

# Parametric Trajectory Representations for Behaviour Classification

Rowland R. Sillito  
r.sillito@ed.ac.uk

Robert B. Fisher  
rbf@inf.ed.ac.uk

Inst. Perception Action and Behaviour  
School of Informatics  
University of Edinburgh, UK

---

## Abstract

This paper presents an empirical comparison of strategies for representing motion trajectories with fixed-length vectors. We compare four techniques, which have all previously been adopted in the trajectory classification literature: least-squares cubic spline approximation, the Discrete Fourier Transform, Chebyshev polynomial approximation, and the Haar wavelet transform. We measure the class separability of five different trajectory datasets - ranging from vehicle trajectories to pen trajectories - when described in terms of these representations. Results obtained over a range of dimensionalities indicate that the different representations yield similar levels of class separability, with marginal improvements provided by Chebyshev and Spline representations. For the datasets considered here, each representation appears to yield better results when used in conjunction with a curve parametrisation strategy based on arc-length, rather than time. However, we illustrate a situation - pertinent to surveillance applications - where the converse is true.

## 1 Introduction

Motion trajectories form the basis for many current approaches to surveillance-related behaviour modelling and classification [5, 11, 12, 14, 23]. While trajectories provide a highly efficient way to summarise surveillance footage, the unbounded dimensionality of trajectory data presents an obstacle when applying machine learning techniques or building statistical models from trajectory data.

It is, of course, possible to sidestep this issue in a variety of different ways. One promising approach is to decompose trajectories into prototype sequences corresponding - for example - to motion vectors [12, 21, 23] or path segments [14], which can then form the basis for constructing symbolic sequence models eg. [8, 14], or for constructing hierarchical models of prototype cooccurrence patterns eg. [21, 23]. Alternatively, it is possible to parametrise trajectories so that a similarity function can be defined: in [17] trajectories are individually fitted with HMMs to define a likelihood-based distance function, while in [11] trajectories are resampled/padded to uniform length so that the Euclidean distance can be evaluated.

However, since anomalous behaviour detection forms a key motivation for trajectory classification algorithms, it is also useful to be able to represent trajectories in a fixed low-dimensional vector form, so that a wide range of current approaches to multivariate outlier detection see eg. [15] could then be applied. Moreover, it may be useful (eg. for incremental/online learning [20]) to be able to represent trajectories in a pre-defined parameter



Figure 1: Representation of trajectory data as separate (X and Y vs time) signals.

space that does not need to be learned from data. Several recent trajectory classification approaches solve this problem by applying 1D signal approximation techniques to describe the X and Y coordinate sequences of a given trajectory with a fixed set of parameters. An early instance of this approach was proposed by Sahouria and Zakhor in [19], where Haar wavelet coefficients are used to represent trajectories for video indexing purposes. More recently, Naftel and Khalid have explored the use of the Discrete Fourier Transform and Chebyshev polynomial approximations for trajectory classification in [16], while Sillito and Fisher have adopted a least-squares cubic spline approximation as the basis for an incremental trajectory modelling algorithm in [20].

Given the wide selection of different techniques, this paper poses and attempts to answer a simple question: what is the best way to represent a set of trajectories with fixed-length vectors? Firstly, Section 2 reviews the preceding trajectory parametrisation techniques and clarifies the manner in which they are implemented in this paper; two useful non-parametric techniques for quantifying class separability are then described in Section 3. Finally, Section 4 presents quantifications of the trade-off between dimensionality and class-separability for each representation, using several different real trajectory datasets.

## 2 Trajectory Representations

As illustrated in Figure 1, a motion trajectory can be considered in terms of two independent signals: X and Y positions vs time. This section briefly reviews the four approaches identified in the preceding section: each provides a means to describe arbitrary-length coordinate vectors  $\vec{X}$  and  $\vec{Y}$ , sampled at times  $\vec{T} = \{t_1, \dots, t_N\}$ , with fixed-length parameter vectors  $\vec{C}^X$  and  $\vec{C}^Y$ , which can then be concatenated as a single vector describing the trajectory:

$$\{\vec{X}, \vec{Y}, \vec{T}\} \rightarrow [C_1^X, \dots, C_M^X, C_1^Y, \dots, C_M^Y] \quad (1)$$

In each case the parameters  $\vec{C}^X, \vec{C}^Y$  extracted for a given coordinate sequence define a curve with respect to an underlying curve parameter, over a fixed interval. As suggested by Figure 1 a natural choice for this parameter is *time*, another possibility - relevant to trajectory representation - is *arc-length* (ie. the total distance traversed at a given point) [9]. We explore both possibilities - thus, in the remainder of this section, the parameter  $s_n$  accompanying the  $n$ th coordinate pair  $x_n, y_n$  could refer interchangeably to one of the following:

1. Proportion of total time:  $s_n = \frac{t_n}{t_N}$
2. Proportion of total arc-length:  $s_n = \frac{\sum_{i=2}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{\sum_{i=2}^N \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}$

Each of the following trajectory representation strategies defines a curve approximation as sum of  $M$  basis functions  $h_1(s), \dots, h_M(s)$  weighted by a corresponding sets of coefficients  $\vec{C}^X$  and  $\vec{C}^Y$ , so that  $X(s) = \sum_{m=1}^M h_m(s) \cdot C_m^X$  and  $Y(s) = \sum_{m=1}^M h_m(s) \cdot C_m^Y$ . What distinguishes each method is the nature of the basis functions, and the manner in which the coefficients are determined. In each case we describe how to determine a set of coefficients  $\vec{C}^X = [C_1^X, \dots, C_M^X]$  approximating an  $X$  coordinate sequence  $\vec{X} = [x_1, \dots, x_N]$  with parameter values  $\vec{S} = [s_1, \dots, s_N]^T$ , noting that exactly the same procedure can be applied to  $\vec{Y}$ .

**Haar Wavelet Coefficients** The Haar wavelet basis set is defined on the interval  $[0, 1]$ , consisting of a scaling function  $\phi(s)$ , and wavelet functions  $\psi_{j,k}(s)$ , where the indices  $k \in [0 \dots 2^j - 1]$  and  $j$  determine location and scale respectively [3, p 60]:

$$\phi(s) = \begin{cases} 1 & \text{if } 0 \leq s < 1 \\ 0 & \text{otherwise} \end{cases} \quad \psi_{j,k}(s) = 2^{\frac{j}{2}} \psi(2^j s - k) : \psi(s) = \begin{cases} 1 & \text{if } 0 \leq s \leq \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} < s \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Given a signal  $\vec{X} = [x_1, \dots, x_N]$ , where  $N$  is an integer power of 2, Haar wavelet coefficients can be calculated using the following closed-form expression [24, p 343] (or, far more efficiently, by using a transformation matrix - see [3] for details):

$$\alpha = \frac{1}{N} \sum_{i=1}^n \phi(s_i) \cdot x_i \quad \beta_{j,k} = \frac{1}{N} \sum_{i=1}^n \psi_{j,k}(s_i) \cdot x_i \quad (3)$$

The requirement that signal length should be an integer power of 2 poses a problem for trajectory parametrisation. We address this by resampling trajectories to an appropriate length (as proposed in [19]): specifically, we linearly interpolate each coordinate sequence to a uniform length of 1024 (ie.  $2^{10}$ ) elements. A vector of  $2^J$  coefficients (where  $J \geq 1$ ) can then be constructed to represent the  $J$ th resolution approximation to the resampled signal:

$$\vec{C}^X = [\alpha, \beta_{0,0}, \dots, \beta_{J-1, 2^J-1}] \quad (4)$$

**Discrete Fourier Transform** The Discrete Fourier Transform (DFT) allows a series of measurements sampled at regular intervals to be described as a weighted set of sinusoidal functions with different frequencies, so that the weights for the constituent frequencies encode the information contained in the original signal. For a signal  $\vec{X}$  of length  $N$ , there are  $\frac{N}{2}$  distinct frequencies; a complex number capturing the magnitude/phase of the  $k$ -th frequency can be calculated as follows (where  $\mathbf{i}$  is the imaginary unit  $\sqrt{-1}$ ) [18, p 607]:

$$f_k(\vec{X}) = \frac{1}{N} \sum_{n=1}^N x_n \exp\left(-\frac{2\pi \mathbf{i}(k-1)(n-1)}{N}\right) \quad (5)$$

Noting that the imaginary component of  $f_1$  is always zero, an approximation to a set of regularly sampled coordinates  $\vec{X} = [x_1, \dots, x_N]$  can thus be represented with a  $2M - 1$  dimensional vector as follows (where  $\Re(z)$  and  $\Im(z)$  denote the magnitude of the real and imaginary components of  $z$ ):

$$\vec{C}^X = [\Re(f_1(\vec{X})), \Re(f_2(\vec{X})), \Im(f_2(\vec{X})), \dots, \Re(f_M(\vec{X})), \Im(f_M(\vec{X}))] \quad (6)$$

For trajectory data regularly sampled in time, the DFT implicitly yields a temporal parametrisation. To achieve spatial parametrisation for this method we linearly interpolate the original trajectory to create a new coordinate sequence regularly sampled in terms of arc-length.

**Chebyshev Polynomial Coefficients** The  $n$ th Chebyshev polynomial function ( $n \geq 0$ ) is defined on the interval  $s \in [-1, 1]$  as follows:

$$T_n(s) = \cos(n \arccos(s)) \quad (7)$$

In order to calculate coefficients to approximate a function  $X(s)$  with the first  $M$  Chebyshev polynomials, it is necessary to evaluate  $X(s)$  and  $T_0(s), \dots, T_{M-1}(s)$  for values of  $s$  where  $T_M(s) = 0$ . There are  $M$  such values for  $T_M(s)$ , with the  $k$ th one given by  $s_{0,k} = \cos(\frac{1}{M}\pi(k + \frac{1}{2}))$  [18, p 233]. Given a discrete sequence  $\vec{X}$  with parameter values  $\vec{S}$  (ranging between 0 and 1), we rescale  $\vec{S}$  to the interval  $[-1, 1]$  and use linear interpolation to find an  $X$  value,  $x_{0,k}$ , for each  $s_{0,k}$ . Finally, each coefficient  $c_j$  can be calculated as follows [18, p 234]:

$$c_0 = \frac{1}{M} \sum_{k=0}^{M-1} x_{0,k} \cdot T_0(s_{0,k}) \quad c_{j>0} = \frac{2}{M} \sum_{k=0}^{M-1} x_{0,k} \cdot T_j(s_{0,k}) \quad (8)$$

**Cubic B-Spline Control Points** The  $i$ th member of a set of  $p$  cubic B-spline functions can be defined as follows [10, p 160], where  $m$  denotes the order of the function ( $m = 4$  for cubic splines) and  $\vec{\tau}$  denotes a ‘‘knot vector’’ with  $p + 4$  elements such that  $\vec{\tau}_{1..4} = [0, 0, 0, 0]$ ,  $\vec{\tau}_{5..p} = \frac{1}{p-3} \cdot [1, \dots, p-4]$ , and  $\vec{\tau}_{p+1..p+4} = [1, 1, 1, 1]$ :

$$B_{i,1}(t) = \begin{cases} 1 & \text{if } \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad B_{i,m}(t) = \frac{t-\tau_i}{\tau_{i+m-1}-\tau_i} B_{i,m-1} + \frac{\tau_{i+m}-t}{\tau_{i+m}-\tau_{i+1}} B_{i+1,m-1} \quad (9)$$

An approximation for a set of  $N$  coordinates  $\vec{X}$  with corresponding parameter values  $\vec{S}$  can be expressed in terms of a vector of  $p$  unknown coefficients  $\vec{C}^X$ , and an  $N \times p$  matrix  $\Phi$  where  $\Phi_{n,i} = B_{i,4}(s_n)$ , so that  $\vec{X} \approx \Phi \vec{C}^X$ . The  $p$  coefficients which minimise the sum of squared errors between the original coordinates and their approximation can then be found using the pseudoinverse operator  $\Phi^\dagger = (\Phi^T \Phi)^{-1} \Phi^T$  as follows [2, p 142]:

$$\vec{C}^X = \Phi^\dagger \vec{X} \quad (10)$$

### 3 Separability Measures

Given a set of trajectories sampled from several different classes of motion, and a potential parametric representation, we wish to quantify the extent to which instances from each class occupy class-specific regions in the space defined by the chosen representation. This section highlights two measures that can be used to provide an indication of class separability.

**Graph-based within-vs-between class edge-weight ratio ( $J_{RNG}$ )** In [25] Zighed et al. proposed a non-parametric class separability measure which works by comparing the sets of within- and between-class edges of a sparse weighted graph spanning the dataset. To measure separability, a Relative-Neighbourhood Graph (see [27]) is constructed from the

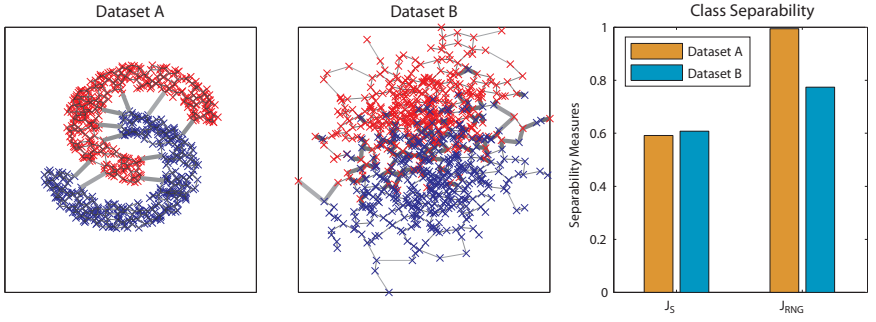


Figure 2: Comparing graph-based separability measure  $J_{RNG}$  [25] with the within-vs-between class scatter ratio [1, p 446]  $J_S = Tr\{S_W^{-1}S_B\}$  (right). Plots of two-class datasets A and B (left, middle) show relative neighbourhood graph edges (thick/thin edges indicate between/within class links, lighter shades indicate lower weights). Although classes in dataset B overlap significantly, it has a higher  $J_S$  value than dataset A which has no overlap. In contrast  $J_{RNG}$  provides a reasonable indication of class overlap.

dataset, where each node represents an individual data point, and edges join only those pairs of nodes that have no mutually closer neighbour. Each edge is then weighted by a similarity function  $\frac{1}{1+d(x_i, x_j)}$ , based on the Euclidean distance  $d(x_i, x_j)$  between two data points.

The resulting graph can be described by a matrix  $W$ , with each element  $w_{ij}$  given by:

$$w_{ij} = \begin{cases} \frac{1}{1+d(x_i, x_j)} & \text{if } d(x_i, x_j)_{i \neq j} \leq \max[d(x_i, x_k), d(x_j, x_k)]_{\forall k \notin \{i, j\}} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Quantities analogous to within and between class scatter can be defined in terms of this matrix, through summation of the weights of edges joining members of the same class/those joining members of different classes as follows (where  $I(\cdot)$  denotes the indicator function, and  $L_n$  denotes the label of the  $n$ th data point):

$$E_{within} = \sum_{i=1}^{N-1} \left( \sum_{j=i+1}^N I(L_i = L_j) \cdot w_{ij} \right) \quad E_{between} = \sum_{i=1}^{N-1} \left( \sum_{j=i+1}^N I(L_i \neq L_j) \cdot w_{ij} \right) \quad (12)$$

Noting that edges are only present for pairs of data points that are (relatively [22]) proximal, it is possible to measure separability in terms of  $E_{within}$  and  $E_{between}$ . In particular, for datasets whose classes are well-separated, the sum of edge weights joining members of the same class ( $E_{within}$ ) will constitute the majority of the total sum of edge weights ( $E_{within} + E_{between}$ ). This leads to the following measure, ranging between 0 and 1, which increases proportionally to class separability:

$$J_{RNG} = \frac{E_{within}}{E_{within} + E_{between}} \quad (13)$$

As illustrated in Figure 2, this provides a better separability measure for non-unimodal datasets than the standard (see eg. [2, 7]) within-vs-between class scatter ratio  $J_S = Tr\{S_W^{-1}S_B\}$ , where  $S_W$  and  $S_B$  denote within and between class covariance matrices respectively.

**Leave-one-out nearest-neighbour classification accuracy ( $J_{NN}$ )** As a supplementary indicator of class separability, we also measure the average leave-one-out nearest-neighbour classification accuracy for the dataset. This classification method assigns to each data point the label of its nearest neighbour: it is thus reasonable to assume the proportion of correct classifications will be proportional to the separation of a set of classes in a given feature space. We quantify classification accuracy in a manner which is invariant to differences in class proportion, by averaging the classification rates calculated separately for each of  $K$  classes as follows (where  $L_n$  and  $P_n$  denote true and predicted labels for the  $n$ th data point):

$$J_{NN} = \frac{1}{K} \sum_{k=1}^K \left( \frac{1}{N_k} \sum_{i \in C_k} I(L_i = P_i) \right) \quad (14)$$

This measure therefore provides a useful supplement to  $J_{RNG}$  since the latter is affected by class proportions (because classes with fewer members contribute fewer graph edges).

## 4 Experiments

**Datasets** We use the approaches described in Section 2 to represent the following trajectory datasets for a range of dimensionalities, before measuring separability as described earlier.

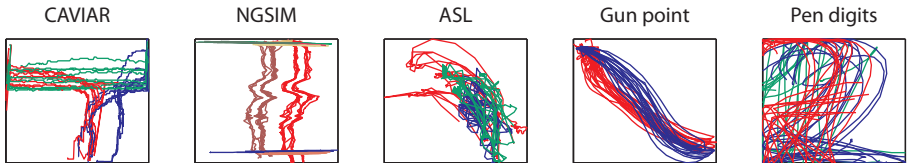


Figure 3: Trajectory datasets (examples coloured by class). See text for description.

1. The *CAVIAR* dataset [6]. We use a set of 111 trajectories corresponding to a side-on view of a shopping centre, each labelled according to one of 6 possible routes (this dataset is used/labelled in a similar fashion in [16]).
2. The *NGSIM* dataset [4]. We use a set of 1758 vehicle trajectories from the “Peachtree St.” scenario, labelled according to their entry/exit point combination, yielding 33 different classes (we disregard tracks lasting  $<2$  seconds / classes with  $<10$  examples).
3. The *Australian Sign Language* dataset [1]. We use a set of 2565 right-hand trajectories corresponding to 95 different words. (Right-hand trajectories from this dataset have previously been used to assess trajectory classification in [16].)
4. The *Gun-point* dataset [13]. This dataset contains 233 hand trajectories with two distinct classes. One class corresponds to the to drawing/pointing of a gun, while a second class corresponds to innocuously pointing forwards without a gun.
5. The *Pen-Based Recognition of Handwritten Digits* dataset [1], which contains 10,992 pen trajectories corresponding to instances of the digits ‘0’ to ‘9’ written on a graphics tablet. We use 2000 randomly selected examples from this dataset (200 per digit).

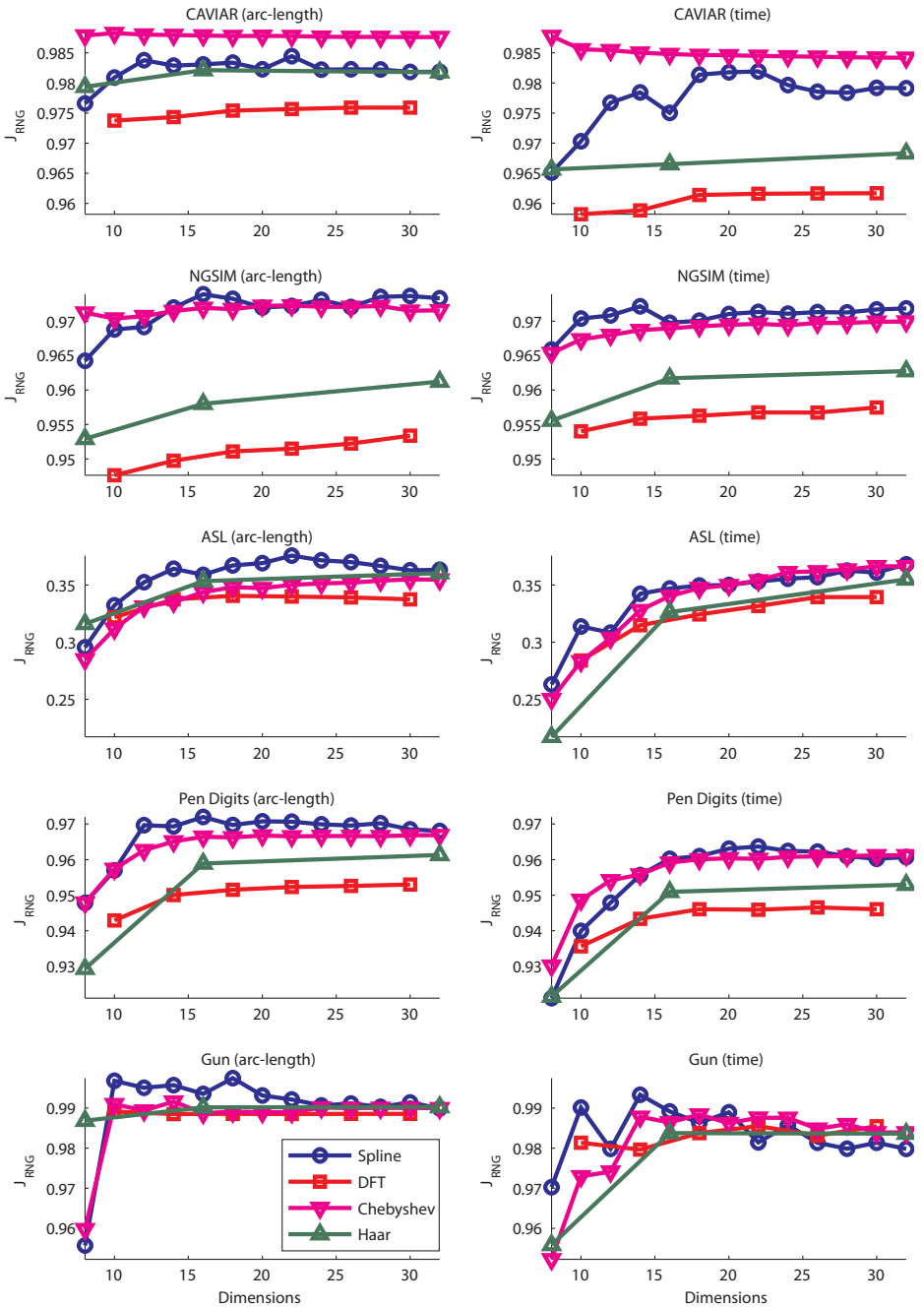


Figure 4: Graph-based separability measure  $J_{RNG}$  calculated for each dataset, using different trajectory representations over a range of dimensionalities. (Please note that the Y axis scale differs between plots - the observed differences in separability are small.) See text for discussion.

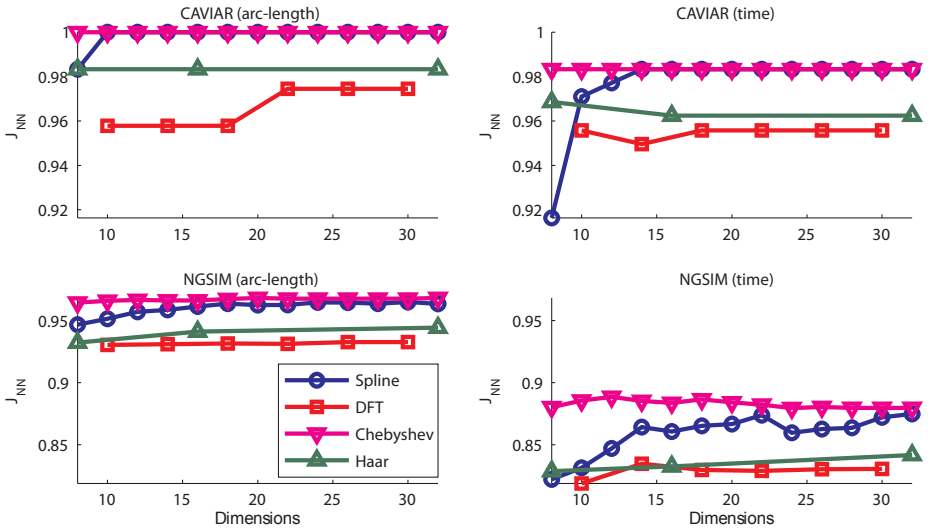


Figure 5: Leave-one-out 1-nearest neighbour classification rate  $J_{NN}$  (corrected for differences in class size) for the CAVIAR and NGSIM datasets (which both have unequal class sizes), using different trajectory representations over a range of dimensionalities. See text for discussion.

**Results** Figure 4 shows the separability values obtained using  $J_{RNG}$  (as described in Section 3) for each representation and its corresponding range of dimensionalities (within the range 8-32). In both figures, each row shows results for a given dataset: the left hand plot shows results for representations parametrised by arc-length while the right hand plot shows results for those parametrised by time; plots within each row have identical Y-axis scale/limits to facilitate the comparison of time/arc-length parametrisations.

It can immediately be seen that in most cases the differences in separability values are small, with the observed values typically spanning a range of around 0.05 (the  $J_{RNG}$  measure assigns values between 0 and 1). The results do not, therefore, indicate that the choice of representation makes more than a small difference to class separability. A natural question to ask, however, is whether there is any persistent “ordering” corresponding to different representations within the small range of observed variation. In most cases, the Haar representation appears to improve upon the DFT representation, and the highest separability values are yielded by either the Chebyshev or Spline representations. For the DFT/Chebyshev/Haar representations, separability appears to increase with dimensionality before reaching a plateau at around 20 dimensions. This trend is less clear for the Spline representation: this may be a consequence of the least-squares fitting procedure, indicating that greater fidelity to the original trajectory (as measured by the sum-of-squared errors) may not directly correspond to better class separation.

The impact of the two different curve parametrisation strategies (ie. either proportion of total time or of total arc-length) can be examined by comparing the left/right hand plots for each dataset. For the CAVIAR, Pen Digits and Gun-point datasets, parametrisation by arc-length appears to improve separability in all cases. For the ASL and NGSIM datasets this trend is less clear, although in both cases the highest observed separability values correspond to arc-length parametrisation. Noting that the CAVIAR and NGSIM datasets contain widely



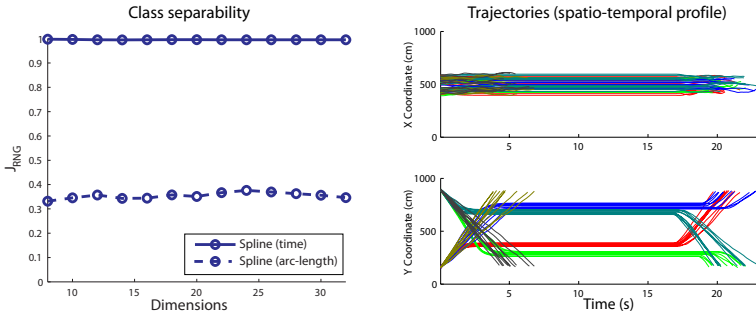


Figure 6: Comparing the separability of simulated trajectories with 6 distinct spatiotemporal classes (right, coloured by class) when represented using splines parametrised by time and arc-length (left). Trajectories follow two possible directions, with three possible temporal profiles: the simulated pedestrian may either pause (in one of two locations) or progress without pausing.

varying class-proportions, we also measure the class-corrected nearest neighbour classification rate  $J_{NN}$  for these datasets, shown in Figure 5. In both cases the relative ordering of separability values is very similar to that observed for the  $J_{RNG}$  plots. Again arc-length parametrisation yields an improvement over time-based parametrisation: however, this is most significantly manifested for the NGSIM dataset where arc-length parametrisation yields a large ( $\sim 10\%$ ) improvement in classification rate.

A clear potential reason for the sharp improvement observed when using arc-length parametrisation to represent the NGSIM dataset pertains to the wide range of different spatio-temporal profiles present in this data: since this data contains vehicle trajectories from a road segment which includes traffic lights etc., it is likely to yield a widely differing set of spatio-temporal profiles for a given route. While one would not automatically rule out the possibility of a single representation which captures spatial and spatiotemporal characteristics, the results obtained for NGSIM dataset indicate that a single spatio-temporal representation may be inadequate if a sufficiently large number of different spatial and spatiotemporal categories are present - regardless of the particular method used.

However, while the results presented here favour the use of arc-length parametrisation, this strategy is likely to be inadequate for many trajectory classification purposes: for surveillance applications, the spatiotemporal characteristics of trajectory may capture the most important information for behaviour classification. To illustrate this point, Figure 6 shows a set of simulated trajectories with 6 distinct spatio-temporal categories, together with its corresponding separability values when represented using splines with time and arc-length parametrisation: it is clear that temporal parametrisation allows near-perfect separation of the 6 classes, while arc-length parametrisation only allows partial separation. This type of spatio-temporal classification problem is likely to be important in a surveillance context, where loitering in different locations may have very different implications.

## 5 Conclusions

This paper has explored the impact of different parametric trajectory representations on the class separability of a range of datasets. The results obtained indicate two things: Firstly, for a given curve parametrisation strategy, the difference between the separability of different

trajectory representation strategies is small, but the Chebyshev and cubic Spline representations appear to provide the best results.

Secondly, although temporal parametrisation is essential if spatio-temporal categories are to be distinguished, curve-parametrisation based on arc-length yields improved separability for the datasets considered here. This is particularly noticeable for the NGSIM dataset, indicating that an all-encompassing spatio-temporal parametric trajectory representation may be an unrealistic prospect for complex real-world surveillance scenarios, where the need to compromise between adequately representing the spatial and temporal characteristics of trajectories may prove unacceptable. Thus, if parametric trajectory representations are to be used, the logic of hierarchical modelling strategies (eg. [11] where spatio-temporal characteristics are modelled separately for different spatially-defined trajectory clusters) becomes apparent.

Finally, it is important to note that the experiments presented in this paper measured class separation based on Euclidean distances between raw/unprocessed coefficient vectors: it is thus possible that further processing of these vectors (eg. by re-weighting or rescaling attributes) may yield improved separation for a given representation. In this light, it is possible that the potential levels of classification performance achievable for each representation may follow a very different pattern to that observed here. In a similar vein, it should be also be noted that the set of trajectory representations examined here merely represents the choices previously adopted in the literature: it is not unlikely that there exist other representations (eg. one of many different families of wavelets) that may be preferable.

## Acknowledgments

RRS was supported by an EPSRC/MRC Neuroinformatics DTC studentship. The authors would like to thank Eamonn Keogh for providing valuable data.

## References

- [1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [3] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms*. Prentice Hall, 1997.
- [4] Cambridge Systematics Inc. NGSIM vehicle trajectory datasets. URL <http://ngsim.camsys.com/>.
- [5] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *Proc. BMVC*, pages 477–486, 2004.
- [6] R. B. Fisher, J. Santos-Victor, and J. Crowley. CAVIAR test case scenarios. URL <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>.
- [7] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [8] A. Galata, N. Johnson, and D. Hogg. Learning Variable-Length Markov Models of behavior. *Computer Vision and Image Understanding*, 81:398–413, 2001.

- [9] A. Gueziec and N. Ayache. Smoothing and matching of 3D space curves. *International Journal of Computer Vision*, 12:79–104, 1994.
- [10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 1st edition, 2001.
- [11] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:1450–1464, 2006.
- [12] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14:609–615, 1996.
- [13] E. Keogh, X. Xi, L. Wei, and C. A Ratanamahatana. The UCR time series classification/clustering homepage, 2006. URL [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/).
- [14] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. Systems, Man, and Cybernetics - Part B*, 35:397–408, 2005.
- [15] M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003.
- [16] A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12:227–238, 2006.
- [17] F. Porikli. Trajectory distance metric using Hidden Markov Model based representation. In *Proc. ECCV PETS Workshop*, 2004.
- [18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [19] E. Sahouria and A. Zakhor. Motion indexing of video. In *Proc. IEEE ICIP*, volume 2, pages 526–529, 1997.
- [20] R. R. Sillito and R. B. Fisher. Semi-supervised learning for anomalous trajectory detection. In *Proc. BMVC*, pages 1035–1044, 2008.
- [21] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:747–757, 2000.
- [22] G. T. Toussaint. The Relative Neighbourhood Graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.
- [23] X. Wang and E. Grimson. Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. In *Proc. CVPR*, 2008.
- [24] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
- [25] D. A. Zighed, S. Lallich, and F. Muhlenbach. A statistical approach to class separability. *Applied Stochastic Models in Business and Industry*, 21:187–197, 2005.