Camera Handoff: Tracking in Multiple Uncalibrated Stationary Cameras

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

Abstract

Multiple cameras are needed to completely cover an environment 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 field of views 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. This helps us disambiguate between possible candidates of correspondence. The proposed approach is very fast compared to camera calibration based approaches.

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 completely 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. However, the purpose here is not to extract depth, rather, simply to completely cover all areas of the environment. 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 by global frame alignment. In a recent paper [4], the authors present a solution based on a combination of camera geometry and image features. They detect faces to find the epipolar geometry between cameras by assuming that the centroid of face boxes are corresponding points. Moreover, they use vertical features in the image to divide the view volume into non-overlapping sets, and matching view-volumes between cameras. They further use the hue

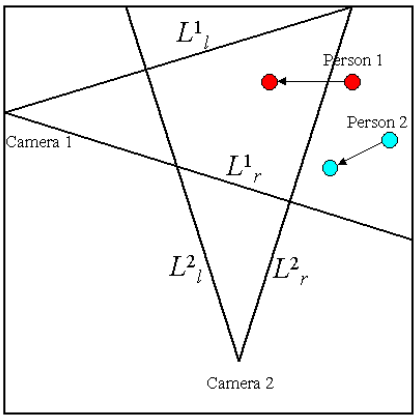


Figure 1: Person 1 and person 2 are both visible in Camera 1 initially. Person 1 then walks into the FOV of Camera 2, and needs to be identified as Person 1, for consistent labeling across views.

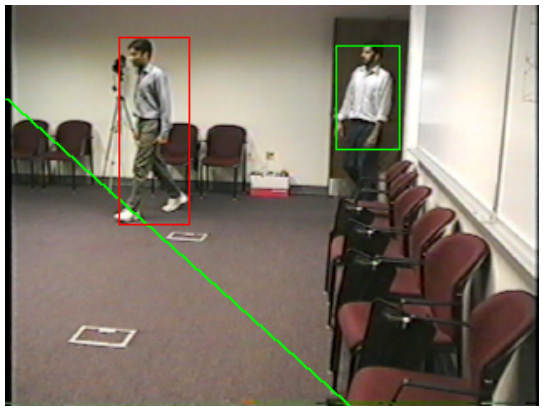
and saturation values of each person as features that help in establishing correspondence. Finally [5, 6] describe approaches which try to establish time correspondences between non-overlapping FOVs. The idea there is not to completely cover the area of interest, but to have motion constrained along 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 [5] or on a freeway [6].

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 only 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 this view. Consider the scenario shown in Figure 1; a room covered by two cameras with two persons walking in it. At time instant 1,



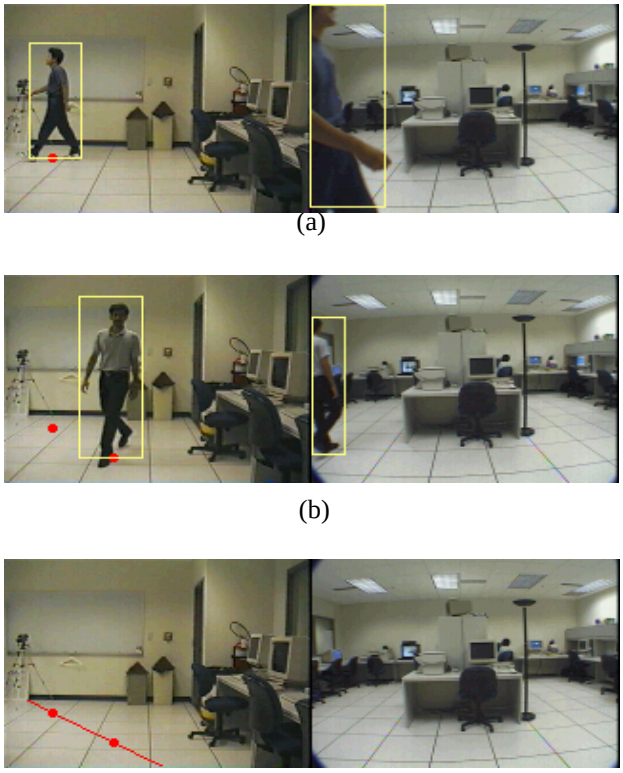
Camera 1

Camera 2

Figure 2: 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.

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. 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 some features (e.g. color) of the two persons visible in camera 1 to the new view in camera 2 for finding the most likely match. However, when the disparity is large, both in camera 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



Camera 1 Camera 2

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).

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.

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 lines L^i_l and L^i_r be the left and right edge of FOV lines 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 C^j be denoted by L^{ij}_x. Note that L^{ii}_x denotes the left or 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 along lines L^{ij}_x and L^{ji}_x. These are up to four lines, possibly two in each camera. When a person enters the FOV of a camera, all that needs to be done is to consider the associated line in the other camera and determine which person is crossing that line. Figure 2 describes this situation in more detail. A person is entering the FOV of C^2 from the left side. 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 that person towards the middle of the image has entered the FOV of C^2. The line that helped us determine this is L^{21}_l. 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.

In the examples given above, it is assumed that when a person is entering the FOV of a camera, he is always visible in the FOV of another camera. This is not always the case. The person may be either entering from a door (in which case he might just "appear" in the middle of the image) or he might be entering the FOV from a point which is not visible in any other camera. If the environment is completely covered by cameras, then the latter case will never happen. However, to keep our formulation general and to not put the constraint of complete coverage of the environment, we have to consider the second case too. For example, in out-door environments, complete coverage of all parts of the site might not be possible.

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. These lines partition the image in C^i into (possibly overlapping) regions, marking the areas of image in C^i that correspond to FOV of other cameras. Thus we will be able to determine all the other cameras in which the current person is visible, by determining the region in which the person's feet are.

Thus, with each edge of FOV line, we also want to store an additional variable which tells us which side of this line falls in the FOV and which side falls outside it. With each line L^{ij}_x, we store a variable δ^{ij}_x. The value of this variable can be either +1 or -1, depending on which side of the line falls inside the FOV of C^j. Then, given an arbitrary point (x', y') in C^i, we can find whether it is visible in C^j or not just by determining if this point is on the correct side of *both* L^{ji}_l and L^{ji}_r. Thus, if L^{ji}_l is represented by $Ax+By+C=0$, then let the value $L^{ji}_l(x', y')$ denote the value of the expression $Ax'+By'+C$. The point (x', y') is visible in C^j if and only if

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

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.

When a person enters the FOV of a new camera, we first determine whether this person is visible in the FOV of some other camera or not. If the person appears in the middle of the frame, we conclude that there must be a door location at this point[1]. If the person is entering from the side of the image, then we can determine all the other cameras in which this person will also be visible (by using Eq. 1). If there is no such camera, then we assign a new label to this person. Otherwise, we need to find the previous track of the person, so that a link can be established between the two views. Say that the person entered from the left side of C^1. Then, we will search the persons visible in all cameras C^j such that i=1, j satisfy Eq. 1. In all such cameras, we find which person is closest to the left edge of FOV line of C^1 in that camera. These two views will then be linked together by entering them in an equivalence table. In general, if we observe a person entering C^i from side x, then the label assigned to the new view will be:

$$label = \arg\min_{k} \left(D\left(P_k^j, L_x^{ij}\right)\right) \quad \text{for all } k, j \text{ such that } i, j \text{ satisfy Eq.1} \tag{2}$$

where k is the set of persons visible in C^j, P^j_k is the label assigned to a person in C^j 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.

```
Repeat for each frame:

For each camera C^i
  If new person appears in the
      middle of the image, assign
      him unique label,
  Else (person appears from side x)
    Find S = set of all cameras C^j
      such that current person is
      visible in C^j (Eq 1)
    If S = ϕ
      Assign current person unique
        label
    Else
      For each camera C^j, j ∈ S
        For each person k in C^j
          d(j,k) = D(P_k^j, L_x^ij)
        end
      end
    end
  end
end
Let s = row of min element of d
Let t = col of min element of d

Then P_t in C^s is the same as new person
in C^i
```

3. Automatic determination of FOV lines

At the beginning 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. 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 4. Once such a line is determined, it can be used to resolve ambiguous cases as described in the previous section.

This calibration is done automatically, even if the number of persons in the room in the beginning is more than one. If ambiguity exists before the lines are computed, then such cases are discarded. However, all correspondences that have only one possible solution contribute an additional constraint for a particular line. Thus if quick self-calibration is desired, only one person should walk around the room a few times, and this should be sufficient for determining the relationship between the cameras.

For cluttered situations where it is hard to find the correspondences to be used for calibration, we propose to use every pair of person views seen as a possible correspondence. The correct correspondences will yield a line in the single orientation, whereas the wrong correspondences will yield lines in scattered orientations. Hough transform can then be used to find the line with the maximum number of votes.

3.1 Confidence Measure:

At each time instant, the current best estimates of the FOV lines can have an associated confidence measure, based on the number of points that we have available to generate the line and the spread of these points. If two points are used to generate an edge of FOV line that are very close together in the image plane, then the line computed has a large error associated with it. Similarly the estimate for the line gets refined, as more points are available to compute the least-squares fit. We define the confidence associated with the line as a product of the

[1] Estimates of door locations are accumulated over time so that we have a higher confidence at a location where multiple observations indicate a door

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

variance in spatial location of the points generating the line and the number of points contributing to it. This confidence measure increases as more points that are well spread out become available.

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 reasoning in our tracker. When foreground regions merge into one another, their velocities are used to predict their positions over future frames, till they separate out again. The blobs are then reassigned their labels based on minimum distance from their predicted positions. 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. 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 problem, we manually corrected this case of wrong tracking for the purposes of our experiments. Other than this one case, tracking was done automatically.

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 sequence and the edge of FOV lines recovered from this step. For the purpose of this experiment, we did not incrementally improve our estimates of lines over time. 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 6 shows all the tracks, which are 4 in C^1, 4 in C^2 and 2 in C^3. Since the actual number of persons in the world is 2, eight links need to be established between these 10 tracks. The images in Figure 7 show the 8 triplets of images that were used to establish the equivalence links between the tracks. In each of these triplets, a person is 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 associated edge of FOV lines that are important in each case are also marked in Figure 7. The arrows in Figure 6 show the equivalence relations found out by our system. Once the arrows are marked, the complete history of tracking of the person is recovered, by linking all the tracks of the same person together. The two different colors in Figure 6 show the globally consistent labels of the two persons. It can be seen from these two figures 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.

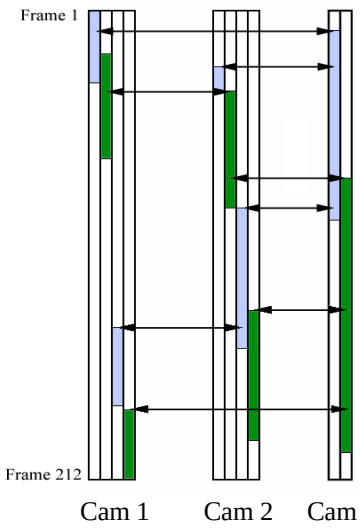


Figure 6: Tracks of two persons as seen in the three cameras. A total of 10 tracks are seen. The first two tracks in Cam 1 are new persons entering from the door. For all other tracks, an equivalence relation is established automatically, shown by the arrows. Because of the equivalence relations, globally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

5. Conclusion

We have described a framework to solve the handoff problem for environments that are covered by multiple cameras. We contend that the problem can be solved without going through the lengthy process of camera calibration, by finding the limits of FOV of a camera in other cameras. We outline a process to automatically find the lines representing these limits, and then use them to resolve the ambiguity between 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.

6. References

[1] P. H. Kelly, *et. al.*, "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", *PAMI*, Vol. 2, No. 11, pp. 1241-1247, Nov 1999

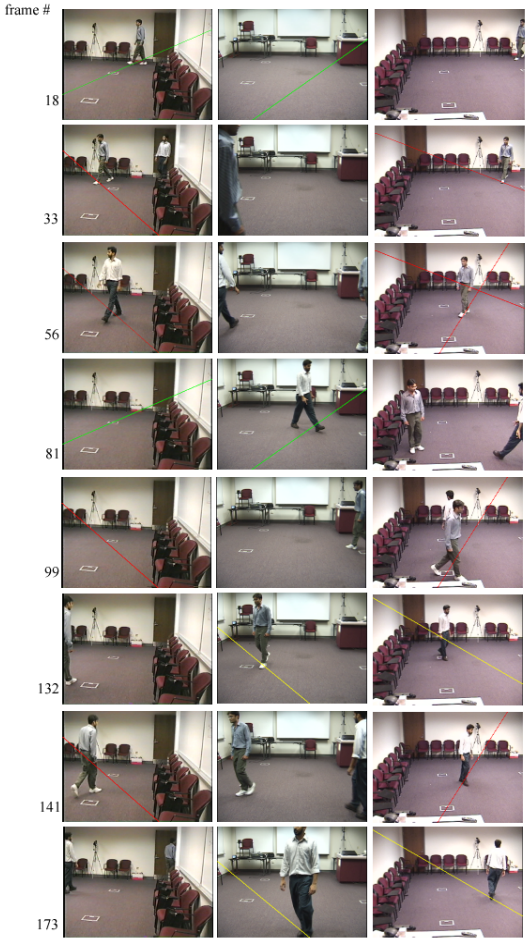


Figure 7: Handoff situations in all three cameras. Each row represents a triplet of images captured at the same time instant. One of the cameras has a person entering in it. The associated FOV lines in the other cameras help disambiguate the label of the new person. These frames result in the equivalence relationships shown in Figure 6.

[3] Gideon P. Stein, "Tracking from Multiple View-Points: Self-calibration of Space and Time", *DARPA IUW*, Montery CA, Nov 1998

[4] T-H. Chang, *et.al.*, "Tracking Multiple People under Occlusion using Multiple Cameras", in *BMVC*, Bristol, UK, Sept 2000

[5] Vera Kettnaker, Ramin Zabih, "Bayesian Multi-Camera Surveillance", *CVPR*, Fort Collins, CO, June 23-25, 1999, pp. 253-259

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