Single camera calibration for trajectory-based behavior analysis

N. Anjum and A. Cavallaro [*]

Multimedia and Vision Group
Queen Mary, University of London
Mile End Road, E1 4NS London (UK)

Abstract

Perspective deformations on the image plane make the analysis of object behaviors difficult in surveillance video. In this paper, we improve the results of trajectory-based scene analysis by using single camera calibration for perspective rectification. First, the ground-plane view is estimated from perspective images captured from a single camera. Next, unsupervised fuzzy clustering is applied on the transformed trajectories to group similar behaviors and to isolate outliers. We evaluate the proposed approach on real outdoor surveillance scenarios with standard datasets and show that perspective rectification improves the accuracy of the trajectory clustering results.

1. Introduction

Camera calibration plays a fundamental role in single and multi-camera surveillance, and when cameras are mounted on unmanned aerial vehicles. In this case, the camera may pitch, roll or rotate thus generating video footage with geometrical distortions. Calibration in such scenarios is highly desirable for consistent and uniform display to survey the monitored scene. Basic calibration techniques require only 2-D point matches in multiple views and work for known cameras parameters or 3-D knowledge of the scene [2, 3, 4]. However, in case of a single camera or a multi-camera network with non-overlapping views, these approaches cannot be used for camera calibration.

Camera calibration is particularly important for analyzing trajectories generated by moving objects, such as vehicles, people, faces, or other body parts (Fig 1). The spatio-temporal information encapsulated in the trajectories provides high-level cues about objects behavior and object interactions. Trajectory analysis also helps determining the lane geometry and type in traffic surveillance and is used for calibration in non-overlapping multi-camera networks [1]. Assuming a calibrated monocular camera that allows the correspondence between the image plane and the top-view reconstruction of the scene where pedestrians can be detected using accumulated trajectory information [5]. In trajectory analysis, geometric effects (perspective) cause deformations in the trajectory shape that degrade the overall performance of the analysis algorithm.

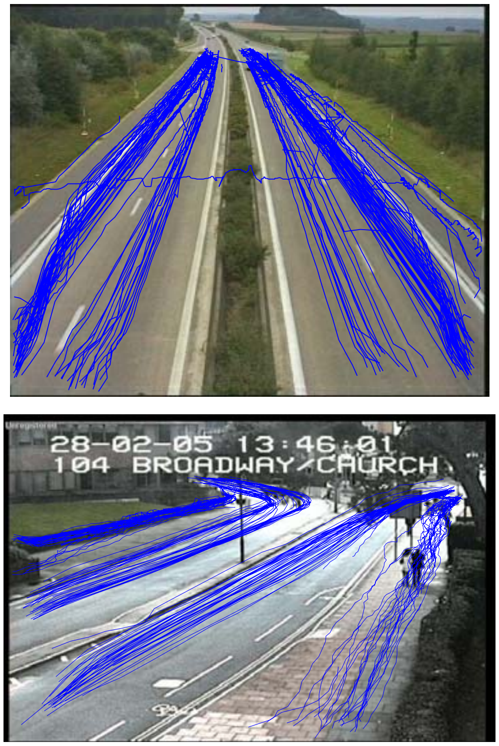


Figure 1: Examples of accumulated object trajectories on the image plane from surveillance videos

In this paper, we improve scene analysis by trajectory clustering using single camera calibration based on perspective rectification. We map the trajectories from the image plane to the ground plane thus reducing the projection effects. Next, we use an unsupervised fuzzy clustering based on Mean-shift to cluster the trajectories.

The remaining sections of this paper are organized as fol-

[*] The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1

lows: Section 2 discusses the perspective view rectification method. Section 3 covers related work in the field of trajectory modeling and clustering and the clustering technique. The experimental results are discussed in Sec. 4. Finally, in Sec. 5 we draw the conclusions.

2 Ground-plane calibration

Instead of assuming a monitored scene as a 3D Euclidean space containing a complete metric structure, we can consider it as being embedded in an affine or even projective space [6]. Under this assumption, Liebowitz describes the geometry, constraints and algorithmic implementation for metric rectification of planes [7]. Let $\aleph_r$ and $\aleph_i$ represent the real and the image plane, respectively. The mapping between the two planes is a general planar homography on the form $\aleph_i$ = H $\aleph_r$, with H a 3×3 matrix of rank 3. This projective transformation can be visualized as a chain of transformations on the form

$$H = \chi\varsigma\rho, \tag{1}$$

where χ represents the similarity transformation, ς represents the affine transformation, and ρ represents the pure projective transformation. The first step is to determine the transformation ρ defiend as

$$\rho = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix}, \tag{2}$$

where $l_\infty = (l_1, l_2, l_3)^T$ is the vanishing line of the plane. Parallel lines on the real-plane intersect at vanishing points in the image plane on the vanishing line. In this work, one vanishing point is calculated, as we assume that the vanishing line is perpendicular to the optical axis of the camera, passing through the vanishing point. A set of real plane parallel lines are identified by selected four points $\{P1, \ldots, P4\}$ on the image plane. The lines are projected to find the intersection or vanishing point for the lines. An illustration of vanishing line construction is given in Fig. 2. Once ρ is determined, the image can be affine-rectified and affine properties can be measured.

To remove affine projection, the transformation ς can be represented with two degree of freedoms:

$$\varsigma = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

where α and β represent the image as of the circular points in the complex domain. The assumption is useful for invariant representation of the image to Euclidean transformation. Liebowitz [7] presented various procedures to solve for the values of the two parameters. To generate constraint circles from known angles and length ratios in the image, we select four points $\{S1, \ldots, S4\}$ (square structure on the real plane) to represent line segments with known length ratio and three points $\{R1, \ldots, R3\}$ to represent right angle formed in real scene. The mentioned two affine invariant properties on the ground plane are sufficient to obtain the values of α and β. Fig. 2 also shows the selected points for square and right-angled structures.

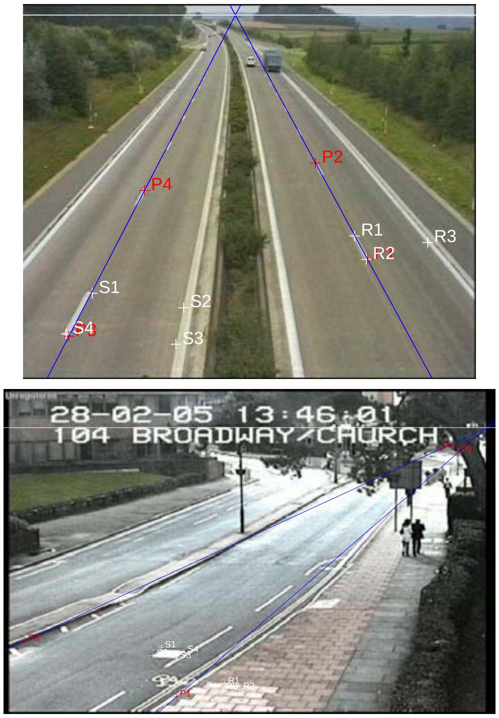


Figure 2: Example of vanishing line construction. The blue lines represent pairs of parallel lines intersecting on the vanishing points. The white line represents the vanishing line. $\{S1, \ldots, S4\}$ and $\{R1, \ldots, R3\}$ are used for known angles and length ratio selection

The circle parameters with known angles are calculated as

$$\begin{cases} c = \left(\frac{a+b}{2}, \quad \frac{a-b}{2} cot\theta \right) \\ r = \left| \frac{a-b}{2sin(\theta)} \right| \end{cases}, \tag{4}$$

where *c* is the 2D coordinate of the center of the constraint circle, *r* is the radius of the constraint circle, *a* and *b* are the directions of lines. In this work, $\theta = \frac{\pi}{2}$ to make circle center on α-axis only.

To calculate circle parameters with known length ratio, let d_{lx} and d_{ly} represent the horizontal and vertical directions of a line l and let s be the known length ratio. Then,

$$\begin{cases} c = \left(\frac{d_{1x}d_{1y} - s^2 d_{2x}d_{2y}}{d_{1y}^2 - s^2 d_{2y}^2}, \quad 0 \right) \\ r = \left| \frac{s(d_{2x}d_{1y} - d_{1x}d_{2y})}{d_{1y}^2 - s^2 d_{2y}^2} \right| \end{cases}, \tag{5}$$

The final values of α and β are calculated by finding a point of intersection of both constraint circles. The affine removal makes it possible to get the metric properties of the plane. The final matrix in the decomposition is a similarity transformation,

$$\chi = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}, \tag{6}$$

where R and t are rotation matrix and translation vector, respectively. Since the similarity transformation changes the coordinates linearly and does not play any role in the perspective view, in this work we have not removed this transformation from the image. The steps for the perspective rectification procedure can be summarized as follows:

- Select $\{P1, \ldots, P4\}$ to define the vanishing line; $\{S1, \ldots, S4\}$ and $\{R1, \ldots, R3\}$ that define known angle and length ratios within a real-world structure. The structures may exist at different heights in the scene.
- Calculate the inverse projection transformation (Eq. 2).
- Rectify the pure projection from the image by applying ρ.
- Calculate the inverse affine transformation (Eq. 4 and Eq. 5).
- Rectify the affine transformation from the image by applying ς.

Sample rectified images transformed using this procedure are shown in Fig. 3.

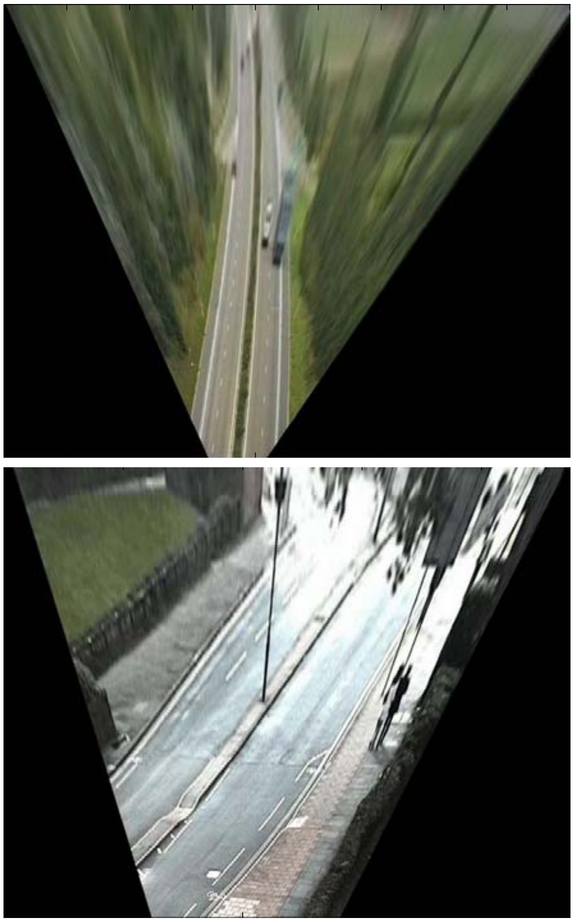


Figure 3: Sample rectified images using single camera calibration

3 Trajectory Clustering

3.1 Related work

*Hidden Markov Model*s (HMM) [13] and their variants like parameterized-HMMs [14] and coupled-HMMs [15] parameter space representations have been extensively applied to activity recognition based on trajectories. These approaches are robust against dynamic time warping of trajectory data, but the structures and probability distributions are highly domain dependent. Moreover, for complex events the size of the parameter space may grow exponentially. Statistical model-based approach for motion trajectory representation [10] are used when each trajectory has different statistical properties for corresponding motion classes. However, they are sensitive to the initial choice of model parameters that may lead to poor clustering results. Unsupervised techniques based on *Self-Organising Maps* (SOMs) [16] attempt to learn behavior patterns from sample trajectories. For real motion sequences, convergence of these techniques is slow and the learning phase is usually carried out offline due to the high dimensionality of the input data space. Recently, *Principal Component Analysis* (PCA) [9], *Independent Component Analysis* (ICA) [8] and *Discrete Fourier Transform* (DFT) coefficients [16] have been used to reduce the dimensionality of the input data to enhance performance of video indexing and retrieval systems. *Trajectory Directional Histograms* (TDH) [17] is another way to represent the statistical directional distribution, and to complement the information from resampled trajectories for vehicle motion trajectory clustering.

After transforming the trajectories into an appropriate feature space, the next step is to organize the data into clusters based on some homogeneity criteria (distance measure). The most common *homogeneity* criteria are conventional (normalized) distance measures. Mean, Maximum, Minimum, modified Hausdorff-type distances ([12]) and *Longest Common Subsequences* (LCSS) ([11]) are also popular similarity measures for trajectory clustering.

3.2 Fuzzy Mean-Shift Clustering

We use the approach presented in Sec. 2 to map the motion trajectories measured on the image plane into real-plane coordinates to reduce perspective deformations. After recovering the ground plane (by finding the projective and affine inverse transformations) the motion trajectories of the objects are reprojected to their ground plane coordinates. Let a trajectory T_j be represented as

$$T_j = \{(x_j^i, y_j^i); i = 1, \ldots, N_j\}, \qquad (7)$$

where (x_j^i, y_j^i) is the estimated position of the target in the ground plane and N_i is the number of trajectory points. Each trajectory needs to be transformed in appropriate feature spaces before clustering. Let $F_m(.)$ be a transformation functions defined as $F_m(T) \rightarrow \Psi_m$, with $m = 1, ..., M$. The transformation $F_m(.)$ maps each trajectory to a *d-dimensional* feature space, Ψ_j, with $j = 1, ..., J$. We use Ψ_1, the space spanned by the first two components of the trajectory data obtained through *PCA*, and Ψ_2, the space spanned by the *average velocity vector* of each trajectory. After transforming the trajectories into the feature spaces, we analyze the trajectory data using Mean-shift in each space to seek the local modes and generate the clusters.

Let $\chi_l \in \Psi_j$; $l = 1, ..., L$ be a set of L data points. The multivariate density estimator $\hat{f}(x)$ is defined as

$$\hat{f}(x) = \frac{1}{Lh^d} \sum_{l=1}^{L} K\left(\frac{x - \chi_l}{h}\right), \qquad (8)$$

where *h* is the bandwidth and *K(.)* is a kernel, defined as

$$K(x) = \left\{ \begin{array}{ll} \frac{1}{2V_d}(d+2)(1 - x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{array} \right. , \qquad (9)$$

with V_d representing the volume of a *d-dimensional* sphere. The density gradient estimate of the kernel can be written as

$$\hat{\nabla} f(x) = \nabla \hat{f}(x) = \frac{1}{Lh^d} \sum_{l=1}^{L} \nabla K\left(\frac{x - \chi_l}{h}\right). \qquad (10)$$

Equation (10) can be re-written as

$$\hat{\nabla} f(x) = \frac{d+2}{h^d V_d} \left(\frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x) \right), \qquad (11)$$

where *S(x)* is a hypersphere of radius h, with volume $h^d V_d$, centered in x and containing L data points. The Mean-shift vector $\zeta_h(x)$ is defined as

$$\zeta_h(x) = \frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x), \qquad (12)$$

and, using Eq. (11), we can express $\zeta_h(x)$ as

$$\zeta_h(x) = \frac{h^d V_d}{d+2} \frac{\hat{\nabla} f(x)}{\hat{f}(x)}. \qquad (13)$$

The output of the Mean-shift procedure is the set of data points associated to each mode.

To refine the clustering results, we apply a Cluster Merging (CM) procedure that fuses two adjacent clusters if the density modes are sufficiently close. The *proximity condition* is defined by the 10% of the kernel bandwidth h.

As a result of this procedure, each trajectory may have a different degree of belongingness to more than one cluster, in multiple feature spaces. To obtain the final clustering, each trajectory is assigned to a particular cluster if its belonginess is consistent across all feature spaces.

Let ξ_k and ξ_{k+1} be the number of clusters in Ψ_k and Ψ_{k+1}, with $\xi_k \leq \xi_{k+1}$ (note that different feature spaces may generate different numbers of clusters). Also, let $C_i^k \in \Psi_k$ and $C_j^{k+1} \in \Psi_{k+1}$ be clusters in the respective feature spaces. The next step is to find the correspondence among the clusters found in feature spaces. Let $\hat{\nu}$ be the index of the cluster in Ψ_{k+1} that has the maximum correspondence with the i^{th} cluster of Ψ_k, i.e.:

$$\hat{\nu} = argmax(C_i^k \cap C_j^{k+1}), \qquad (14)$$

with $j = 1, ..., \xi_{k+1}$. Let the cluster $B_i = C_i^k \cap C_{\hat{\nu}}^{k+1}$ contain the overlapping elements in $C_{\hat{\nu}}^{k+1}$ and C_i^k. If $\{\ \Delta\ \}_i^q$, with $q = 1, 2$, represents non-overlapping elements, then

$$\Delta_i^1 = C_i^k - (C_i^k \cap C_{\hat{\nu}}^{k+1}) \qquad (15)$$

and

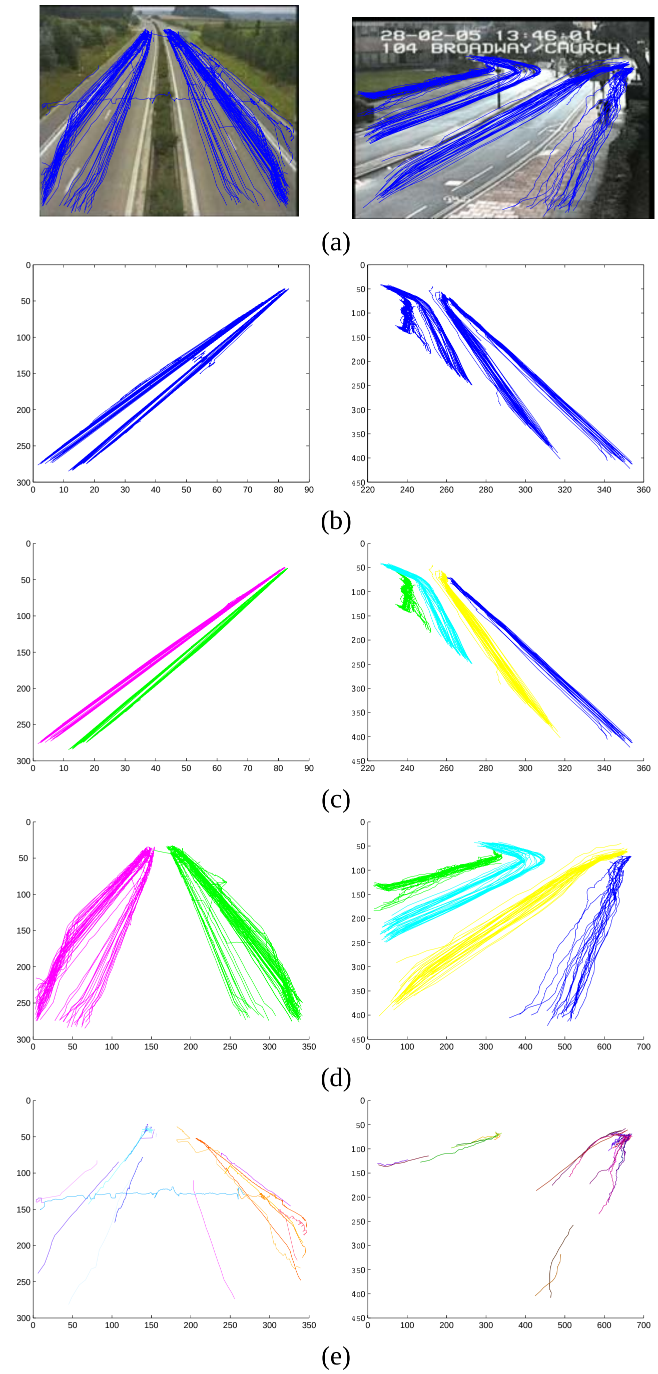
$$\Delta_i^2 = C_{\hat{\nu}}^{k+1} - (C_i^k \cap C_{\hat{\nu}}^{k+1}), \qquad (16)$$

with Δ_i^1 and Δ_i^2 forming new independent clusters. Finally, CM is applied again to merge adjacent clusters based on the *proximity condition* and the modes associated to too few data points are considered outliers. The outlier condition is set as the 5% of the maximum peak in the dataset.

Trajectories far from all clusters' center or belonging to small clusters are considered as generated by outlier object behaviors and then removed.

4. Experimental results

We demonstrate the trajectory clustering results on two real outdoor traffic scenes for both image and ground plane view


(a)

(b)

(c)

(d)

(e)

Figure 4: Trajectory clustering and outlier detection on $SEQ1$ (left) and on $SEQ2$ (right). (a) Original trajectories on the image plane; (b) rectified trajectories; (c) clustered trajectories on the ground plane; (d) clustering results backprojected on the image plane; (e) outliers backprojected on the image plane

of the trajectories. The following test sequences are used: $SEQ1$, a highway surveillance sequence from the *MPEG-7* dataset with 134 object trajectories, and $SEQ2$, a traffic monitoring sequence from the *VACE* evaluation dataset with 159 trajectories [18]. Both sequences are captured at $25Hz$. The trajectories are the ground-truth trajectories generated by the center points of the bounding box of each object.

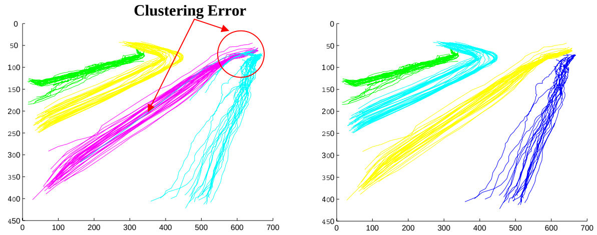


Figure 5: Comparison of clustering results. (left) Clusters obtained on the image plane without rectification; (right) Clusters obtained on the ground plane and reprojected on the image plane

We compare the trajectory clustering results obtained directly on the image plane using the method presented in Sec. 3.2 with the results obtained with the same method on the ground plane and reprojected back to the image plane for visualization and comparison. Fig. 4 shows the transformed trajectories along with the clustering results. The corresponding image plane view of the results is also shown with the detected outliers. It is possible to notice that in both dataset outliers are identified that correspond either to objects moving abnormally (e.g., an object crossing the highway and an object moving backwards in $SEQ1$) or to objects that have not completed their trajectory due to the limited length of the test sequence. In both cases they represent unusual behaviors compared to the majority of other objects.

Fig. 5 compares the clustering results obtained on the image plane without rectification with the results obtained after transformation on the ground plane and subsequent reprojection on the image plane. This second approach succeeded in identifying the correct cluster memberships for the trajectories and to overcome a few misclassification generated when the clustering was performed on the image plane. The ground plane view also helped in identifying outlier trajectories especially at the boundaries of the image plane where they are strongly affected by the perspective projection.

5. Conclusions

We have proposed a two-step approach for object trajectories analysis to compensate for the distortions introduced by the perspective view in surveillance video. In first step, the trajectories are mapped on the ground plane in order to rectify the perspective view on image plane. The rectified trajectories are then analyzed by an unsupervised fuzzy clustering algorithm based on Mean-Shift. The procedure is validated on real outdoor traffic scenarios from standard test sequences. Experiments indicate that the proposed method increases the accuracy of the clustering results. Our current work is focused on utilizing the ground plane object's trajectories information for non-overlapping multi-sensor calibration.

References

[1] N. Anjum, M. Taj, A. Cavallaro, *Relative position estimation of non-overlapping cameras*, In Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu (USA), April 2007.

[2] S. J. Maybank and O. D. Faugeras, *A theory of self-calibration of moving camera*, Intl. Journal of Computer Vision, Volume 8, Issue 2, pp 123 - 151, August 1992.

[3] A. Basu, *Active calibration: alternative strategy and analysis*, In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, New York (USA), June 1993.

[4] Q. Ji and S. Dai, *Self-calibration of a rotating camera with a translational offset*, IEEE Trans. on Robotics and Automation, Volume 20, Issue 1, pp 1- 14, February 2004.

[5] G. Antonini and M. Bierlaire, *A Discrete choice framework for acceleration and direction change behaviors in walking pedestrians*, SPIE, Lecture Notes in Computer Science, pp 145 - 156, February 2005.

[6] O. Faugeras, *Stratification of 3-dimensional vision: Projective, Affine, and Metric representations*, JOSA-A, Volume 12, Number 3, pp 465 - 484, March 1995.

[7] D. Liebowitz and A. Zisserman, *Metric rectification for perspective images of planes,* In Proc. of the Conf. on Computer Vision and Pattern Recognition, Santa Barbara (CA USA), June 1998.

[8] G. Antonini and J. P. Thiran, *Counting pedestrians in video sequences using trajectory clustering*, IEEE Trans. on Circuits and Systems for Video Technology, Volume 16, Issue 8, pp 1008 - 1020, August 2006.

[9] F. I. Bashir, A. A. Khokhur and D. Schonfeld, *Segmented trajectory based indexing and retrieval of video data*, In Proc. of Intl. Conf. on Image Processing, Chicago (IL USA), September 2003.

[10] S. Gaffney and P. Smyth, *Trajectory clustering with mixtures of regression models*, In Proc. of Intl. Conf. on Knowledge Discovery and Data Mining, California (USA), 1999.

[11] H. Fashandi and A. M. E. Moghaddam, *A new invariant similarity measure for trajectories*, In Proc. of IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation, Espoo (Finland), June 2005.

[12] J. Melo, A. Naftel, A. Bernardino and J. S. Victor, *Retrieval of vehicle trajectories and estimation of lane geometry using non-Stationary Traffic Surveillance Cameras*, Advanced Concepts for Intelligent Vision Systems, Brussels (Belgium), August 2004.

[13] F. Porikli, *Trajectory pattern detection by HMM parameter space features and eigenvector clustering*, Mitsubishi Electric Research Laboratories, Technical Report TR2004-032, January 2004.

[14] A. D. Wilson and A. F. Bobick, *Recognition and interpretation of parametric gesture*, In Proc. of IEEE Intl. Conf. on Computer Vision, Bombay (India), January 1998.

[15] N. M. Oliver, B. Rosario and A. P. Pentland, *A Bayesian computer vision system for modeling human interactions*, IEEE Trans. on Pattern analysis and machine intelligence, Volume 22, Issue 8, pp 831-843, August 2000.

[16] A. Naftel and S. Khalid, *Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space*, In Trans. of Multimedia Systems, Volume 12, Number 3, pp 45 - 52, September 2006.

[17] X. Li, W. Hu and W. Hu, *A coarse-to-fine strategy for vehicle motion trajectory clustering*, In Proc. of Intl. Conf. on Pattern Recognition, August 2006.

[18] R. Kasturi, *Performance evaluation protocol for face, person and vehicle detection tracking in video analysis and content extraction, VACE-II*, Computer Science & Engineering University of South Florida (USA), Tampa, January 2006.