Modelling Spatio-Temporal Trajectories and Face Signatures on Partially Recurrent Neural Networks

Alexandra Psarrou†, Shaogang Gong‡, Hilary Buxton§
† *School of Computer Science, University of Westminster, London*
‡ *Department of Computer Science, Queen Mary and Westfield College, London*
§ *School of Cognitive and Computer Sciences, University of Sussex, Sussex*

ABSTRACT

We address the problem of trajectory prediction in machine vision applications using variants of Elman's partially recurrent networks. We use dynamic context to constrain the representation learnt by a network and explore the characteristics of various input representations. Network stability and generalisation from training on complex 2D trajectories are tested. We train such networks to encode knowledge about "trajectories" in dynamic face recognition using an extended "temporal signature" eigenface representation of face image sequences. Eigenvector decomposition on each time step of a motion sequence allows for natural variations in view and scale. This application makes use of on-line head detection and face tracking from image sequences and achieves a high success rate when tested on sequences of known and unknown individuals with large viewpoint differences.

1. Introduction

Trajectory prediction is an important capability for many computer vision applications, e.g. visual surveillance [3] or biomedical sequence understanding [11]. Multi-layer perceptrons with supervised learning are very popular for applications which can use static representations, but time is important in many domains, e.g. vision, speech and motor control. Dynamic neural networks can be constructed by adding recurrent connections to form a contextual memory for prediction in time [2, 8]. In learning to predict a trajectory with a recurrent net, it is important to have the input representation reflect the geometric and topological features of that trajectory. This partly determines the effectiveness of learning and the network's ability to generalise and predict. Techniques for learning finite state machines using partially recurrent networks were explored by Elman and Cleeremans [1, 2] and it was shown that such vector coding of trajectories allows generalisation across a range of positions, sizes and speeds in picking up the essential state changes [5, 12]. A topological invariant representation of a shape can be measured by chord length distribution along the boundary of a shape [15]. We use an error measure in restoring such a topological distribution to estimate the effectiveness of representation schemes in trajectory prediction. One of the main problems in face recognition is dimensionality reduction to remove redundant information in the original images. A well known example is the "eigenface" approach [16] which is widely acknowledged for its potential in practical applications. However, the need for representations at a range of scales and orientations causes extra complexity and updating the representations with new data can be a problem [9, 10, 13]. Here we address the problem of scale and orientation by exploiting motion trajectory knowledge to form extended "temporal signature" eigenfaces for face recognition. This encodes the natural variations in orientation and scale [4]. In what follows, we first present, in section 2, the architecture of augmented Elman recurrent neural networks and in section 3, we explore the use of context exponential memory and the effect of having different data representation in predicting complex trajectories. Then, in section 4, we study recurrent network based knowledge representation in predicting spatio-temporal trajectories of moving faces before we conclude in section 5.

2. Elman Recurrent Neural Networks

Recurrent neural networks have both feedforward and feedback connections. Here we consider only partially recurrent networks in which the majority of connections are feedforward and adaptable with a few selected fixed feedback connections to a set of "context" units. Several architectures have been suggested [2, 6, 8] which have in common this use of a set of context units to receive the feedback signals and act as memory for the recent past required in dynamic tasks.

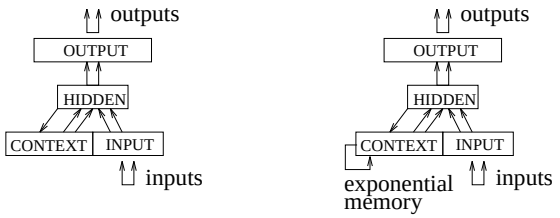

Fig. 1: *The basic Elman network and a modified network with exponential memory.*

Elman [2] suggested the simple architecture that we have modified for the trajectory prediction tasks here (see Fig. 1). We can see that the network consists of (1) four sets of units: the input, hidden, output and context units, (2) a set of feedforward

connections and (3) a set of fixed feedback connections from the hidden to the context units. The context units hold a copy of the activations of the hidden units from the previous time step and thus help to remember the past internal state. At the same time, the hidden units "encode" input patterns so that the layer interconnections build an internal representation of the relationship between successive inputs in a time series.

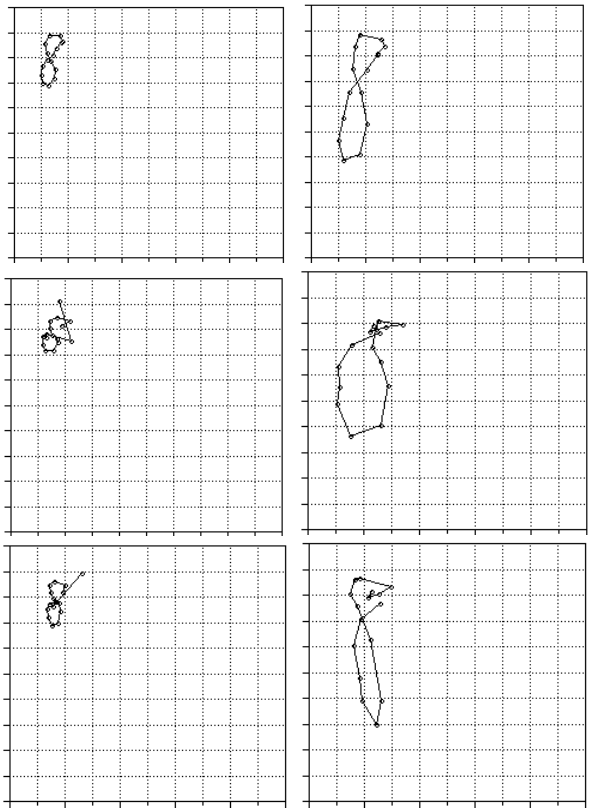

Fig. 2: *Complex trajectories with generalised "8" shape (top row). Predictions of these trajectories by a network without (middle row) and with (bottom row) exponential memory.*

The extra connections in recurrent networks develop internal representations that are sensitive to temporal context i.e. they provide a dynamic memory. However, due to the increased number of weights and more complex dynamics, there is generally a problem with the stability of networks of this kind. For complex trajectories where the next input depends not only on the previous time step but also ones earlier in the series, an augmented context layer can be used. This may explicitly save internal states in further layers or (see the right hand side of Fig. 1), use an exponential memory decay where each context unit saves a bit of the past internal state using an additive function. Thus, the main component of the context unit activation is from the previous time $t-1$ as usual but a secondary component is due to $t-2$, a third due to $t-3$ and so on until the effect is negligible (depending on the decay constant). The exponential encoding of the hidden unit activation is formulated according to the equation $c(t) = (1-\alpha)\, h(t) + \alpha\, c(t-1)$, where the decay constant α lies in the interval $[1, -1]$, $h(t)$ represents the vector of the hidden units activation values at time t and, $c(t)$ represents the context vector at time t.

3. Prediction of Complex Trajectories

Based on Elman's partially recurrent architecture [2], variations were compared for predicting simple circular trajectories [5, 12]. The top row in Fig. 2 shows a couple of examples of complex trajectories we used here. The main difference between these and the circular trajectories is that the curvature of the trajectories is not constant but instead varies along their length. Predicting such trajectories from given starting points depends on the spatio-temporal position of those points on the trajectory. Hand drawing the trajectories introduced noise and shape variation that resulted in the creation of a more realistic, and statistically sound, data set. Each trajectory was drawn inside a grid in a clockwise order starting from a similar relative position but could be centered around any position inside the grid and vary in orientation and size. The generation order of each point of the trajectories was recorded to create the set of time sequences required for the experiments. Sampling produced a set of trajectories of length 16. From a set of 100 trajectories, 70 were used to train the networks and the remaining 30 were used for testing the generalisation ability of the networks. The objective of our experiment is to determine the appropriate representation and possible spatio-temporal invariant of the trajectories.

First, we train an Elman network (2 input, 8 context, 8 hidden, 2 output) with the coordinate (x, y) representation of trajectories. The network was trained by back-propagation with the learning rate of 0.2 and the momentum of 0.5. No conventional error limit was set but instead, the training was terminated according to the network's ability to restore the topological shape of the trajectories (since they are closed). The network converged in 1500 epochs, but achieved the best generalisation after 600 - 1000 epochs. The results given by the middle row of Fig. 2 show that the network was able to predict the evolution of the trajectories independent of their position and scale using a single starting position. However, it gives poor performance where a trajectory has high curvature. Now, a network with exponential memory was trained with exponential parameter $\alpha = 0.5$. The network converged faster and its generalisation ability was increased. Results given by the bottom row in Fig. 2 show that the network also improves its prediction at locations of high curvature along trajectories compared with those predicted by the network without the exponential memory.

In the next experiment, curvature was directly used to represent trajectories. The angle between three consecutive sample points on the trajectory was computed and normalised between [0.1, 0.9]. The one-step-ahead prediction of the trajectories is being performed after two consecutive starting points were given. An Elman network (1 input, 6 hidden, 6 context and 1 output) with exponential parameter $\alpha = 0.5$ gave the best results. The network was trained with the learning rate of 0.1 and momentum of 0.5. The network did not converge but the error was momentarily low. The results are

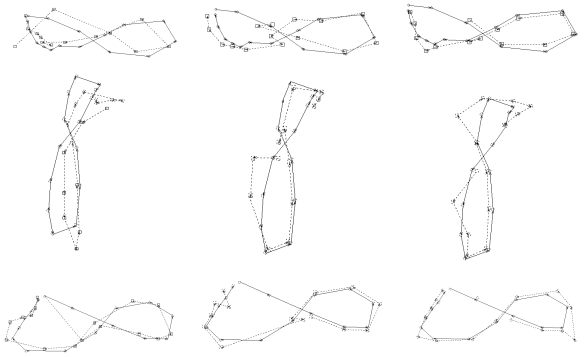

Fig. 3: *The input (solid contours) and predicted (dotted contours) trajectories represented by coordinate (left column), continuous (middle column) and quantised (right column) curvature.*

given by the middle column of Fig. 3. The network was able to give most of its predictions within 45 degrees of accuracy. However, in some cases the prediction varied between 135 to 270 degrees.

In our last experiment, we represented the curvature along a trajectory as a finite state machine coded with vectors 0 to 7 in a binary format. The states correspond to locations of sharp curvature change. The symbols correspond to a set of allowed curvature changes between two successive sample points on a trajectory. The advantage of this representation is that it is independent of the position and scale of the trajectory. An Elman network architecture with 3 input, 10 context, 10 hidden and 3 output units was trained with learning rate of 0.001, momentum of 0.5 and exponential parameter α set to 0.3. After a training session of 140000 epochs the network was able to give a good qualitative prediction of the evolution of the trajectory. The results are given by the right column of Fig. 3. In most cases when the network failed to predict the next curvature measurement it did so by 45 to 90 degrees. However, the network was able to recover the next curvature measure and adapt to the different shapes of each trajectory. The advantages of such representation becomes more apparent when trajectories of different lengths are used.

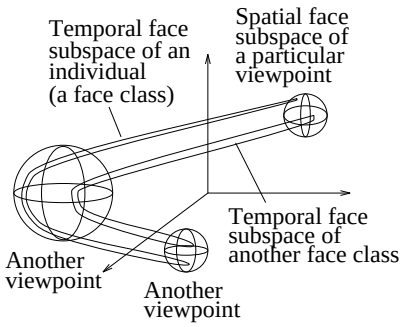

Fig. 4: *A face space can be divided into many sub-spaces according to either viewpoint, or face class (faces of different people) or time.*

4. Prediction in Face Sequences

In the second part of this work, we exploit Elman networks' capacity for predicting spatio-temporal trajectories in a rather more difficult context, registering and recognising dynamic faces. It is important to notice however that the notion of "spatio-temporal trajectories" here is somehow different from what we have used so far. It is more implicit and implies rather the spatio-temporal contextual constraint under which a dynamic face can move. This is illustrated in Fig. 5.

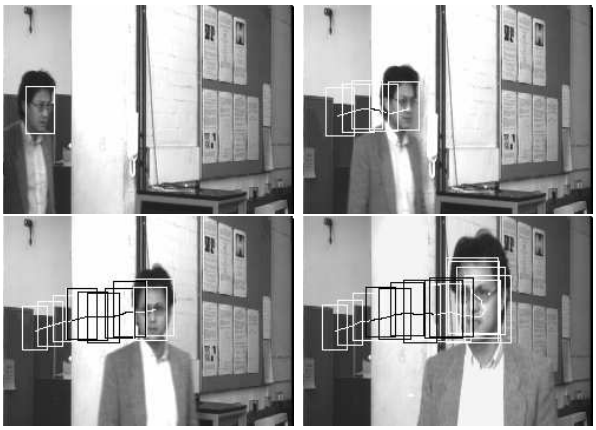

Fig. 5: *Effective face recognition should be carried out within a given context because environmental layout and physical freedom in human head movement not only limit but also correlate changes in face images of a moving face.*

Past studies on applying neural networks to recognise face images [7] have shown to be computationally impractical. This is largely due to direct use of images as network input patterns which resulted in huge networks that are difficult to train. Eigenface representation vastly reduces the dimension of input patterns. With eigenface approach, a face image is represented by a weighted linear sum of a set of orthogonal *eigenfaces*. These eigenfaces are grey-level scaled eigenvectors which are the principal components of a given face image set, the "face space" (illustrated in Fig. 4). An eigenface characterises one distinctive global probabilistic variation in the given face image set. In general, for a set of face images that have an unified size $N = m \times n$ and (m, n) are the width and height of an image, a N-dimensional face space can be defined where face images are points in this hyperspace. Let a set of face images be $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \ldots, \mathbf{\Gamma}_M$, then an average image can be computed as:

$$\mathbf{\Psi} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{\Gamma}_i \qquad (1)$$

and a new set $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_M$ is given by $\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi}$. This simply translates the original face images by $\mathbf{\Psi}$ in the face space [1]. Then the principal

[1] For calculating eigenvectors, $\mathbf{\Phi}_i$ are computationally more stable since their values are much smaller compared with those of $\mathbf{\Gamma}_i$.

Fig. 6: *Face detection and segmentation on a camera input sequence.*

components of the new face space given by $\mathbf{\Phi}_i$ are the eigenvectors of the following covariance matrix:

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Phi}_i \mathbf{\Phi}_i^T = \mathbf{A}\mathbf{A}^T$$

where $\mathbf{A} = [\mathbf{\Phi}_1 \mathbf{\Phi}_2 \dots \mathbf{\Phi}_M]$ and $\mathbf{C}$ are $N \times N$ matrices.

For a typical image size [2], computing the N eigenvectors of $\mathbf{C}$ is computationally hard. However, since the number of the face images in the set is much smaller than the dimension of the face space ($M \ll N$), there only exists $M-1$ nontrivial eigenvectors with the remaining ones associated with negative or zero eigenvalues. Now, if we consider $\mathbf{V}_i$ to be the eigenvectors of matrix $\mathbf{A}^T\mathbf{A}$ whilst $\mathbf{A}^T\mathbf{A}$ is only a $M \times M$ matrix, i.e. $(\mathbf{A}^T\mathbf{A})\mathbf{V}_i = \lambda_i \mathbf{V}_i$ where λ_i are the eigenvalues, then $\mathbf{A}(\mathbf{A}^T\mathbf{A})\mathbf{V}_i = \mathbf{A}(\lambda_i \mathbf{V}_i)$ and $(\mathbf{A}\mathbf{A}^T)(\mathbf{A}\mathbf{V}_i) = \lambda_i(\mathbf{A}\mathbf{V}_i)$ where $\mathbf{A}\mathbf{V}_i$ are the eigenvectors of $\mathbf{C} = \mathbf{A}\mathbf{A}^T$. Therefore, the eigenvectors of $\mathbf{C}$ are given by:

$$\mathbf{U}_i = \mathbf{A}\mathbf{V}_i = [\mathbf{\Phi}_1 \dots \mathbf{\Phi}_k \dots \mathbf{\Phi}_M] \begin{bmatrix} v_1^i \\ \vdots \\ v_k^i \\ \vdots \\ v_M^i \end{bmatrix} = \sum_{k=1}^{M} v_k^i \mathbf{\Phi}_k$$

where $(i = 1, \dots, M-1)$ and v_k^i is the kth element of $\mathbf{V}_i$. The eigenfaces are just the grey-level scaled and shifted eigenvectors which fall into the band of pixel values.

For a given set of eigenfaces $\mathbf{U}_k$, a face image $\mathbf{\Gamma}$ can be projected onto the eigenfaces by:

$$\omega_k = \frac{\mathbf{U}_k^T(\mathbf{\Gamma} - \mathbf{\Psi})}{\lambda_k} \qquad k = 1, \dots, M' \qquad (2)$$

where $\mathbf{\Psi}$ is the average face image given by Eq. (1), λ_k are the eigenvalues and $M' \leq M - 1$ [3]. Now, we have a weight distribution vector $\mathbf{\Omega} = [\omega_1 \, \omega_2 \, \dots \, \omega_{M'}]$, known as the *pattern vector* of $\mathbf{\Gamma}$. A face image can therefore be represented by its pattern vector and the first M' eigenfaces of the face space. If we regard the eigenfaces as the basis of the face space, then $\omega_k \in (-1, 0) \cup (0, 1)$. Now, for all the face images in the given set that belong to the same face class, one computes an average pattern vector $\overline{\mathbf{\Omega}}$. This can be regarded as the "signature" of that face class. For a new face image, one calculates the Euclidian distance ε between its pattern vector and the $\overline{\mathbf{\Omega}}$ of a known face class $\varepsilon = \| \mathbf{\Omega} - \overline{\mathbf{\Omega}} \|$. Then, this face image can be identified with the known face class if ε falls within a given threshold.

[2] For an image size of 256×256, the dimension of the face space N is 65536.

[3] With $M' < M - 1$, the representation is approximate.

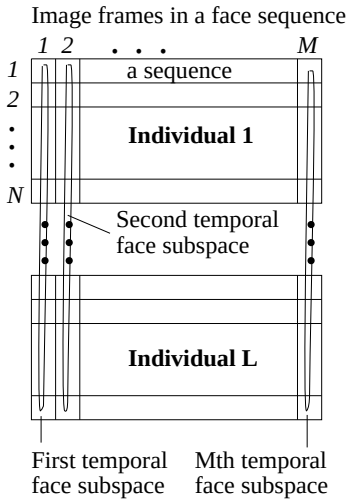

Fig. 7: *Temporal face subspaces are associated with image frames of specific time indices in face sequences. Note, M is the number of frames in a face sequence.*

Computationally, however, this approach is scale and viewpoint dependent. Current eigenface based face recognition models are limited to register only single face images all taken from a very similar viewpoint, most commonly, the frontal view. This substantially limits a model's robustness and effectiveness [14]. A common approach to overcome the scale dependency problem is to normalise (i.e. unifying size) face images. With our system, we first detect and track a moving head from an on-line camera input before segmenting and normalising the face images using our on-line face detection system [4] (see Fig. 6). This process also improves the accuracy in encoding the face images. If large areas in the images that are used to form a face space contain "significant" background rather than face information, the extracted eigenfaces would "say" more about the statistical properties of the background than that of the faces.

The focus of this work is about encoding view-invariant face features and recognising face images from large viewpoint differences. Temporal information in time sequence provides important constraints in data registration and association. Environmental layout and physical freedom in human head movement limits possible changes of a moving face. Dividing face space in time ties this implicit contextual constraints to temporal correlations (possible invariants) between successive face images in a sequence. If we capture a set of image sequences of moving faces where all the sequences start from and finish at the same "two distinctive" regions in a given context, each image frame from every sequence contributes to one "similar orientation" in time. Then we can divide face space into a

set of subspaces that correspond to groups of face orientations associated with time index. We call these groups "temporal face subspaces" (illustrated in Fig. 7).

To extract the temporal signature of a face class we do the following: First, we track and segment a set of face sequences with a fixed number of frames (M) taken from the head movement of one person. We then compute M temporal face subspaces for that individual with each subspace corresponding to one time frame (see Fig. 7). The ith temporal face subspace is represented by the few significant eigenfaces of the ith image frames from the sequence set. The ith image frame of a given face sequence can then be represented by a pattern vector given by Eq. (2). In the face sequence, we can then measure the temporal change in the pattern vectors of successive frames by their temporal-Euclidian-distance:

$$\varepsilon = \| \mathbf{\Omega}_t - \mathbf{\Omega}_{t+1} \| \tag{3}$$

where $t = 1, 2, \ldots, M' - 1$. We regard this information relevant to a "temporal signature" of a face class in a given spatio-temporal context.

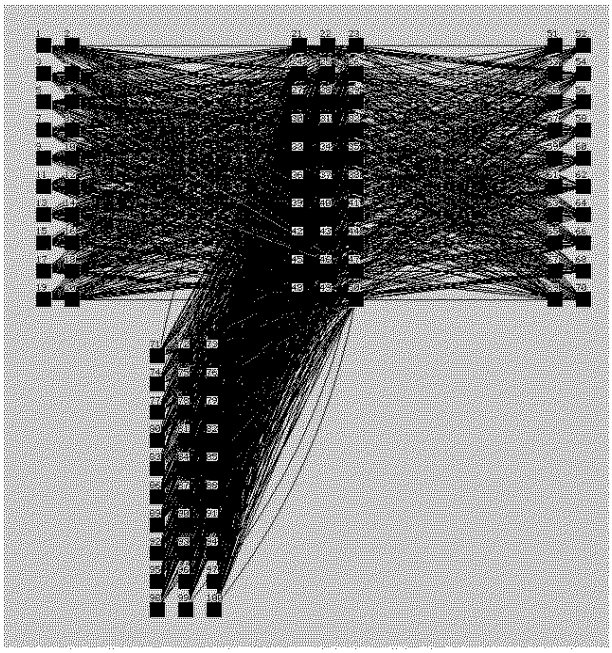

Fig. 8: *An Elman recurrent neural net for learning temporal signatures.*

Second, we train a set of Elman networks to learn possible temporal signatures of a set of face classes (see Fig. 8). Each Elman net is trained to learn any temporal signatures of one face class and it has 20 inputs and 20 outputs. The input and output patterns are the face image pattern vectors and each temporal face subspace is represented by the first 20 eigenfaces.

In the following, we describe one of our experiments in order to highlight the computational procedures involved. In this experiment, we take 30 face sequences of 5 image frames for each of 3 different individuals, "John", "Pascal" and "Katerina". We represent each image by a pattern vector for the first 20 eigenfaces and train 3 Elman networks (20 inputs, 30 hidden, 30 context and 20 outputs). The networks were trained after 6000 epochs and were used to test the the following task: "Recognising John". We use Model-John, Model-Pascal and Model-Katerina to refer to the Elman nets for the face classes. We then take a new face sequence of John (Sequence-John) and compute 3 pattern vector sequences by projecting the frames of Sequence-John to the 3 face class subspaces represented by the three models. Each pattern vector sequence is then applied to the Elman network associated with the corresponding subspace and the Euclidian distance between successive outputs (predictions) of the Elman network is computed as shown in Fig. 9.

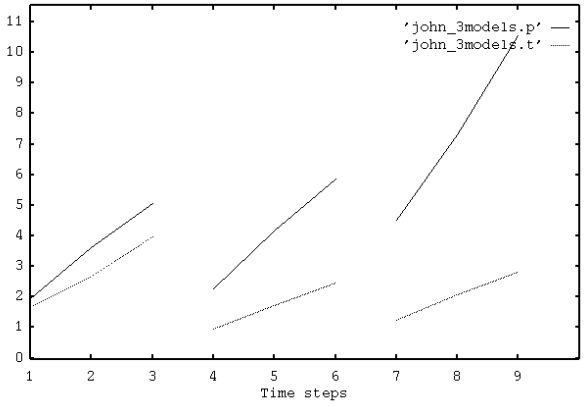

Fig. 9: *The temporal-Euclidian-distance between successive pattern vectors that belong to Sequence-John and the temporal-Euclidian-distance between the predictions of Model-John, Model-Pascal and Model-Katerina respectively.*

It is clear that the temporal-Euclidian-distance of Sequence-John is "much closer" to the temporal-Euclidian-distance predicted by Model-John. In other words, the temporal gradients of the pair of temporal-Euclidian-distance lines on the left hand side of Fig. 9, which are respectively for Sequence-John (ε_s) and Model-John (ε_m), have similar values, i.e.

$$\frac{\partial \varepsilon_s}{\partial t} \simeq \frac{\partial \varepsilon_m}{\partial t}$$

This suggests that the rate of temporal change in the pattern vectors of face sequences in a given context could be used as a measure of temporal signature for a dynamic face. Similar results were obtained from 5 other test sequences of different face classes.

5. Conclusion

In the first part of this paper we exploit different data representations for predicting trajectories with generalised "8" shape using partially recurrent networks. Fig. 3 gives comparative results from coordinate, continuous and quantised curvature representations. The experiments show that although a network trained with coordinate representation can generalise well in predicting trajectories of random position and size, it can not predict accurately sharp changes in curvature along a trajectory. Better results are obtained by using either

the continuous or quantised curvature representation. Training an Elman network with the quantised curvature representation was able (1) to generalise well by recovering fast from sharp curvature change, (2) to provide a good qualitative prediction and overall (3) to outperform a network trained with continuous representation. However, this may be the result of the coarse representation of the trajectories we used here. Further experiments using trajectories of different length and more frequent sampling are required to establish a better quantitative understanding of the difference between these two schemes. The second part of this paper addresses the scale and orientation problem in the eigenface approach to face recognition. We studied a novel approach that exploits temporal correlation and invariance among face images in order to recognise face appearances of large viewpoint difference. With our preliminary experiment, we illustrated how to use contextual constraints in data registration and association in face recognition. Instead of recognising a single face "snapshot", a temporal sequence of a moving face is used to selectively cluster face space according to time and space and register a "dynamic" face where possible temporal invariance of a moving face in a given spatio-temporal context can be learnt and extracted as a form of "temporal signature" in recognition. More extensive and systematic experiments will be undertaken shortly to provide a quantitative measure of this approach.

References

[1] A. Cleeremans. "Finite state automata and simple recurrent networks". *Neural Computation*, 1, 1989.

[2] J. Elman. "Finding structure in time". *Cognitive Science*, 14, 1990.

[3] S. Gong and H. Buxton. "Advanced visual surveillance using Bayesian nets". In *IEEE Workshop on Context-Based Vision*, Cambridge, MA., 1995.

[4] S. Gong, A. Psarrou, I. Katsoulis, and P. Palavouzis. "Head tracking and dynamic face recognition". In *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, Hamburg, Germany, 1994.

[5] Y. Ho. Classification and Prediction of Motion Trajectories using ANNs. Master's thesis, School of Cognitive and Computing Sciences, University of Sussex, 1994.

[6] M. Jordan. "Serial Order: A Parallel Distributed Processing Approach". In *Advances in Connectionist Theory*. Erlbaum, 1989.

[7] T. Kohonen. *Self-organization and Associative Memory*. Springer-Verlag, Berlin, 1989.

[8] M. Mozer. "Neural Net Architectures for Temporal Sequence Processing". In A. Weigend and N. Gershenfeld, editors, *Time Series Prediction: Predicting the Future and Understanding the Past*. Addison-Wesley, 1993.

[9] A. Pentland, B. Moghaddam, and T. Starner. "View-based and modular eigenspaces for face recognition". In *Computer Vision and Pattern Recognition*, 1994.

[10] N. Petkov, P. Kruizinga, and T. Lourens. "Biologically motivated approach to face recognition". In *International Workshop on Artifical Neural Networks*, Sitges, Spain, 1993.

[11] A. Psarrou and H. Buxton. "Hybrid architecture for understanding motion sequences". *Neurocomputing*, 5, 1993.

[12] A. Psarrou and H. Buxton. "Motion analysis with recurrent neural nets". In *International Conference on Artificial Neural Networks*, Sorrento, Italy, 1994.

[13] R. Rao and D. Ballard. "Natural basis functions and topographic memory for face recognition". In *International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.

[14] G. Robertson and I. Craw. "Testing face recognition systems". In *British Machine Vision Conference*, Guildford, England, 1993.

[15] C. Taylor and D. Cooper. "Shape verification using belief updating". In *British Machine Vision Conference*, Oxford, England, 1990.

[16] M. Turk and A. Pentland. "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, 3(1), 1991.