Motion Trajectory Clustering for Video Retrieval using Spatio-Temporal Approximations

Shehzad Khalid and Andrew Naftel

School of Informatics, University of Manchester, PO Box 88, Sackville Street, Manchester M60 1QD, United Kingdom
{s.khalid-2@postgrad.manchester.ac.uk, a.naftel@manchester.ac.uk}

Abstract. A new technique is proposed for clustering and similarity retrieval of video motion clips based on spatio-temporal object trajectories. The trajectories are treated as motion time series and represented either by least squares or Chebyshev polynomial approximations. Trajectory clustering is then carried out to discover patterns of similar object motion behaviour. The coefficients of the basis functions are used as input feature vectors to a Self-Organising Map which can learn similarities between object trajectories in an unsupervised manner. Encoding trajectories in this manner leads to efficiency gains over existing approaches that use point-based flow vectors to represent the whole trajectory as input vector. Experiments on two different motion datasets – vehicle tracking and pedestrian surveillance - demonstrate the effectiveness of our approach. Applications to motion data mining in video surveillance databases are envisaged.

1 Introduction

An increasing number of systems are now able to capture and store data about object motion such as those of humans and vehicles. This has acted as a spur to the development of sophisticated content-based visual data management techniques. General purpose tools are now urgently required for motion data search and retrieval, discovery and grouping of similar motion patterns, detection of anomalous behaviour, motion understanding and prediction.

Much of the recent research focus has been on representation schemes for motion indexing and retrieval [1]-[11]. This work presupposes the existence of some low-level tracking scheme for reliably extracting object-based motion trajectories. A description of relevant tracking algorithms is not within the scope of this paper but recent surveys can be found in [12], [13]. The literature on trajectory-based motion understanding and pattern discovery is less mature but advances using learning Vector Quantization (LVQ) [14], Self-Organising Feature Maps (SOMs) [15], [16], hidden Markov Models (HMMs) [17], and fuzzy neural networks [18] have been reported. Most of these techniques attempt to learn high-level motion behaviour patterns from sample trajectories using point-based flow vectors as input to the learning phase. In this paper, we show how to circumvent this requirement by proposing a trajectory classification approach based on a simple spatio-temporal representation scheme used for motion indexing and retrieval.

Related work within the temporal database community on approximation schemes for indexing time series data is highly relevant to the parameterisation of object trajectories. However, computer vision researchers have been slow to adopt this work. For example, spatiotemporal trajectories have been successfully modelled using discrete Fourier transforms (DFT) [19], wavelet transforms (DWT) [20], adaptive piecewise constant approximation (APCA) [21], and Chebyshev polynomials [22], to name but a few.

In this paper, we aim to apply time series indexing of spatiotemporal trajectories to the problem of trajectory classification and show how to learn motion patterns by using the indexing scheme as an input feature vector to a neural network learning algorithm. The remainder of the paper is organized as follow. We review some relevant background material in section 2. In section 3, we present our trajectory modelling approach. The algorithm for learning trajectories is then presented in section 4 within the framework of a self-organising map. This is applied in the context of clustering motion trajectories and experimental results for both vehicle and pedestrian object tracking databases are reported in section 5. The paper concludes with a discussion of the advantages of our proposed technique over competing approaches and outlines further work.

2 Review of Previous Work

Motion trajectory descriptors are known to be useful candidates for video indexing and retrieval schemes. Previous work has sought to represent moving object trajectories through piecewise linear or quadratic interpolation functions [1], [2], motion histograms [4] or discretised direction-based schemes [3], [8], [9]. Flexible spatiotemporal representations using piecewise polynomials were proposed by Hsu [6], although consistency in applying trajectory-splitting across query and searched trajectories can be problematic. Affine and more general spatiotemporal invariant schemes for trajectory retrieval have also been presented [5], [7], [10]. The importance of selecting the most appropriate trajectory model and similarity search metric has received relatively scant attention [11].

In addition to polynomial models, a wide variety of basis functions have been used to approximate object trajectories [19]-[22]. Efficient indexing schemes can then be constructed in the coefficient space of the basis functions. These have been compared with respect to search pruning power, CPU and I/O efficiency costs [21], [22].

It is surprising to find that many of these candidate spatiotemporal trajectory indexing schemes have not yet been applied to the problem of motion data mining and trajectory classification. Recent work has either used probabilistic models such as HMMs [17] or point-based trajectory flow vectors [14], [16], [18] as a means of learning patterns of motion activity. Flow vectors consist of spatial coordinates augmented by instantaneous object velocities and optionally accelerations. These can be normalised to account for variation in trajectory lengths.

The contribution of this paper is to show that a trajectory-encoding scheme based on input feature vectors consisting of basis function approximation coefficients can be used to learn motion patterns. Hence, clustering and classification processes can be carried out effectively in the coefficient space.

3 Trajectory Representation

The output of a motion tracking algorithm is usually a set of (noisy) 2-D points (x_i, y_i) representing the object's motion path over a sequence of n frames, where $i = 1,\dots,n$. In this case, the representative point is taken to be the centroid of the object's minimum bounding rectangle. The motion trajectory can be considered as two separate 1-dimensional time series, $<t_i, x_i>$ and $<t_i, y_i>$, the horizontal and vertical displacement against time where $t_1 < \dots < t_n$. We consider two alternative trajectory models, Least Squares (LS) and Chebyshev polynomial approximations. LS polynomials are suitable for modelling simple motion trails, e.g. vehicles moving smoothly along highways, whilst Chebyshev approximations are more appropriate for modelling complex spatiotemporal trajectories such as pedestrian tracking exhibiting stop-start and looping motions.

3.1 Least Squares Polynomials

The trajectory can be approximated by a polynomial $P_m(t)$ of degree $m < n$ as

$$[x \mid y] \approx P_m(t) = a_0 + a_1 t + \dots + a_m t^m \tag{1}$$

The x, y displacements are modelled as independent polynomials P_m^x, P_m^y in t. Note that separate 1-D trajectories are created for each spatial coordinate [x | y]. The unknown 2(m+1) coefficients {a_{xi}, a_{yi}}, i = 0,...,m can be determined using LS by minimising the function E with respect to a_0, a_1, ...

$$E(a_0, a_1, \dots, a_m) = \sum_{i=1}^{n} \{[x_i \mid y_i] - (a_0 + a_1 t + \dots + a_m t_i{}^m)\}^2 \tag{2}$$

The motion trajectories are thus indexed by vector of coefficients $\{(a_{x0},\dots,a_{xm}),(a_{y0},\dots,a_{ym})\}$.

3.2 Chebyshev Polynomials

Alternatively, a spatiotemporal trajectory can be approximated by a function $f(t)$ expressed as a weighted sum of Chebyshev polynomials $C_k(t)$ up to degree m, defined as

$$[x \mid y] = f(t) \approx \sum_{k=0}^{m} b_k C_k \tag{3}$$

where $C_k(t) = \cos(k \cos^{-1}(t))$ and

$$b_0 = \frac{1}{m}\sum_{k=1}^{m} f(t_k)\,,\ b_i = \frac{2}{m}\sum_{k=1}^{m} f(t_k)C_i(t_k) \tag{4}$$

for $t \in [-1,1]$ and $i = 1,..., m$. The k roots of $C_k(t)$ are given by t_j for $1 \le j \le k$. Implementation details can be found in [23].

Occasionally it is necessary to approximate the motion trail (spatial trajectory shape) in the xy plane. In this case, we replace t by x or y in one of the above equations depending on the choice of principal axis [6]. This would only be worthwhile if all trajectories can be aligned with the same principal axis.

3.3 Similarity Search Metric

A Euclidean distance is used as the basis for comparing the similarity of motion trajectories. Each polynomial produces a vector of coefficients which can be used to index a 2-dimensional spatiotemporal trajectory. Given two trajectories Q and S, we can index these by a vector of $2(m+1)$ coefficients $Q = \left\{\overrightarrow{q_0},....,\overrightarrow{q_m}\right\}$ and $S = \left\{\overrightarrow{s_0},....,\overrightarrow{s_m}\right\}$, where $\overrightarrow{q_i}$, $\overrightarrow{s_i}$ are $\vec{q_i} = [q_{xi}, q_{yi}]^T$ and $\vec{s_i} = [s_{xi}, s_{yi}]^T$ ($i = 0,..., m$).

A Euclidean distance function (ED) on the coefficient space can be expressed as

$$ED(Q,S) = \sqrt{\sum_{i=0}^{m} \left(\vec{q}_i - \vec{s}_i\right)^2} = \sqrt{\sum_{i=0}^{m} \left(q_{xi} - s_{xi}\right)^2 + \left(q_{yi} - s_{yi}\right)^2} \tag{5}$$

4 Learning Trajectory Patterns Using Self-Organizing Maps

Self-organised maps (SOMs) have been previously used for motion trajectory classification [15], [16] with trajectories encoded as flow vectors. This step can be replaced by the proposed coefficient indexing scheme. A SOM discovers the underlying structure of motion trajectory data through unsupervised learning.

4.1 Network Model

The architecture chosen for the SOM is very simple with a layer of input neurons connected directly to a single 1-dimensional output layer. Each input neuron is connected to every output neuron with the connection represented by a weight vector. A similar architecture was used in [16] for learning vehicle trajectories as a means for accident prediction.

In a SOM network, physically adjacent output nodes encode the patterns in the trajectory data that are similar and, hence, it is known as a topology-preserving map. Consequently, similar object trajectories are mapped to the same output neuron. The number of input neurons is determined by the size of the feature vector which in this case relates to the selected number of coefficients for the basis functions. The degree of the polynomial can be chosen by setting a threshold on the maximum deviation of the approximation from the data or mean-squared error. The number of output neurons represents the number of groups in the trajectory data and this is selected manually.

4.2 Learning Algorithm

The algorithm used to cluster the trajectories differs slightly from the original SOM proposed by Kohonen [24]. The number of output neurons is initially set to a higher value than the desired number of clusters which we wish to produce. After training the network, clusters representing the most similar patterns are merged until the cluster count is reduced to the required number. The weights are initialised to linearly spaced values lying within the range of input values. Neighbourhood size is initially set to cover over half the diameter of the output neurons.

Let B be the input feature vector representing the set of trajectory basis function coefficients, and W the weight vector associated to each output neuron. The learning algorithm comprises the following steps:

1. Determine the winning output node k (indexed by c) such that the Euclidean distance between the current input vector B and the weight vector W_k is a minimum amongst all output neurons, given by the condition

$$\| B - W_c(t) \| \leq \| B - W_k(t) \| \qquad \forall k \tag{6}$$

2. Train the network by updating the weights. A subset of the weights constituting a neighbourhood centred around node c are updated using

$$W_k(t+1) = W_k(t) + \alpha(t)\eta(k,c)(B - W_k(t)) \tag{7}$$

   where $\eta(k, c) = \exp(-|r_k - r_c|^2 / 2\sigma^2)$ is a neighbourhood function that has value 1 when $k=c$ and falls off with distance $|r_k - r_c|$ between nodes k and c in the output layer, σ is a width parameter that is gradually decreased over time and t is the training cycle index.

3. Decrease the learning rate $\alpha(t)$ linearly over time.

4. After a pre-determined number of training cycles, decrease the neighbourhood size.

5. At the end of the training phase, merge the most similar cluster pairs until the desired number of groupings is achieved. Clusters are merged by calculating the weighted mean of the weights associated with each neuron taking into account the number of input samples allocated to the cluster. Assuming W_a and W_b are the weight vectors associated with output neurons representing the most similar clusters, and m, n are the number of sample trajectories mapped to these neurons respectively, a new weight value W_{ab} for the merged cluster can be calculated as

$$W_{ab} = \frac{mW_a + nW_b}{m + n} \tag{8}$$

5 Experiments

We now present some results to indicate the effectiveness of the proposed trajectory clustering technique. The algorithm was first tested on a highway traffic surveillance sequence. The video sequence was recorded with a stationary camera having a resolution of 176 x 144 pixels and a video capture rate of 15 frames/second. The individual trajectories of 284 vehicles were extracted using the PTMS tracking algorithm [25] illustrated in Fig. 1(a). This dataset was chosen due to its simple underlying structure and the fact that cluster visualisation could be easily interpreted in terms of vehicle lane classification.

Given the uniformity of the vehicles' trajectory shape, it was decided to fit a least squares polynomial of degree 3, in the form $x = P(y)$ to the motion trail. The coefficient vectors $\{a_{i0},a_{i1},a_{i2},a_{i3}\}$ for each sample trajectory i were input to the SOM training network. Initially, the trajectories were grouped into 10 clusters (output neurons). Similar clusters were then merged in a hierarchical fashion until just 4 clusters remained. The results can be seen in Fig. 1(b)-(e). Notice that a number of trajectories represent vehicle lane changes and these are usually the most distant from the cluster centre. By setting a threshold on the distance between the final adjusted weight vector and each sample trajectory, anomalous trajectories can be detected as shown in Fig. 1(f). It should be noted that some anomalies result from tracking errors and vehicle occlusions.

Given that labelled ground truth was available, the effect on classification accuracy of partitioning the trajectory samples into training and test data was investigated. The training set was obtained by selecting samples at random from each labelled group. The classification errors are shown in Table 1 for different sized training/test data. The SOM achieves 100% classification accuracy for modest sized training sets demonstrating the robustness of using polynomial coefficients as inputs to the learning algorithm.

In the second example, we evaluate the performance of the trajectory clustering algorithm using the CAVIAR visual tracking database [26]. The database consisted of hand annotated video sequences of moving and stationary people and are intended to provide a testbed for benchmarking vision understanding algorithms. Semantic descriptions of the target object behaviours and motion have been generated and stored in XML files. These have been parsed these to extract ground truth-labelled object trajectories. The dataset contains 102 trajectories as shown in Fig. 2.

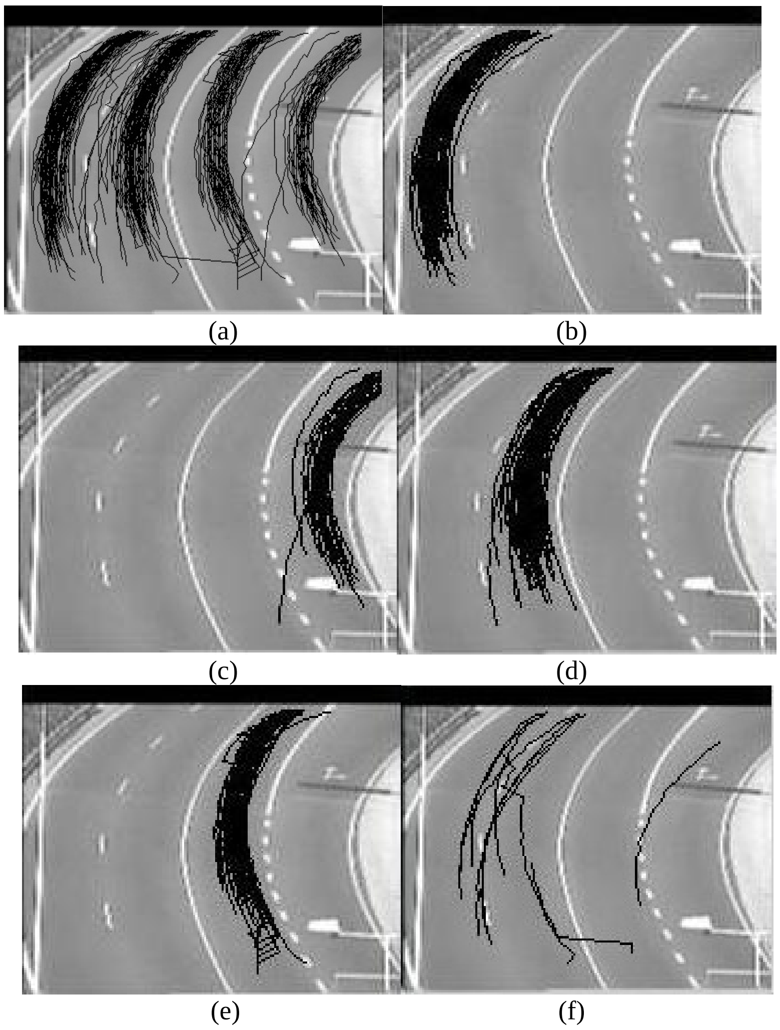

Fig. 1. Clustering of vehicle trajectories using SOM network. (a) Motion trajectories extracted using vehicle tracking. (b)-(e) Trajectories clustered into groups based on lane membership. (f) Anomalous trajectories obtained by setting threshold on distance from final weight vector.

Since this database contains more complex trajectories, these are modelled spatio-temporally using Chebyshev approximations. Temporal shift and scale invariance has been used to normalise the time series datasets[6]. Polynomials of degree 8 and above provide adequate model fidelity to the data generating input feature vectors with 18 coefficients (9 for each spatial coordinate). Orthogonality properties ensure that high order approximations ($m > 4$) do not yield oscillatory polynomials. We initially train a SOM network with 25 output neurons and then reduce these to 12 using the agglomerative clustering method described in section 4.2. The resulting trajectory groups are shown in Fig 3. Qualitatively similar motion trajectory patterns appear to have been grouped together quite successfully.

Table 1. Vehicle trajectory classification errors for SOM training based on highway lane membership. Sample trajectories have been partitioned into different sizes of training and test datasets.

Size of training set	Size of test set	Correct classification (%)
5	259	86.7
8	256	87.1
10	254	94.4
15 onwards	249	100.0

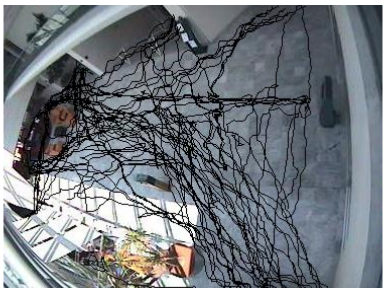

Fig. 2. Background scene containing database of ground truth labelled object trajectories.

6 Discussion and Conclusion

This paper presents a neural network learning algorithm for classifying spatiotemporal trajectories. Global features of motion trajectories are represented well by polynomial approximations and this is apparent in the cluster visualizations. Using coefficients of basis functions as input feature vectors to a neural network learning algorithm offers an efficient alternative to the use of flow vectors for trajectory classification.

A current disadvantage is the handling of partial trajectory matching which is unsuited to a polynomial-based representation. One possibility is to parameterise the trajectory length and augment the feature vector with additional entries. In future work we would like to extend this approach to the autonomous detection of anomalous trajectories and prediction of unusual motion behaviour.

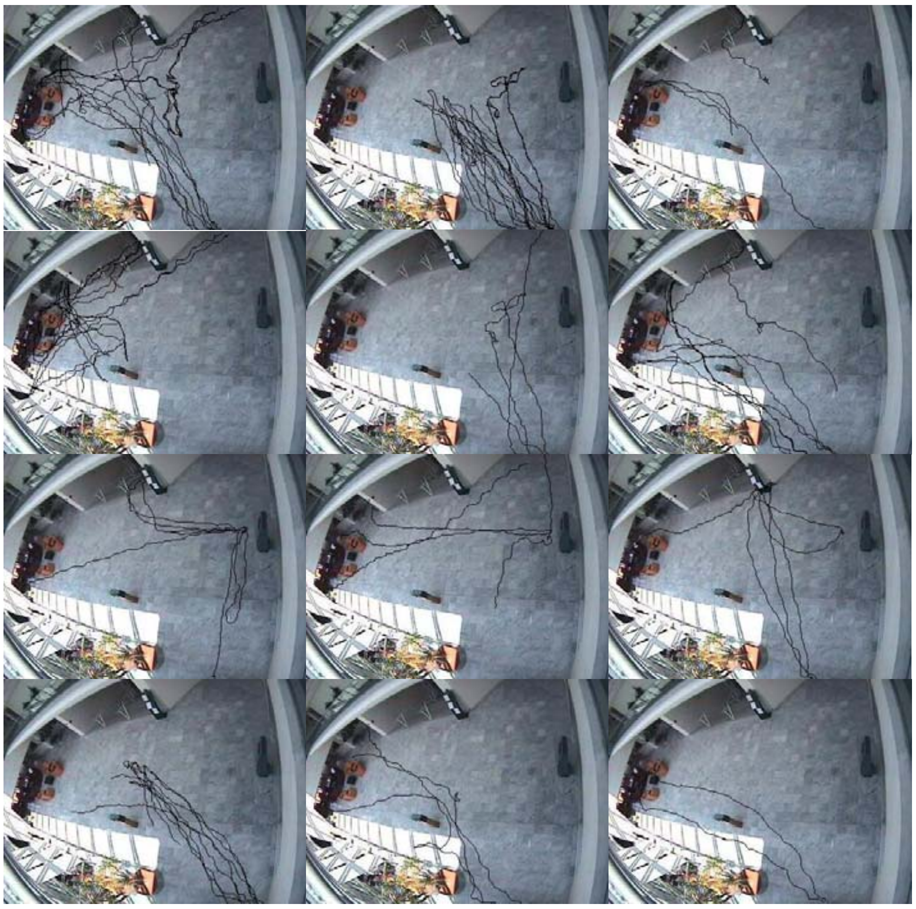

Fig. 3. Clustering of spatiotemporal object trajectories from CAVIAR database into 12 distinct groups using SOM network.

References

1. S-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 602-615, Sept. 1998.
2. S. Jeannin and A. Divakaran, "MPEG-7 visual motion descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 720-724, June 2001.
3. S. Dagtas, W. Ali-Khatib, A. Ghafor, and R.L. Kashyap, "Models for motion-based video indexing and retrieval," *IEEE Trans. Image Proc.*, vol. 9, no. 1, pp. 88-101, Jan 2000.
4. Z. Aghbari, K. Kaneko, and A. Makinouchi, "Content-trajectory approach for searching video databases," *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 516-531, Dec. 2003.
5. F. Bashir, A. Khokhar, and D. Schonfeld, "Segmented trajectory-based indexing and retrieval of video data," in *Proc. IEEE Int. Conf. Image Processing*, Spain, 2003, pp. 623-626.

6. C-T. Hsu and S-J. Teng, “Motion trajectory based video indexing and retrieval,” in *Proc. IEEE Int. Conf. Image Processing*, pt 1, 2002, pp. 605-608.
7. F. Bashir, A. Khokhar, and D. Schonfeld, “A hybrid system for affine-invariant trajectory retrieval,” in *Proc. MIR’04*, 2004, pp. 235-242.
8. C. Shim and J. Chang, “Content-based retrieval using trajectories of moving objects in video databases,” in *Proc. IEEE. 7th Int. Conf. Database Systems for Advanced Applications*, 2001, pp. 169-170.
9. C. Shim and J. Chang, “Trajectory-based video retrieval for multimedia information systems,” in *Proc. ADVIS*, LNCS 3261, 2004, pp. 372-382.
10. Y. Jin and F. Mokhtarian, “Efficient video retrieval by motion trajectory”, in *Proc. BMVC’04*, 2004.
11. S. Khalid and A. Naftel, “Evaluation of matching metrics for trajectory-based indexing and retrieval of video clips,” in *Proc. IEEE WACV*, Colorado, USA, Jan. 2004.
12. L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognition*, vol. 36, no. 3, pp. 585-601, 2003.
13. W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Trans. Systems, Man & Cybernetic*, Part C, vol.34, no.3, pp. 334-352, August 2004.
14. N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition,” *Image Vis. Comput.*, vol. 14, no. 8, pp. 609-615, 1996.
15. J. Owens and A. Hunter, “Application of the self-organising map to trajectory classification,” in *Proc. IEEE Int. Workshop Visual Surveillance*, pp. 77-83, 2000.
16. W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, “Traffic accident prediction using 3-D model-based vehicle tracking,” *IEEE Trans. Vehicular Tech*., vol. 53, no. 3, pp. 677-694, May 2004.
17. J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, “Discovering clusters in motion time-series data,” in *Proc. IEEE CVPR*, June 2004.
18. W. Hu, D. Xie, T. Tan, and S. Maybank, “Learning activity patterns using fuzzy self-organizing neural networks,” *IEEE Trans. Systems, Man & Cybernetic*, Pt. B, vol. 34, no. 3, pp. 1618-1626, June 2004.
19. C. Faloutsas, M. Ranganathan, and Y. Manolopoulos, “Fast subsequence matching in time-series databases,” in *Proc. ACM SIGMOD Conf.*, 1994, pp. 419-429.
20. K. Chan and A. Fu., “Efficient time series matching by wavelets,” in *Proc. Int. Conf. Data Engineering*, Sydney, March 1999, pp. 126-133.
21. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrota, “Locally adaptive dimensionality reduction for indexing large time series databases,” in *Proc. ACM SIGMOD Conf.*, 2001, pp. 151-162.
22. Y. Cui and R. Ng, “Indexing spatio-temporal trajectories with Chebyshev polynomials”, in *Proc. ACM SIGMOD Conf.*, Paris, June 2004, pp. 599-610.
23. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.* Cambridge, England: Cambridge University Press, 1992.
24. T. Kohonen, *Self-Organizing Maps,* 2nd ed. New York: Springer-Verlag, 1997, vol. 30.
25. J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, “Viewpoint independent detection of vehicle trajectories and lane geometry from uncalibrated traffic surveilllance cameras,” in *Proc. Int. Conf. Image Anal. Recog.* (*ICIAR* ’04), Porto, Portugal, 2004, pp 454-462.
26. CAVIAR: Context aware vision using image-based active recognition. (2004, Jan. 10). [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR