Human Tracking in Multiple Cameras

Sohaib Khan, Omar Javed, Zeeshan Rasheed, Mubarak Shah
Computer Vision Lab
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
{ khan, ojaved, zrasheed, shah}@cs.ucf.edu

ABSTRACT

Multiple cameras are needed to cover large environments for monitoring activity. To track people successfully in multiple perspective imagery, one needs to establish correspondence between objects captured in multiple cameras. We present a system for tracking people in multiple uncalibrated cameras. The system is able to discover spatial relationships between the camera fields of view and use this information to correspond between different perspective views of the same person. We employ the novel approach of finding the limits of field of view (FOV) of a camera as visible in the other cameras. Using this information, when a person is seen in one camera, we are able to predict all the other cameras in which this person will be visible. Moreover, we apply the FOV constraint to disambiguate between possible candidates of correspondence. We present results on sequences of up to three cameras with multiple people. The proposed approach is very fast compared to camera calibration based approaches.

Keywords: Tracking in multiple cameras, multi-perspective video, surveillance, camera handoff, sensor fusion

1. INTRODUCTION

Tracking humans is of interest for a variety of applications such as surveillance, activity monitoring and gait analysis. With the limited field of view (FOV) of video cameras, it is necessary to use multiple, distributed cameras to completely monitor a site. Typically, surveillance applications have multiple video feeds presented to a human observer for analysis. However, the ability of humans to concentrate on multiple videos simultaneously is limited. Therefore, there has been an interest in developing computer vision systems that can analyze information from multiple cameras simultaneously and possibly present it in a compact symbolic fashion to the user.

To cover an area of interest, it is reasonable to use cameras with overlapping FOVs. Overlapping FOVs are typically used in computer vision for the purpose of extracting 3D information. The use of overlapping FOVs, however, creates an ambiguity in monitoring people. A single person present in the region of overlap will be seen in multiple camera views. There is need to identify the multiple projections of this person as the same 3D object, and to label them consistently across cameras for security or monitoring applications.

In related work, [1] presents an approach of dealing with the handoff problem based on 3D-environment model and calibrated cameras. The 3D coordinates of the person are established using the calibration information to find the location of the person in the environment model. At the time of handoff, only the 3D *voxel-occupancy* information is compared to achieve handoff, because multiple views of the same person will map to the same voxel in 3D. In [2], only relative calibration between cameras is used, and the correspondence is established using a set of feature points in a Bayesian probability framework. The intensity features used are taken from the centerline of the upper body in each projection to reduce the difference between perspectives. Geometric features such as the height of the person are also used. The system is able to predict when a person is about the exit the current view and picks the best next view for tracking. A different approach is described in [3] that does not require calibrated cameras. The camera calibration information is recovered by observing motion trajectories in the scene. The motion trajectories in different views are randomly matched against one another and plane homographies computed for each match. The correct homography is the one that is statistically most frequent, because even though there are more incorrect homographies than the correct one, they lie in scattered orientations. Once the correct homography is established, finer alignment is achieved through global frame alignment. Finally [4, 5] describe approaches which try to establish time correspondences between non-overlapping FOVs. The idea there is not to completely cover the area of interest, but to have motion constrained along a few paths, and to correspond objects based on time from one camera to another. Typical applications are cameras installed at intervals along a corridor [4] or on a freeway [5].

The luxury of calibrated cameras or environment models is not available in most situations. We therefore tend to prefer approaches that can discover a sufficient amount of information about the environment to solve the handoff problem. We contend that camera calibration is unnecessary and an overkill for this problem, since the only place where handoff is required is when a person enters or leaves the FOV of any camera. By building a model of the relationship between FOV lines of various cameras can provide us sufficient information to solve the handoff problem.

In the next section we formalize the handoff problem and describe how the relationship between the FOV of different cameras can be used to solve the handoff problem. In Section 3, we describe how this relationship can be automatically discovered by observing motion of people in the environment. Finally we present results of our experiments in Section 4.

2. EDGE OF FIELD OF VIEW LINES

The handoff problem occurs when a person enters the FOV of a camera. At that instant we want to determine if this person is visible in the FOV of any other camera, and if so, assign the same label to the new view. If the person is not visible in any other camera, then we want to assign a new label to this person. Consider the following scenario; a room with two cameras has two persons walking in it. At time instant 1, both persons are visible in Camera 1. At time instant 2, Person 1 walks into the FOV of Camera 2. Since we have already assigned labels to both persons (Person 1 and 2), we need to figure out at this instant which of the persons is entering the FOV of Camera 2. There are three possibilities to consider here. The new person seen in Camera 2 could be Person 1, Person 2 or a new person entering the environment. Since we do not know any 3D information about the environment or the camera calibration matrices, we cannot determine what label to assign to the new view seen in Camera 2.

Note here that we could have matched color features of the two persons visible in Camera 1 to the new view in Camera 2 to find the most likely match. However, when the disparity is large, both in location and orientation, feature matches are not reliable. After all, a person may be wearing a shirt that is different colors at front and back. The reliability of feature matching decreases with increase in disparity, and it is not uncommon to have surveillance cameras looking at an area from opposing directions. Moreover, different cameras can have different intrinsic parameters as well as photometric properties (like contrast, color-balance etc.). Lighting variations also contribute to the same object being seen with different colors in different cameras.

Camera 1 Camera 2

Figure 1: Example of correct handoff: There are two persons visible in Camera 1. When one of them enters the FOV of Camera 2, the left edge of FOV of Camera 2 as seen in Camera 1 ($L^{21}{}_{l}$) helps us disambiguate between the labels.

For shallow mounted cameras each FOV's footprint can be described by two lines on the floor-plane, the left and the right limit of FOV. Let L^{i}_{l} and L^{i}_{r} be the left and right limits of FOV of the i^{th} camera (C^{i}) on the ground plane (Figure 1). Let the projection of L^{i}_{x} ($x \in \{l, r\}$) in Camera j be denoted by L^{ij}_{x}. Note that L^{ii}_{x} denotes the left and the right sides of the image in C^{i}. As far as the camera pair i, j is concerned, the only locations of interest in the two images for handoff are L^{ij}_{x} and L^{ji}_{x}. These are up to four lines, possibly two in each camera. Let us currently assume that a person already visible in one of the cameras is entering the FOV of another camera. In this case, all that needs to be done is to look at the associated line in the *other* camera and see which person is crossing that line. Figure 1 describes this situation in more detail. A person is entering the FOV of C^{2}. There are two persons visible in C^{1} at this instant. Both these persons are being tracked and we have a bounding box around them. By looking at the bottom part of the bounding box, we can determine quite easily which person has entered the FOV of C^{2}. The line that helped us determine this is $L^{21}{}_{l}$ i.e. the left FOV of C^{2} as seen in C^{1}. The new person in C^{2} is therefore assigned the same label as the one it was assigned in C^{1}. Note that we are considering only the left and right edges of FOV in this formulation, which is sufficient for cameras mounted at a low angle of depression. However, there is nothing in this analysis which prevents it from being extended to considering all four limits of the camera footprint, which will be necessary for images shot at a high angle of depression.

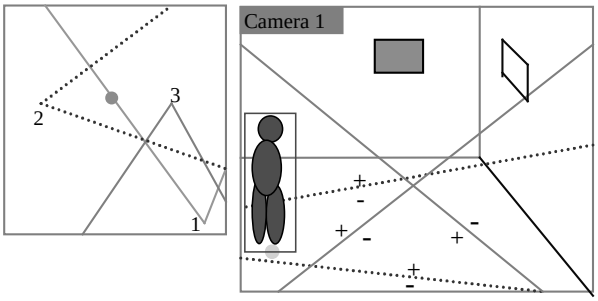

Figure 2: (Left) Three cameras setup in a room, with their FOVs shown by different lines. A person is entering the FOV of Camera 1. (Right) By looking at the FOV lines of Cameras 2 and 3 in Camera 1, we can determine that this person is visible in Camera 2 but not in Camera 3.

Detection of New Persons

In the example given above, it is assumed that when a person enters the FOV of a camera, he must be visible in the FOV of another camera. This is not always the case. A person might be entering from the door (in which case he might just "appear" in the middle of the image) or he might be entering the FOV from a point that is not visible in any other camera. If the camera setup is such that the environment is completely covered, then the latter case will never happen. However, to keep the formulation general, the second case has to be considered too.

In the previous case, we looked at the FOV lines of the current camera as seen in other cameras. To find whether a person is visible in other cameras or not, we look at the FOV lines of other cameras as seen in the current camera. Consider the scenario when a person is entering the FOV of C^i. Whether this person is visible in any other camera (C^j, $j \neq i$) or not can be determined by looking at all the FOV lines that are of the form L^{ji}_x , i.e. edge of FOV lines of other cameras as visible in this camera (C^i). These lines partition the image C^i into (possibly over lapping) regions, marking the areas of image C^i that correspond to FOV of other cameras. Figure 2 illustrates this situation symbolically. Thus all the cameras in which current person is visible can be determined by acquiring the region of the person's feet.

Thus with each line L^{ji}_x, an additional variable δ^i_x is stored. The value of δ^i_x can either be +1 or –1, depending upon which side of the line falls inside the FOV of C^j. Then, given an arbitrary point (x', y') in C^i, the point's visibility in C^j can be determined by just determining if this point is on the correct side of both L^{ji}_l and L^{ji}_r. If L^{ji}_l is represented by A x' + B y' + C. The point (x', y') is visible in C^j if and only if

$$\text{sgn}(L^{ji}_l(x', y')) = \delta^{ji}_l \text{ and } \text{sgn}(L^{ji}_r(x', y')) = \delta^{ji}_r \quad (1)$$

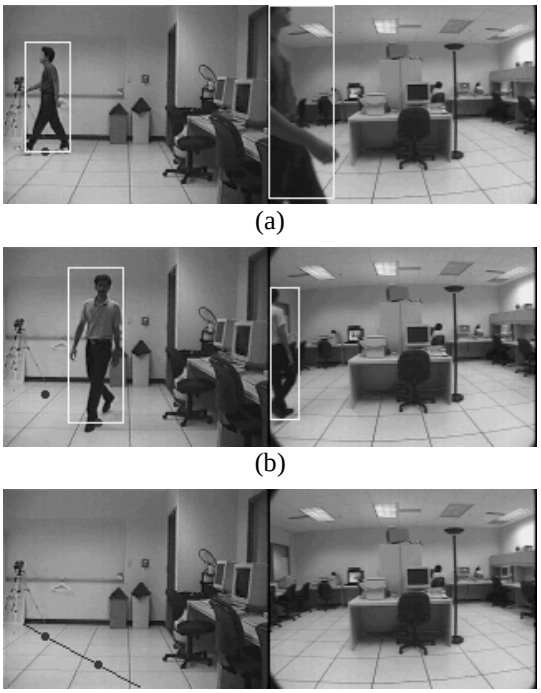

Figure 3: (a) Person entering the FOV of C^2 from left yields a point on line L^{21}_l in image taken from C^1. (b) Another such correspondence yields another point, which are joined to find the complete line L^{21}_l shown in (c).

In the case when only one of the left or right lines of C^j is visible in C^i, the condition in Eq. 1 is simplified to only one of the anded terms.

Establishing Correspondence Between Views

When a person enters the FOV of a new camera, it can be determined whether this person is visible in the FOV of some other camera or not. Whenever a person is in the image all the other cameras in which this person will also be visible can be found out by using Eq 1. If there is no such camera, then a new label is assigned to this person. Otherwise the previous track of this person is found so that a link can be established between the two views. This is done by finding the person closest to the appropriate edge of FOV line. Say that the person entered from the left side of C^1. Then, the persons visible in all cameras other than C^1 will be searched and the person that is closest to the left edge of FOV line of C^1 in that camera will be found. These two views will then be linked together by entering them in an equivalence table. In general, if a person enters C^i from side x, then the label assigned to the new view will be:

```
Repeat for each frame
    For each camera C^i
        If person appears from side x
            Find S = {j | current person is visible in C^j }
                (from Equation 1)

            If S = φ
                then assign current person a unique label
            else
                For each camera C^j s.t. j∈S
                    For each person k in C^j
                        Compute d(j,k)= D(P^j_k, L^ij_x)
                    end
                end
            end
        end
    end
    Let s = row of minimum element in d
    Let t = column of minimum element in d
    Then P^s_t ( in C^s ) is the same as the new person in C^i
end
```

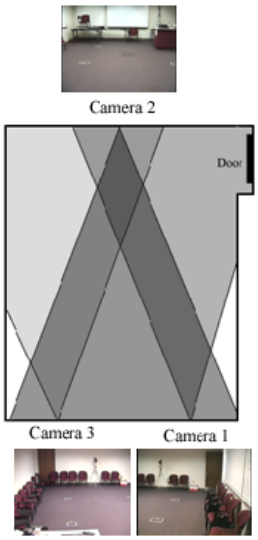

Figure 4: Experimental setup: 3 cameras are set up in a room to cover most of the area. There is only one door, which is visible in camera 1.

$$\text{label} = \arg\min_{k}(D(P^k, L_x^{ij})) \quad \forall j \neq i$$
$$\text{where } k = \text{set of persons visible in } C^j \tag{2}$$

where P^k is the label assigned to a person and $D(P, L)$ returns the absolute distance between the center of the bottom line of the rectangular bounding box of person P and the line L.

The complete algorithm for ambiguity resolution of new views is given in the inset.

3. AUTOMATIC DETERMINATION OF FOV LINES

When tracking is initiated, there is no information provided about the FOV lines of the cameras. The system can, however, find this information by observing motion in the environment. Whenever there is a person entering or exiting one camera, he actually lies on the projection of the FOV line of this camera in all other ones in which he is visible. Suppose that there is only one person in the room. Then, when this person enters the FOV of a new camera, we find one constraint on the associated line. Two such constraints will define the line, and all constraints after that can be used in a least squares formulation. This concept is visually described in Figure 3. However it is not always possible to have only one person walking in the scene. Therefore, for cluttered situations where it is hard to find the correspondences to be used for initial setup, we propose another method. When multiple people are in the scene and if someone crosses the edge of FOV, all persons in other cameras are picked as being candidates for the projection of FOV line. Since the false candidates are randomly spread on both sides of the line where as the correct candidates are clustered on a single line, correct correspondences will yield a line in a single orientation, whereas the wrong correspondences will yield lines in scattered orientations. We use Hough transform to find the best line in this case.

Thus there are two options for initial setup of FOV lines. Quick self-calibration can be achieved by having only one person walk around the room a few times. This should be sufficient for determining the relationship

Figure 5: Determination of Edge of FOV lines using a short sequence of person walking in the room. The first 3 columns show triplets of sample images taken at same time instant. The last column shows the recovered lines

between the cameras. All lines of interest should be crossed at least twice during such a walk, which is often easily established during a 30-40 second random walk around the room. However, if the environment is busy and cannot be cleared of people, we can use the second method, which finds the statistical best line, treating every correspondence as a potentially correct one. This method needs more points for a reliable estimate of the lines and will therefore take longer to be setup correctly. However, it is completely automatic and does not need even the simple setup step required in the first method.

4. EXPERIMENTS AND RESULTS

To verify this formulation, we setup 3 cameras in room to cover most of the floor area. The setup is shown in Figure 4. To track persons, we used a simple background difference tracker. Each image was subtracted from a background image, and the result thresholded, to generate a binary mask of the foreground objects. We performed noise cleaning heuristically, by dilating and eroding the mask, eliminating very small components and merging components likely to belong to the same person. Occlusion is frequent in indoor environments, and to deal with occluding cases, we incorporated constant-velocity-based assumption in our tracker. Our tracker could not deal with one case of occlusion where a person exited from the image and at the same time another person entered the image from the same location, generating ambiguity. Since the emphasis of this paper is not to develop a robust technique for tracking during person to person occlusion, but rather to demonstrate the solution to the handoff

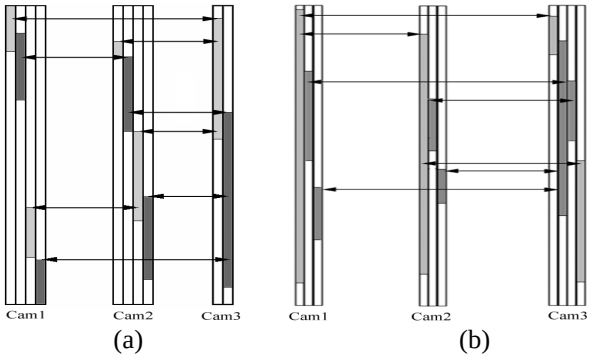

Figure 6: (a) Tracks of two persons as seen in the three cameras. A total of 10 tracks are seen. The first two tracks in Camera 1 are persons entering from the door. For all other tracks, an equivalence relation is established automatically, shown by the arrows. Because of the equivalence relations, globally correct labeling is achieved, shown by the different colors of the tracks. **(b)** Track of three persons as seen in three cameras in sequence 2

problem, we manually corrected this case of wrong tracking for the purposes of our experiments. Other than this one case, tracking was done automatically for all experiments.

To determine the FOV lines initially, we had one person walk around the room briefly. All significant edge of field of view lines were recovered from a short sequence of a single person walking in the room for only about 40 sec. Figure 5 shows some sample frames from this

Figure 7: Handoff examples in Sequence 2. In each of the cases in column 1, a person is entering a new camera. By looking at images in the 2nd column, we can correctly identify this person.

sequence and the edge of FOV lines recovered from this step. The lines found in this first step were used for the remaining experiment.

Next, two persons entered the room, walked among the cameras and exited. The tracking module tracked each view of these persons separately and assigned a unique label to each track in every camera. Overall, 10 different tracks of these persons were seen in the three cameras. Figure 6a shows all the tracks, which are 4 in C^1, 4 in C^2 and 2 in C^3. Our algorithm identified 8 situations where a new view of an existing person was observed. In each of these situations, a person was seen entering a new camera. The distance of all other persons from the edge of FOV of that camera is used to find the previous view of the person. The arrows in Figure 6a show the equivalence relations found out by our system. Once the arrows are marked, the complete tracking history of the person is recovered, by linking all the tracks of the same person together. The two different colors in Figure 6a show the globally consistent labels of the two persons. It can be seen that all handoffs were handled correctly, and the global tracking information was consistent at all times. The whole analysis part is very fast, as only the information about bounding boxes of the images and the lines is used in establishing the equivalence between tracks.

We performed another experiment involving three persons in a different environment. Figure 6b shows the recovered relationships between the 10 tracks seen in three cameras. In this case, our system correctly identified that these 10 tracks actually represented three different persons, with Person1 entering in Camera 1, then moving to Cameras 2 and 3 before exiting the room while seen by Camera 1, and so on. Figure 7 shows some of the handoff scenarios seen in this sequence.

CONCLUSION

We have described a framework to solve the camera handoff problem. We contend that camera calibration and 3D reconstruction is unnecessary for solving this problem. Instead, we present a system based on edge of FOV lines of cameras that can handle handoffs. We outline a process to automatically find the lines representing these limits, and then using them to resolve the ambiguity between multiple tracks. This approach does not require feature matching, which is difficult in widely separated cameras. The whole approach is simple and fast. We show results for a three-camera setup and resolve the handoff problem correctly.

References

[1] P. H. Kelly, A. Katkere, D. Y. Kuramura, S. Moezzi, S. Chatterjee, R. Jain, "An architecture for multiple perspective interactive video", *Proc. ACM Conf. Multimedia*, pp. 201-212, 1995

[2] Q. Cai, J. K. Aggarwal, "Tracking Human Motion in Structured Environments Using a Distributed-Camera System", *IEEE PAMI*, Vol. 2, No. 11, pp. 1241-1247, Nov 1999

[3] L. Lee, R. Romano, G. Stein, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame", *IEEE Trans on PAMI*, Aug 2000, pp. 758-768

[4] Vera Kettnaker, Ramin Zabih, "Bayesian Multi-Camera Surveillance", *Proceedings of Computer Vision and Pattern Recognition*, Fort Collins, CO, June 23-25, 1999, pp. 253-259

[5] Hanna Pasula, Stuart Russell, Michael Ostland, Ya'acov Ritov, "Tracking Many Objects with Many Sensors" In *Proc. IJCAI-99*, Stockholm 1999