Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 103 (2006) 33–51
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Integrating multiple levels of zoom to enable activity analysis

Paul Smith *, Mubarak Shah, Niels da Vitoria Lobo

Computer Vision Laboratory, School of Computer Science, University of Central Florida, Orlando, FL 32816, USA

Received 10 July 2004; accepted 20 February 2006
Available online 18 April 2006

Abstract

In this paper, we present a multi-zoom framework for activity analysis in situations requiring combinations of both detailed and coarse views of the scene. The epipolar geometry is employed in several novel ways in the context of activity analysis. Detecting and tracking objects in time and consistently labeling these objects across zoom levels are two necessary tasks for such activity analysis. First, a multiview approach to automatically detect and track heads and hands in a scene is described. Then, by making use of epipolar, spatial, trajectory, and appearance constraints, objects are labeled consistently across cameras (zooms). Finally, we demonstrate how multiple levels of zoom can cooperate and complement each other to help solve problems related to activity analysis.

Keywords: Activity analysis; Action recognition; Multiple cameras; Multiple zoom; Multi zoom

1. Introduction

Activity recognition is being aggressively studied in computer vision for a range of applications from surveillance to gesture recognition. A growing number of surveillance applications are utilizing forests of sensors for increased monitoring over large areas. These cameras generally have a low zoom to cover as much area as possible, yielding valuable tracking information and overall scene context. Other work in activity recognition has focused on facial expression analysis and gesture recognition using highly zoomed cameras of the face and hands. By sacrificing overall scene context the higher zooms gain valuable detailed, subtle cues about specific events in the scene. There is currently a gap between the surveillance class of applications, where the cameras generally have a low zoom and subjects are tracked simply as blobs, and the gesture analysis applications which analyze highly detailed images of faces, hands, and eyes. Several researchers have hinted at the possibility of combining multiple cameras with different levels of zoom for improved activity analysis [1,2]. Therefore, we propose an approach for activity analysis to integrate the strengths in coarse views, mid level views, and fine views.

In many problem domains there are certain regions in the scene where detailed (highly zoomed) monitoring is needed. In other areas only a coarser view of the scene is needed. Consider an office environment where someone is working at a desk. Many actions one would perform in this environment involve the head, such as talking, using the phone, looking at something, eating, coughing, putting on eye glasses, etc. A coarse view of the scene can give information about the origin and destination of hand-held objects and about such matters as how fast the hands are approaching the face. A finer view around the facial region would be able to provide more detailed information such as where on the face the action occurred, where the person is looking, whether the person is talking or not, what kind of object is being brought to the face and so on. In this context, it would be helpful to have multiple cameras employing varying degrees of zoom to accomplish activity analysis.

To achieve multi-zoom activity analysis, the following problems need to be solved. (1) The head and hands need

* Corresponding author. Fax: +1 407 823 5419.
E-mail addresses: rsmith@cs.ucf.edu (P. Smith), shah@cs.ucf.edu (M. Shah), niels@cs.ucf.edu (N. da Vitoria Lobo).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.02.006

Fig. 1. Example of scene showing zoom 1, zoom 2, and zoom 3 views.

to be automatically detected and tracked in each view. (2) Objects need to be consistently labeled across views. (3) The cameras must cooperate to perform activity analysis. We have experimented with a camera configuration in which there is a hierarchy of $N \geqslant 3$ zooms which give various degrees of detail in the scene shown in Fig. 1. The non-planarity of the scene requires these problems to be solved using the epipolar geometry. We assume the epipolar geometry of the scene is known, but it could be learned as in [3].

Our key contributions are the following: we first demonstrate a bootstrapping process that automatically finds the head and hands in the video sequences. Our approach utilizes dynamic color models and multi camera cooperation to achieve better recognition than was possible with independent cameras. Then a method for consistently labeling objects across multiple cameras (each camera with a different zoom) is presented. Innovations of our algorithm include incorporating not only epipolar, spatial, and appearance information, but also integrating trajectory matching. Finally, it is shown how the zoom levels can be combined to give better activity analysis capability. Preliminary results of this method appeared in [4].

The organization of the paper is as follows. Related work is discussed in Section 2 and some mathematical conventions are given in Section 3. Section 4 presents the method of detecting and tracking the heads and hands. Section 5 demonstrates how to establish uniform labeling of objects across cameras. Section 6 presents the action analysis module. In Section 7, results are discussed and finally we conclude.

2. Related work

Activity recognition is an important problem in computer vision, and there has been an increasing amount of research done in this field in recent years [5,6]. The problem of integrating multiple levels of detail (MLOD) to improve activity recognition is not as well studied. This paper provides a formulation for studying MLOD in the context of activity analysis.

In [7], multiple cameras are used to cover non-overlapping regions to recognize activities. They introduce the Abstract Hidden Markov mEmory Model to analyze activities, which allows them to utilize the inherent hierarchical structure of activities. Their approach is used to cover large spatial environments, however they do not attempt to use multiple levels of detail to perform finer action recognition. In [8], a large scene is monitored and people and vehicles are tracked automatically. Three dimensional world coordinates are determined for all objects. Though the system does not make any inferences as to what kinds of activities are occurring. All information is passed to an operator for evaluation.

An active vision system is presented in [9] using one static and one Pan-Tilt-Zoom (PTZ) camera to identify and track multiple people. This approach makes a number of restrictive assumptions on the color of people's clothes and number of people present. No activity analysis capabilities are demonstrated.

By combining multiple cameras in an active vision system with stereo vision, Hongo et al. [10] is able to perform head and hand tracking and limited gesture recognition. Their correspondence only considers horizontal epipole line information and object size. A multiple camera approach is given in [11] to detect events for an intelligent meeting room, however they do not use the high zoomed cameras for activity analysis. In both these systems the camera positions are known beforehand. We have tried to avoid active vision systems (i.e., PTZ and foveating cameras) in our approach to focus on integrating multiple zooms levels simultaneously.

A key element of any multi camera activity analysis system is the consistent labeling of objects across cameras. An obvious option would be to compute the full 3D alignment using stereo. Basic stereo methods will fail because the assumption of the standard stereo setup is violated [12]. Even after applying polar rectification [13] to our image pairs and then attempting the methods in [14, 15], these direct methods fail because polar rectification cannot resolve the ambiguities in occlusion and illumination changes across the cameras.

In [16], a feature based method is used, in which the feature point matches are picked randomly. Then a homography is estimated and an error function is minimized which allows the best guesses to help contribute to a better estimate in the next round. In our case however, we do not have a ground plane to work with, which they require, and we have a full 3D scene. As noted in [17], the approach is also sensitive to noise and match ambiguities. Work presented in [18] attempts to find the fundamental matrix and establish trajectory correspondences in 3D scenes. However, their method does not take full advantage of appearance, trajectory, and spatial properties, which we have

found adds more robustness to finding the consistent labeling across cameras.

In [19], the rank constraint is used to find linearly dependent trajectories. In this way similar trajectories can be grouped together for classification. While they achieve good results, if multiple trajectories in multiple views move similarly then there is ambiguity between which trajectories are most similar. Further, the method could not be extended to our trajectory matching because it cannot handle matching degenerate trajectories, like stationary objects.

A method is presented in [20] to track across wide field of views. They use epipolar, homography, landmark, apparent height, and apparent color to resolve ambiguities. However, the system assumes common illumination across the cameras. We use a better appearance comparison using energy minimization. They neglect to use trajectories themselves, which also provide us a valuable cue to alignment. Further, their approach would have problems without ground plane calibration.

Work done by [21] show how depth and color information are combined to track multiple people in a scene using a pair of stereo rigs. Appearance and spatial information are both used to acquire matching trajectories across views. In [22], range data are acquired from stereo pairs to match trajectories across views. Pixel data from multiple views are integrated in a late-segmentation strategy. Each pixel is checked against all trajectories estimated over time.

In [23], correspondences are acquired using segmentation and epipolar geometry with information combined from multiple cameras. Their method relies on ground plane calibration and will not work as we have no ground plane. Multiple views with widely different zooms are not considered.

While there has been much work done in multicamera surveillance integrating multiple zooms simultaneously has not been well studied. Our work provides an algorithm for making high level inferences about activities using multiple zooms. Further, because no consistent labeling (i.e., correspondence) algorithms were successful in our test cases a new method needed to be developed.

3. Definitions and conventions

There are many good references on the details of multiview geometry. Refs. [24, 25] provide good introductory knowledge. Only the minimum foundations needed for our purposes are presented here. A pair of cameras are related by the fundamental matrix, so all points in image I can then be transferred to their corresponding epipolar line in I' by $l = p \cdot \mathbf{F}$, where $l = [\alpha \quad \beta \quad \gamma]$ are the coefficients of the line

$$\alpha \cdot r + \beta \cdot c + \gamma = 0, \tag{1}$$

p is any point in I, $\mathbf{F}$ is the fundamental matrix, and r, c are the row and column indices of point p. All epipolar lines will pass through the epipole, found from $\mathbf{F}$ by taking its singular value decomposition, $\mathbf{F} = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^{\mathrm{T}}$. The epipoles are obtained by normalizing the last columns of $\mathbf{V}$ and $\mathbf{U}$. To transfer an epipolar line to image coordinates normalize l. For lines with slope $|m| > 1$ apply Eq. (2):

$$p_1 = l \times [0 \quad 1 \quad 0]^{\mathrm{T}} \quad \text{and} \quad p_2 = l \times [0 \quad -1/Y \quad 1]^{\mathrm{T}}, \tag{2}$$

where Y is the height of the image and p_1, p_2 are the intersection points of the image with the epipolar line l. The slope m is the ratio of the coefficients $\frac{\alpha}{\beta}$. A slightly modified operation gives the intersection points for lines with slope $|m| \leqslant 1$:

$$p_1 = l \times [1 \quad 0 \quad 0]^{\mathrm{T}} \quad \text{and} \quad p_2 = l \times [-1/X \quad 0 \quad 1]^{\mathrm{T}}, \tag{3}$$

where X is the width of the image and p_1, p_2 are the intersection points of the image with the epipolar line l.

Now consider N cameras (we show 3) using zoom 1 through zoom N (for us, zoom 1 through zoom 3). Let C_i be the camera number with zoom i. Define $I_{i,f}$ to be a color image at frame, f, taken from camera C_i. Define the set of objects in a particular image frame as $X_{i,f} = \{x^1_{i,f}, \ldots, x^m_{i,f}\}$, where i is the camera number and f, $1 \leqslant f \leqslant Z$, is the frame number. m represents the number of objects in a particular frame. An object is defined by its bounding box (top left, bottom right corners). The centroid of $x^k_{i,f}$ can be represented as the vector: $[\hat{x}^k_{i,f} \quad \hat{y}^k_{i,f}]^{\mathrm{T}}$. We would like to determine the consistent labeling between all objects in the various sequences. For a given frame f we have the set $T = \{X_{1,f}, \ldots, X_{N,f}\}$ expanded as $T = \{\{x^1_{1,f}, \ldots, x^{m_1}_{1,f}\}, \{x^1_{2,f}, \ldots, x^{m_2}_{2,f}\}, \ldots, \{x^1_{N,f}, \ldots, x^{m_N}_{N,f}\}\}$, which is the set of all objects for frame f. We would like to find the mapping

$$w(x^k_{n,f}) = \left\{x^{a_1}_{b_1,f}, x^{a_2}_{b_2,f}, \ldots, x^{a_p}_{b_p,f}\right\}$$

which takes a particular object k in frame f viewed from camera n, and finds the corresponding object a_k with $1 \leqslant a_k \leqslant m_{b_i}$ for all cameras b_i, $1 \leqslant b_i \leqslant N$, $b_i \neq n$, if the object is visible. m is subscripted to stress that the number of objects can vary between frames and/or cameras.

4. Detection and tracking of heads and hands

For activity analysis, the heads and hands first need to be detected, tracked, and labeled across cameras. This section deals with detection and tracking of heads and hands. Our approach first finds the head regions and then builds color models of these regions which are used to find the hands. The head regions are detected independently for each camera, C_a, employing the object detector described in [26].

Using the RGB pixel values of the head region, a color model, h_a, is built for each, C_a, as in [27]. However in [27], the remaining color pixel values are treated as negative samples. This will not produce a good color model in our case because the hand regions will count as negative samples. To overcome this limitation, after building an initial color model using the positive sampled regions, the final color model is only negatively weighted by those samples

which did not show up positively in the initial color model. This prevents the hand regions from contributing adversely to the final color model and provides better segmentation. An appropriate threshold can be chosen to make a binary decision,

$$H_a(r,g,b) = \begin{cases} 1, & h_a(r,g,b) > a, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

which can then be used to segment the images.

Since the head detector is for frontal head regions only, the color model will be helpful for detecting hands and heads with small variations in viewpoints. Fig. 2 shows the input images in column one. Color segmentation output and head detection is in column two. Detected heads were drawn with rectangles around them.

Once a detected head given by the head detector has been present for more than four frames, a mean shift [28] tracker is initialized around this head region, which will provide tracking information in subsequent frames. There is no limitation to how many heads can be in the scene at one time. An alternative approach would be to initialize mean shift trackers around head regions whose centroids project to epipolar lines that intersect found head regions in all other views.

Next the hands must be found and tracked in each view. We could simply track all skin colored regions that were found from the head color model, but this has problems as there are many spurious skin regions marked. Better detection is possible using multiple cameras. First for frame f, all possible hand candidates are independently labeled in each camera, C_a, using H_a. Hand candidates are those connected components that have size

$$\sum_i H_a(I_{a,f}(x_i, y_i)) \geqslant \delta \cdot \Phi_1,$$

where Φ_1 is the average head size in this camera, $\delta = .05$, and $\sum_i$ occurs over the connected component. The computation is performed at all levels of detail.

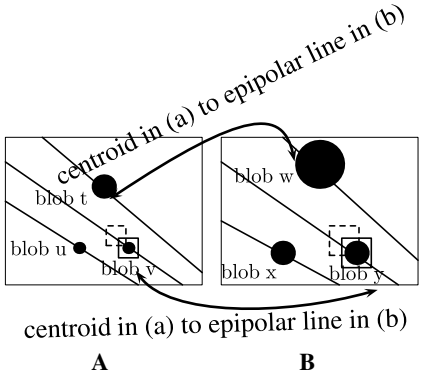

Fig. 3. Unambiguous hand labeling. Three blobs are shown in (A). Blob t is the head and has already been identified in the first stage. It is shown (along with its epipolar line projections in both views for completeness). Blobs u and v are hand candidates. Blob v in (A) has its centroid projected to its epipolar line in (B). This line in (B) is searched for a matching, unambiguous hand candidate. It can be seen that there is a single hand candidate (blob y) on this epipolar line. This is an unambiguous match. Since the match is unambiguous, a mean shift tracker would be initialized around blob v in (A) and blob y in (B). This process starts the tracking for the matching hand candidates in both views. Similarly, the hand candidate blob u in (A) has its centroid projected to its epipolar line in (B). This line is then searched and since a single hand candidate, blob x, is on this epipolar line, mean shift trackers would be initialized around each of these blobs in both views.

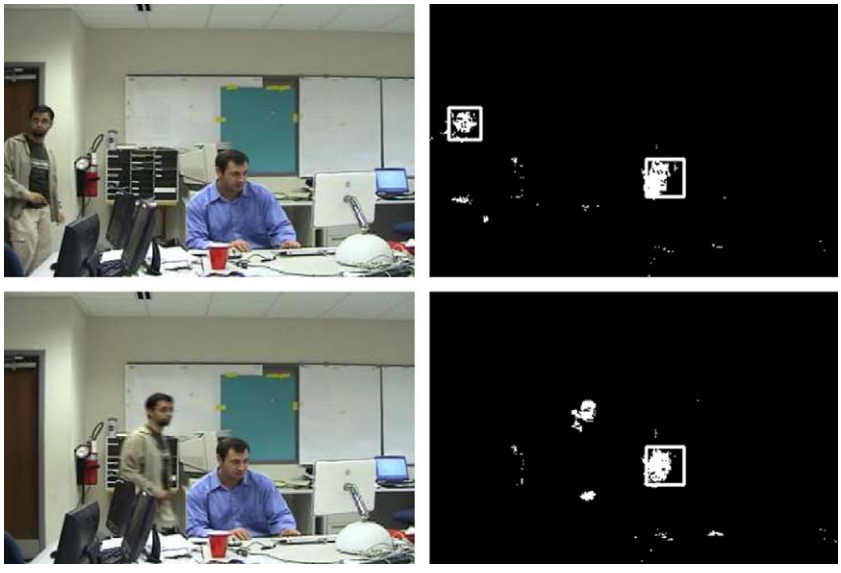

Fig. 2. Output from the head detector and color segmentation. Found head regions are marked by rectangular boxes, and color pixels belonging to the head color model are marked as white. The first row is frame 3 in zoom 1. The second row shows frame 162 in zoom 1. Though no explicit color model has been generated for the hands, they show up reliably even for multiple people. In row 1 both heads are found, but later in the sequence (row 2) the head detector misses one head, though the color segmentation still finds both head regions as skin regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Once all candidate hand regions are labeled, the epipolar geometry is used to confirm or reject the presence of a hand on an epipolar line in another view. Fig. 3A represents a lower zoomed image and Fig. 3B represents a higher zoomed image. Three objects (one head and two hands) and the corresponding epipolar lines of the objects from the other view are shown in each image. For each hand candidate in C_a its centroid is projected to an epipolar line in C_b. The epipolar line is searched for a region with size $\epsilon \cdot \Phi_2$, where Φ_2 is the average head size in this camera and ϵ is a small positive constant. If only one region is found on the corresponding epipolar line in C_b then a mean shift tracker is initialized around these regions in both views, and the regions are tracked. If there are multiple hand candidates along this line, the search is deemed ambiguous, and no mean shift trackers are introduced. This can be seen in Fig. 4. This method is able to successfully detect and track the head and hands. Figs. 5–8 show automatic initialization of the hands. In all cases subfigures (a) and (c) are the same frame, f, taken from zoom 1 and 2, respectively. Subfigures (b) and (d) are the same frame, f', taken from zoom 1 and 2, respectively.

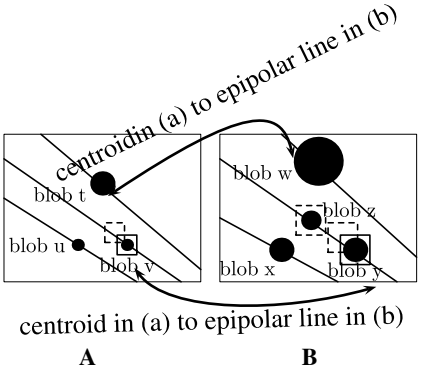

Fig. 4. Ambiguous hand labeling. Three blobs are shown in (A). Blob t is the head and has already been identified in the first stage. Blob v in (A) has its centroid projected to its epipolar line in (B). This line is searched and it is found that there are two hand candidates, blobs y and z on this epipolar line, thus a mean shift tracker would not be initialized around any of these regions. This is so because there is an ambiguity as to which hand candidate in (B) corresponds to the hand candidate in (A).

It often happens that the hand partially overlaps or occludes the face. When one or both hands overlap with the face the mean shift tracks will find the same candidate region (shown in Fig. 9). In this case, the algorithm will use one of the tracks for all the overlapping regions. Once the regions separate, the proposed initialization procedure will find and reinitialize the regions. The algorithm then can continue tracking these regions using geometrical domain knowledge based on which side of the face the hand was on.

When there are multiple head and hand regions and when there are other objects that need to be tracked, a consistent set of labels across cameras for all objects will be necessary. A method to establish these consistent labels across cameras is presented next.

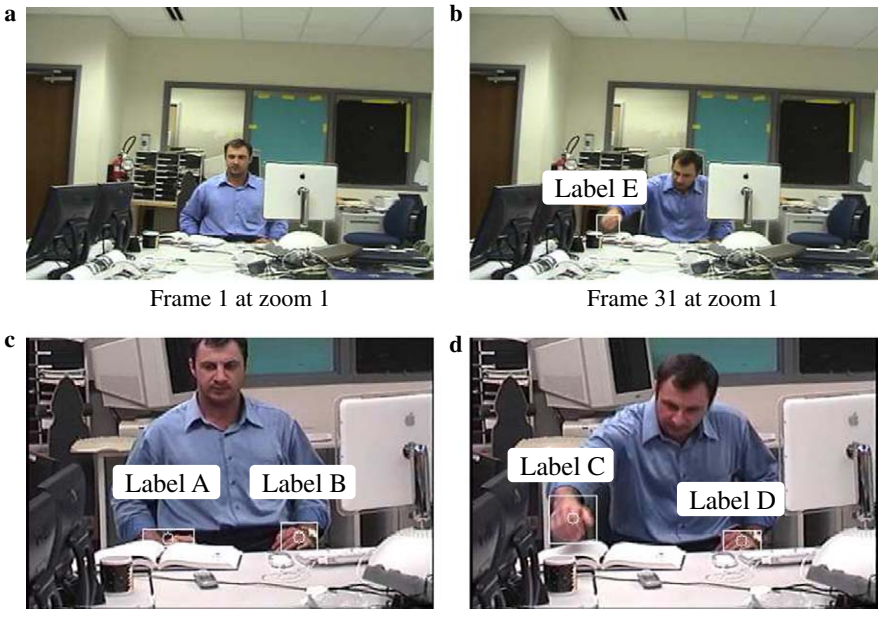

Fig. 5. In (c) Labels A and B indicate the found two hand candidates. Each hand candidate has a box around it. Since no matching hand candidates have been found in (a), these hand candidates are not tracked in subsequent frames. For frame 31 in (d) two hand candidates, Labels C and D, are found. In (b) a single hand candidate Label E is also found. Labels C and E are not ambiguous (according to the detection method), so mean shift tracks are initialized around both of these corresponding regions. Since Label D in (d) has no corresponding hand candidate in (b) no mean shift tracker is initialized around Label D.

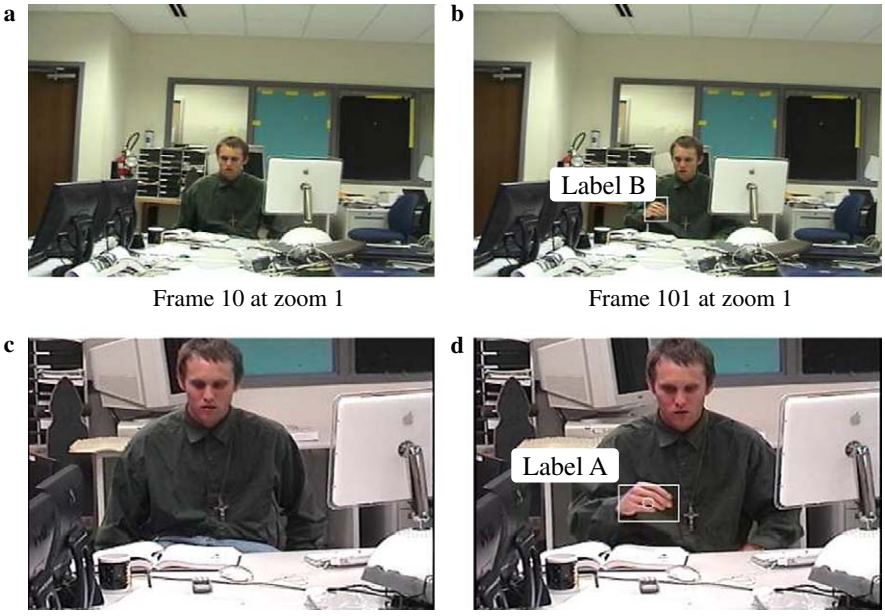

Fig. 6. In frame 10 there are no hand candidates in either (a) or (c). In frame 101 in (d) the hand candidate labeled A is found. In (b) a hand candidate, Label B, is also found. These hand candidates are not ambiguous so mean shift tracks are initialized around both of these hand candidates in both zooms.

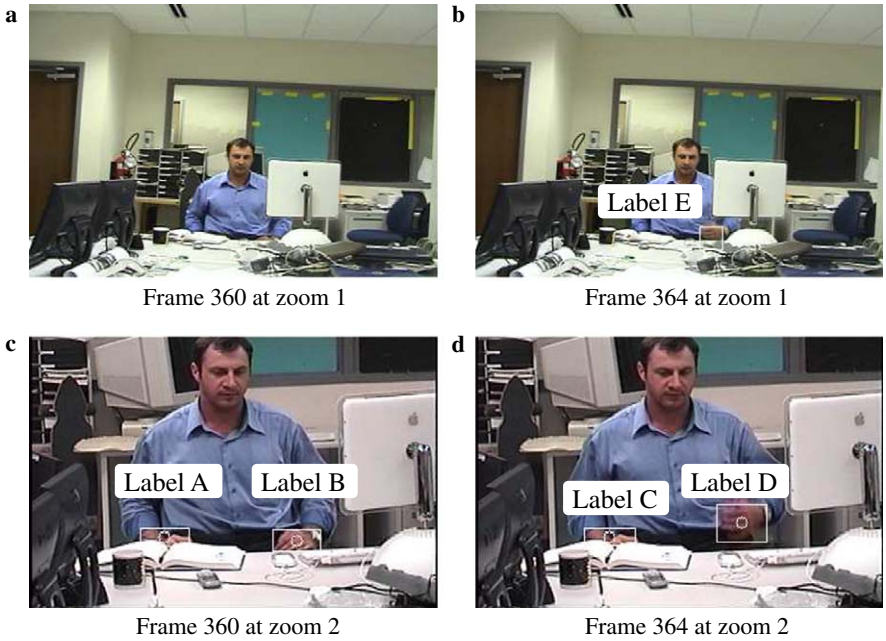

Fig. 7. In (c) Labels A and B indicate the found hand candidates. Since no hand candidates that match have been found in zoom 1 (a), these hand candidates are not tracked in subsequent frames. For frame 364 in (d) Labels C and D indicate the found hand candidates. In (b) a single hand candidate, Label E, is also found. Since Labels D and E are unambiguous, mean shift tracks are initialized around both of these corresponding regions in (b) and (d). Since hand candidate Label C in (d) has no corresponding hand candidate in (b) no mean shift tracker is initialized around Label C in (d).

5. Establishing consistent set of labels across cameras

To allow the cameras to communicate object information to one another, a method to determine the consistent set of labels across cameras needs to be found. For simplicity we will describe our method using two cameras. The ideas can easily be extended to work with additional cameras. Given two cameras, C_a and C_b, we want to determine the consistent set of labels for objects between cameras for frame j (see Section 3 for a precise definition).

Our approach uses the following constraints:

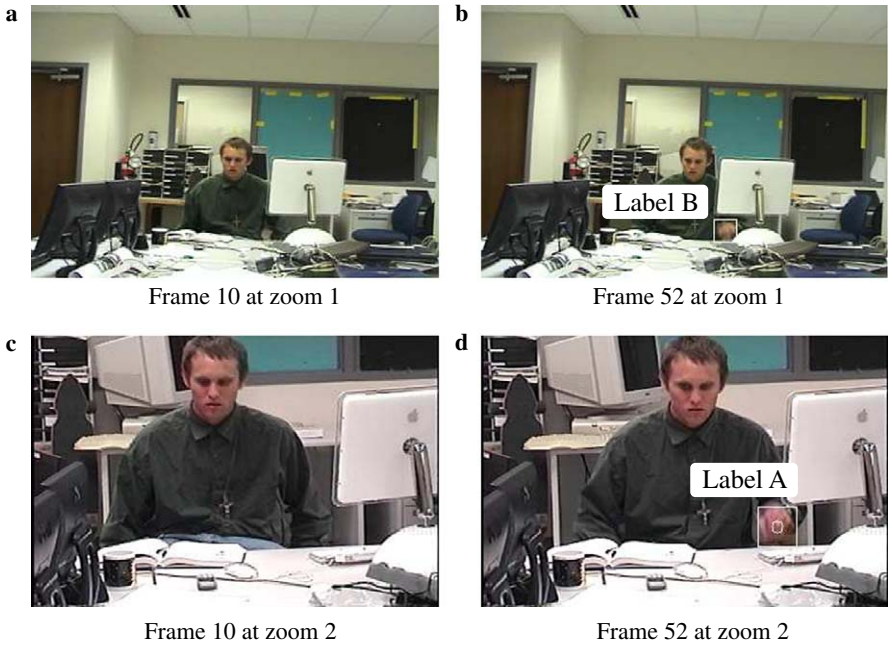

Fig. 8. In frame 10 there are no hand candidates in either (a) or (c). For frame 52 in (d) hand candidate, Label A, is found. In (b) Label B is also found. These hand candidates are not ambiguous so mean shift tracks are initialized around both of these hand candidates in both zooms.

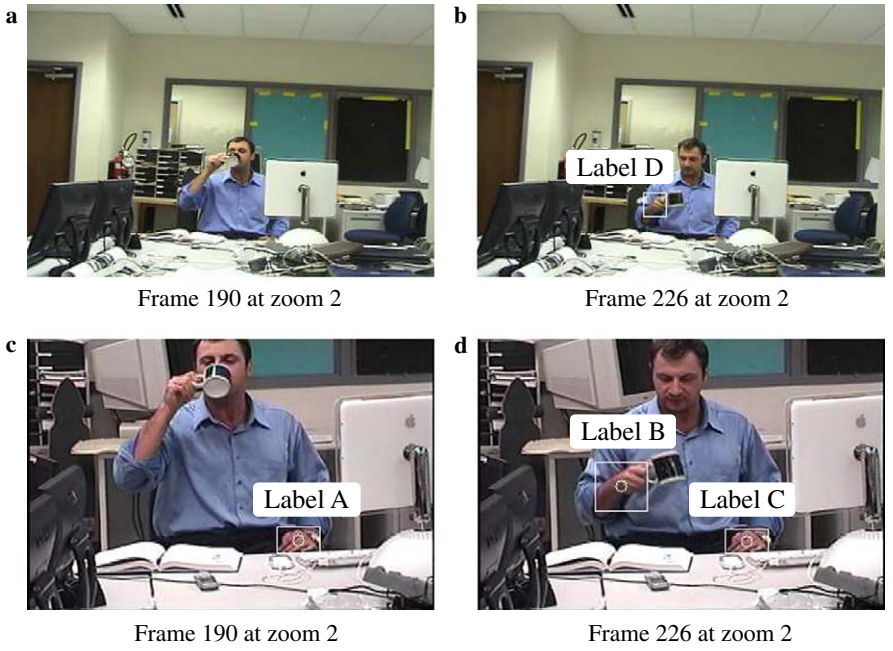

Fig. 9. In (c) the Label A is found as a hand candidate, though this hand candidate cannot be seen in (a). Since there is a partial overlap occurring with the head and other hand, this hand is not considered a hand candidate in either (a) or (c). Since no matching hand candidates have been found in (a), the hand candidates are not tracked in subsequent frames. Frame 226 occurs after the occlusion. In (d) hand candidates Labeled B and C are found. In (b) a single hand candidate, Label D, is also found. Since Labels B and D are unambiguous, mean shift tracks are initialized around both of these corresponding regions. Label C in (d) has no corresponding hand candidate in (b) so no mean shift tracker is initialized around it.

- epipolar line projections for each object
- spatial constraints
- trajectory constraints
- appearance constraints for each object.

The algorithm starts by transferring the object centroids in C_a to their corresponding epipolar lines in C_b. The distance between each epipolar line and each centroid in C_b can be accumulated and thought of as a matching error

between the object in C_a that generated the epipolar line and the object in C_b. A distance of zero indicates a good match. This is done for every frame in the sequence. The best match can be selected as the epipolar line/centroid pair with the lowest error. This leads to the following algorithm.

1. For the fth frame $\forall m$ objects: $X_{a,f} = \{x^1_{a,f}, \ldots, x^m_{a,f}\}$ make a set of all centroids, $\mathbf{P_a} = \{[\hat{x}^1_{a,f} \quad \hat{y}^1_{a,f}]^{\mathrm{T}}, \ldots, [\hat{x}^m_{a,f} \quad \hat{y}^m_{a,f}]^{\mathrm{T}}\}$ in camera C_a. Transfer these centroids using the fundamental matrix to get the set, $\mathbf{A}$, of corresponding epipolar lines

$$\{l_1, \ldots, l_m\} = \{\{[\hat{x}^1_{a,f} \quad \hat{y}^1_{a,f} \quad 1]\mathbf{F_{a,b}}\}, \ldots, \{[\hat{x}^m_{a,f} \quad \hat{y}^m_{a,f} \quad 1]\mathbf{F_{a,b}}\}\}$$

   in camera C_b that correspond to the centroids $\mathbf{P_a}$ from C_a.
2. Make a set of centroids $\mathbf{P_b} = \{[\hat{x}^1_{b,f} \quad \hat{y}^1_{b,f}]^{\mathrm{T}}, \ldots, [\hat{x}^n_{b,f} \quad \hat{y}^n_{b,f}]^{\mathrm{T}}\}$ in camera C_b $\forall n$ objects: $X_{b,f} = \{x^1_{b,f}, \ldots, x^n_{b,f}\}$. There is no requirement for $n = m$. If the ith object of $C_a, x^i_{a,f}$ is visible in C_b it will lie on some epipolar line l_k. So $\forall [\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f}]^{\mathrm{T}} \in \mathbf{P_b}$ and $\forall l \in \mathbf{A}$ the error for this match is the Euclidean distance between the centroid and the epipolar line

$$\mathrm{d}(l, [\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f} \quad 1]) = \frac{|l_\alpha \cdot \hat{x}^j_{b,f} + l_\beta \cdot \hat{y}^j_{b,f} + l_\gamma|}{\sqrt{l_\alpha^2 + l_\beta^2}}. \tag{5}$$

   This distance is the error to match the object, $x^i_{a,f}$, in C_a (whose epipolar line is l) with the object, $x^j_{b,f}$, in C_b. l_α, l_β, l_γ are the coefficients of l, the epipolar line with parameters described in Eq. (1). We can compute the accumulated distance error for every centroid $p \in \mathbf{P}_b$ in C_b with every epipolar line for every frame and match the objects that had the lowest error.

More formally given an object x^i_a in C_a, to find the corresponding object in all other cameras C_b compute:

$$\forall b \neq a \text{ obtain } \operatorname*{argmin}_j \frac{1}{N^{i,j}_{a,b}} \times \sum_{f=1}^{N^{i,j}_{a,b}} \mathrm{d}([\hat{x}^i_{a,f} \quad \hat{y}^i_{a,f} \quad 1]\mathbf{F_{a,b}}, [\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f} \quad 1]), \tag{6}$$

where b is the index of the bth camera. $N^{i,j}_{a,b}$ is the number of frames for which objects i, j had valid tracks in cameras a, b, respectively. $\mathbf{F_{a,b}}$ is the fundamental matrix between cameras a and b. The function d is given in Eq. (5). We have verified that slightly better results can be achieved by modeling the error measure as a gaussian zero mean random variable

$$\mathrm{d}'(l, [\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f} \quad 1]) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\left(\frac{q^2}{2\sigma^2}\right)}, \tag{7}$$

where q is the value of Eq. (5). We used Eq. (7) in our experiments.

In the above algorithm a method was presented that finds the labeling going from C_a to C_b. It is desirable for the matching to be commutative (if the number of objects differ across cameras, i.e., $n \neq m$, then the matching occurs only in the direction with less objects), so that

$$(x^i \text{ in } C_a \text{ matches } x^j \text{ in } C_b) \Longleftrightarrow (x^j \text{ in } C_b \text{ matches } x^i \text{ in } C_a).$$

Unfortunately, if the algorithm is computed from C_b to C_a the labeling might not be the same. Eq. (6) can give different minimums going in different directions. This can happen, for example, when multiple centroids in C_a lie on nearly coincident epipolar lines in C_b. The next three constraints provide additional restrictions on matched objects to help reduce the incorrect labelings due to these ambiguities.

5.1. Spatial constraints

When two object centroids in one view project to nearly coincident epipolar lines in another view, it is difficult to determine which line belongs to which object using solely a Euclidean based distance criteria. In the case of coincident epipolar lines, the distance metric described in the previous section might not match the correct objects. In our camera setup the spatial ordering of objects across cameras must be preserved. We can use this fact to make a better determination as to which match is correct. The difficulty is in determining which object matches are to be penalized. Consider Fig. 10B to illustrate the difficulty. The red box

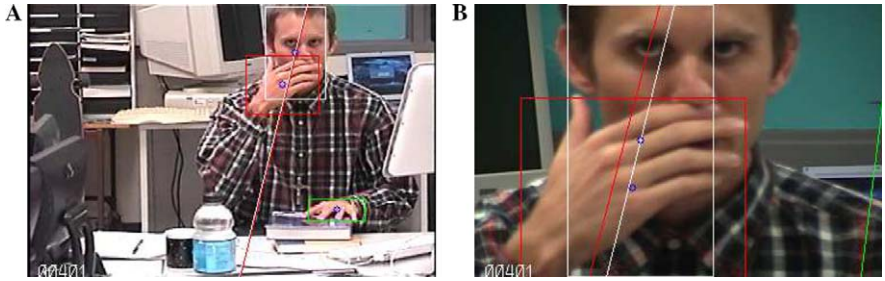

Fig. 10. One type of spatial inconsistency. The head and hand bounding boxes intersect in both views. The first spatial constraint tests for intersecting bounding boxes. If the boxes intersect in one view, then intersecting boxes in other views are checked for consistency and penalized if necessary. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

indicates the hand track. The white box indicates the head track. The small blue circles indicate the centroids of each of these bounding boxes (the lower centroid corresponds to the hand). The red and white lines correspond to the similarly colored bounding boxes in 10A. Using the distance criteria both blue centroids are closest to the white epipolar line (corresponding to the head in 10A). Which match is the correct one and which should be penalized? To aid in resolving this ambiguity, we consider the two cases: in the first case, the bounding boxes of the objects intersect each other in both views, shown in Fig. 10. In this case, the object matches to be penalized can be easily identified.

Concretely, we proceed in the following manner. Given two objects in C_a: $x^{\mathrm{d}}_{a,f}$, $x^{e}_{a,f}$ and two objects in C_b: $x^{g}_{b,f}$, $x^{h}_{b,f}$, check the following condition

$$x^{\mathrm{d}}_{a,f} \bigwedge x^{e}_{a,f} \text{ and } x^{g}_{b,f} \bigwedge x^{h}_{b,f} \text{ with } x^{\mathrm{d}}_{a,f} \Psi p x^{e}_{a,f} \text{ implies } x^{g}_{b,f} \Psi p x^{h}_{b,f},$$

where $\bigwedge$ represents intersection between bounding boxes. Ψ represents an operator that compares the ordering of the bounding boxes along the axis (x or y) that the bounding boxes are furthest apart on, and p is the parity indicating the direction of the comparison operator Ψ. If this condition is not met, the spatial constraint has been violated and the match between $x^{\mathrm{d}}_{b,f}$ and $x^{g}_{b,f}$ is penalized by the Euclidean distance between the centroids: $\sqrt{[\hat{x}^{g}_{b,f} \quad \hat{y}^{g}_{b,f}][\hat{x}^{h}_{b,f} \quad \hat{y}^{h}_{b,f}]^{\mathrm{T}}}$. It will be shown later in this section how to integrate this penalty into the original error minimization.

In the second case, shown in Fig. 11, the bounding boxes of the skateboard and book do not intersect but the epipolar lines are almost coincident, which will result in the epipolar distance minimization possibly selecting the incorrect labels. To resolve this, recall that every epipolar line was generated by a known object in C_a. The two objects in C_b nearest the coincident epipolar lines and the original centroids in C_a that generated the coincident epipolar lines are checked for spatial consistency. The object matches to be penalized can, thus, be easily identified. Concretely, the minimum distance between the two epipolar lines

$$[\hat{x}^{\mathrm{d}}_{a,f} \quad \hat{y}^{\mathrm{d}}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}} \text{ and } [\hat{x}^{e}_{a,f} \quad \hat{y}^{e}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}}$$

is computed. The minimum distance between points on the line will occur at one of the end points of the image, which can be found from Eqs. (2) and (3). If $\min < \varepsilon$, then the lines are nearly coincident and the respective centroids in C_b that are closest to either of the epipolar lines $[\hat{x}^{\mathrm{d}}_{a,f} \quad \hat{y}^{\mathrm{d}}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}}$ or $[\hat{x}^{e}_{a,f} \quad \hat{y}^{e}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}}$ are identified. Only the situation where two centroids in C_b have as their closest epipolar line either $[\hat{x}^{\mathrm{d}}_{a,f} \quad \hat{y}^{\mathrm{d}}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}}$ or $[\hat{x}^{e}_{a,f} \quad \hat{y}^{e}_{a,f} \quad 1]\mathbf{F}_{\mathbf{a},\mathbf{b}}$ is considered because it allows us to unambiguously identify which objects to compare ($x^{g}_{b,f}$ and $x^{h}_{b,f}$) and penalize. If the condition

$$x^{\mathrm{d}}_{a,f} \Psi p x^{e}_{a,f} \rightarrow x^{g}_{b,f} \Psi p x^{h}_{b,f}$$

does not hold then the match between $x^{\mathrm{d}}_{b,f}$ and $x^{g}_{b,f}$ is penalized by the Euclidean distance between the centroids: $\sqrt{[\hat{x}^{g}_{b,f} \quad \hat{y}^{g}_{b,f}][\hat{x}^{h}_{b,f} \quad \hat{y}^{h}_{b,f}]^{\mathrm{T}}}$. Again, it will be shown later in this section how to integrate this penalty into the original error minimization.

5.2. Trajectory constraints

The spatial constraints may not resolve all ambiguities due to inaccuracies in the tracking or fundamental matrix. The spatial constraints work well, but stringent requirements must be satisfied to make use of them. Therefore, a more broadly applicable trajectory constraint is introduced. From a high level, the trajectory constraint looks at all possible pairs of objects across views and penalizes them according to how dissimilar their motion is (based on the previous 30 frames). We address the following three cases: (1) if the motion of both objects is negligible, no penalty is assessed as the motion vectors cannot be reliably obtained. (2) If the motion of both objects is large, then a penalty is assessed based on the relative direction of the motion vectors. (3) If the motion of one is negligible and the other is large, a penalty is assessed based on the current match score (as one of the motion vectors cannot be reliably obtained).

The correct correspondences across cameras will be penalized least since their motion is most similar. This constraint ensures that moving objects in one view match with similarly moving objects in another view.

Fig. 11. A second type of spatial inconsistency. In this case the bounding boxes of the skateboard and book do not intersect but the epipolar lines are almost coincident. This could result in incorrect labeling. The second spatial constraint penalizes label matches that overturn the order of the centroids.

Formally, the trajectory constraint penalizes object x^i_{af} in C_a matching object x^j_{bf} in C_b by adding to $S(i,j)$ the amount

$$T_{i,j,f} = \begin{cases} 0 & \text{for } M^i_{a,f} < 1 \text{ and } M^j_{b,f} < 1 \\ \Delta\theta_{i,j,f} S(i,j) * (.00001) & \text{for } M^i_{a,f} > 1 \text{ and } M^j_{b,f} > 1 \\ S(i,j) * \ .00001 & \text{otherwise,} \end{cases} \tag{8}$$

where

$$S(i,j) = \sum_{s=1}^{f} \Big(\mathrm{d}'([\hat{x}^i_{a,s} \quad \hat{y}^i_{a,s} \quad 1]\mathbf{F_{a,b}}, [\hat{x}^j_{b,s} \quad \hat{y}^j_{b,s} \quad 1]) + \Gamma_A(s) \sqrt{[\hat{x}^j_{b,s} \quad \hat{y}^j_{b,s}][\hat{x}^{h_{s,l}}_{b,s} \quad \hat{y}^{h_{s,l}}_{b,s}]^{\mathrm{T}}} + T_{i,j,s} \Big) \tag{9}$$

is the current cumulated un-normalized match score between objects x^i_a and x^j_b. $\Gamma_A(s)$ is an indicator function, A is the set of frames in which the spatial constraint is met and $h_{s,l}$ is the index of the centroid that violated the spatial constraint. It is subscripted by l to emphasize that it is possible for a single object pair to be involved in multiple spatial constraint violations.

$$M^i_{a,f} = \sqrt{[\hat{x}^i_{a,f} \quad \hat{y}^i_{a,f}][\hat{x}^i_{a,f+j} \quad \hat{y}^i_{a,f+j}]^{\mathrm{T}}}$$

represents the maximum motion in a 30 frame sliding window for any single object i in a particular camera C_a for a particular frame f. $M^i_{a,f} = \emptyset$ when, there is no bounding box information for x^i_a in this 30 frame window. To find j compute

$$j = \underset{j}{\operatorname{argmax}} \sqrt{[\hat{x}^i_{a,f} \quad \hat{y}^i_{a,f}][\hat{x}^i_{a,f+j} \quad \hat{y}^i_{a,f+j}]^{\mathrm{T}}}$$

$\Delta\theta_{i,j,f}$ represents the difference between the angle of the maximum motion vectors for each object

$$\Delta\theta_{i,j,f} = \arccos \frac{\mathbf{M}^i_{a,f} \cdot \mathbf{M}^j_{b,f}}{\|\mathbf{M}^i_{a,f}\| \|\mathbf{M}^j_{b,f}\|}.$$

$\mathbf{M}^i_{a,f}$ is the maximum motion vector computed from $M^i_{a,f}$:

$$\mathbf{M}^i_{a,f} = [\hat{x}^i_{a,f} - \hat{x}^i_{a,f+j} \quad \hat{y}^i_{a,f} - \hat{y}^i_{a,f+j}]^{\mathrm{T}}.$$

Since $0 \leqslant \arccos(\Delta\theta_{i,j,f}) \leqslant \pi \ \forall \ \Delta\theta_{i,j,f}$, there is no issue of angles becoming imaginary or wrapping around 2π.

5.3. Appearance constraints

Previous methods have considered color similarity of objects between views to increase the accuracy of the label assignments. This is important when there are small errors in the track data or epipolar geometry which cause the objects to be matched incorrectly. Directly comparing the appearance of objects can present difficulties especially when the cameras are not color calibrated. Relative color similarity between objects still can give useful information. At an abstract level we can consider all permutations of object matches from one camera to another. Suppose there are two objects, A, B in Camera 1 and two objects A', B' in Camera 2. One permutation would be A matches to A' and B matches to B'. Another permutation would be A matches to B' and B matches to A'. Given a permutation we can find the appearance score of this match by computing the average intensity difference between the corresponding objects in the permutation.

Concretely, after applying the previous constraints to all frames, if there are still ambiguous matches (i.e., those objects for which there is not a 1-1 mapping), then collect these ambiguous objects into two lists. The ambiguous objects in C_a are $A = \{x^1_a, \ldots, x^q_a\}$ and those in C_b are $B = \{x^1_b, \ldots, x^q_b\}$, where q is the number of ambiguous objects. To get the correct matches, find the permutation of superscript indices in B to minimize the relative error:

$$p = \underset{P}{\operatorname{argmin}} \sum_{i=1}^{|A|} \left(\frac{1}{M} \sum_{x \in x^i_a}^{M} \bar{I}_a(x) - \frac{1}{N} \sum_{x \in x^{p_i}_b}^{N} \bar{I}_b(x) \right)^2, \tag{10}$$

where P is the set of all permutations of the indices of ambiguous objects in B. Each p is a set of indices of objects in B. $\bar{I}(x)$ represents the image intensity at x. Fig. 12 shows some results of the labeling algorithm. The tracks that are colored the same were matched across views. In Section 4, the method automatically finds the heads and hands. To test the accuracy of the labeling algorithm, we have manually introduced additional bounding boxes around other objects. The algorithm correctly labels all objects across all views. More results are presented in Section 7. We show the final function that needs to be minimized to satisfy all constraints. Given an object x^i_a in C_a, to find the corresponding object in all other cameras C_b compute:

$$\forall b \neq a \text{ obtain } \underset{j}{\operatorname{argmin}} \frac{1}{N^{i,j}_{a,b}} \sum_{f=1}^{N^{i,j}_{a,b}} \Big(\mathrm{d}'([\hat{x}^i_{a,f} \quad \hat{y}^i_{a,f} \quad 1]\mathbf{F_{a,b}}, [\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f} \quad 1]) + \Gamma_A(f) \sqrt{[\hat{x}^j_{b,f} \quad \hat{y}^j_{b,f}][\hat{x}^{h_{f,l}}_{b,f} \quad \hat{y}^{h_{f,l}}_{b,f}]^{\mathrm{T}}} + T_{i,j,f} \Big) + \Gamma_C(x^i_a, x^j_b) \Omega(P), \tag{11}$$

where b is the index of the bth camera. $\Gamma_A(f)$ and Γ_C are indicator functions, A is the set of frames in which the spatial constraint is met and C is the set of objects for which the appearance constraint is met. $h_{f,l}$ is the index of the centroid that violated the spatial constraint. It is subscripted by l to emphasize that it is possible for a single object pair to be involved in multiple spatial constraint violations. $\Omega(P)$ is the penalty amount found from Eq. (10).

6. Combining multiple zooms for improved activity analysis

After performing tracking and labeling across cameras, the next step is to use the multiple levels of detail for improved activity analysis. We demonstrate with three scenarios the capability of our system to use multiple levels of

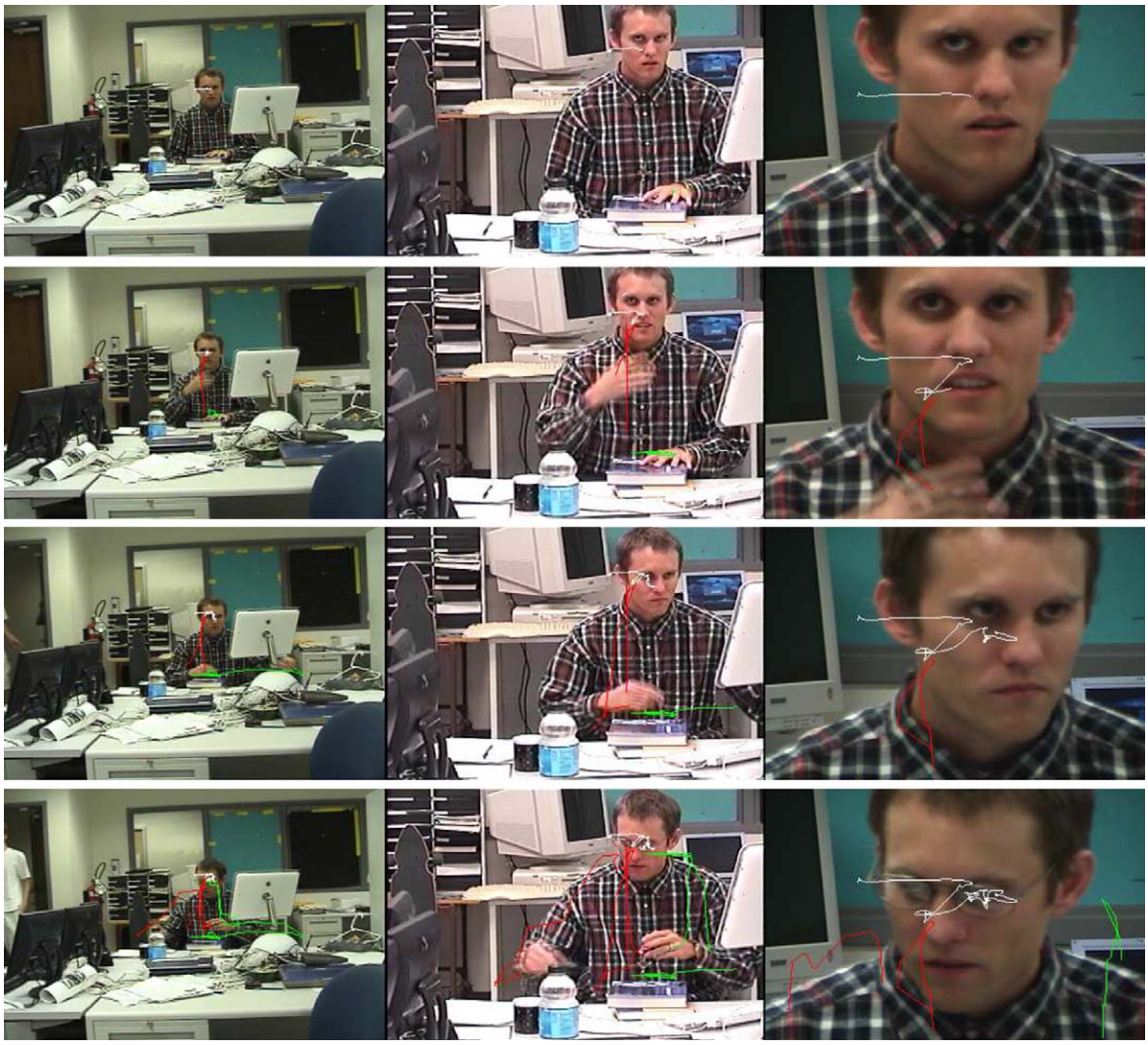

Fig. 12. Output of consistent labeling. Each row is a particular time unit in the sequence. For each row zoom 1, zoom 2, and zoom 3 are shown, respectively. The previous object trajectories are superimposed on the current frame in the sequence. The matched trajectories across views are shown in similar colors. All objects were labeled across views correctly. Row 3 shows a frame after the head has moved. Notice that this generates a white line, similarly the white line appears in the other zooms indicating it is the same trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

scene detail to improve activity analysis. In Section 7, detailed results are presented showing the effectiveness of our integrated multi-zoom analysis.

6.1. Object segmentation

The first scenario we consider is determining whether there is an object in the hand as it comes to the face. Using only a view such as zoom 1 will present several challenges because there is not enough detail to determine whether the hand had an object in it, and whether it went to the mouth or the ear. In a higher zoomed view such as zoom 3, there is no way to know where the object originally came from in the scene or where and when to look for the object, but zoom 1 and zoom 2 both can provide this information to zoom 3. Thus, multiple zooms need to be combined in a manner such that each zoom level answers the questions that it is best able to answer. We show how to combine multiple levels of detail to detect and analyze these objects that are difficult to detect with a single level of zoom. In previous sections, C_a and C_b were denoted as arbitrary cameras. Here, the strengths of each zoom are used, so C_l and C_h denote the lower and higher zoomed cameras, respectively. This notation will be used throughout this section.

To identify if there is an object in either hand, the hands in C_l are analyzed for motion by computing $I_{t,l,f}$. I_t indicates temporal derivative for camera, C_l and frame f. Significant motion of non-skin colored pixels indicates that a potential object is found. $\mathbf{p}$ is denoted as the centroid of this potential object in the lower zoom, C_l. If a significant amount of motion generated by non-skin colored pixels is

found in C_h near the intersection of the epipolar line, $\mathbf{p} \cdot \mathbf{F}_{l,h}$, with the image plane, then an object is assumed to be in the hand. The flowchart (including the auto-correct step in Section 6.1.1) is shown in Fig. 13. Concretely, for each hand region $B_i = x_a^i$, in C_l found using the method in Section 4 compute:

$$\sum_{p \in B_i} \overline{H}_l(I_{l,f}(p))\hat{I}_{t,l,f}(p) > \alpha_0, \tag{12}$$

where p is an image point, $\overline{H}_l$ is the negated color model of the head and hands for C_l, presented in Section 4. $I_{l,f}$ is the image from camera C_l for this frame f. $\hat{I}_{t,l,f}$ is a binary valued motion segmentation produced from $I_{t,l,f}$ in the following way:

$$\forall p, \quad \hat{I}_{t,l,f}(p) = I_{t,l,f}(p) > \alpha_1. \tag{13}$$

If Eq. (12) holds then C_h can be notified as to where an object may be present by finding the corresponding epipolar line $c \cdot \mathbf{F}_{l,h}$, where c is the centroid in C_l

$$c = \frac{\sum_{p \in B_i} p \cdot \overline{H}_l(I_{l,f}(p))\hat{I}_{t,l,f}(p)}{\sum_{p \in B_i} \hat{I}_{t,l,f}(p)}. \tag{14}$$

While it is true that the epipolar geometry maps points to lines (for orthogonal, perspective cameras), the search for the object can be reduced to two regions. This reduction in the search space is possible since we know the hand and object are not yet in C_h. Since we have an object position in C_l, we can find its epipolar line l in C_h. Then intersect this line with the image plane, and only look at these intersection points, P_i, for entering objects. There will be at most two points because the images are planar. With the predicted intersection points P_i, regions around these points, R_i are searched using a modification of Eq. (12):

$$\sum_{p \in R_i} \overline{H}_h(I_{h,f}(p))\hat{I}_{t,h,f}(p) + H'_l(I_{h,f}(p))\hat{I}_{t,h,f}(p) > \alpha_2. \tag{15}$$

The main differences here are that (1) we have a predicted region R_i which gives the probable location of where to look in C_h and (2) we can use H'_l which is the object color model from C_l, transferred to C_h, to find additional moving non-skin pixels. H'_l is built (using [27]) from the lower zoomed camera, C_l, using the pixels $p \in B_i$ such that $\overline{H}_l(I_{l,f}(p))\hat{I}_{t,l,f}(p) > 0$. Since the cameras are not color calibrated the quality of the transferred color model H'_l could be increased by performing a color space transform such as [29]. H_l and H_h are the color models for the head and hands in C_l and C_h, respectively (presented in Section 4). The object color model of C_l could be updated based on the object color model in C_h. If inequality 15 does not hold then it means no objects appeared in C_h at location P_i, with the predicted color H'_l and C_h assumes a false positive was observed. This allows for a bad segmentation in C_l to be auto corrected in C_h. The bad segmentation in C_l will not yet be eliminated but the propagation of the error is halted. Section 6.1.1 details how C_l can then be notified of its error to correct the bad segmentation.

If the object is confirmed in C_h, then segmentation in C_h can proceed. By passing location and color information between cameras, we can achieve better object segmentation. This allows early identification of objects in C_h. By passing this updated color and spatial information back to C_l we can update its color and spatial parameters for the object in question, which will allow for better segmentation in the lower zooms. Results from our multi camera segmentation have demonstrated that we are able to correctly determine when an object is in the hand and further, when C_l gives an incorrect result the method is able to determine this in C_h and notify C_l. Results are shown in Fig. 14. In Fig. 14, C_l triggers that an object is present in the hand because the segmentation is not perfectly correct. This can be seen by observing the hole in the segmented skin image. The C_h segmentation is correct and it does not observe any significant motion of non-skin colored

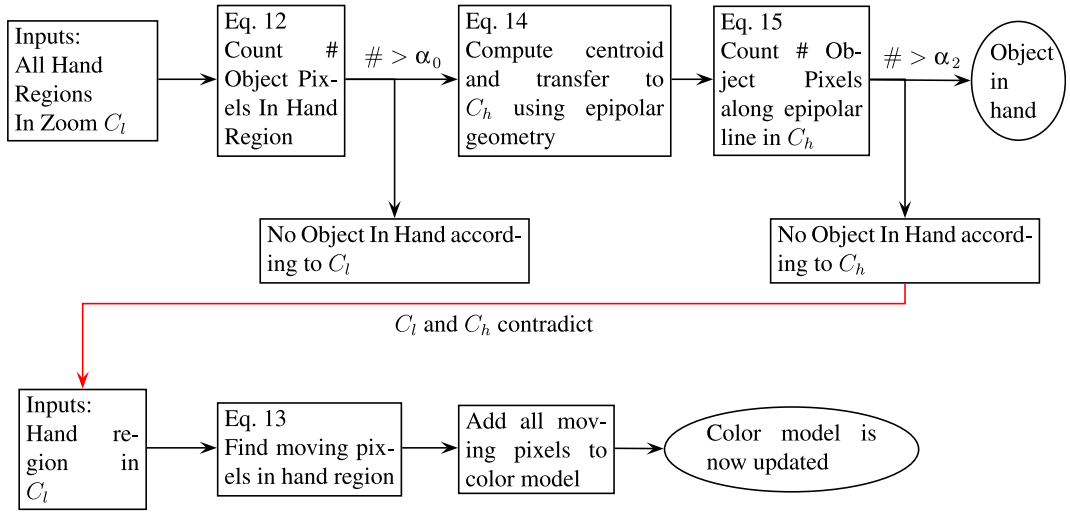

Fig. 13. Flowchart for Section 6.1 (top row) and 6.1.1 (bottom row).

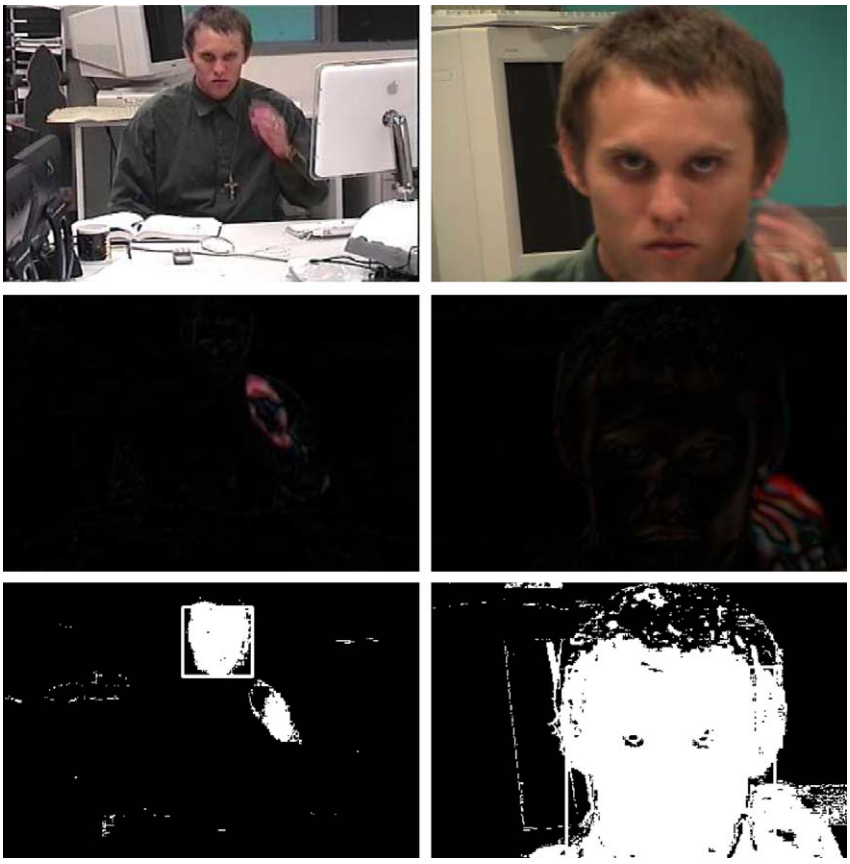

Fig. 14. Zoom 2 images are in column one and zoom 3 images are in column two. Row one is the input images. Row two is the $I_{t,l,f}$ images, and the third row is the color segmentation images. In zoom 2, a poor color model does not correctly segment all of the hand (column one, row three). Thus, zoom 2 incorrectly concludes that an object is present in the hand. However, in zoom 3, the color segmentation is correct, it can override zoom 2's decision. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

objects, thus it overrides C_l's decision and notifies C_l of the incorrect segmentation.

6.1.1. Automatically correcting incomplete segmentations

As stated in Section 4, the color model is built using the RGB values of the head pixels. This gives a good color model for the hands, but it is not always complete. In Fig. 14, the reason that zoom 2 incorrectly determines that an object is in the hand is because of the incomplete color model that is built using the head's color information. It was already shown in Section 6 how the color models of the object can be transferred across cameras to allow for improved object segmentation. Here, we show how the color model of the head and hands in C_l can be auto corrected. This is a consequence of the multicamera detection scheme. When C_l incorrectly determines that an object is in the hand, C_l can go back to this particular frame (which can be done once C_h notifies C_l of the error) and put these hand pixels that were detected as an object in the color model for the head and hands.

Concretely, the following steps must be taken: (1) C_h notifies C_l of the incorrect segmentation. (2) C_l then goes back to the frames that it (incorrectly) determined an object was in the hand. (3) Any moving pixels in this region are then treated as hand pixels and they are added to the color model for the head and hands. This process greatly increases the accuracy of the color model and Fig. 15 shows the segmentation using the corrected color model.

6.2. Determining number of hands in head region

For action analysis another important subtask is determining how many hands are at the face. Certain actions require a certain number of hands to be present. Putting on eyeglasses requires two hands whereas drinking a beverage involves one hand coming to the face. Utilizing multiple zoom levels aids in the task of determining the number of hands in the head region. Zooms one and two cooperate in this task. The flowchart for this scenario is shown in Fig. 16. The first step is to compute the distance between the head and hand for each zoom as shown in Fig. 17. This results in d_1, d_2, d_3, d_4. Then the likelihood of the hand being near the face is computed as $\frac{1}{\sqrt{2\pi}\sigma} e^{-(\frac{d_i^2}{2\sigma^2})}$. Because the hand tracks are noisy we add the distance between the hand and head for zooms one and two: $\frac{1}{\sqrt{2\pi}\sigma} e^{-(\frac{(d_i+d_j)^2}{2\sigma^2})}$. The plot of this measure over time is shown in Fig. 18. The red plot is $\frac{1}{\sqrt{2\pi}\sigma} e^{-(\frac{(d_1+d_3)^2}{2\sigma^2})}$(for hand 1 and hand 3). The green

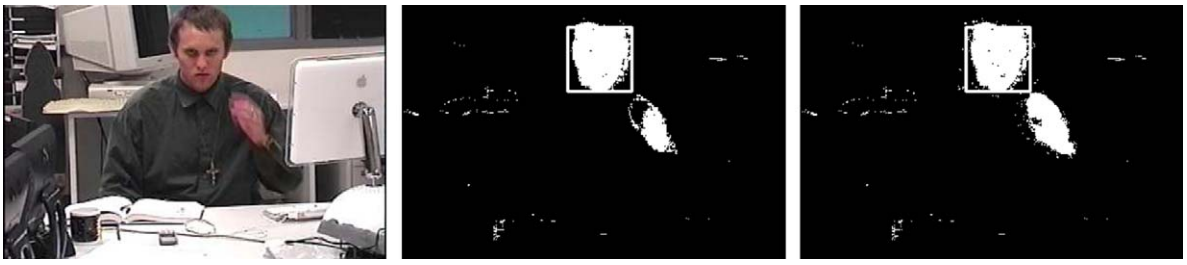

Fig. 15. This figure shows how an incorrect result in one zoom can be used to correct future bad segmentations. Column 1 shows the input image. Column 2 shows the segmentation using the incomplete color model. This figure is the same as Fig. 14 (column one, row three). Column 3 shows the segmentation of the same image after the notification and update process. This update of the color model allows for much better segmentation of the hand. This is an interesting consequence of the multi-zoom cooperation among cameras. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

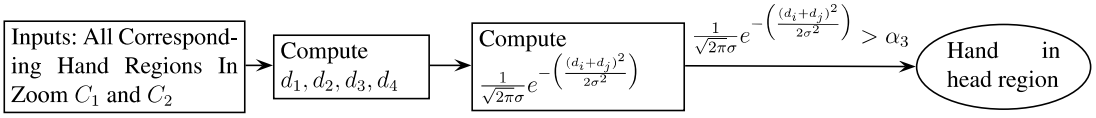

Fig. 16. Flowchart for Section 6.2.

Fig. 17. Computing distance between the hand and head.

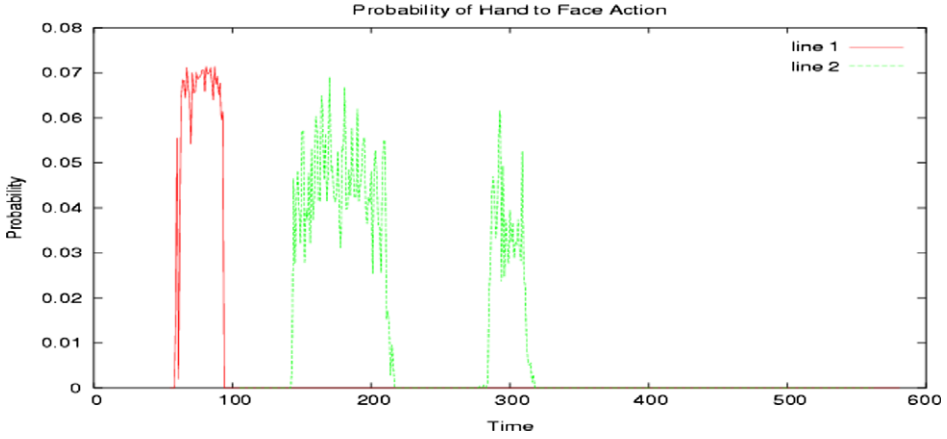

Fig. 18. Probability of hand in head region.

plot is $\frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(d_2+d_4)^2}{2\sigma^2}\right)}$ (for hand 2 and hand 4). When $\frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(d_i+d_j)^2}{2\sigma^2}\right)} > \alpha_3$ the hand is near the face. Results of the method are shown in Fig. 19.

6.3. Localizing hand on face

The final scenario we present is determining where on the face the hand is. Many actions can be distinguished based on where the hand is on the face, such as using the

phone and drinking . We split the head into six regions shown in Fig. 21A. The computation is similar to that in Section 6.1 with the difference being all moving pixels are used. Thus Eq. (12) becomes

$$\sum_{p\in B_i} \hat{I}_{t,l,f}(p) > \alpha_4 \tag{16}$$

Eq. (14) becomes

$$c = \frac{\sum_{p\in B_i} p \cdot \hat{I}_{t,l,f}(p)}{\sum_{p\in B_i} \hat{I}_{t,l,f}(p)}. \tag{17}$$

Eq. (15) is similarly modified. The region with the maximum number of hand pixels is taken to be the location of the hand. Results of hand localization are shown in Figs. 21B and C. The flowchart for this scenario is presented in Fig. 20.

6.4. Other directions for integrating multiple levels of zoom

We have given details on three techniques to combine multiple levels of zoom with applications for action analysis. There are many other possible ways to use multiple levels of zoom. For instance one technique would be to measure the temporal duration the hand was in the head region. Another technique would be to determine what object a person was looking at using the detailed head position in zoom 3 combined with the scene details (possible objects) in zooms 1 and 2. We are exploring these and other methods to combine zoom levels.

7. Quantitative results

The proposed overall method has been formulated in the context of activity analysis for cameras with multiple levels of zoom. The guidelines for experimental design and evaluation are discussed next. The cameras were placed so that all were facing the same scene with different levels of zoom. The successively higher zoom levels each viewed a subset of the scene taken at lower zoom levels. There were no strict camera placement protocols. Datasets taken at different times did not have to have identical zooms/placement as the initial experiments. The zoom and camera placement were different for most of the tests. Because our method uses the fundamental matrix, we did not need strict camera placement protocols. We wanted this flexibility to make the system less restrictive and more useful to others. To compute the fundamental matrix, 14 point correspondences were used. This was sufficient cali-

Fig. 19. Automatic results of determining the number of hands in head region.

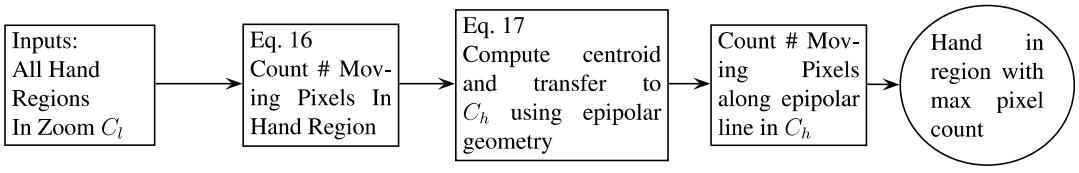

Fig. 20. Flowchart for Section 6.3.

Fig. 21. Automatic results of hand localization.

bration for our purposes. The experiments were all in a normal office environment and no special illumination calibration across cameras was performed. We first present results of the correspondence algorithm (Section 5) and then show results for activity analysis (Section 6).

In evaluating our consistent labeling algorithm the main task to evaluate was how many objects were correctly labeled across the video sequences. A number of constraints were used (see Sections 5.1–5.3) to make the matching more robust. A valid question arises: is there any benefit of the constraints. We show in the following tables how the correspondence matching performed with various combinations of constraints. Results using only the epipolar distance minimization (Eq. 6) are presented in Table 1. Table 2 shows the effect of using Eq. (7) for the error instead of Eq. (5). Results using only the epipolar distance minimization and spatial constraints are presented in Table 3. Results using only the epipolar distance minimization and trajectory constraints are presented in Table 4. Next results are presented, in Table 5, using only the epipolar distance minimization and appearance constraints. When we combine all four constraints together we achieve 100% accuracy as presented in Table 6. Table 7 lists a summary of the average score for each algorithmic setup. This average was obtained by summing the individual percentage scores for each sequence and dividing by the total number of sequences. The appearance constraints did well overall. However in sequence 2, the appearance constraints failed, and this was a sequence that the spatial constraints performed well on. Though in sequence 2, it was only in combining all the constraints that the algorithm achieved 100%. Thus, we can see how the various constraints work together to achieve better results. Data Sets 1–7 are multiple level of zoom sequences. Data Set 8, shown in Fig. 22, is a sequence with partially overlapping FOVs as found in many surveillance papers [30]. Data Set 9 is a three camera sequence with partially overlapping FOVs. This labeling algorithm was tested on a number of different camera configurations to show the robustness of the proposed approach. The proposed labeling algorithm has been tested on eight such three camera sequences, and one two camera sequence for a total of over 18,500 video frames with over 160 objects corresponded correctly with 100% accuracy.

Results from the multiple levels of detail activity analysis module are now presented. Multiple camera configurations were tested with various camera placement. The module in Section 6.1 was tested on 15 video clips. The method was required to automatically determine whether there was an object in the hand for each sequence. In all cases the hand came to the face either with an object in the hand (eight times) or without an object in the hand (seven times). In all the trials there were only two bad decisions (one in each category). Some of the clips were challenging. For instance, the method was successful in determining that there was an object in the hands when eye glasses were being brought to the head. With one low zoom it would be hard to see the eyeglasses. Further, unconstrained search in zoom 3 would have too many false

Table 1
Only epipolar minimization using Eq. (5)

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	85	17/20
Sequence 2	7	7	2	72	13/18
Sequence 3	9	9	2	91	20/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	85	17/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	67	8/12
Sequence 9	6	9	3	100	18/18

Table 2
Only epipolar minimization using Eq. (7)

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	90	18/20
Sequence 2	7	7	2	83	15/18
Sequence 3	9	9	2	91	20/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	85	17/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	75	9/12
Sequence 9	6	9	3	100	18/18

Table 3
Epipolar and spatial constraints

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	80	16/20
Sequence 2	7	7	2	89	16/18
Sequence 3	9	9	2	91	20/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	85	17/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	67	8/12
Sequence 9	6	9	3	100	18/18

Table 4
Epipolar and trajectory constraints

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	90	18/20
Sequence 2	7	7	2	83	15/18
Sequence 3	9	9	2	91	20/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	85	17/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	75	9/12
Sequence 9	6	9	3	100	18/18

Table 5
Epipolar and appearance constraints

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	100	20/20
Sequence 2	7	7	2	83	15/18
Sequence 3	9	9	2	100	22/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	100	20/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	100	12/12
Sequence 9	6	9	3	100	18/18

Table 6
All constraints

Sequence #	Objects in Camera 1	Objects in Camera 2	Objects in Camera 3	% Matched	# Matched
Sequence 1	7	7	3	100	20/20
Sequence 2	7	7	2	100	18/18
Sequence 3	9	9	2	100	22/22
Sequence 4	6	6	2	100	16/16
Sequence 5	7	7	3	100	20/20
Sequence 6	11	10	2	100	24/24
Sequence 7	4	4	2	100	12/12
Sequence 8	6	6	0	100	12/12
Sequence 9	6	9	3	100	18/18

positives. The system was able to successfully recognize when only the hands were coming to the head. Fig. 23 shows a case in which the user enters the scene, though his hand is not put to his head. In this case, there was significant skin (face) and non-skin (hair, eyes, etc...) in zoom 3, which would have given a false alarm if the system only looked at zoom 3 or used some other naive method.

The module in Section 6.2 was tested on 10 video clips. The method was required to determine how many hands were at the face (if any). Five clips had one hand coming

Table 7
Summary for all algorithmic setups

Algorithm setup	Average sequence score
Only Eq. (5)	88.8
Only Eq. (7)	91.5
Epipolar and spatial constraints	90.2
Epipolar and trajectory constraints	91.5
Epipolar and appearance constraints	98.1
All constraints	100

to the face and five clips had two hands coming to the face. The method correctly determined the number of hands coming to the face in 9 of the clips. The system never said there were hands at the face when there were none. The module in Section 6.3 was tested on 10 video clips. In all clips the hand came to one of the six regions shown in Fig. 21A. The method was required to determine which region the hand was in. The method had only two incorrect decision. That is we made the correct determination as to which region the hand was in eight clips.

Fig. 22. Other camera configurations that we tested the labeling algorithm on (Section 5). There were two cameras in this setup. One input image from each camera is shown.

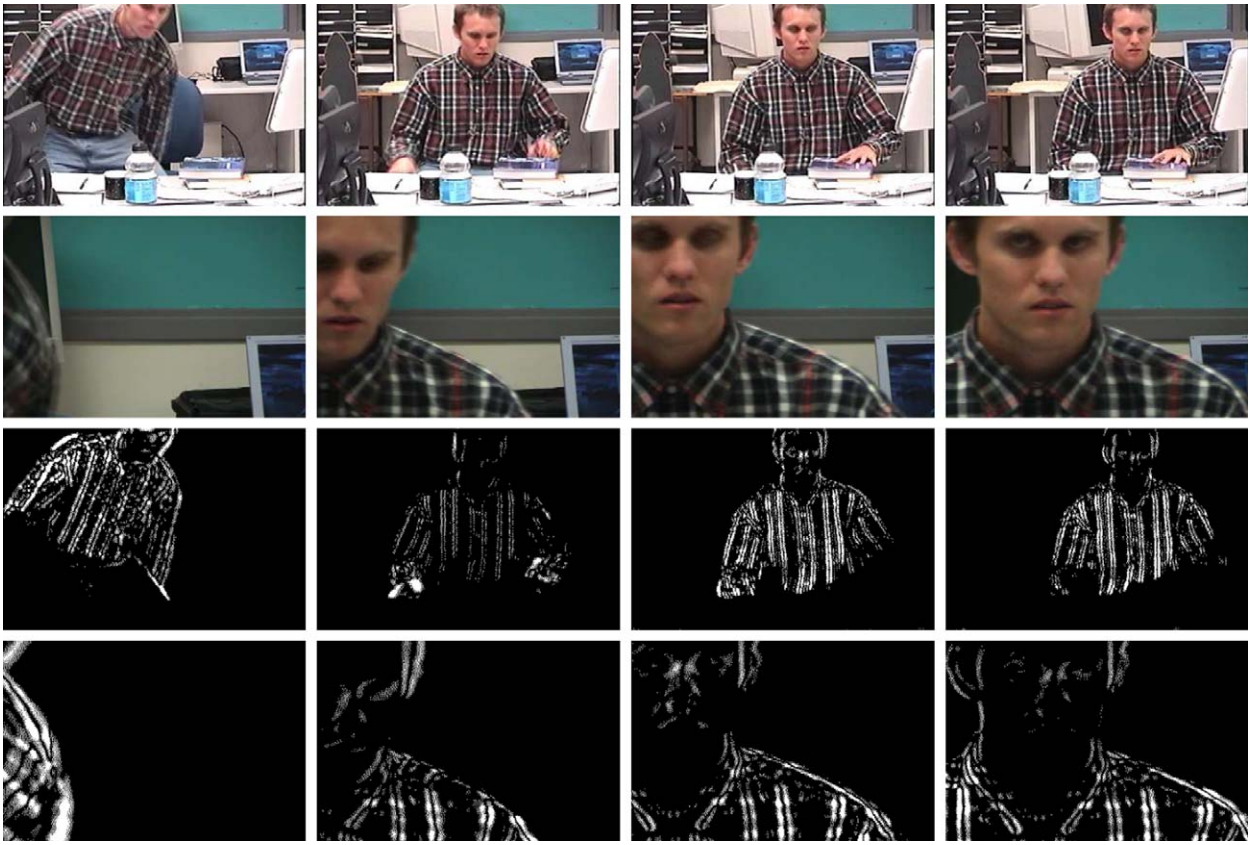

Fig. 23. Multi-zoom segmentation. The first row shows the input images in zoom 2. The second row shows the input images in zoom 3. Rows three and four show the $I_{t,l,f}$ images in zoom 2 and zoom 3, respectively. Though there is significant non-skin motion, the system is able to infer from the context of zoom 2 that the hand is not near the head.

In all these scenarios the multicamera formulation is able to discern the *context* of the scene. The term refers to the low level tracking information available and coarse object information in the lower zooms combined with the object detail present in the high zooms. Also present in the high zooms is more detailed (but spatially limited) tracking information. Because there is no hand near the head when the user is entering, which is known from the context of zoom 2, the method is able to disregard the significant non-skin motion which would have otherwise signaled a false positive that the hand was near the head.

8. Conclusion

We have developed a robust multi-zoom framework to enable activity analysis. The presented framework is able to combine information from cameras in multiple ways to increase overall system performance. Heads and hands are automatically found and tracked using multiple levels of detail. We have presented a method which is able to incorporate epipolar, spatial, trajectory, and appearance together into a unified framework to achieve consistent object labeling across multiple cameras. The activity analysis module is able to integrate multiple levels of detail to determine whether there is an object in the hand, a task which would be rather difficult with a single view. In the future we hope to explore additional ways of combining multiple levels of scene detail to recognize complex actions.

References

[1] A. Pentland, Looking at people: sensing for ubiquitous and wearable computing, in: TPAMI, 2000.
[2] W.E.L. Grimson, L. Lee, C. Stauffer, K. Tieu, The activity perception project (2003).
[3] Y. Wexler, A.W. Fitzgibbon, A. Zisserman, Learning epipolar geometry from image sequences, in: CVPR, 2003.
[4] P. Smith, M. Shah, N. da Vitoria Lobo, Integrating and employing multiple levels of zoom for activity recognition, in: CVPR, 2004.
[5] J.K. Aggarwal, Q. Cai, Human motion analysis: a review, CVIU 73 (3) (1999) 428–440.
[6] D.M. Gavrila, The visual analysis of human movement: a survey, CVIU 73 (1) (1999) 82–98.
[7] N. Nguyen, H. Bui, S. Venkatesh, G. West, Recognising and monitoring highlevel behaviours in complex spatial environments, in: CVPR, 2003.
[8] R. Collins, A. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for cooperative multisensor surveillance, Proceedings of the IEEE 89 (10) (2001) 1456–1477.
[9] S. Stillman, R. Tanawongsuwan, I. Essa, A system for tracking and recognizing multiple people with multiple cameras, GIT-GVU 98-25, Georgia Institute of Technology (August 1998).
[10] H. Hongo, M. Ohya, M. Yasumoto, Y. Niwa, K. Yamamoto, Focus of attention for face and hand gesture recognition using multiple cameras, in: Automatic Face and Gesture Recognition, 2000.
[11] I. Mikic, K. Huang, M. Trivedi, Activity monitoring and summarization for an intelligent meeting room, in: IEEE Workshop on Human Motion, 2000.
[12] S.M. Seitz, J. Kim, The space of all stereo images, in: IJCV, 2001.
[13] M. Pollefeys, R. Koch, L.V. Gool, A simple and efficient rectification method for general motion, in: International Conference on Computer Vision, 1999, pp. 496–501.
[14] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: ICCV, 2001.
[15] Y. Boykov, O. Veksler, R. Zabih, Markov random fields with efficient approximations, in: CVPR, 1998.
[16] G.P. Stein, Tracking from multiple view points: self-calibration of space and time, in: DARPA IU Workshop, 1998, pp. 1037–1042.
[17] M. Antone, S. Teller, Scalable, absolute position recovery for omnidirectional image networks, in: CVPR, 2001.
[18] Y. Caspi, D. Simakov, M. Irani, Feature-based sequence-to-sequence matching, in: ECCV Vision and Modelling of Dynamic Scenes Workshop, 2002.
[19] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, in: IJCV, 2002.
[20] T.-H. Chang, S. Gong, Bayesian modality fusion for tracking multiple people with a multi-camera system, in: Proc. European Workshop on Advanced Video-based Surveillance Systems, 2001.
[21] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, S. Shafer, Multi-camera multi-person tracking for easy living, in: 3rd IEEE International Workshop on Visual Surveillance, 2000.
[22] T. Darrell, D. Demirdjian, N. Checka, P. Felzenszwalb, Plan-view trajectory estimation with dense stereo background models, in: IEEE International Conference on Computer Vision, 2001.
[23] A. Mittal, L.S. Davis, M2tracker: a multi-view approach to segmenting and tracking people in a cluttered scene, in: International Journal of Computer Vision, 2003, p. 189203.
[24] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, MA, 2000.
[25] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review, in: International Journal of Computer Vision, 1998.
[26] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: CVPR, 2001.
[27] R. Kjedlsen, J. Kender, Finding skin in color images, in: Face and Gesture Recognition, 1996, pp. 312–317.
[28] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-regid objects using mean shift, in: CVPR, 2000.
[29] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley, Color transfer between images, in: Computer Graphics and Applications, 2001.
[30] Q. Cai, J.K. Aggarwal, Tracking human motion in structured environments using a distributed-camera system, in: PAMI, vol. 21, 1999.