segments thus formed are merged to form one segment consisting of the two extreme end points. These segments from one camera view are matched to the segments in the other view. 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. Even if one segment matches to more than one segment in the other view, we do not select among these matches but consider all of the matched pairs as possible matches.

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 6).

The motivation for taking the point of intersection of the diagonals of the quadrilateral is that, 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 for 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 that 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 plance after projecting the 3D points onto it. It is also possible to do clusterng in 3D (as done by (Krumm et. al., 2000)) 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. Likelihood maps are

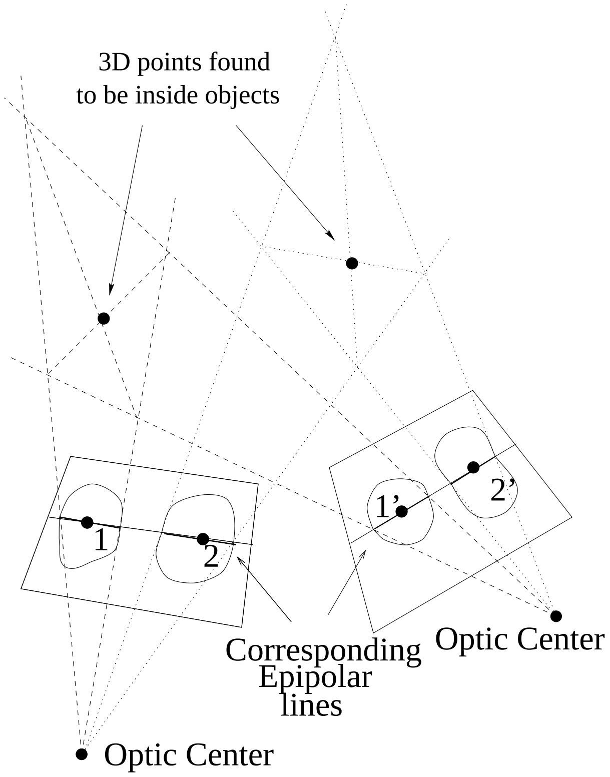

Figure 6. The point of intersection of the quadrilateral formed by back-projecting the end-points of the matched segments yields a 3D point lying inside an object. The matching segments are 1 and 1', and 2 and 2' respectively.

developed for each camera pair and are then combined in a robust manner using the occlusion information available.

7.1. Likelihood from a Single Camera Pair

We develop likelihood maps on the ground plane using the ground plane points found from region matching. For each of the 2D ground points, we add a Gaussian kernel to the likelihood map. The weight and 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). This helps us give higher weight to points originating from longer segments than from smaller ones. The likelihood from all these gaussian kernels is then integrated to obtain a likelihood 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 likelihood associated with any point x on the 2D plane is given by

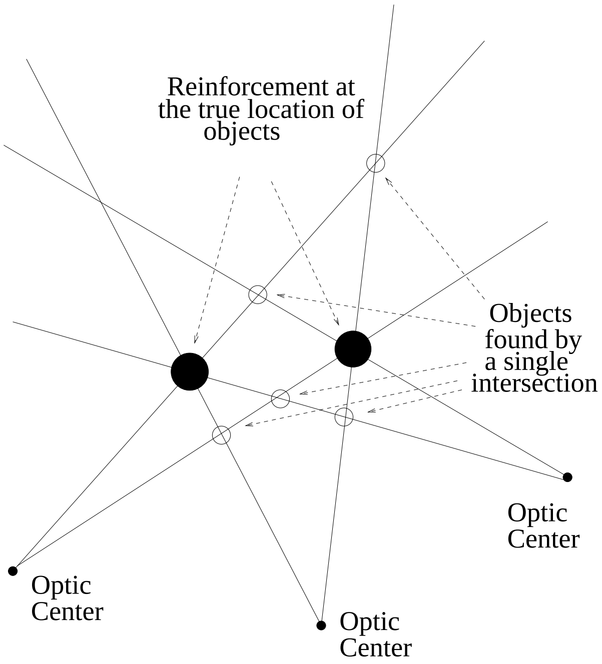
Reinforcement at the true location of objects
Objects found by a single intersection
Optic Center
Optic Center
Optic Center

Figure 7. 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.

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

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

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. The simplest method would average the likelihood estimates obtained from different pairs. This method does yield good results because the likelihood 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 7 where false matches are not reinforced but good ones are.

Although simple averaging of likelihood values yields good results, we can improve likelihood estimates significantly by determining (probabilistically) whether a particular location is visible or occluded from a camera and weighing the likelihood values accordingly.

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 $L_j()$ 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

$$P_c(\mathbf{x}) = \prod_{j \textit{ occludes } \mathbf{x}} (1 - L_j(w_j)) \tag{6}$$

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

$$wt_{(c1,c2)}(\mathbf{x}) = P_{c1}(\mathbf{x})P_{c2}(\mathbf{x}) \tag{7}$$

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

$$Lk(\mathbf{x}) = \frac{\sum_{(c1,c2)} wt_{(c1,c2)}(\mathbf{x}) Lk_{(c1,c2)}(\mathbf{x})}{\sum_{(c1,c2)} wt_{(c1,c2)}(\mathbf{x})} \tag{8}$$

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

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

$$\mathbf{x}_{centroid} = \frac{\sum_{\mathbf{x}, \mathbf{x} \in region} \mathbf{x} * Lk(\mathbf{x})}{\sum_{\mathbf{x}} Lk(\mathbf{x})} \tag{9}$$

where $Lk(\mathbf{x})$ is the likelihood at point $\mathbf{x}$.

These centroids of the object blobs are tracked over time using a different Kalman filter for each of the blobs. The state vector of the filter consists of position and velocity of the object and prediction is made using the assumption of constant velocity during the time step. The current observation of the object location is obtained by finding the centroid over a circular region around the current predicted object location.

Some simple heuristics are used to eliminate false detections, maintain robust person identities and to reidentify lost people who might be seen again. We keep track of the amount of time that a particular person has been tracked. If a person is not found near his predicted position by the ground plane position finding algorithm, then he is eliminated or retained depending on the amount of time he has been tracked continuously. Also, if two objects are too close to each other, then the one tracked for lesser duration is removed unless both of them have been tracked for some predefined duration. If a person that would normally have been removed is retained because of time considerations, then he is predicted to be at the position predicted by the Kalman filter for some time and this position and his models are used in the segmentation algorithm. However, after some time of non-detection, the person is removed from the list of tracked people and his model is stored in the "lost" people list. When a new person is detected, his model (mainly the color profile) is matched to the models for lost people and if it matches one of them, then he is identified as this "lost" person. We calculate the dissimilarity between two probability distribution functions P_1 and P_2 using the L_1 distance, i.e.

$$diss_{L_1}(P_1, P_2) = \int \mid P_1(\mathbf{x}) - P_2(\mathbf{x}) \mid d\mathbf{x} \tag{10}$$

The dissimilarity from the models at different heights is then simply summed up to get the dissimilarity measure between the color profiles of two people.

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. In order to develop accurate models, we want to identify pixels for which we are very sure that they belong to a particular person. To do so, for each pixel, we calculate the "presence" probabilities L_j for each person as described earlier. We use the observed probability method only when the number of observed pixels for a particular width w is above a certain threshold. Then 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

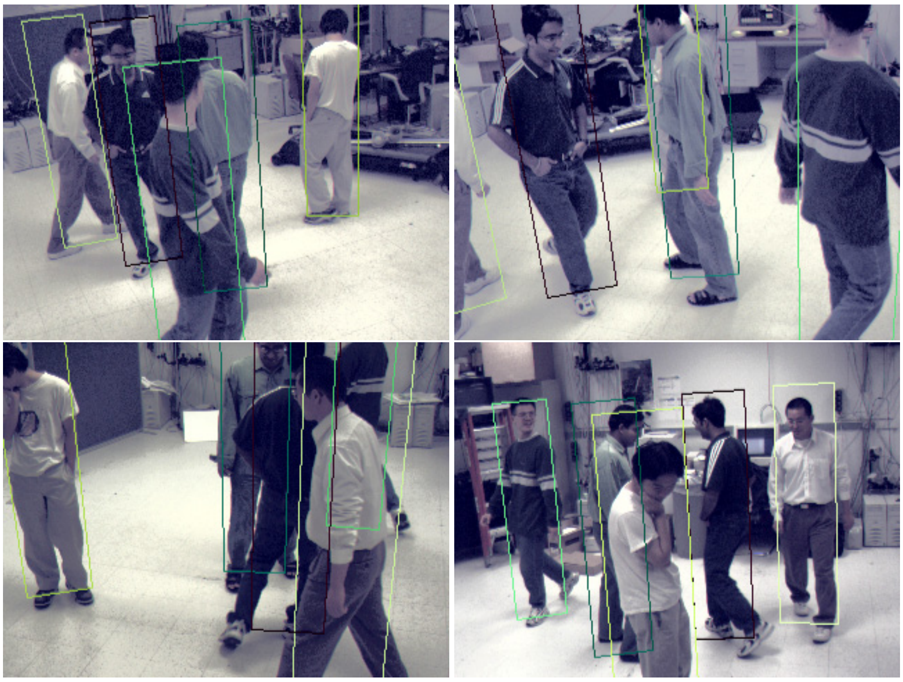

Figure 8. The results of detection and tracking as seen in four of the images shown in Fig. 1.

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 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.

For a new person, since we do not know the true height of the person, we develop models up to a maximum possible height. The true height of the person is estimated by observation over time and is used to delete the slices above that height.

10. Implementation and Experiments

Image sequences are captured using up to 16 color CCD cameras (Kodak ES-310C). 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. Eight cameras are located in approx-

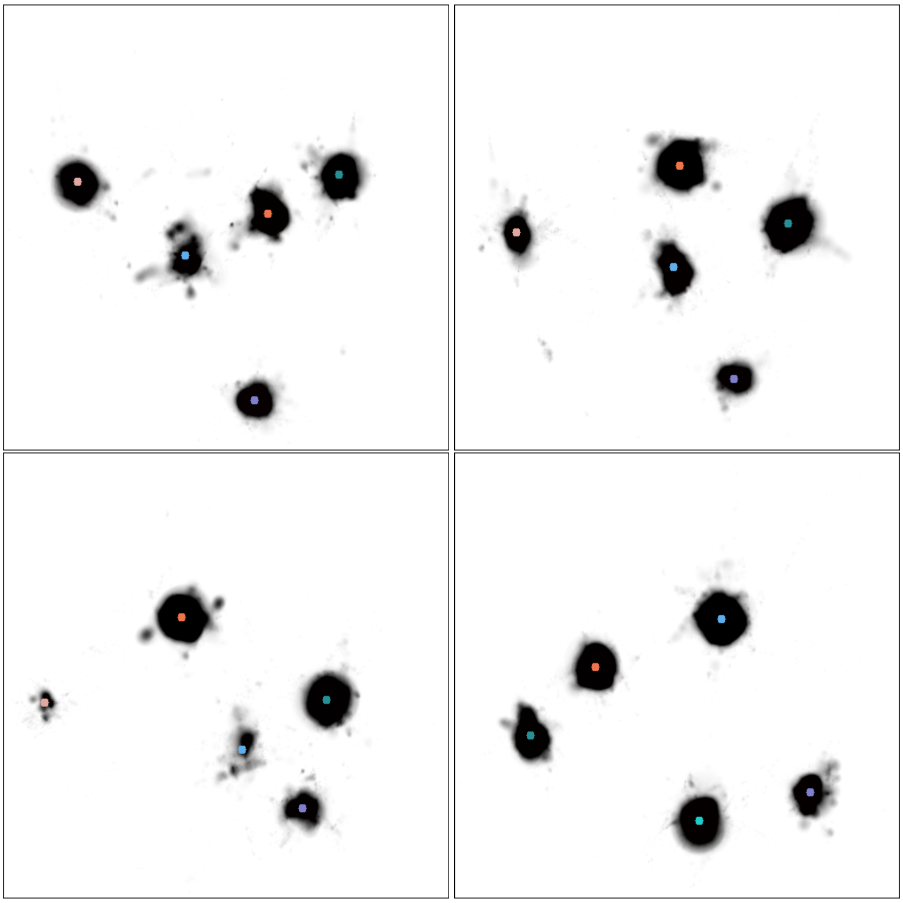

Figure 9. The first image is the likelihood map obtained for the image set shown in Figure 1 by applying the occlusion-analysis weighting scheme. The rest of the images are maps obtained for the sequence at other time steps. The dots show the position state variable of the Kalman filter tracking the person. Visit www.umiacs.umd.edu/˜anurag for the full sequence and some additional results.

imately a circle at a lower (in height) level and the other eight cameras are located directly on top of them at a higher level. Therefore, the angle between adjacent cameras at the same height is approximately 45 degrees. 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. For details see (Cutler et. al., 2000).

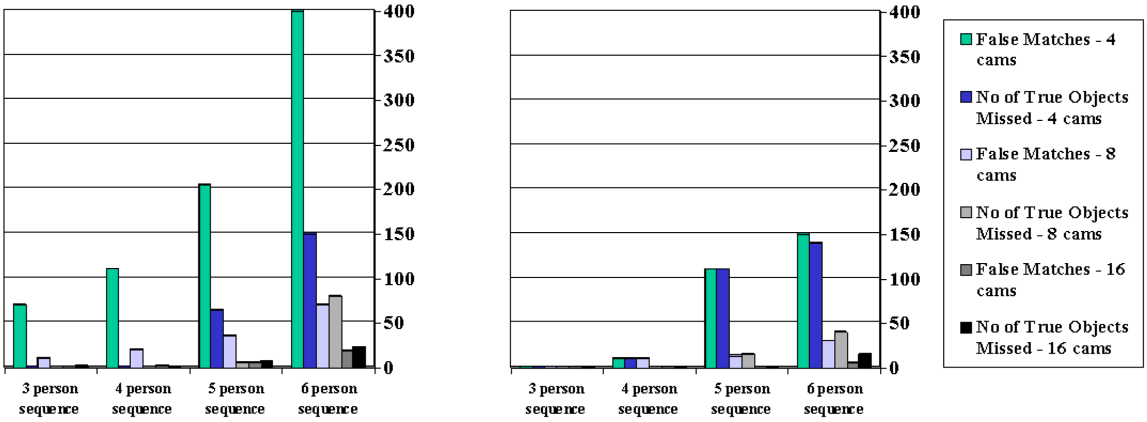

Figure 10. Cumulative errors for four sequences of 200 time steps each by (a) averaging likelihoods and using no occlusion analysis, and (b) using occlusion analysis

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 currently 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 describe experiments on four sequences containing 3, 4, 5 and 6 people respectively. Each sequence consisted of 200 frames taken at the rate of 10 frames/second and people were constrained to move in a region approximately 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, thus enabling us to determine the minimum number of cameras required to properly identify and track a certain number of objects. For the experiment with four cameras, cameras were placed at an angular separation of 90 degrees, for the eight camera experiment, only the top row of cameras were used (so the angular separation between adjacent cameras is

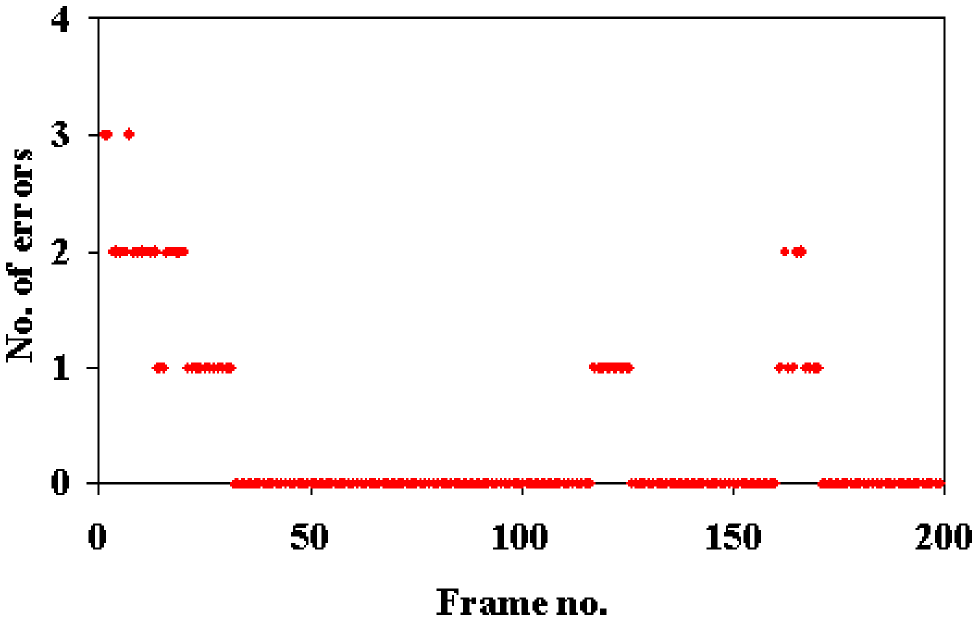

Figure 11. Total errors as a function of time for the sequence with 5 people using 8 cameras. Note how the errors decrease with time as the models become more robust. Errors after the initial period occur mainly because of people coming too close to each other.

about 45 degrees) and for the sixteen camera experiment, all cameras were used (again the angular separation between adjacent cameras is 45 degrees). The angular separation for us is at least 45 and up to 90 degrees, which is much more than that used by short baseline stereo cameras where the separation is of the order of 5 to 15 degrees. The cumulative errors over the 200 frames are shown in Figure 10(b). Also shown in Figure 10(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.

Behavior during Initialization

In the beginning, we have no information as to where the people are, or what their models are, so the algorithm is trying to find people based solely on color matching across views. The results from an algorithm that does detection and tracking based solely on color matching can be found in our earlier paper (Mittal and Davis, 2001). The results are decidedly of lower quality but are used to initialize the algorithm presented in this paper. Once an object has been detected, however, he can be tracked easily since we are able to build the models for him. Due to undetected people in the scene, the models developed for the detected people are sometimes not very accurate because pixels belonging to the undetected people are assigned to detected ones and are included in their models. Apart from inaccurate detection and tracking results, this also results in an increase in the number of iterations required. This is a potential problem, especially when the number of people is very large and occlusions are severe. However, for cases of moderate occlusions, we have found that this stage is very short and the undetected people are also found after some time, allowing the algorithm to build more accurate models and obtain very accurate results.

11. 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. It is fully automatic and does not require any manual input or initializations. It is able to handle occlusions and partial occlusions caused by the dense location of these objects and should be useful in practical surveillance applications.

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. (For a non-convex object, this point lies inside the convex hull of the object.)

We prove this with the help of an illustration showing the plane corresponding to the epipolar lines. (see Figure 12). 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

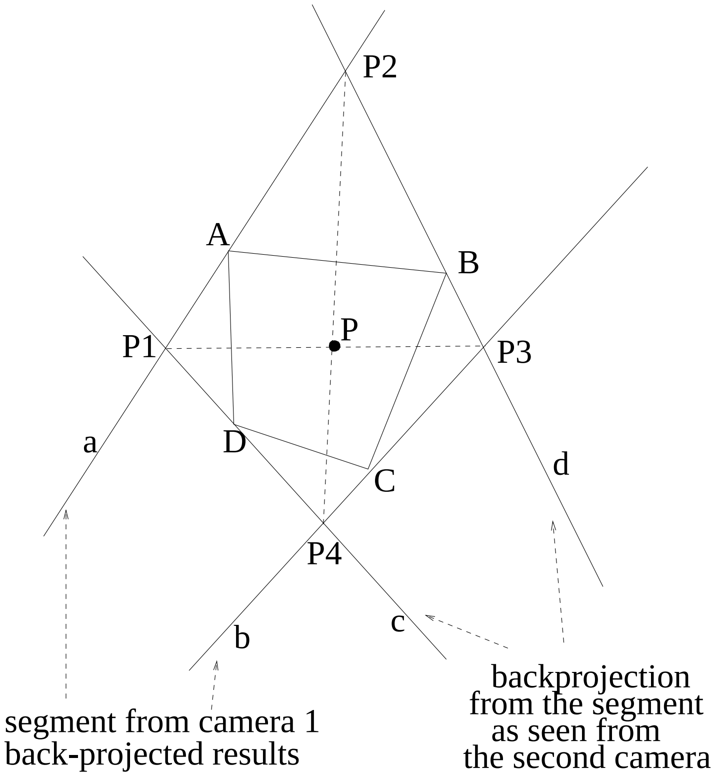
P2
A
B
P
P1
P3
a
D
C
d
P4
b
c
backprojection from the segment as seen from the second camera
segment from camera 1 back-projected results

Figure 12. 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

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

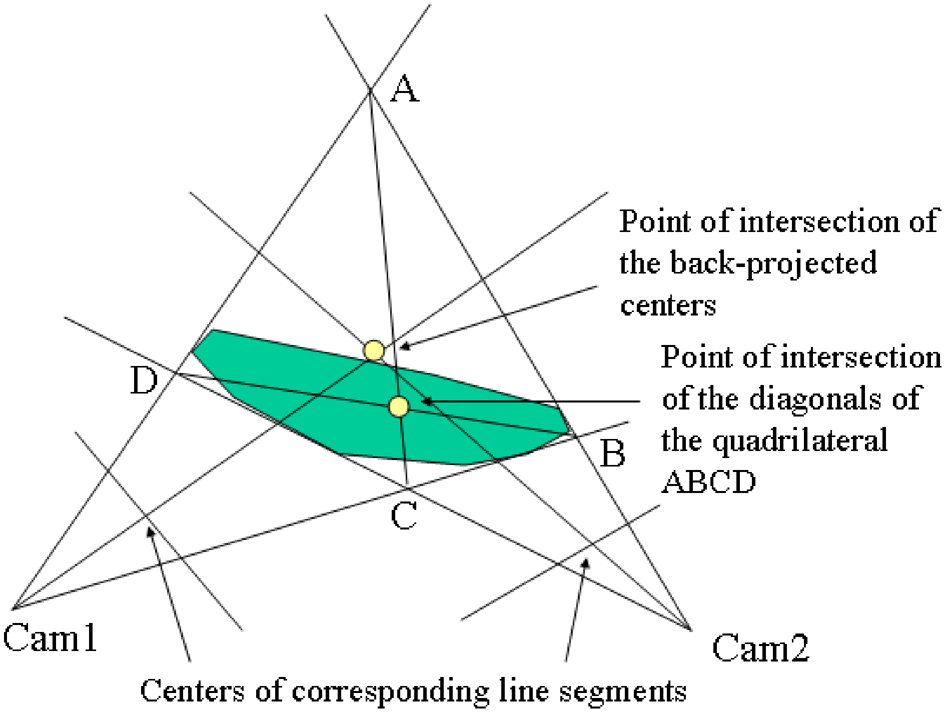
A
Point of intersection of the back-projected centers
Point of intersection of the diagonals of the quadrilateral ABCD
D
B
C
Cam1
Cam2
Centers of corresponding line segments

Figure 13. Shows that the standard method of intersecting the projection of the centers of corresponding segments may result in a 3D point outside the object, but the point of intersection of the diagonals of quadrilateral ABCD is always inside the object if the object is convex.

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.

Fig. 13 shows a case where the standard method of intersecting the projection of the centers of corresponding segments results in a 3D point outside the convex object, but the point of intersection of the diagonals of quadrilateral ABCD is inside it.

References

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.

Collins R.T., Lipton A.J., Fujiyoshi H. and Kanade T. 2001. Algorithms for Cooperative Multi-Sensor Surveillance. *Proceedings of the IEEE, Vol 89(10), October 2001,* pp. 1456-1477.

Cutler R.G., Duraiswami R., Qian J.H. and Davis L.S. 2000. Design and Implementation of the University of Maryland Keck Laboratory for the Analysis of Visual Movement. Technical Report, UMIACS, University of Maryland.

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.

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.

Elgammal A., David Harwood and Larry Davis. 2000. Non-parametric Model for Background Subtraction. In *6th European Conference on Computer Vision*, Dublin, Ireland.

Faugeras, O.D. and Keriven, R. 1998. Complete dense stereovision using level set methods. *European Conference on Computer Vision*,.

Gavrila D.M. and Davis L.S. 1996. 3D Model-Based Tracking of Humans in Action: A Multi-View Approach. *IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, CA, pp. 73-80.

Georgis N., Petrou M., and Kittler J.V. 1995. Obtaining correspondences from 2D perspective views with wide angular separation of non-coplanar points. In *Proceedings of the European- Chinese Workshop on Computer Vision,* pages 376-379.

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.

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.

Haritaoglu I., Harwood D., and Davis L.S. 1999. Hydra: Multiple People Detection and Tracking Using Silhouettes. *International Conference on Image Analysis and Processing,* Venice, Italy, pp 280-295.

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.

Intille S. S. and Bobick A. F. 1995. Closed-World Tracking. *5TH International Conference on Computer Vision,* Cambridge, MA, pp. 672-678.

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.

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.

Kettnaker V. and Zahih R. 1999. Bayesian Multi-camera Surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition,* Fort Collins, CO. pp. 117-123.

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.

Kutulakos K.N. and Seitz S.M. 2000. A Theory of Shape by Space Carving. *International Journal of Computer Vision,* Vol. 2000, 38(3), pp. 199-218.

MacCormick J. and Blake A. 1999. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *7th International Conference on Computer Vision,* Kerkyra, Greece, pp. 572-578.

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.

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.

Mittal A. and Davis L.S. 2002. M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo. In *Seventh European Conference on Computer Vision,* Copenhagen, Denmark.

Okutomi M., and Kanade T. 1993. A Multiple-Baseline Stereo. In *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 15, No. 4.

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.

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.

Pritchett P., and Zisserman A. 1998. Wide Baseline Stereo Matching. In *Sixth International Conference on Computer Vision*, Bombay, India, pp. 754-760.

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.

D.W.Scottt, *Multivariate Density Estimation*. Wiley-Interscience, 1992.

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.

Swain M.J. and Ballard D.H. 1991. Color Indexing. International Journal of Computer Vision, vol. 7, pp. 11-32.

Want R., Hopper A., Falcao V., and Gibbons J. 1992. The Active Badge Location System. *ACM Transactions on Information Systems*, vol. 10, no. 1, pp 91-102.

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.

List of Figures

List of Algorithms