RECOVERING FIELD OF VIEW LINES BY USING PROJECTIVE INVARIANTS

Senem Velipasalar, Wayne Wolf

Princeton University
Electrical Engineering Department
Princeton, NJ 08544
{svelipas,wolf@princeton.edu}

ABSTRACT

Establishing correspondences between moving objects is an important problem in multiple camera tracking, and field of view (FOV) lines have been introduced in literature as an efficient tool to resolve the consistent labeling issue. We introduce a new and robust method to find the FOV lines, which uses projective invariants, thus does not rely on the object movement in the scene, and does not require information about camera parameters. As the labeling scheme suggested before is based on the distance of an object to an FOV line, accurate recovery of the FOV lines provides reliable labeling, and makes the process less error prune. We present results on different sequences, obtained from the PETS2001 database, which show the robustness of the algorithm in recovering all visible FOV lines of another camera in the current camera view.

1. INTRODUCTION

Efficient tracking of moving objects in complex environments is an important and challenging problem which finds wide-ranging application areas such as video surveillance, indexing and compression, gathering statistics from sports videos, traffic flow monitoring, and smart rooms. Although single-camera tracking may be sufficient for some applications, most of the time multiple cameras are needed. One reason is that a single camera has limited field of view (FOV) and cannot cover a wide enough area to track objects for longer periods of time. Certain applications, such as monitoring of areas for surveillance and statistics gathering purposes, require the coverage of larger areas and longer tracking times. In addition, occlusion can be an important problem for a single camera which may be overcome by using multiple cameras. Moreover, if a specific part of an object is of interest, such as the face of a person, a single camera cannot guarantee that this region will be in view.

Yet, tracking multiple objects with multiple cameras poses the challenge of providing the correspondences between tracks in different camera views. In multiple camera systems, rather than treating each camera individually, it is important to establish communication between cameras in order to obtain history of the object movements, and hand off some of the processing from one camera to the other. The latter makes the consistent labeling an interesting problem also for distributed camera systems which work better when less communication between cameras is required.

Several approaches have been taken to tackle the multiple camera tracking problem. Kelly [1] et al. assume that all cameras are calibrated to construct a $3D$ environment using voxel features, and track objects as a group of voxels. In [2] and [3], feature matching is used to establish correspondences, while camera calibration information is also needed. In [4], only the neighboring cameras are calibrated to their relative coordinates. $3D$ motion estimation is employed and the points on the middle line of the human body are matched by Bayesian classification schemes to provide correspondence.

Observation intervals and transition times of objects across cameras are used for tracking in [5], and the centroids of the tracked objects are used to estimate the homography between ground planes in [6]. In a more recent work, Khan et al. [7] present a new method, for cameras with overlapping FOVs, which does not require camera calibration. They first discover the *"FOV lines"*, which are described briefly in Section 3, and use a labeling scheme based on the smallest distance between a line and the middle point of the bottom of the boundary box of an object. They show that when FOV lines are recovered, the consistent-labeling problem can be solved successfully. But it is also stated that the feet of the people, and the portions of the ground have to be visible, and there needs to be enough traffic across a particular FOV line to be able to recover it. In addition, the method in [7] relies on the performance of the background subtraction (BGS) algorithm. Depending on the size of the objects, they may not be detected instantly and this effects the location of the FOV lines. But, as the labeling scheme in [7] is based on the distance between the line and the objects, the precision in locating the FOV line is very important especially in crowded scenes. In this paper, we present a ro-

This work has been funded by NSF ITR grant 325119.

bust and reliable way of finding the FOV lines, which uses projective invariants in P^2, and does not rely on the movement of the objects in the scene, and thus BGS. This way, all visible FOV lines in a view can be recovered even if there is no traffic at the corresponding region. In addition, with the proposed method, if the ground plane is not visible, i.e. if the cameras are watching some plane other than the ground plane, e.g. a wall, the FOV lines can still be obtained. The method we present in this paper, augmented with the labeling scheme in [7], provides a robust and reliable system to solve the correspondence problem in multi-camera tracking.

Invariant theory is a classical mathematical theory and has been applied to various computer vision problems such as object recognition and indexing into databases, optical tracking [8], shape reconstruction, and Euclidean reconstruction [9]. We present a method which applies the invariant theory to the area of multi-camera tracking by finding the limits of field of view of one camera as seen in the other cameras. In this method, there is no need to know the intrinsic or extrinsic parameters of the camera and no requirement about the camera disparities. As in [7], we assume the scene includes planar surfaces.

2. PROJECTIVE INVARIANTS IN P^2

A projective invariant is a measurement that does not change under the projective transformations. On the projective plane P^2, five points in general position, i.e. no three of them are collinear, have two independent projective invariants which are defined as follows [10]:

$$I_1 = \frac{|M_{421}^{(1)}||M_{532}^{(1)}|}{|M_{432}^{(1)}||M_{521}^{(1)}|} = \frac{|M_{421}^{(2)}||M_{532}^{(2)}|}{|M_{432}^{(2)}||M_{521}^{(2)}|}, \quad (1)$$

$$I_2 = \frac{|M_{421}^{(1)}||M_{531}^{(1)}|}{|M_{431}^{(1)}||M_{521}^{(1)}|} = \frac{|M_{421}^{(2)}||M_{531}^{(2)}|}{|M_{431}^{(2)}||M_{521}^{(2)}|}, \quad (2)$$

where $|M_{abc}^{(i)}|$, $\{a,b,c\} \in \{1,\ldots,5\}$, $i \in \{1,2\}$, denotes the determinant of the matrix

$$M_{abc}^{(i)} = \begin{bmatrix} x_a^{(i)} & x_b^{(i)} & x_c^{(i)} \\ y_a^{(i)} & y_b^{(i)} & y_c^{(i)} \\ 1 & 1 & 1 \end{bmatrix}$$

for image i, and $(x_a^{(i)}, y_a^{(i)}, 1)$ are the homogeneous coordinates of the point $p_a^{(i)}$ on image i.

In image 1, I_1 and I_2 can be calculated from five image points, $p_1^{(1)},\ldots,p_5^{(1)}$, which are the projections of the 5 coplanar points in space. Then if four of the five corresponding points, $p_1^{(2)},\ldots,p_4^{(2)}$, in image 2 are known, the fifth one can be recovered by using equations (1) and (2). It can be easily shown that by rewriting (1) and (2), and then solving for $x_5^{(2)}$ and $y_5^{(2)}$ we get the following two equations:

$$a_1 x_5^{(2)} - b_1 y_5^{(2)} + c_1 = 0, \quad (3)$$

$$a_2 x_5^{(2)} - b_2 y_5^{(2)} + c_2 = 0, \quad (4)$$

where

$$a_1 = I_1 \frac{|M_{432}^{(2)}|}{|M_{421}^{(2)}|}(y_2^{(2)} - y_1^{(2)}) - (y_3^{(2)} - y_2^{(2)}),$$

$$b_1 = I_1 \frac{|M_{432}^{(2)}|}{|M_{421}^{(2)}|}(x_2^{(2)} - x_1^{(2)}) - (x_3^{(2)} - x_2^{(2)}),$$

$$c_1 = I_1 \frac{|M_{432}^{(2)}|}{|M_{421}^{(2)}|}(x_2^{(2)}y_1^{(2)} - x_1^{(2)}y_2^{(2)}) - (x_3^{(2)}y_2^{(2)} - x_2^{(2)}y_3^{(2)}),$$

$$a_2 = I_2 \frac{|M_{431}^{(2)}|}{|M_{421}^{(2)}|}(y_2^{(2)} - y_1^{(2)}) - (y_3^{(2)} - y_1^{(2)}),$$

$$b_2 = I_2 \frac{|M_{431}^{(2)}|}{|M_{421}^{(2)}|}(x_2^{(2)} - x_1^{(2)}) - (x_3^{(2)} - x_1^{(2)}),$$

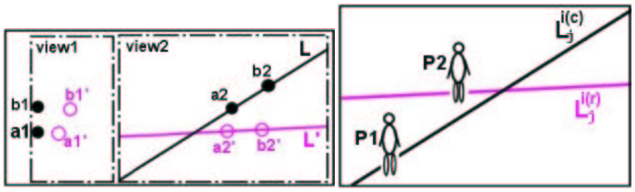
$$c_2 = I_2 \frac{|M_{431}^{(2)}|}{|M_{421}^{(2)}|}(x_2^{(2)}y_1^{(2)} - x_1^{(2)}y_2^{(2)}) - (x_3^{(2)}y_1^{(2)} - x_1^{(2)}y_3^{(2)}).$$

Solving for the coordinates, $x_5^{(2)}$ and $y_5^{(2)}$, of the fifth coplanar point will assist in recovering the FOV lines as explained in Section 4.

3. FIELD OF VIEW LINES

The FOV of a camera C^i is a rectangular pyramid in space whose apex is at the center of projection of the camera, and its base (the far clipping plane) and the image plane are parallel to each other. The intersection of each planar side s of this pyramid with the planes of the scene in space defines $3D$ FOV lines denoted by $L^{i,s}$ [7] where $s \in \{r, l, t, b\}$ and r, l, t, b correspond to the four sides of the image plane. When two cameras have overlapping FOVs, some or all of the lines $L^{i,s}$ will be visible by the other camera C^j. The projection of $L^{i,s}$ on C^j will result in $2D$ lines which will be denoted by $L_j^{i,s}$, and called *FOV lines* from now on.

As discussed in [7], knowing the boundaries of the FOVs of cameras are very important to solve the consistent labeling and hand-off problems in multi-camera multi-object tracking. In [7], it is argued that, if the single-camera tracking algorithm performs reasonably well, the consistent-labeling can be established if the FOV lines are recovered, and a system is introduced which finds the FOV lines of the uncalibrated cameras by observing moving objects in different camera views. However, it is stated that if the feet of the people in the scene are not visible, e.g. because of a cluttered scene or occlusion, the performance may break down, and if there is no traffic across a particular FOV line, it cannot be recovered. Moreover, to be able to determine the line, at least two passes from different points across the FOV boundary is necessary, and waiting for this may delay the detection of the line. Another requirement is that the background subtraction(BGS) algorithm should be able to detect the moving object as soon as it becomes visible, which may not be the case especially if the size of the object is too small and the BGS algorithm has a size threshold.



(a) (b)

Fig. 1. (a)$a1$-$a2$, $b1$-$b2$ are the corresponding point pairs in views 1 and 2. (b)Locating the FOV line reliably is very important for correct labeling.

Figures 1a and 1b are motivated by the comparison of the recoveries of a particular FOV line obtained by the method in [7], and the method presented in this paper. As shown in Fig. 1a, detecting the entrance of an object with delay (i.e, detecting them at $a1'$, $b1'$ instead of $a1$ and $b1$ respectively) causes the corresponding points on the other view to be marked at $a2'$, $b2'$ instead of $a2$ and $b2$, and the FOV line obtained, line L', is significantly deviant from its correct location. In Fig. 1b, lines $L_j^{i(c)}$ and $L_j^{i(r)}$ denote the correct and the recovered location of the FOV line of C^i as seen in C^j respectively, and $P1$ and $P2$ denote two moving people in C^j. When a new person enters the view of C^i, the label of the object in C^j, that is on the visible side of $L_j^{i(r)}$ and closest to it, is given to the new object in C^i. Let's consider the case in which the new person in C^i is detected with no delay, i.e, its size is large enough. Then its corresponding label in the view of C^j is $P1$. But when $L_j^{i(r)}$ is used, the label assigned to it will be $P2$. This example shows the importance of the reliable recovery of the FOV line. The alternative way of finding FOV lines, introduced in Section 4, augmented with the labeling scheme described in [7] makes the consistent labeling procedure in multi-camera tracking more robust.

4. RECOVERING FIELD OF VIEW LINES

We have developed a system that recovers all the limits of the FOV of a camera that are visible in the views of the other cameras by using the projective invariants in P^2. In order to achieve this, the algorithm partially interacts with the user. The system presents the user with the images of the different camera views. The user selects four points in one of the views whose FOV lines will be recovered (Fig. 2a). We call the image of this view the *field image*. (In general, any four points that are coplanar in space can be used as stated above). If any three of the selected four points in the *field image* are collinear the user is alerted and is asked for a new point. Then the user clicks on the four corresponding points in the other view (Fig. 2b). After pairs of four corresponding points for each view are obtained, the system finds two points on one of the boundaries of the *field image*, in a way that each one of them with four points chosen by the user will be in general position, and checks with the user that this point is coplanar with the other four points (for now, let's assume that such a point exists). In the same way, two points are found on each boundary (Fig. 2a). With five coplanar points, one on the boundary and four selected by the user, two independent P^2 invariants can be calculated by using equations (1) and (2). As we also know the four corresponding points in the other view, we can solve for the fifth one, which corresponds to the boundary point on the *field image*, by solving equations (3) and (4). Similarly, we solve for the corresponding point of the other point on the same boundary of the *field image*. These two points define the FOV line for that boundary (Fig. 2b). Let the *field image* and the image in which FOV lines are recovered be denoted by i and j respectively. As in [7], $L_j^{i,s}$ denotes the projection of $L^{i,s}$ onto camera j and is represented by the equation of the line. Let $p = (x, y)$ be one of the four chosen points on image j. Then a point (x', y') on the image j will be visible if and only if $sgn(L_j^{i,s}(x, y)) = sgn(L_j^{i,s}(x', y'))$. The side on which initially chosen points lie is the visible side and vice versa.

(a)

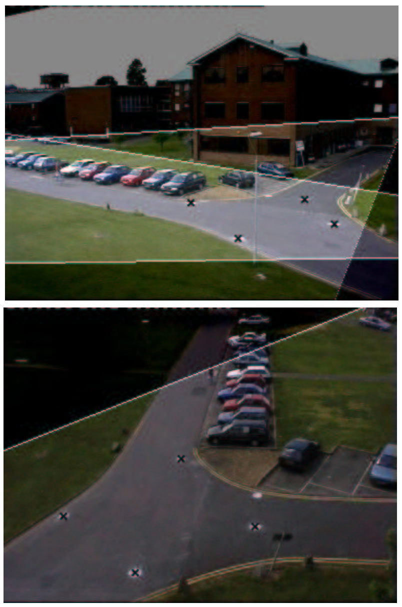
(b)

Fig. 2. Black and white crosses in (a) are the points chosen by the user and the points found on the image boundaries by the system, respectively. White crosses in (b) are recovered by the algorithm and correspond to the boundary points in (a).

There are cases in which there may not be any two points on a particular boundary, coplanar with the user chosen points, e.g. none of the points on the top boundary of Fig. 2b are coplanar with the chosen points on the ground. In these cases the corresponding FOV line is not found.

5. EXPERIMENTAL RESULTS

We have tried the proposed method on two different sequences obtained from the PETS2001 data set. Once the user chooses four pairs of corresponding points in two camera views, the FOV lines for both views are recovered automatically by the algorithm. The results are shown in Figures 3, and 4. Although there was no traffic along the right boundary of Fig. 3b, the FOV line corresponding to it is succesfully recovered as shown in Fig. 3a. Although the calculation of the projective invariants can be sensitive to positional errors, in our experiments we have observed that we can obtain the point positions and hence the FOV lines with sufficient accuracy.



(a) (b)

Fig. 3. (a) and (b) show the recovered FOV lines. The shaded regions are outside the FOV of the other camera.

(a) (b)

Fig. 4. (a) and (b) show the recovered FOV lines. The shaded regions are outside the FOV of the other camera.

6. CONCLUSION

We have introduced a new method for recovering FOV lines by using the projective invariants in P^2, and presented the results obtained on different sequences. This method does not rely on the object movement in the environment, and thus the detection of the FOV lines is instantaneous. In addition, there is no need to know the camera parameters. As the results are very robust and reliable, the user interaction to get four pairs of corresponding points is being tolerated for now. Our next goal is to determine the corresponding points in the other view automatically, once four points are chosen in the *field image*.

7. REFERENCES

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

[2] T.-H. Chang and S.Gong, "Tracking multiple people with a multi-camera system," *Proc. of IEEE Workshop Multi-Object Tracking*, pp. 19–26, July 2001.

[3] A. Utsumi, H. Mori, J. Ohya, and M. Yachida, "Multiple-camera-based human tracking using non-synchronous observations," *Proc. of Asian Conf. on Computer Vision*, pp. 1034–1039, 2000.

[4] Q. Cai and J.K. Aggarwal, "Tracking human motion in structured environments using a distributed camera system," *IEEE Trans. on PAMI*, vol. 21, no.11, pp. 1241–1247, Nov. 1999.

[5] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," *Proc. of CVPR*, pp. 253–259, 1999.

[6] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Trans. on PAMI*, pp. 758–768, Aug. 2000.

[7] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Trans. on PAMI*, vol. 25, no.10, pp. 1355–1360, Oct 2003.

[8] J.D. Mulder R. van Liere, "Optical tracking using projective invariant marker pattern properties," *Proc. of IEEE Conf. on Virtual Reality*, March 2003.

[9] D. Fofi, J. Salvi, and El M. Mouaddib, "Uncalibrated vision based on structured light," *Proc. IEEE Conf. on Robotics & Automation*, pp. 3548–3553, May 2001.

[10] C.A. Rothwell, *Object Recognition Through Invariant Indexing*, Oxford Science Publications, 1995.