Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 145715, 11 pages
doi:10.1155/2008/145715

Research Article

Multiview Trajectory Mapping Using Homography with Lens Distortion Correction

Gabin Kayumbi and Andrea Cavallaro

Multimedia and Vision Group, Department of Electronic Engineering, Queen Mary University of London, Mile End Road, London E1 4NS, UK

Correspondence should be addressed to Gabin Kayumbi, gabin.kayumbi@elec.qmul.ac.uk

Received 17 March 2008; Accepted 8 September 2008

Recommended by Jian Zhang

We present a trajectory mapping algorithm for a distributed camera setting that is based on statistical homography estimation accounting for the distortion introduced by camera lenses. Unlike traditional approaches based on the direct linear transformation (DLT) algorithm and singular value decomposition (SVD), the planar homography estimation is derived from renormalization. In addition to this, the algorithm explicitly introduces a correction parameter to account for the nonlinear radial lens distortion, thus improving the accuracy of the transformation. We demonstrate the proposed algorithm by generating mosaics of the observed scenes and by registering the spatial locations of moving objects (trajectories) from multiple cameras on the mosaics. Moreover, we objectively compare the transformed trajectories with those obtained by SVD and least mean square (LMS) methods on standard datasets and demonstrate the advantages of the renormalization and the lens distortion correction.

1. INTRODUCTION

Monitoring large areas such as airports and underground stations requires a set of distributed cameras to capture common patterns of activities and detect unusual events or anomalous behaviors [1]. Albeit object detection is traditionally carried out in each camera separately, there is an opportunity to coordinate and integrate information on activities across all cameras in order to improve the performance of the overall system. Multicamera settings with overlapping views provide elements of redundancy that help minimize ambiguities due to occlusions and increase the accuracy of the estimated objects' positions [2]. Moreover, after object extraction, data originating from individual cameras can be registered into a common frame of observation to facilitate scene analysis. The use of multiple cameras raises several issues such as image registration, camera calibration, object correspondence, and the fusion of visual information.

Once objects of interest have been extracted [1, 3], the spatial mapping (image registration) between images taken at different viewpoints is usually performed following three main steps [4]: (a) *feature detection*, which manually or automatically selects salient primitives such as contours, line intersections, or corners (*control points*); (b) *feature matching*, which establishes the correspondence between the detected features in the different images; (c) *transform model estimation*, which aligns the images. Although the image registration problem has been addressed in the literature [2, 4–6], misalignments in trajectory mappings are still a critical issue in multiview scene analysis [7].

In this paper, we present an algorithm for registering views and trajectories from multiple uncalibrated cameras with overlapping field-of-views, when the relative positions of the cameras are unknown. In particular, we overcome two major causes of trajectory misalignments due to the inaccuracies in the registration process and the distortions introduced by the camera lenses in the image acquisition process itself.

In most works involving trajectory mapping [2, 4–7], the authors adopt the linear pinhole camera model which assumes the principle of collinearity [8]. However, this model is only an approximation of the real camera projection. The centers of curvature of lens surfaces are not always perfectly collinear and this may result in the need for an extension of the model to include correction of the distortion introduced

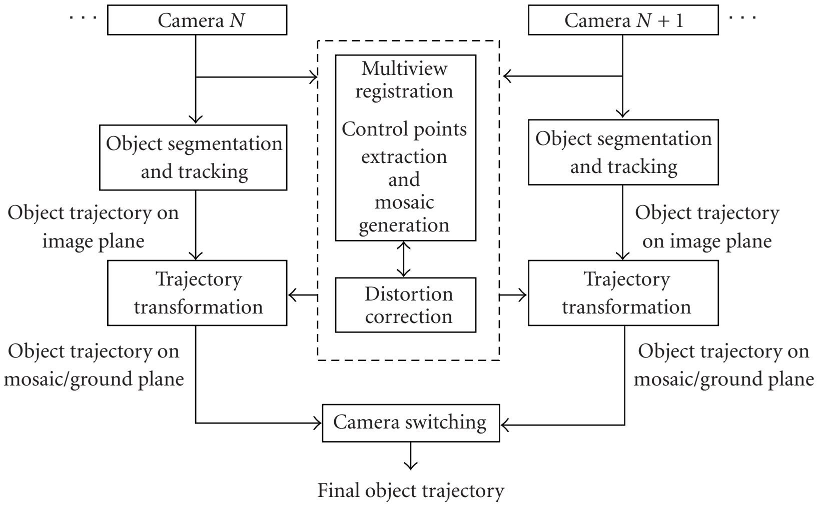

· · ·
Camera N
Camera $N+1$
· · ·
Multiview registration
Control points extraction and mosaic generation
Object segmentation and tracking
Object segmentation and tracking
Object trajectory on image plane
Object trajectory on image plane
Trajectory transformation
Distortion correction
Trajectory transformation
Object trajectory on mosaic/ground plane
Object trajectory on mosaic/ground plane
Camera switching
Final object trajectory

Figure 1: Block diagram of the proposed multicamera trajectory mapping approach.

by camera lenses [9]. Although lens distortion does not affect the quality of the image, it has, however, an impact on the image geometry. Hence a correction will be required. We present a framework that embeds lens distortion correction into a homography estimation algorithm that attains the theoretical accuracy bound in homography estimation by minimizing residual misalignments. The functional diagram of the proposed framework is shown in Figure 1.

The rest of the paper is organized as follows. In Section 2, we cover the related work. Section 3 describes the algorithms used to extract and track objects, and map them across views. Experiments are presented in Section 4. Finally, Section 5 presents the conclusions and future work.

2. RELATED WORK

Black and Ellis [5] identify corresponding blobs generated by the same object in different viewpoints and extract the 3D measurements of the object location using epipolar geometry to define correspondences. However, this method requires calibrated cameras. Additionally, since epipolar geometry may correspond a point to a line, this may lead to ambiguous matches [10]. Khan and Shah [11] use homography between views and a training phase during which one person enters and exits the field-of-views of the cameras. A limit of this approach is, for example, when a person enters or exists from the bottom of the image. Their ground location cannot simply be computed by taking the bottom point of the detected blob. Since the ground location of the objects is extracted at the camera hand-off, false correspondences will be generated because the assumption on homography ground plane is not valid anymore. A normalization technique is proposed by Kanatani et al. [12] to enhance the numerical stability of the homography since most linear methods are particularly sensitive to the accuracy of the correspondence as well as to the condition numbers of the matrices.

The process of matching control points across views to estimate homography can be affected by lens distortions, in particular at the image periphery [13]. Works on online distortion estimation often use straight lines in the scenes to provide constraints on the distortion parameters [14–16]. This implies the a priori availability of such lines and their detection. Such methods work under the assumption that a straight edge, bent by lens distortion, will deviate from a fitted segment. An optimization is performed on the distortion parameters to minimize the deviation of edges from straight lines. Unlike this approach, Stein [17] requires neither the 3D location of points nor the camera calibration but uses point correspondences in multiple views to recover epipoles and epipolar lines considering lens distortion. A cost function defined as the root mean square of the distances is computed from the feature points to the epipolar lines. Zhang [18] describes the epipolar geometry between two images with lens distortion by matching a point to its correspondent on the other image. The corresponding point is considered as lying on a curve rather than a straight line as it is the case in a distortion-free camera. Swaminathan et al. [19] derive a metric to measure distortions in multi-viewpoint images, but this method requires scene priors such as spheres, cylinders, or planes to be defined.

3. MULTIVIEW OBJECT TRACKING

3.1. *Object detection and tracking*

Prior to trajectory transformation, trajectories have to be estimated with a robust method to extract foreground objects from video, followed by a multiple object tracker. The best trajectory transformation process will remain bounded by the quality of the object segmentation process. A faulty segmentation in one camera generates a truncated blob that will undermine the point-to-point correspondence with the corresponding blob in another camera.

Object extraction can be achieved with an object detector or change detector that segments regions corresponding to moving objects. We use background subtraction [20] to extract foreground objects. The background of a stationary camera is modeled with a pixel-wise mixture of three Gaussian (MOG) distributions and each Gaussian is weighted in relationship with the frequency with which it explains the observed background. Gaussians with the highest weight that together explain over 50% of past data are considered as background.

Object tracking is performed by applying a graph matching algorithm on the detected objects [21]. Objects are treated as nodes of a bi-partitioned digraph (i.e., a directional graph), whereas edges are determined by all possible object combinations (track hypotheses) in adjacent frames and weighted using multiple object features, namely, position, direction, and size. The graph is formed by iteratively creating new edges from the detected targets. The optimal set of tracks is generated by computing the maximum weight path cover of the graph. Since graph matching links nodes based on the highest weights, a gating window is used to avoid connecting two trajectory points far from each other.

3.2. Homography estimation

The homography matrix, H, that relates the corresponding views is estimated based on a set of control points and it is computed as

$$\mathbf{x}' = H\mathbf{x}, \tag{1}$$

where $\mathbf{x} = (x, y, 1)$ is a point in one view and $\mathbf{x}' = (x', y', 1)$ is the corresponding point in the second view. The homography maps points lying on a plane in one view onto points on the same plane imaged on another view. Using homogeneous coordinates, (1) can be expressed as

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}, \end{aligned} \tag{2}$$

where the unknowns, h_{ij}, are entries of matrix

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}. \tag{3}$$

Rewriting (1) under a vector cross-product form

$$\mathbf{x}' \times H\mathbf{x} = 0, \tag{4}$$

and H as

$$\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}), \tag{5}$$

enables a simple linear solution for H to be derived [8]. n corresponding points will generate the following matrix:

$$A = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -y_2x_2' & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2' & -y_2y_2' & -y_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' & -y_n' \end{pmatrix}, \tag{6}$$

where $(x_i, y_i, 1)$ and $(x_i', y_i', 1)$ are the coordinates of the corresponding points that generate each 2×9 matrix A_i forming A. The linear form yields

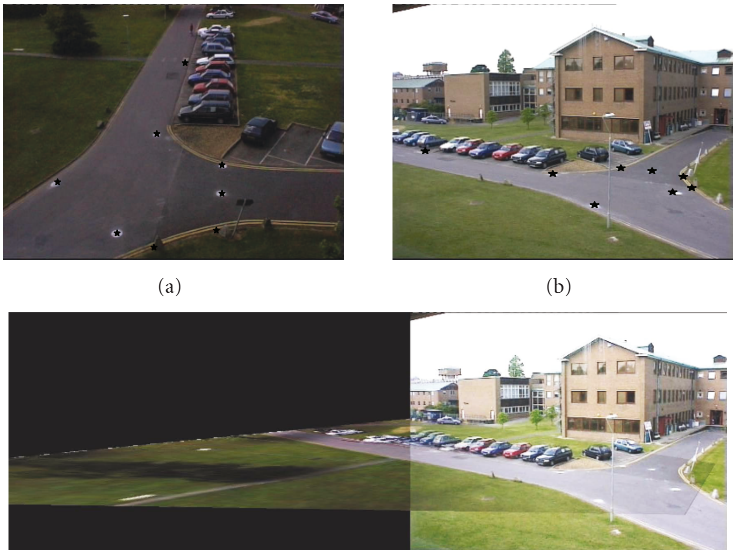
$$A\mathbf{h} = 0. \tag{7}$$

Most works [2, 5, 11] compute $\mathbf{h}$ using linear methods. The key idea in these methods is that given a pair of corresponding pixels, (7) is linear in the unknown $\mathbf{h}$, whereas the entries of A_i are quadratic in the known coordinates of the points. This means that given enough equations, it is possible to implement linear algebra methods to compute the coefficients of H. Direct linear transformation (DLT) [8] is a widely used linear method that minimizes the algebraic residuals $\|A\mathbf{h}\|$, subject to $\|\mathbf{h}\| = 1$. The solution is the unit eigenvector corresponding to the smallest eigenvalue of A^TA. The superscript indicates the transpose matrix. This eigenvector can be obtained directly by the singular value decomposition (SVD) of A. If the vector $\epsilon = A\mathbf{h}$ is the residual vector, then its components are derived from the individual correspondences that generate each row of A. Each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$ contributes a partial error vector ϵ_i (algebraic error vector), toward the full error vector ϵ, whose norm is the algebraic distance. Although DLT, based on SVD, has the advantage of easy implementations, it is quite sensitive to noise. In addition to this, one cause of error in the matrix computation lies in the value being minimized as it does not account for the noise in the geometry of the corresponding points [22]. To overcome this problem, we use a renormalization technique which attains the theoretical accuracy bound in homography estimation [12, 23].

In this approach, the uncertainty of data points (x_α, y_α) and (x_α', y_α') is described by their respective covariance matrices Σ_α and Σ_α'. It follows that the vectors $\mathbf{x}$ and $\mathbf{x}'$ of (4) have the following singular covariance matrices

$$\begin{aligned} V[\mathbf{x}_\alpha] &= \frac{1}{f^2}\begin{pmatrix} \Sigma_\alpha & 0 \\ 0^T & 0 \end{pmatrix}, \\ V[\mathbf{x}_\alpha'] &= \frac{1}{f^2}\begin{pmatrix} \Sigma_\alpha' & 0 \\ 0^T & 0 \end{pmatrix}, \end{aligned} \tag{8}$$

where f is a scale factor. If the covariance matrices are known up to that scale, then $V[\mathbf{x}_\alpha] = \epsilon^2 V_o[\mathbf{x}_\alpha]$ and $V[\mathbf{x}_\alpha'] = \epsilon^2 V_o[\mathbf{x}_\alpha']$, where ϵ is the noise level. The normalized covariance matrices $V_o[\mathbf{x}_\alpha]$ and $V_o[\mathbf{x}_\alpha']$ indicate the relative dependence of noise occurrence on positions



(a)

(b)

(c)

Figure 2: (a)-(b) Two views with corresponding control points (dark landmarks) on the overlapping area; (c) image mosaic generated after homography transformation.

and orientations. By modeling the uncertainties in geometric inference [23], the theoretical accuracy bound is attained in its first order by minimizing the squared Mahalanobis distance $\mathcal{M}$ as:

$$\mathcal{M} = \sum (\mathbf{x}'_\alpha \times H\mathbf{x}, W_\alpha(\mathbf{x}'_\alpha \times H\mathbf{x}_\alpha)), \tag{9}$$

where W_α denotes the matrix

$$W_\alpha = (\mathbf{x}'_\alpha \times HV_o[\mathbf{x}_\alpha]H^T \times \mathbf{x}'_\alpha + (H\mathbf{x}_\alpha) \times V_o[\mathbf{x}_\alpha] \times (H\mathbf{x}_\alpha))_2^{-}. \tag{10}$$

Details on the algorithm are presented in [12].

3.3. Mapping

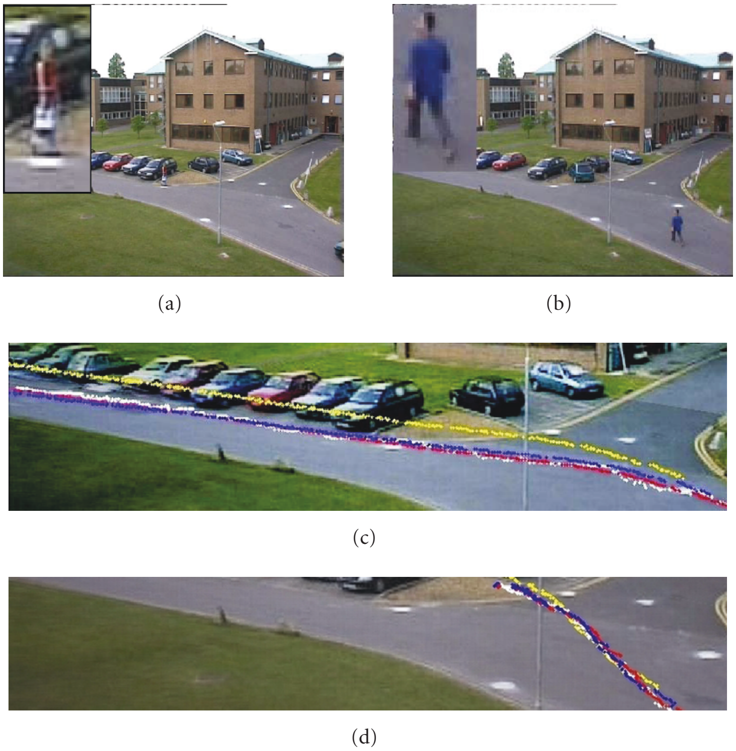
We generate the mosaics from overlapping views to create a larger cameras' field-of-view that represents the common coordinate frame where we map the objects' trajectories. A mosaic allows an uninterrupted observation of objects that enter and exit individual camera's field-of-views. We proceed in a pairwise mode by first aligning the two images then by applying image stitching to generate the composite image [24]. The alignment is obtained by warping one image onto the other, considered as reference view, using the estimated homography transformation. Although aligned, a simple juxtaposition of the two images would create visible photometric artifacts such as inconsistencies in pixel colors in the resulting mosaic. We apply image stitching by pixel selection and center-weighting [24]. We blend pixel colors in the overlapping area by interpolating the pixel intensities in that region. Since we pursue a seamless merging, the colors of the pixels in the overlapping areas are weighted through averaging. For this purpose, we calculate the centers of the images and use them as coefficients to weight pixels intensities in the overlap. Let I_1 and I_2 represent the pixel intensity in the first and the second images , respectively. Furthermore, let a_1 be the Euclidean distance between the center of the first image to the pixel. The same computation is carried out for a_2 with respect to the second image. The resulting pixel intensity, I, of the composed image is given by

$$I = \left(\frac{a_1}{a_1 + a_2}\right) I_1 + \left(\frac{a_2}{a_1 + a_2}\right) I_2. \tag{11}$$

An example of homography estimation based on manually selected control points and the resulting mosaic after image alignment and stitching is shown in Figure 2. The dark dots in Figures 2(a) and 2(b) indicate the selected control points. The view illustrated in Figure 2(a) is warped onto Figure 2(b) which is the reference view. Colors in the overlap are blended and the mosaic is shown in Figure 2(c). However, when using 2D planar homography to warp images we still can observe undesirable effects for objects that do not lie on the plane of reference. On the warped image, these objects are bent toward the plane as a result of the strong coplanarity assumption of the homography transformation [8]. The image mosaic shown in Figure 2(c) illustrates this effect. After warping Figure 2(a) onto Figure 2(b), the tree on the top left of the image is bent in Figure 2(c).

As areas corresponding to overlapping field-of-views, there are concurrent object observations, a decision is taken that results in a single observation on the mosaic [25]. We assume that the most reliable measurement of an object spatial location is given by the observation from the closest camera (*camera switching*). At a time t, given a detected object located at spatial coordinates (x_1^t, y_1^t) in camera 1 and (x_2^t, y_2^t) in camera 2, the closest camera is the one whose y object coordinate is closer to the bottom of the image plane.

Figure 3 compares the trajectories of two targets from the PETS2001 dataset (http://peipa.essex.ac.uk/ipa/pix/pets/PETS2001/) superimposed on the mosaic (Figure 3(c))



(a)

(b)

(c)

(d)

Figure 3: Comparison of trajectory transformations for two targets, (a) P_1 and (b) P_2, on the image mosaic of Figure 2; (c)-(d) zoom on the transformed trajectories: ground truth (white), proposed approach (red), SVD (yellow) and LMS (blue).

Table 1: Trajectory transfer error with and without renormalization. μ and σ indicate the mean and the standard deviation of the resulting transfer errors.

	Without renormalization				With renormalization	
	SVD		LMS		Proposed approach	
Target	μ	σ	μ	σ	μ	σ
P_1	5.97	3.11	3.86	2.49	3.05	2.12
P_2	4.51	1.99	2.87	1.86	2.29	1.01

generated with 3 different methods: the proposed algorithm, the SVD-based and the LMS-based algorithm [8]. Figures 3(c) and 3(d) show a close-up on the targets' transformed trajectories, illustrating the displacement between the expected location (ground-truth: white line) of the target and their actual measurements after transformation. The quantitative evaluation of the results is based on the transfer error e defined as

$$e = d(\tilde{\mathbf{q}}_i, H\mathbf{q}_i), \qquad (12)$$

where $d(\cdot)$ is the Euclidean distance; $\mathbf{q}_i$ is the trajectory point to be transformed; H the transformation and $H\mathbf{q}_i$ is the estimated value resulting from the transformation of the trajectory point $\mathbf{q}_i$. $\tilde{\mathbf{q}}_i$ is the ground-truth location of the transformed trajectory point. Table 1 shows the improvement in the trajectory mapping accuracy when using the renormalization technique as opposed to SVD and LMS, which do not use renormalization. The reported values are expressed as mean values, with their corresponding variance.

We compare the robustness of the three approaches against noise and report mean values of the variation of the transfer error in Table 2. The robustness test is performed by corrupting the selected control points with varying magnitudes of Gaussian noise N and by then estimating the subsequent transfer error on the transformed trajectories. The results show a smaller increase in the transfer error for the proposed approach than for the SVD-based and in LMS-based methods. This is due to the fact that the renormalization accounts for the geometric noise in the point-to-point correspondence.

In the next section, we will take into account another source of errors, namely, the distortion introduced by the camera lenses.

3.4. Embedding lens distortion in the homography estimation

Radial lens distortion can be a significant factor introducing errors typically in the range of 10–100 pixels at the edges of the image [17]. To overcome this problem, we take into account the radial distortion that can be introduced by the camera lenses when estimating the point-to-point correspondence between views.

Let an undistorted image point $(x_u, y_u, 1)$ be subject to a radial distortion and let $(x_d, y_d, 1)$ be the resulting

Table 2: Influence of noise on the trajectory transfer error. N indicates the noise amplitude (Gaussian) used to corrupt the control points; μ and σ denote the mean and the standard deviation of the resulting transfer errors for target P_1 and target P_2.

	N	SVD		LMS		Proposed approach	
		μ	σ	μ	σ	μ	σ
P_1	1	6.85	4.24	4.16	2.59	3.65	2.19
	2	7.69	5.32	5.35	2.98	4.08	2.31
	4	8.94	6.37	5.96	3.81	4.57	2.75
	5	9.35	5.92	6.66	4.41	5.38	2.93
	7	13.38	7.03	7.49	4.57	6.01	3.18
	8	14.84	7.19	9.01	5.05	6.72	4.01
	10	17.27	8.25	12.11	5.45	8.12	4.37
	12	19.97	9.33	13.89	5.88	9.25	4.49
	15	21.91	10.39	16.41	7.09	11.36	6.28
P_2	1	5.98	3.97	3.49	2.20	2.73	1.46
	2	6.71	4.56	3.89	2.25	2.97	1.80
	4	8.85	5.90	4.97	2.91	4.13	2.34
	5	11.34	6.90	6.92	3.53	5.37	2.93
	7	15.27	8.94	7.72	3.79	6.35	3.45
	8	19.97	10.66	10.17	4.66	7.64	3.95
	10	20.03	11.50	11.84	5.42	8.39	4.41
	12	20.89	12.18	13.91	5.95	9.44	5.06
	15	23.29	14.15	15.31	7.10	12.26	6.62

```
Goal: Given corresponding points in two views, estimate
the homography relating the two images.
(1) Compute the set of distorted corresponding control
    points pairs
        $\begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} \longleftrightarrow \begin{pmatrix} x'_d \\ y'_d \\ 1 \end{pmatrix}.$
(2) Scale the control points by subtracting the center
    and then normalizing by the sum of the image
    width and height.
(3) Compute $[V, A^{-1}] = \mathrm{polyeig}(D_1^T D_3, D_1^T D_2, D_1^T D_1)$,
    where $V$ is the matrix of eigenvectors and $A^{-1}$ the
    corresponding inverse eigenvalues.
(4) Discard imaginary and null eigenvalues and select the
    median value $k_1$ from the above remaining eigenvalues.
(5) Compute corresponding pairs of undistorted control
    points
        $\begin{pmatrix} x_d \\ y_d \\ 1 + k_1 r^2 \end{pmatrix} \longleftrightarrow \begin{pmatrix} x'_d \\ y'_d \\ 1 + k_1 {r'}^2 \end{pmatrix},$
    where $r = \sqrt{(x_d)^2 + (y_d)^2}$ and $r' = \sqrt{(x'_d)^2 + (y'_d)^2}$.
(6) Minimize the squared Mahalanobis distance $\mathcal{M}$:
        $\mathcal{M} = \sum((\mathbf{x}'_\alpha)_\mathbf{u} \times H\mathbf{x}_\mathbf{u}, W_\alpha((\mathbf{x}'_\alpha)_\mathbf{u} \times H(\mathbf{x}_\alpha)_\mathbf{u})),$
    updating $W_\alpha$ using the renormalization technique,
    where $W_\alpha$ denotes the matrix
$W_\alpha = (\mathbf{x}'_\alpha \times H V_o[\mathbf{x}_\alpha] H^T \times \mathbf{x}'_\alpha + (H\mathbf{x}_\alpha) \times V_o[\mathbf{x}_\alpha] \times (H\mathbf{x}_\alpha))_2^-.$
(7) Obtain the estimated homography $H$ from step (6).
```

Algorithm 1: Undistorted homography estimation.

distorted point. The distorted and undistorted points satisfy the following:

$$x_u = x_d + x_d(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots),$$
$$y_u = y_d + y_d(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots), \tag{13}$$

where $r = \sqrt{(x_d)^2 + (y_d)^2}$ and k_i are coefficients of the radial distortion. As including more coefficients increases the risk of numerical instability in the distortion model [26], we consider only the first term of the radial distortion.

We embed the division distortion model [13] in the correspondence algorithm. The geometric constraints in the homography matrix estimation (1) is augmented to include the first term, k_1, of the radial lens distortion

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \\ 1 + k_1(x_d^2 + y_d^2) \end{pmatrix}, \tag{14}$$

where $\mathbf{p} = (x_u, y_u, 1)$ is the distortion-free point, $\mathbf{x} = (x_d, y_d, 1)$ the distorted point, and k_1 the distortion parameter. Thus,

$$\mathbf{p} = \mathbf{x} + k_1 \mathbf{z}, \tag{15}$$

where $\mathbf{z} = (0, 0, (x_d^2 + y_d^2))$. The homography constraint in (1) can be expressed in terms of vector cross product for each corresponding pair $(\mathbf{p}_i, \mathbf{p}'_i)$ using (15) as

$$(\mathbf{x}'_d + k_1 \mathbf{z}') \times H(\mathbf{x}_d + k_1 \mathbf{z}) = 0, \tag{16}$$

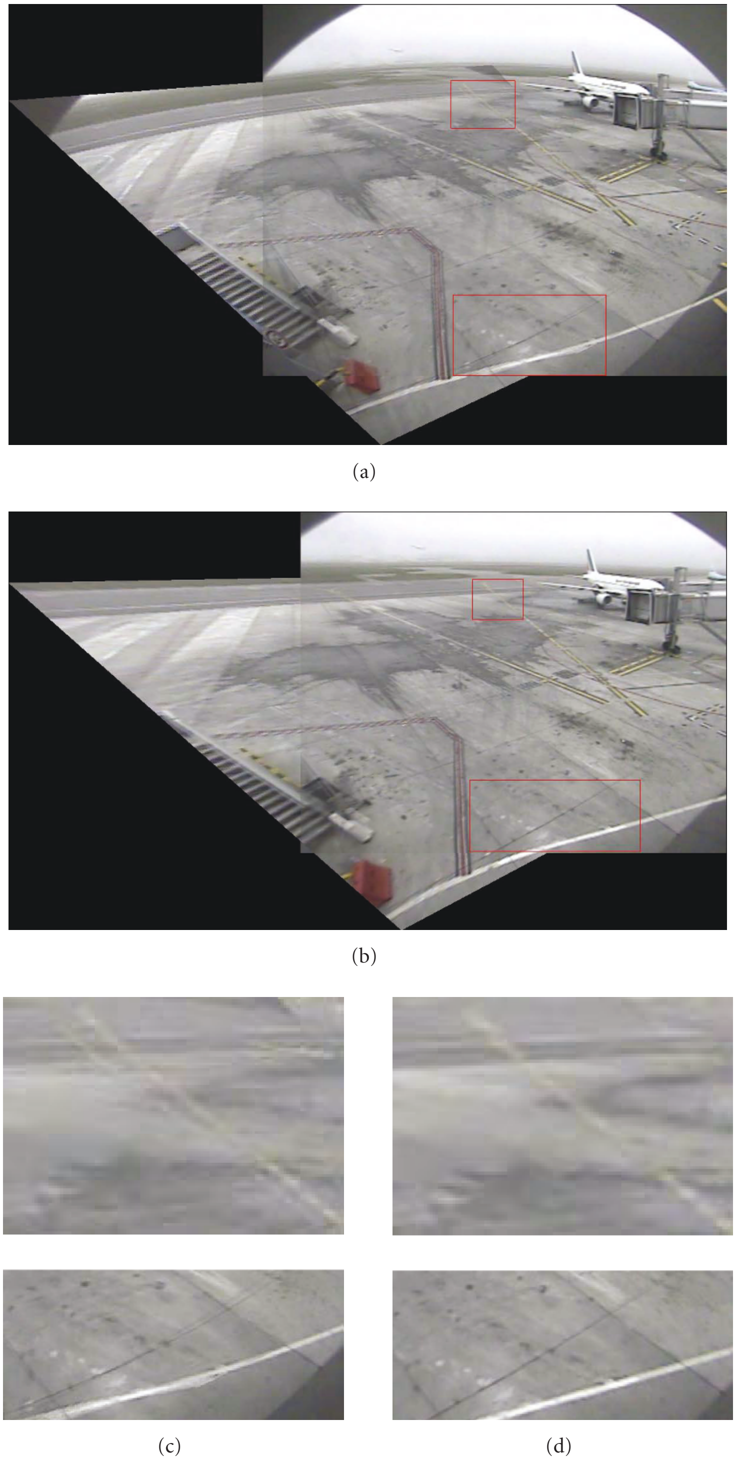
which is quadratic in k_1 and linear in H. Expanding with the coordinates, we obtain

$$(D_1 + k_1 D_2 + k_1^2 D_3)\mathbf{h} = 0, \tag{17}$$

where $\mathbf{h}$ is the vector in (5) and the coefficients D_r are such that

$$D_1 = \begin{pmatrix} 0 & 0 & 0 & -x'_d & -y'_d & -1 & y_d x'_d & y_d y'_d & y_d \\ x'_d & y_d & 1 & 0 & 0 & 0 & -x_d x'_d & -x_d y'_d & -x_d \end{pmatrix},$$
$$D_2 = \begin{pmatrix} 0 & 0 & 0 & -r x'_d & -r y'_d & -r - r' & 0 & 0 & y_d r' \\ r x'_d & r y'_d & r' + r & 0 & 0 & 0 & 0 & 0 & -x_d r' \end{pmatrix},$$
$$D_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -r' r & 0 & 0 & 0 \\ 0 & 0 & r' r & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{18}$$

However, (17) is a quadratic eigenvalue problem (QEP). The solution of this equation yields 4–6 nonimaginary, nonnull values. The best values of k_1 have been determined as corresponding to the median value of the set of solutions. In summary, the embedded distortion correction is performed according to Algorithm 1. Figure 4 visualizes the benefits of the correction of radial lens distortion. A mosaic from two views with overlapping areas is shown before and after lens distortion correction. Because of the radial distortion,



(a)

(b)

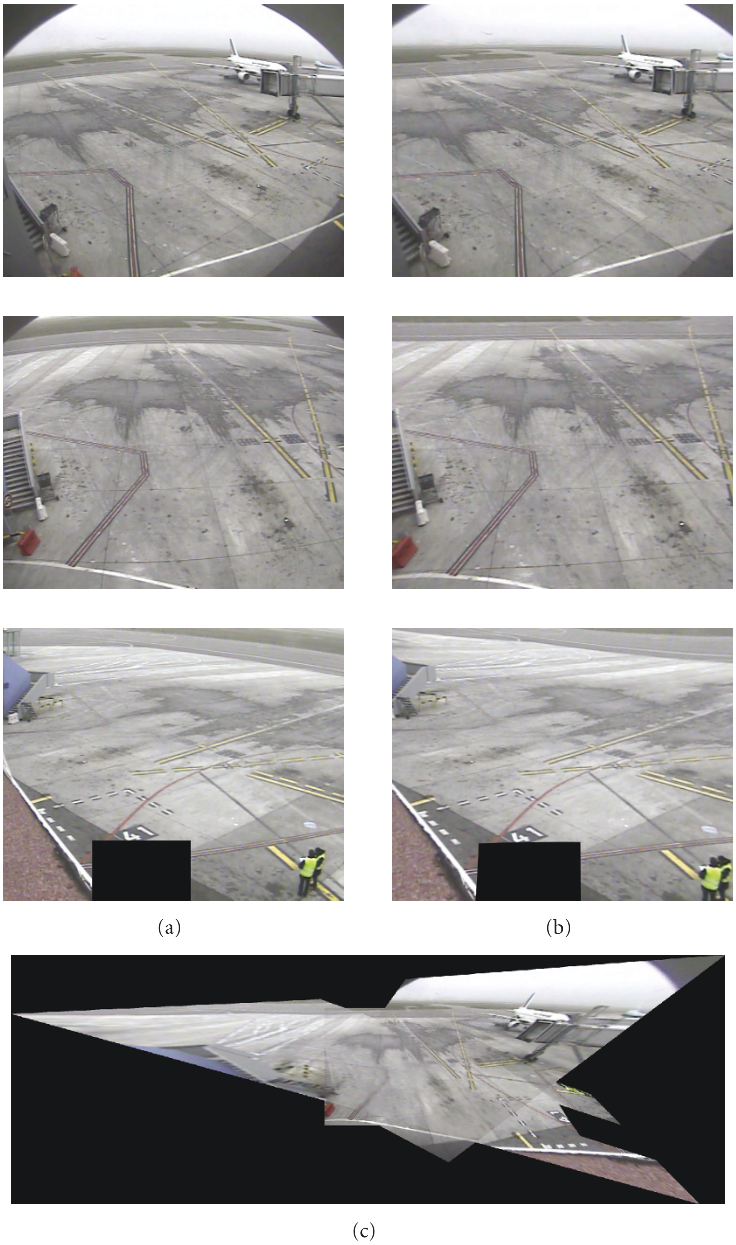
(c)

(d)

Figure 4: Example of distortion correction in image mosaics. (a) Image mosaic without distortion correction; (b) image mosaic after lens distortion correction; (c) zoom from (a); (d) zoom from (b). (c) and (d) refer to the areas enclosed by the red rectangles in (a) and (b). Note the difference in the residual misalignment between (c) before and (d) after lens distortion correction.

residual misalignments are visible on Figure 4(a) (before lens distortion correction), particularly with the white and yellow lines located at the borders of the image. A significant improvement is obtained after correction as illustrated by the alignment in Figure 4(b).

Based on the above, the estimation of the homography transformation that embeds lens distortion correction for N cameras ($N \geq 3$) proceeds on a pairwise basis. The homography H_{ij} will relate camera C_i to C_j; H_{jk}, camera C_j to C_k; and so on. The distortion parameter computation



(a)

(b)

(c)

Figure 5: Example of three-view correspondence. (a) Original images (distorted); (b) images after radial distortion correction; (c) generated mosaic from the three views (after distortion correction).

is image-to-image dependent. This means the correction obtained for camera C_j while estimating the homography H_{ij} is only applied to the image C_j in the correspondence $C_i \leftrightarrow C_j$, and the radial correction parameter for camera C_j will be computed when estimating the homography H_{jk} in the correspondence $C_j \leftrightarrow C_k$. Figure 5 shows the original, distorted images (Figure 5(a)) and the corrected images (Figure 5(b)), as well as the corresponding mosaic after a three-view correspondence.

4. EXPERIMENTAL RESULTS

We demonstrate the proposed approach for trajectory transformation with lens distortion correction on the ETISEO dataset (http://www-sop.inria.fr/orion/ETISEO/) and compare the results with those of SVD and LMS. We analyze examples of resulting object detection and tracking across multiple views and of trajectory mapping on mosaics whose distorted images have been corrected.

Four sequences of 110, 300, 100, and 170 frames (the image size is 720 × 576 pixels) with moving pedestrians have been used. For fairness of comparison, the same distortion correction is applied to all methods. Figure 6 shows the results from object segmentation and tracking of targets E_1, E_2, E_3 and E_4.

Figure 7 shows the correction of two objects' trajectories (target E_1 and target E_2). Note the difference between the distorted (red) and corrected (blue) trajectory when the

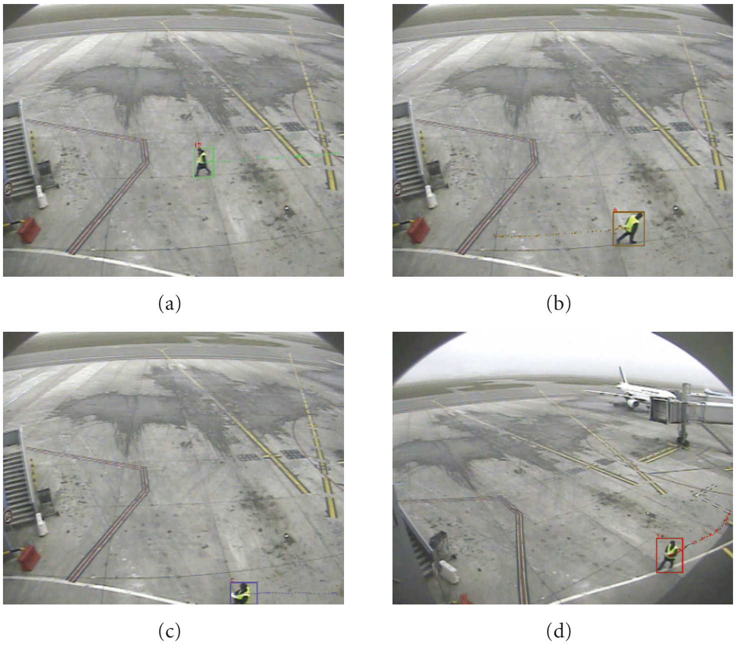


FIGURE 6: Sample targets from the ETISEO dataset. (a) Target E_1; (b) target E_2; (c) target E_3; (d) target E_4.

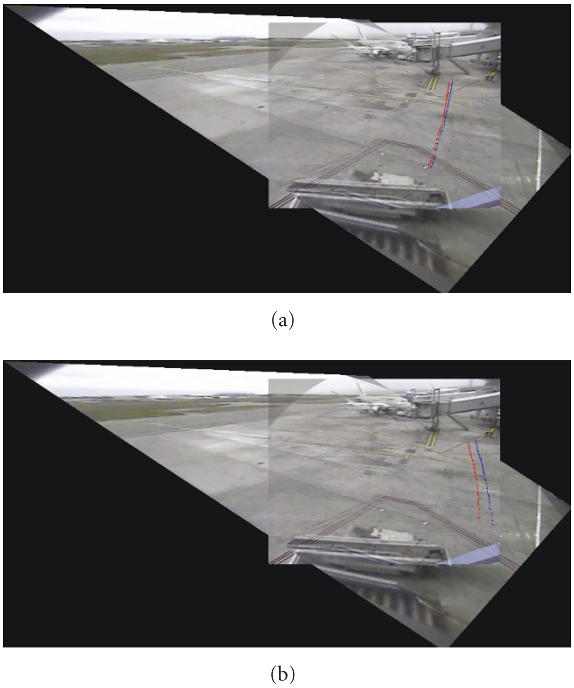


FIGURE 7: Example of distorted trajectory (in red) and corrected (in blue) on the mosaic. (a) Target E_1; (b) target E_2.

target moves closer to the image periphery. With lens distortion correction, we rectify the trajectory points location with distances that reach 10 pixels for E_1, 45 pixels for E_2, 62 pixels for E_3, 88 pixels for E_4. These quantities measure the differences between the trajectory points before and after lens distortion correction.

Figure 8 illustrates the variation of the transfer errors over time for the ETISEO targets. The results show that the linear SVD method presents larger errors than both LMS and the proposed approach. The linear estimation has its lowest errors when a target is close to the control points. As targets move away from the control points, the drift between the estimated and the expected location becomes more and more significant. The main reason behind the relatively lower performance of SVD resides in the way this technique estimates the homography transformation matrix. Indeed, its linear estimation consists of a pure algebraic solution to the geometric problem of fitting noisy corresponding points in a homography relationship. The absence of geometric constraints to relate the corresponding points is likely to degrade the homography in those areas.

LMS presents a better fitting due to the use of a geometric cost function that minimizes the transfer error. However, one disadvantage of this model is that it requires an additional phase in the homography estimation that consists of an initialization with the linearly estimated homography matrix. Besides, LMS assumes that the entire data can be interpreted by only one parameter vector of a given model and even when the data contain only one bad datum, LMS estimates may be completely perturbed [27]. This limitation was also confirmed by the experiments on perturbations generated by a Gaussian noise introduced on the control points location (see Table 2).

5. CONCLUSIONS AND FUTURE WORK

We presented an algorithm for trajectory transformation for wide-baseline multicamera scene analysis with embedded lens distortion correction. The proposed approach first estimates homography from multiple overlapping uncalibrated

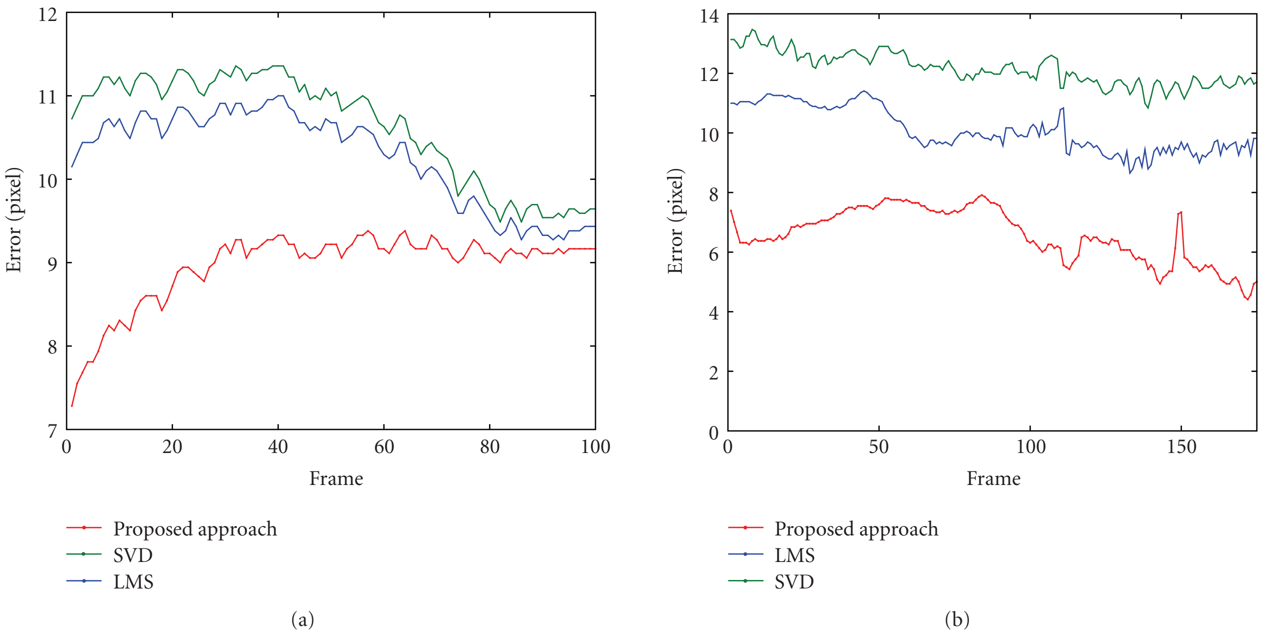


Figure 8: Comparison of transfer errors over time for (a) target E_3 and (b) target E_4. The distortion correction is applied to all 3 methods.

cameras and then blends them to generate mosaics on which object trajectories are registered. Using the transfer errors, we demonstrated that the proposed method improves the accuracy of the trajectory transformation compared to state-of-the-art methods. Moreover, we have demonstrated that this approach is more robust to errors in the estimation of the control points and that the perturbation in the trajectory transformation is smaller than that of traditional approaches using linear (SVD) or nonlinear (LMS) homography estimation.

Future work includes improvements in the fusion of the trajectory data from the cameras observations and operating in the image gradient domain to improve the removal of artificial edges in mosaics [28]. We also plan to address the issue of off-plane objects by combining homography with epipolar geometry [29].

ACKNOWLEDGMENT

The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPRSC), under Grant EP/D033772/1.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[2] T. Ellis, J. Black, and P. Rosin, "Multi-view image surveillance and tracking," in *Proceedings of the Workshop on Motion and Video Computing (MOTION '02)*, pp. 169–174, Orlando, Fla, USA, December 2002.

[3] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 95–105, 2007.

[4] A. Goshtasby and A. Ardeshir, *2-D and 3-D Image Registration for Medical, Remote Sensing and Industrial Applications*, John Wiley & Sons, New York, NY, USA, 2005.

[5] J. Black and T. Ellis, "Intelligent image surveillance and monitoring," *Measurement and Control*, vol. 35, no. 8, pp. 204–208, 2002.

[6] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.

[7] W. Nunziati, J. Alon, S. Sclaroff, and A. Del Bimbo, "View registration using interesting segments of planar trajectories," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 75–80, Como, Italy, September 2005.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2004.

[9] C. C. Salma, *Manual of Photogrammetry*, American Society of Photogrammetry, Falls Church, Va, USA, 4th edition, 1980.

[10] J. Black, *Multi-view image surveillance and tracking*, Ph. D. thesis, City University, London, UK, April 2004.

[11] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355–1360, 2003.

[12] K. Kanatani, N. Ohta, and Y. Kanazawa, "Optimal homography computation with a reliability measure," *IEICE Transactions on Information and Systems*, vol. E83-D, no. 7, pp. 1369–1374, 2000.

[13] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 125–132, Kauai, Hawaii, USA, December 2001.

[14] F. Devernay and O. Faugeras, “Straight lines have to be straight,” *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, 2001.

[15] S. B. Kang, “Semiautomatic methods for recovering radial distortion parameters from a single image,” Tech. Rep. CRL 97/3, Digital Equipment Corporation, Cambridge Research Lab, Cambridge Mass, USA, May 1997.

[16] R. Swaminathan and S. K. Nayar, “Nonmetric calibration of wide-angle lenses and polycameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1172–1178, 2000.

[17] G. P. Stein, “Lens distortion calibration using point correspondences,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’97)*, p. 602, San Juan, Puerto Rico, USA, June 1997.

[18] Z. Zang, “On the epipolar geometry between two images with lens distortion,” in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 1, pp. 407–411, Vienna, Austria, August 1996.

[19] R. Swaminathan, M. D. Grossberg, and S. K. Nayar, “A perspective on distortions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’03)*, vol. 2, pp. 594–601, Madison, Wis, USA, June 2003.

[20] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[21] M. Taj, E. Maggio, and A. Cavallaro, “Multi-feature graph-based object tracking,” in *Proceedings of the 1st International Evaluation Workshop on Classification of Events, Activities and Relationships (CLEAR ’06)*, vol. 4122 of *Lecture Notes in Computer Science*, pp. 190–199, Southampton, UK, April 2006.

[22] F. L. Bookstein, “Fitting conic sections to scattered data,” *Computer Graphics and Image Processing*, vol. 9, no. 1, pp. 56–71, 1979.

[23] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, New York, NY, USA, 1996.

[24] R. Szeliski, “Image alignment and stitching: a tutorial,” *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[25] G. Kayumbi and A. Cavallaro, “Robust homography-based trajectory transformation for multi-camera scene analysis,” in *Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC ’07)*, pp. 59–66, Vienna, Austria, September 2007.

[26] G.-Q. Wei and S. D. Ma, “Implicit and explicit camera calibration: theory and experiments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 469–480, 1994.

[27] O. Faugeras, Q.-T. Luong, and T. Papadopoulou, *The Geometry of Multiple Images: The Laws that Govern the Formation of Images of a Scene and Some of Their Applications*, MIT Press, Cambridge, Mass, USA, 2001.

[28] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, “Seamless image stitching in the gradient domain,” in *Proceedings of the 8th European Conference on Computer Vision (ECCV ’04)*, vol. 3024 of *Lecture Notes in Computer Science*, pp. 377–389, Prague, Czech Republic, May 2004.

[29] S. Calderara, A. Prati, and R. Cucchiara, “HECOL: homography and epipolar-based consistent labeling for outdoor park surveillance,” *Computer Vision and Image Understanding*, vol. 111, no. 1, pp. 21–42, 2008.