Learning Intrinsic Video Content using Levenshtein Distance in Graph Partitioning

Jeffrey Ng and Shaogang Gong

Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
{jeffng,sgg}@dcs.qmul.ac.uk

Abstract

We present a novel approach for automatically learning models of temporal trajectories extracted from video data. Instead of using a representation of linearly time-normalised vectors of fixed-length, our approach makes use of Dynamic Time Warp distance as a similarity measure to capture the underlying ordered structure of variable-length temporal data while removing the non-linear warping of the time scale. We reformulate the structure learning problem as an optimal graph-partitioning of the dataset to solely exploit Dynamic Time Warp similarity weights without the need for intermediate cluster centroid representations. We extend the graph partitioning method and in particular, the Normalised Cut model originally introduced for static image segmentation to unsupervised clustering of temporal trajectories with fully automated model order selection. By computing hierarchical average Dynamic Time Warp for each cluster, we learn warp-free trajectory models and recover the time warp profiles and structural variance in the data. We demonstrate the approach on modelling trajectories of continuous hand-gestures and moving objects in an indoor environment.

Key words: Automatic Model Order Selection, Dynamic Time Warping, Graph-Partitioning, Modelling Video Content, Normalised Cut, Levenshtein Distance, Trajectory Modelling,Unsupervised Clustering

1 Introduction

Recognition of temporal trajectories in video plays an important role in a host of interpretational tasks in computer vision, in particular human-computer interfaces and surveillance of moving objects. Learning effective and computationally viable models from complex sets of trajectories however highlights one of the fundamental problems of visual learning: unsupervised clustering of temporal structures with arbitrary model order. Johnson and Hogg [6] learned trajectory vertices in an unconnected state space and used a temporal pattern formation to reconstruct the ordered correlation using a mechanism similar to the leaky neuron memory. The model order was user-defined. Jebara and Pentland [5] automatically learned the correlation of action-reaction pairs of trajectories through

sliding windows. Entropy [2] and Minimum Description Length [16] based criteria have also been proposed for the automatic holistic discovery of intrinsic classes in the learning set. Both algorithms treat continuous temporal trajectories as fixed-length data vectors and do not address the effects of localised non-linear warping of the time scale, a key characteristic of the time-based generative processes which underpin temporal trajectories.

In extracting a trajectory of an object in motion, it is often the case that local variations, in the speed of evolution of the continuous observation values, arise while the underlying spatio-temporal structure of the trajectory remains the same. A similar problem exists in speech recognition; the two main sources of variations across time-based voice data are from non-linear warping of the time scale and the variance of the pronunciation itself after the time scale has been optimally restored [4].

However, measuring DTW distance between trajectories alone cannot be adopted directly for clustering using traditional methods such as the k-means or EM algorithms which require clusters represented by their centroids. To overcome this problem, we exploit recent developments in Spectral Graph Theory which have been originally proposed for image segmentation and perceptual grouping. Weiss [17] provides a review of approaches exploiting the information found in the eigenvectors of the affinity matrix. The matrix consists of similarity weights which link pairs of training elements. Sarkar and Boyer [11] determine the number of clusters and cluster membership from the positive same-sign eigenvectors of the affinity matrix. Our experiments have shown that the method does not yield any meaningful results when inter-cluster similarity is not insignificant. Robles-Kelly and Hancock [10] use Sarkar and Boyer's method as an initial estimation of the clustering process and refine the clustering with an EM-like re-estimation of cluster memberships. As such, the technique suffers from the same problem of high inter-cluster affinity. Scott and Longuet-Higgins [12] relocalise the first k eigenvectors to preserve the dominant cluster-membership information but inter-cluster similarity information is lost. Shi and Malik [14] show that the second generalised eigenvector yields the solution to a "normalised cut" which minimises the disassociation resulting from splitting a weight-linked dataset into two. However, the value of this threshold, a regularisation parameter, which controls the quality of the clustering (the goodness of the model) and the number of clusters obtained (the model order), were determined *ad hoc*.

The main focus of this paper is therefore to develop an unsupervised clustering technique which can capture and model intrinsic structures of arbitrary order from a dataset of temporal trajectories with nonlinear temporal variations. This is achieved by first removing non-linear time warping effects from the data. In Section 2, we describe the use of Levenshtein distance, a Dynamic Time Warp (DTW) distance [7], which determines the optimal time warp between two trajectories through dynamic programming and provides a measure of the dissimilarity between the trajectories after the warping effects have been removed. However, the DTW distance does not obey the metric axioms and thus cannot be directly used in traditional clustering methods which rely on the

computation of cluster centroids. In Section 3, we exploit the inverse DTW distance as a warp-free metric for a similarity graph representation and extend Shi and Malik's [14] approach by formulating the normalised cut with an automatic threshold parameter using a self-validation principle based on the discriminant analysis. This is to allow for a fully automatic graph partitioning to perform unsupervised clustering of trajectories of any order. We show in Section 4 how DTW mean trajectories, local time warping profiles and DTW trajectory vertex covariances can be learned for a better cluster representation. Experimental results are provided in Section 5 to illustrate the structures learned from clustering hand-gesture trajectories and moving objects in an indoor environment. We also compare our approach with Robles-Kelly's and Hancock's EM-like clustering via eigendecomposition of the affinity matrix [10].

2 Trajectory Representation and Distance Metrics

Any learning process would inherit the problem of non-linear time warping from directly sampling continuous temporal trajectories into discrete vectors of vertices, often hiding the underlying structure and mutual relationships among visual trajectories [4]. Hidden Markov Models aim to automatically perform dynamic time warping during the recognition phase through the use of the forward-backward or Viterbi algorithm. However, during the learning of the hidden states, using either Vector Quantisation or other forms of clustering, the input trajectory vectors are required to be of fixed-length. A linear time-normalisation of the trajectories results in models that suffer from extra "dimensional" variance. Structure-discovery methods such as Entropic minimisation and Minimum Description Length would attempt to learn models of both non-linear warping and the underlying structure of the trajectories when used on such a representation. In this section, we exploit a representation that captures the underlying structural similarity relationships between trajectories while factoring out the effects of non-linear local time-warping.

2.1 Dynamic Time Warp Distance

Computing the dissimilarity between two temporal trajectories can be formulated as a dynamic programming problem that finds the best time warping from one trajectory $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ to another $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$, consisting of vertices $\mathbf{a}_i \in \Re^N$ and $\mathbf{b}_j \in \Re^N$. A simple DTW distance, as mostly used in computer vision [9, 15], would attempt to find the correspondence between the discrete vertices of the two trajectories. However, sampling drifts and inaccuracies can cause captured trajectories to be out of phase. In situations where the distance between sampled vertices across trajectories are small compared to the distance between consecutive vertices on the same trajectory, skewed sampling can cause the DTW distances to be excessively large while the two trajectories are conceptually, i.e. structurally, very similar. Alternatively, let us introduce Kruskal and Liberman's [7] version of DTW distance, adopted based on the Levenshtein

distance used in string-edit comparisons [8]. To compensate the problem associated with other DTW distances, the interpolated Levenshtein distance matches trajectories according to which vertices are linked with interpolated points on the opposite trajectories, as shown in Fig. 1. The linkage cost $w[\mathbf{a}_i, (\mathbf{b}_j, r))]$ is defined as the Euclidean distance from $\mathbf{a}_i$ to an interpolated point with ratio r, $0 \leq r \leq 1$, between $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$ and respectively for $w[(\mathbf{a}_i, r), \mathbf{b}_j]$,

$$w[\mathbf{a}_i, (\mathbf{b}_j, r))] = || \, \mathbf{a}_i - [\mathbf{b}_j + r \cdot (\mathbf{b}_{j+1} - \mathbf{b}_j)] \, || \tag{1}$$

$$w[(\mathbf{a}_i, r), \mathbf{b}_j] = || \, [\mathbf{a}_i + r \cdot (\mathbf{a}_{i+1} - \mathbf{a}_i)] - \mathbf{b}_j \, || \tag{2}$$

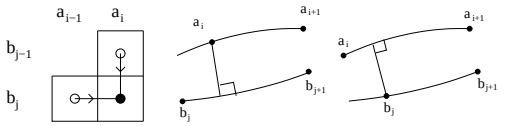


Fig. 1. The Levenshtein distance: the accumulated cost of mapping a vertex from a trajectory to an interpolated point from another trajectory.

The space of all possible links between vertices of the two trajectories can be represented as a two-dimensional grid where each axis represents a trajectory and each cell a vertex-interpolation point link. The cost of each link is inserted into the corresponding cell in the grid and the task of finding the optimal time warp is formulated as the dynamic programming problem of finding the path with the minimum cost from one end of the grid mapping the first vertices of the trajectories to the other end where the last vertices are mapped. More precisely, the Levenshtein distance for computing the cost of the optimal time warp between two trajectories $\mathbf{A}$ and $\mathbf{B}$ is defined as in [7],

$$D(\mathbf{A}, \mathbf{B}) = D_{m-1,n-1} + \min \begin{cases} \min_{0 \leq r \leq 1} w[\mathbf{a}_m, (\mathbf{b}_{n-1}, r)] \\ \min_{0 \leq r \leq 1} w[(\mathbf{a}_{m-1}, r), \mathbf{b}_n] \end{cases} \tag{3}$$

where

$$D_{i,j} = \min \begin{cases} D_{i-1,j} + \min_{0 \leq r \leq 1} w[\mathbf{a}_i, (\mathbf{b}_j, r)] \\ D_{i,j-1} + \min_{0 \leq r \leq 1} w[(\mathbf{a}_i, r), \mathbf{b}_j] \end{cases} \tag{4}$$

2.2 Reformulating the Interpolation Constraint for DTW

Kruskal and Liberman [7] formulated the cost of linking a vertex on one trajectory to an interpolated point between two vertices on another trajectory as the shortest Euclidean distance between the former vertex and the line segment formed by the latter two vertices. The distance is therefore constrained by the geometric configuration of the three vertices. An interpolation ratio r used in the recurrent time warp cost equations to control this distance was defined.

The dimensionality of the vertices and the smoothness of the trajectories contribute to regularise the time warp correspondence matching process. Depending on the underlying structure of the two trajectories, more than one vertex on one trajectory may be linked to the same interpolated line segment on the other. In worst case scenarios, the line segment may be sufficiently long so that a significant portion of the other trajectory is arbitrarily matched against the degenerate line segment. Computing a mini time-warp for the line segment would introduce a local nested minimisation search in the dynamic programming task.

To overcome this problem, we introduce a shifting constraint r to prevent subsequent matching to the same segment from violating the order of the linkage. In Fig 2, $\mathbf{a}_i$, $\mathbf{a}_{i+1}$ and $\mathbf{a}_{i+2}$ all map to interpolation points on the same segment between $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$. While the linkage from $\mathbf{a}_i$ involves the minimisation $\min_{0 \leq r1 \leq 1} w[\mathbf{a}_i, (\mathbf{b}_j, r1)]$, further minimisations along the same segment have a lower limit of the same value as the previous r, i.e. $\min_{r1 \leq r2 \leq 1} w[\mathbf{a}_{i+1}, (\mathbf{b}_j, r2)]$, $\min_{r2 \leq r3 \leq 1} w[\mathbf{a}_{i+2}, (\mathbf{b}_j, r3)]$.

In practical terms, the shifting constraint r is implemented into the dynamic programming task by creating another grid $R_{i,j}$ to store the constraints used for past cells. Another trace grid $T_{i,j}$ is also required to store tokens describing which vertices have been linked to which segments during the computation of past cells, e.g. $\text{Token}_{0,1}$ and $\text{Token}_{1,0}$. For a new cell, the dynamic programming algorithm checks whether a vertex has been already linked to the current line segment and automatically sets the lower bound on the interpolation constraint to r_{min} as follows,

$$\bar{r} = \frac{(\mathbf{a}_i - \mathbf{b}_j) \cdot (\mathbf{b}_{j+1} - \mathbf{b}_j)}{(\mathbf{b}_{j+1} - \mathbf{b}_j) \cdot (\mathbf{b}_{j+1} - \mathbf{b}_j)}, \; r = \begin{cases} 1, & \text{if } \bar{r} > 1, \\ r_{min}, & \text{if } \bar{r} < r_{min} \\ \bar{r}, & \text{otherwise} \end{cases} \tag{5}$$

where the added constraints are

$$r_{min} = \begin{cases} R_{i,j-1}, & \text{if } T_{i,j-1} = \text{Token}_{0,1}, \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

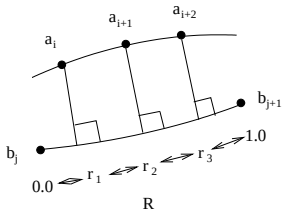


Fig. 2. Constraining the range of the interpolation point through subsequent linkage to the same segment.

2.3 A Representation: Warp-Free DTW Affinity Matrix

However, this DTW distance is computed by finding the cost of the optimal time-warp through dynamic programming. As such, it does not obey the metric axioms, in particular $d(x,z) < d(x,y) + d(y,z)$ is not guaranteed. It is therefore not a suitable representation for traditional clustering techniques, such as k-means or Expectation Maximisation, where centroids are required to represent clusters. In order to utilise DTW distance as a representation for unsupervised clustering, we introduce a pairwise DTW dissimilarity measure to compute an affinity matrix of link-weight similarities between any two trajectories in the training set of N trajectories,

$$W_{s,t} = exp\left(-\frac{D(\mathbf{A}_s, \mathbf{A}_t)}{2\sigma}\right) \tag{7}$$

where σ is the standard deviation of the DTW distance for the whole dataset, $1 \leq s \leq N$ and $1 \leq t \leq N$.

Once this pairwise affinity matrix for a training set of warp-free trajectories is established, applying graph partitioning methods to the matrix is analogous to clustering the data-elements based purely on the structural similarity between trajectories irrespective of any models used.

3 Clustering by Automated Graph Partitioning

Recently, a number of authors have adopted approaches related to Spectral Graph Theory for image segmentation and perceptual grouping [17]. An affinity matrix consists of similarity link-weights of pairs of training elements. For static image segmentation, Shi and Malik [14] attempted to find the minimal (best) "normalised cut" which partitions a graph into two sub-graphs and minimises their inter-cluster affinities. They used a recursive binary-partitioning algorithm together with a normalised cut threshold to stop the partitioning process. The value of this threshold, a regularisation parameter, controls the quality of the clustering and the number of clusters obtained. However, this is rather *ad hoc* and cannot be set in a principled manner. In the following, we extend Shi and Malik's approach to the problem of clustering temporal trajectories with fully automated model-order selection. We formulate the normalised cut threshold parameter using self-validation based on the discriminant analysis principle.

3.1 Normalised Cut of Temporal Trajectories

We consider the trajectory learning problem as that of partitioning the graph whose nodes consist of the trajectories themselves and the link-weights (edges) corresponding to entries in the affinity matrix. Following Shi and Malik, the normalised cut of partitioning a graph $V = \{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ into two sub-graphs C and D is defined as,

$$NCut(C,D) = \frac{cut(C,D)}{assoc(C,V)} + \frac{cut(C,D)}{assoc(D,V)} \tag{8}$$

where, for simplicity, $W_{\mathbf{A}_s,\mathbf{A}_t} \equiv W_{s,t}$,

$$cut(C,D) = \sum_{\mathbf{A}_s \in C, \mathbf{A}_t \in D} W_{\mathbf{A}_s,\mathbf{A}_t},\ assoc(C,V) = \sum_{\mathbf{A}_s \in C, \mathbf{A}_t \in V} W_{\mathbf{A}_s,\mathbf{A}_t} \quad (9)$$

The second generalised eigenvector of the affinity matrix, $(\mathbf{D} = \mathbf{W})x = \lambda \mathbf{D}y$, yields a real-valued solution to the normalised cut. The best (minimal) normalised cut can be obtained by finding the threshold α which partitions the eigenvector into two separate sub-graphs and minimises the Normalised Cut (*NCut*) value. This process is recursively applied to an initial graph until the minimal *NCut* value at each recursion exceeds a preset *NCut* threshold n. The results were disjoint partitions $\{C_1, \ldots, C_k\}$ of V.

3.2 Unsupervised Normalised Cut

The components of the second generalised eigenvector of the affinity matrix can either lie around two easily separable discrete values or they can be continuous [14]. The *NCut* method can only find a solution in the former case. As the partitions are reduced to the intrinsic classes embedded in the data, the latter case becomes the norm as the partitions become harder to split accurately. Shi and Malik proposed a stability criteria based on ratios of histogram bins to detect when the values become continuous so that the splitting process can be stopped. However, ratios of histogram bins do not perform well when the values of the second generalised eigenvector lie between the discrete and continuous state. The method is therefore not guaranteed against over-segmentation of the dataset, especially when the *NCut* threshold n is set high.

Moreover, the *NCut* method attempts to minimise the disassociation caused by splitting graphs and implicitly maximise the association within sub-graphs. In our case, this association is defined in terms of the affinity or similarity of trajectories. An analogous method is Linear Discriminant Analysis where inter-cluster variance is maximised while intra-cluster variance is minimised [3] . However, a rather arbitrary parameter n, the *NCut* threshold, controls the depth of the splitting process and the quality of the clustering. To overcome this problem, let us formally define a cost function to find the optimal threshold $n_{optimal}$ which maximises intra-cluster affinity and minimises inter-cluster affinity,

$$f(n) = -ln(I(n)) + ln(B(n)) \quad (10)$$

where standard deviation functions for the intra-cluster $I(n)$ and between cluster affinities $B(n)$ are computed from the partitioning resulting from parameter n,

$$I(n) = \sqrt{\frac{1}{\sum_{z=0}^{k} \eta(C_z)^2} \sum_{z=0}^{k} \sum_{s \in C_z} \sum_{t \in C_z} (W_{s,t})^2} \quad (11)$$

$$B(n) = \sqrt{\frac{1}{N^2 - \sum_{z=0}^{k} \eta(C_z)^2} \sum_{z=0}^{k} \sum_{s \in C_z} \sum_{t \in C'_z} (W_{s,t})^2} \quad (12)$$

$$\eta(C) = \text{number of elements in set } C \quad (13)$$

We show in Section 5 how an increasing n threshold causes increasing intra-cluster $I(n)$ and between cluster affinities $B(n)$ on real data. The cost function $f(n)$, expressed in terms of the inverse log likelihood of the former and the log likelihood of the latter, becomes a convex function which can be minimised.

4 Learning Warp-Free Models of Trajectories

Given an unsupervised clustering of warp-free temporal trajectories with automatic model order selection, we want to learn representative models for different classes of trajectories. DTW has been used in speech recognition [4] and in computer vision [15] to learn models of classes by warping each instance of the class to a reference voice pattern or trajectory and computing the average. The reference trajectory is often manually selected using domain-specific heuristics to control the characteristics of the model. Furthermore, the time warping is asymmetric and biased towards the reference trajectory.

4.1 Hierarchical DTW Mean of Trajectories

To overcome the above problem, we adopt the weighted DTW average [7] in a hierarchical merging algorithm to learn a warp-free trajectory representation of the underlying structure of each cluster of trajectories. The warp-free trajectory can subsequently be used as the "reference" to recover the time warping profile and remaining structural covariance of each trajectory vertex for recognition.

Given a cluster of trajectories, the average Levenshtein trajectory can be hierarchically computed in any order. However, to minimise interpolation errors, we only average the two closest trajectories at each instant. More specifically,

1. Given a cluster of $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ trajectories, assign each with weight 1.
2. Temporarily make a set $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_n\}$ of weight-normalised trajectories.
3. Find the two normalised trajectories with the smallest Levenshtein distance and remove their non-normalised counterparts from the original cluster.
4. Compute the Levenshtein mean trajectory of the two trajectories according to their weights and compute the weight of the mean trajectory from the sum of the weights of the previous trajectories.
5. Insert the new mean trajectory into the original cluster.
6. Repeat from step 2 until only one trajectory remains.

4.2 DTW Covariance and Time Warp Profiles of Clusters

We compare the warp-free model trajectory of a cluster to its trajectories to learn typical time warping profiles and the remaining structural covariance for each vertex of the model trajectory. The final Levenshtein mean trajectory $\{\mathbf{c}_1, \cdots, \mathbf{c}_h\}$ of a cluster does not retain the time warp links with the elements of the individual trajectories because of the hierarchical weight-warping process. We recompute the linkage of each vertex in the mean trajectory to vertices of

the cluster trajectories by finding the optimum time warp. A trajectory of covariances $\{\Sigma_1, \cdots, \Sigma_h\}$ for each vertex of the mean trajectory is then compiled. We also compute a trajectory of local average time warping by finding the difference in time-stamps of the linked vertices to the mean vertex, e.g. $t(\mathbf{a}_i) - t(\mathbf{c}_k)$. This local time warping information and covariance are used to better guide the scaling dynamics of the correlation particles used in CONDENSATION-based trajectory recognition systems [1] . To this end, we summarise the algorithm as:

1. Find the Levenshtein time warps from the mean sequence $\{\mathbf{c}_1, \cdots, \mathbf{c}_h\}$ to all the trajectories in the cluster $S = \{\mathbf{A}_1, \cdots, \mathbf{A}_z\}$.
2. For each vertex $\mathbf{c}_k$ in the mean trajectory, find the set of time warp links from the vertex to interpolated points on the other trajectories $I_k = \{(\mathbf{c}_k, (\mathbf{a}_i, r)) : \mathbf{a}_i \in \mathbf{A}, \mathbf{A} \in S\}$.
3. For each vertex $\mathbf{c}_k$, find the covariance

$$\Sigma_k = \frac{1}{\eta(I_k)} \sum_{(\mathbf{c}_k, (\mathbf{a}_i, r)) \in I_k} \{((\mathbf{a}_i, r)) - \mathbf{c})^{\mathrm{T}} \cdot ((\mathbf{a}_i, r)) - \mathbf{c})\} \tag{14}$$

and local time warp factor

$$tw_k = \frac{1}{\eta(I_k)} \sum_{(\mathbf{c}_k, (\mathbf{a}_i, r)) \in I_k} |t(\mathbf{a}_i) - k| \tag{15}$$

$$\eta(C) = \text{number of elements in set } C \tag{16}$$

5 Experiments

The warp-free trajectory-learning algorithm was tested on a dataset of 500 hand gestures from [16], illustrated in Fig. 3. They were performed in random order from a repertory of 7 descriptive gestures and captured by a head and hand tracking system developed by Sherrah and Gong [13]. The trajectories were sampled 15 times per second and segmented by a multi-scale pause detection algorithm developed by Walter [16], which was reported to yield 19 inherent atomic gesture classes from visual inspection. We show the clustering solutions obtained from different values of the *NCut* threshold in Fig. 5. We also plot the average intra-cluster and inter-cluster affinity values as well as the number of clusters found for each *NCut* threshold in Fig. 7. The value of n for the optimal partitioning can be seen to lie somewhere between 0 and 1.0. Of particular interest is that the threshold n controls to what extent sub-graphs or clusters are further split and thus defines the quality of the clustering as illustrated in the reordered affinity matrices in Fig. 5. As n increases, the average intra-cluster affinity increases while the average inter-cluster affinities also increases but by a smaller factor. The number of clusters also increase as n is increased. The relationship between the three values is shown in Fig. 7.

For comparison, we first implemented and tested Robles-Kelly and Hancock's [10] grouping method which use Sarkar and Boyer's [11] method as an initial

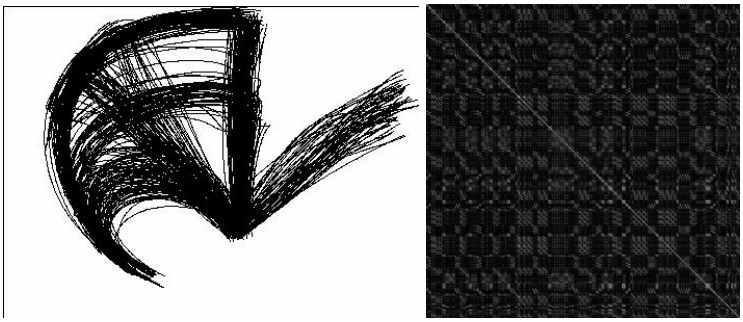
estimate for the clustering. In both methods, the model order was obtained from the number of same-sign eigenvectors of the affinity matrix with positive eigenvalues. The non-zero same-sign values of the eigenvector were interpreted as cluster membership indicators. Robles-Kelly and Hancock observed that the method only works on datasets where the inter-cluster affinities are close to zero. On our gesture data, we obtained a model order of 1. The values of the only positive same-sign largest eigenvector are plotted in Fig. 4 and it can be seen that the non-zero values encode the membership of the first cluster while the near-to-zero values encode the combined membership of the other clusters. Our dataset is therefore too complex for such a clustering algorithm with holistically determined (directly from the eigenvectors) model order.

We compare the results of our unsupervised *NCut* to Shi and Malik's threshold value n of 0.04. This value is too small to break the strong between-class similarities of the gesture dataset into significant structures as shown in Fig. 6. By minimising the cost function in Eqn. (10), our unsupervised *NCut* searches for the best compromise between finding partitions of high intra-class similarities and splitting clusters with similar trajectories into partitions of high between-class similarities. For the gesture dataset, the value of $n_{optimal}$ obtained is 0.5037 and the number of clusters found is 18. The unsupervised *NCut* has found the intrinsic structure of the trajectory classes in Fig. 8. From a cursory look on the trajectory classes obtained, our algorithm has identified three single-element classes Fig. 8(p-r) as outliers. A few outliers have been included in some of the clusters (g, h and j), while the segmentation of the remaining clusters match the conceptual atomic gestures quite closely. Fig. 8 has been ordered to present trajectories of similar shape but different directions next to each other, e.g. a and e, i and j, k and l, etc.

We have also tried the warp-free trajectory-learning algorithm on the Queen Mary "can shop" experiment, as illustrated in Fig. 9. Customers are tracked as they enter the shop from the left, browse the array of cans in the centre of the image, then move on to pay the shop-keeper at the right-end and exit from the left again. There are 22 trajectories in the dataset with high semantical content and the customers move according to their state of mind, i.e. entering, browsing, more browsing, paying and leaving, sometimes not in any sensible order. Owing to lack of space, only the reordered affinity matrices with 4 detected clusters from our unsupervised *NCut* are shown in Fig. 10.

6 Conclusion

We presented a comprehensive, fully automatic unsupervised technique for the clustering of temporal trajectories of arbitrary model order. Levenshtein-distance based Dynamic Time Warping was used to remove non-linear warping of the time scale in temporal trajectories, which is a key characteristic of most stochastically generated temporal data. Owing to the fact that DTW distance is not a Euclidean metric distance, traditional centroid clustering techniques such as k-means or EM cannot be used. An affinity matrix was built from the inverse of



(a) (b)

Fig. 3. From left to right: (a) a hand-gesture dataset of 500 trajectories and (b) the pairwise 500×500 affinity matrix of the elements of the dataset where black is the most dissimilar and white is the most similar.

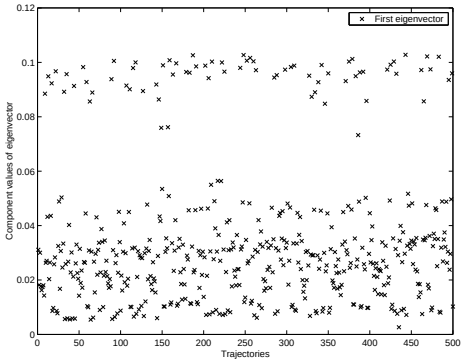


Fig. 4. Results from Kelly and Hancock's method: The component values of the first eigenvector of the affinity matrix of the gesture dataset. The top row (around 0.1) indicates the membership of the trajectories to the first cluster while the other values do not clearly separate the membership of the remaining trajectories to the other clusters.

the pairwise DTW distance between trajectories. The clustering problem was then treated as the optimal partitioning of the graph where the nodes consist of the trajectories and the link weights consist of the affinities. We extended the Normalised Cut graph partitioning method for unsupervised discovery of intrinsic structures in the affinity matrix and the dataset by selecting the free *NCut* threshold parameter n to maximise the intra-cluster affinity and minimise the inter-cluster affinity of the final partitioning solution. Comparative experiments with existing techniques including Kelly and Hancock [10] and Shi and Malik [14] indicate that the partitions yielded a much more accurate clustering of the trajectory dataset when other Graph Theoretical methods have failed to construct any meaningful clustering. Our results corresponded well to the con-

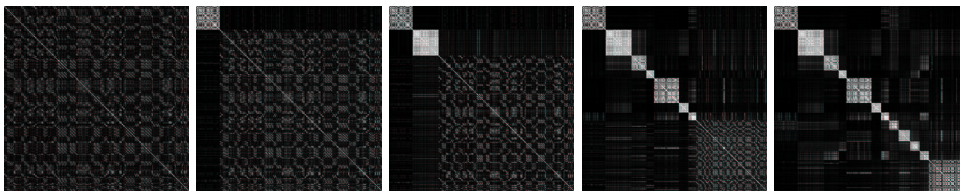
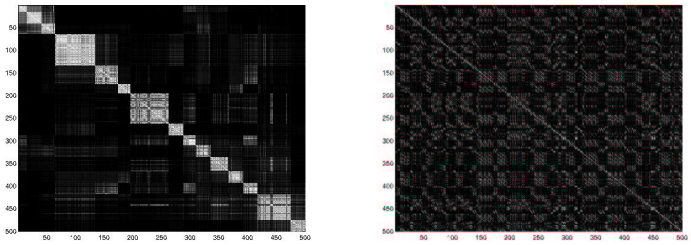


Fig. 5. From left to right: The affinity matrix of the gesture dataset with elements of the same cluster reordered to be adjacent to each other for *NCut* threshold n: 0.025, 0.075, 0.150, 0.275, 0.325.



(a) (b)

Fig. 6. From left to right: The reordered affinity matrix for (a) our unsupervised *NCut* method ($n_{optimal}$ estimated to be 0.5037) and (b) Shi and Malik's *NCut* with original n threshold manually set to 0.04.

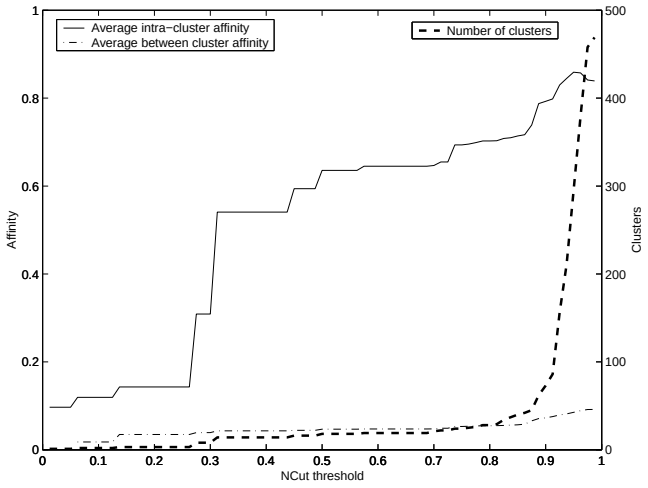


Fig. 7. A plot of the average intra-cluster affinity, inter-cluster affinity and no. of clusters found for varying values of the *NCut* threshold n for the gesture dataset.

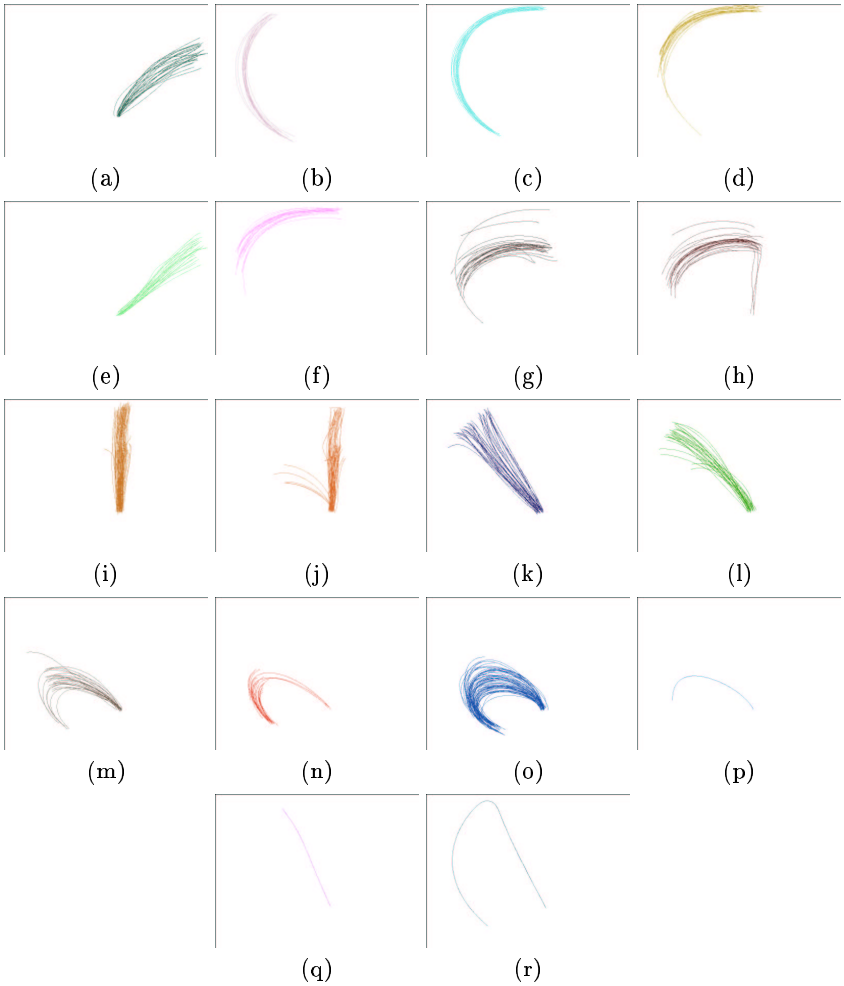


Fig. 8. From left to right, top to bottom: The clusters obtained from the unsupervised *NCut* ($n_{optimal}$ estimated to be 0.5037) on the gesture dataset. Of note is that most of the gestures are performed in both directions and are separated into two clusters.

ceptual classes of observed trajectories. We also showed how warp-free DTW mean trajectories, time-warp profiles and trajectory vertex covariances can be learned from the data.

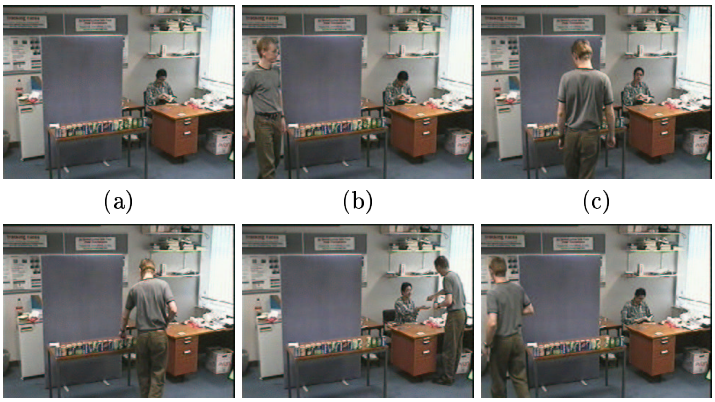


Fig. 9. From left to right, top to bottom: (a) The QMUL "can shop". To illustrate customer behaviour, we show selected frames for 5 typical events: (b) enter from left, (b) browse cans in the middle, (d) take a can (optional), (e) pay shopkeeper on the right (optional), and (f) leave shop through the exit on the left.

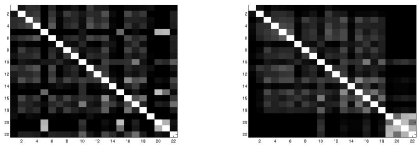


Fig. 10. From left to right: (a) The affinity matrix of the object centroid trajectories of the QMUL "can shop" (b) the reordered affinity matrix for $n_{optimal} = 0.2361$ and 4 clusters obtained from unsupervised *NCut*.

Bibliography

[1] M.J. Black and A.D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *ECCV*, pages 909–924, Freiburg, 1998.

[2] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE PAMI*, 22(8):844–851, August 2000.

[3] R.A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[4] S. Furui. Vector-quantization-based speech recognition and speaker recognition techniques. In *Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 954–958, Los Alamitos, USA, 1991.

[5] T. Jebara and A. Pentland. Automatic visual analysis and synthesis of interactive behaviour. In *International Conference on Vision Systems*, pages 273–292, Berlin, Germany, 1999.

[6] N. Johnson and D.C. Hogg. Learning the distribution of object trajectories for event recognition. *IVC*, 14(8):609–615, 1996.

[7] J.B. Kruskal and M. Liberman. The symmetric time-warping problem: From continuous to discrete. In *Time Warps, String Edits, And Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161. CSLI Publications, 1999.

[8] V.I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Cybernetics and Control Theory*, 10(8):707–710, 1965.

[9] S.Y. Lee M.K. Shan. Content-based video retrieval via motion trajectories. *SPIE*, 3562:52–61, 1998.

[10] A. Robles-Kelly and E.R. Hancock. An em-like algorithm for motion segmentation via eigendecomposition. In *BMVC*, pages 123–132, Manchester, U.K., 2001.

[11] S. Sarkar and K.L. Boyer. Quantitative measures of change based on feature organisation: Eigenvalues and eigenvectors. *CVIU*, 71(1):110–136, July 1998.

[12] G.L. Scott and H.C. Longuet-Higgins. Feature grouping by relocalisation of eigenvectors of the proximity matrix. In *BMVC*, pages 103–108, 1990.

[13] J. Sherrah and S. Gong. Resolving visual uncertainty and occlusion through probabilistic reasoning. In *BMVC*, pages 252–261, Bristol, UK, 2000.

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, August 2000.

[15] T. Wada and T. Matsuyama. Multiobject behavior recognition by event driven selective attention method. *IEEE PAMI*, 22(8):873 –887, 2000.

[16] M. Walter, A. Psarrou, and S. Gong. Data driven gesture model acquisition using minimum description length. In *BMVC*, pages 673–683, Manchester, UK, 2001.

[17] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, pages 975–982, Los Alamitos, USA, 1999.