

Shift-invariant Dynamic Texture Recognition

Franco Woolfe¹ and Andrew Fitzgibbon²

¹ Yale University, Newhaven, CT
franco.woolfe@yale.edu
<http://www.yale.edu>

² Microsoft Research, Cambridge, UK
awf@microsoft.com

<http://www.research.microsoft.com/~awf>

Abstract. We address the problem of recognition of natural motions such as water, smoke and wind-blown vegetation. Such dynamic scenes exhibit characteristic stochastic motions, and we ask whether the scene contents can be recognized using motion information alone. Previous work on this problem has considered only the case where the texture samples have sufficient overlap to allow registration, so that the visual content of the scene is very similar between examples. In this paper we investigate the recognition of entirely non-overlapping views of the same underlying motion, specifically excluding appearance-based cues.

We describe the scenes with time-series models—specifically multivariate autoregressive (AR) models—so the recognition problem becomes one of measuring distances between AR models. We show that existing techniques, when applied to non-overlapping sequences, have significantly lower performance than on static-camera data. We propose several new schemes, and show that some outperform the existing methods.

1 Recognition from motion

Motion is a powerful cue for visual recognition of scenes and objects. Johansson’s moving dot displays [1] show that objects which are highly ambiguous from a single view are readily recovered once motion is supplied. In computer vision, the classification of scenes from motion information has seen considerable research, summarized in the recent survey of Chetverikov and Péteri [2]. In this paper, we focus on classification of objects using the class of state-space *dynamic texture* models introduced by Doretto and Soatto [3, 4] and Fitzgibbon [5].

Dynamic textures are image sequences of moving scenes which exhibit characteristic stochastic motion. Examples include natural scenes such as water, wind-blown flowers and fire. State-space models [5, 4] view a dynamic texture as a realization of a time-series model such as an autoregressive process. By determining the model parameters for such sequences, we can hope to recognize similar motions by comparing the models represented by the parameters. Our goal in this paper is to define a distance measure between pairs of image sequences which is low for models representing the same motion (or motion class), and high for models derived from motions of different classes. Such distance

measures can be used in kernel-based or nearest-neighbour classifiers; and as the basis of clustering algorithms for the unsupervised learning of dynamic texture classes. Some of the distance measures we propose are based on feature vectors extracted from the state-space models, and are thus also suitable for density estimation or regression.

Two important new aspects of our work are that we require shift invariance, and that we want to investigate recognition using motion alone, for reasons we now explain. **Shift invariance.** Previous authors [6, 7] have investigated only the case where the temporal sequences are captured by a camera at a single viewpoint, so that the same area of the scene is viewed in (part of) each sequence. In some cases [5, 8] the camera is panning across the scene, or the textures compared are in overlapping tiles [9], but there remains the constraint of overlap between the textures. However, in order to separate the appearance and dynamic components of recognition we compare images of the scene where there is no spatial overlap between the example dynamic textures. Recognition rates for this configuration are much lower than for the single-viewpoint case, but are significantly higher than either baseline methods or chance, and thus confirm that motion can provide a useful cue for recognition.

Recognition from motion alone. As noted by Chan and Vasconcelos [7], much of the recognition performance on typical test data may be attributed to appearance cues. Thus comparisons between the recognition schemes conflate appearance and motion, and this conflation is of a form that is hard to disentangle. Furthermore, the appearance component of these schemes is not representative of the current state of the art in appearance-based recognition, being based essentially on a principal components analysis of the image sequence. Thus a practical scheme for recognition including motion should combine a state-of-the-art appearance-based scheme and the best possible motion-based scheme. By considering motion-only schemes, we hope to allow this selection to be more carefully performed, and to allow the balance between motion and appearance to depend on the training set for any given real-world system.

The remainder of the paper is structured as follows: a discussion of the state of the art also serves to introduce the DT model and the notation of the paper. We then discuss the construction of motion-based distance measures between such models, and introduce some novel measures. We conduct experiments comparing these and existing distance measures in section 4, and conclude with a discussion of the relative merits of the various models.

2 Background

General-purpose automated recognition of motions in video sequences may be attributed to Polana and Nelson, who considered two classes: stochastic motions and “activities”. For activities they considered periodicity measures on edges in xyt slices [10]. Subsequent research on activity recognition has been considerable, using optical flow [11], features in the spatiotemporal volume [12, 13], spatiotemporal correlation [14], parametrized models [15] and exemplars [16]. In



Fig. 1. Single frames from the database sequences. Although many of the sequences are easily distinguished using colour information, the goal of this paper is to explore how well they can be distinguished using motion information alone.

addition, models of videotextures [17] may be considered to be related to activity models. These perform well for regular motions, but are less well suited to stochastic motions of the types we consider.

Stochastic models of temporal texture may be divided into local and global: local models include Polana and Nelson’s co-occurrence statistics of optical flow vectors [18]; and the spatiotemporal autoregressive models of Szummer and Picard [19], which model stochastic regularity by expressing each pixel of the sequence as a linear combination of its spatial and temporal neighbours. By fitting the model to an example sequence, and assuming the AR model parameters are constant over the sequence, each temporal texture is represented by a small number of model parameters. Comparison of such parameters may be achieved using the methods reviewed in the current paper. Fablet and Bouthemy [20] first quantize certain motion-related per-pixel measurements, and then model the spatiotemporal cooccurrence of the quantized labels as a Gibbs distribution. A model is learned for each class to be recognized and recognition proceeds by measuring the likelihood of the labels of a novel sequence under each class model. These local models allow robust classification, but strongly bind together the appearance and motion of the texture, limiting their applicability to textures which are both spatially and temporally stationary; yet offering limited shift and viewpoint invariance.

State-space models [3, 5] on the other hand, model the image sequence more globally, and have been used for recognition [6], image segmentation [21, 9, 8], image registration [5] and videotexture synthesis [4, 5]. The core of such models is a spatiotemporal autoregressive (AR) model, and recognition depends on computing the similarity of pairs of AR models. Saisan *et al.* [6] propose the Martin distance between AR model parameters, and Chan and Vasconcelos [7] measure the Kullback-Leibler (KL) divergence between the realization distributions defined by the models. In both of these previous cases however, the sequence

appearance plays an important role in the recognition performance, and indeed, as we show, swamps the motion-based results.

2.1 The state-space model

Dynamic texture models [4, 5] represent the image using a state-space model. Images are represented by column vectors \mathbf{y} . A sequence of T images is the matrix $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$. Under the state-space model of such a sequence, each \mathbf{y}_t is assumed to be a linear projection of a low-dimensional state vector $\mathbf{x}_t \in \mathbb{R}^N$, with typical values of N in the range 5 to 35. The observed \mathbf{y} are corrupted with zero-mean Gaussian noise with covariance matrix \mathbf{R} , yielding

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (1)$$

The matrix \mathbf{C} is sometimes termed the *output matrix*. The temporal evolution of \mathbf{x}_t is modelled by the first-order time-series, or autoregressive (AR) model,

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (2)$$

where \mathbf{A} is the $N \times N$ *state matrix*, and \mathbf{Q} is the $N \times N$ *driving noise covariance matrix*. The model from which a given sequence is drawn will be represented by its parameters $\theta = (\mathbf{C}, \mathbf{A}, \mathbf{Q})$, where \mathbf{C} models the sequence *appearance* and \mathbf{A} and \mathbf{Q} its *motion*. A sequence such as \mathbf{Y} which is generated from the model is called a *realization* of the model. Figure 2 shows some example trajectories.

2.2 Fitting the model

Given an example sequence \mathbf{Y} , we would like to estimate the parameters $\theta = (\mathbf{C}, \mathbf{A}, \mathbf{Q})$ of the model of which it is a realization. We adopt the approach of [4, 5], described here for completeness.

We ensure that input sequences have zero mean $\sum_t \mathbf{y}_t = \mathbf{0}$, by subtracting the mean from each frame. The matrix \mathbf{C} is determined via principal components analysis of the sequence, i.e. assuming \mathbf{Y} is much taller than it is wide, compute the eigendecomposition $\mathbf{Y}^\top \mathbf{Y} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$ and set $\mathbf{C} = \mathbf{Y}\mathbf{V}\mathbf{D}^{-\frac{1}{2}}$. From this we may immediately obtain estimates of the state vectors

$$\mathbf{x}_t = \mathbf{C}^\top \mathbf{y}_t, \quad t = 1 \dots T \quad (3)$$

The state matrix \mathbf{A} is found as the minimizer

$$\mathbf{A} = \operatorname{argmin} \sum_{t=2}^T \|\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}\|^2 \quad (4)$$

which is easily computed. Finally the driving covariance \mathbf{Q} is approximated as the sample covariance of the residuals $\mathbf{r}_t = \mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}$, given by

$$\mathbf{Q} = \frac{1}{T-1} \sum_{t=2}^T \mathbf{r}_t \mathbf{r}_t^\top \quad (5)$$

As we are interested in recognition schemes which depend only on motion, and not appearance, we shall not be required to estimate \mathbf{R} .

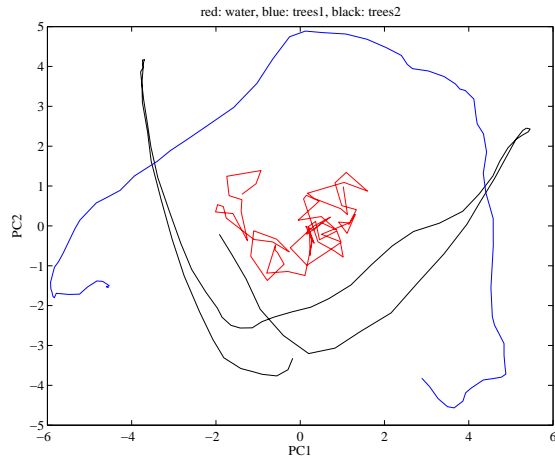


Fig. 2. Example 2D state-space trajectories $\mathbf{x}_{1..T} \subset \mathbb{R}^2$ for three example sequences. Red: water flowing over stone; Black and Blue: tree blowing in the wind. We characterize the sequences using auto-regressive models, and wish to compare the model parameters to identify similar models. Any distance metric must be invariant to changes of basis in the state space (see §2.3).

2.3 Comparing state-space models

For recognition, we will need to determine whether two models $\theta = (\mathbf{C}, \mathbf{A}, \mathbf{Q})$ and $\theta' = (\mathbf{C}', \mathbf{A}', \mathbf{Q}')$ represent the same dynamic texture. It is not sufficient to check for equality of the parameters, because a given sequence may be generated by an equivalence class of models [3]. Specifically, for any invertible $N \times N$ matrix \mathbf{M} the model

$$(\mathbf{C}\mathbf{M}^{-1}, \mathbf{M}\mathbf{A}\mathbf{M}^{-1}, \mathbf{M}\mathbf{Q}\mathbf{M}^{\top}) \quad (6)$$

generates image sequences drawn from the same distribution as $(\mathbf{C}, \mathbf{A}, \mathbf{Q})$. Thus any metric for comparing AR models must be invariant to this class of transformations of the model parameters. In this paper we explore three classes of distance measure which (sometimes approximately) obey this property: measures of divergence between the distributions of the model realizations [7], spectral methods [6], and techniques which operate directly on invariant functions of the AR model parameters. Each of these will now be discussed.

2.4 Time-series spectrum

Several of the distance measures previously proposed in the literature, as well as those we introduce, may be expressed in terms of the Fourier transform of the autocovariance of the time series, or its spectral density [22, Ch3]. For an infinite time-series $(\mathbf{C}, \mathbf{A}, \mathbf{Q})$, the spectral density matrix is a matrix function of

frequency ν , $\mathbf{F}(\nu) \in \mathbb{C}^{N \times N}$ defined as

$$\mathbf{F}(\nu) = (\mathbf{I} - \mathbf{A}e^{-2\pi i\nu})^{-1}\mathbf{Q}(\mathbf{I} - \mathbf{A}e^{2\pi i\nu})^{-1}.$$

and for a finite time-series of length T it will suffice to evaluate this on the finite set $\nu_k = k/T$, $k = 0, \dots, T-1$. Thus the spectral density of a length T time-series is a set of T matrices. We refer to this method for estimating the spectral density matrices as the AR method, since it is computed from the auto-regressive model parameters. We may also directly estimate the spectrum $\mathbf{F}(\nu_k)$ using the fast Fourier transform of the raw time-series as follows. Given the $N \times T$ matrix of state values \mathbf{X} , compute the componentwise FFT \mathbf{f}_k (i.e. FFT each component $f(i, \cdot) = \text{fft}(X(i, \cdot))$, and set $\mathbf{f}_k = f(\cdot, k)$.) Then compute the periodogram $\mathbf{G}_k = \mathbf{f}_k \mathbf{f}_k^*$. The spectrum is then given by smoothing \mathbf{G} with a window of size $2H + 1$, yielding $\mathbf{F}(\nu_k) = \sum_{i=k-H}^{k+H} \mathbf{G}_k$. We refer to this as the time-series or TS method, and show that it can give better results than the AR method for appropriate choices of smoothing parameter H .

3 Distance measures between dynamic textures

We are now in a position to define distance measures between dynamic textures. We consider distances of three forms. The first class compares the probability densities over all possible sequences generated by the time-series under comparison. We present a new formulation of the KL metric and introduce the Chernoff distance. The second class of measure is based on a multivariate definition of the Cepstrum. The final class is based on computing a set of features from the fitted AR model parameters.

3.1 Distances between realization distributions

We consider the set of all possible realizations of time-series generated by the AR model $(\mathbf{C}, \mathbf{A}, \mathbf{Q})$. Following [7], it suffices to consider only sequences of a certain length T . This is a probability density over the set of sequences \mathbf{Y} , which we may write as $p(\mathbf{Y})$ or $p(\mathbf{y}_T, \dots, \mathbf{y}_1)$. As the \mathbf{y}_t are linear transformations of \mathbf{x}_t , it is sufficient to characterize the distribution of the \mathbf{x}_t , written $p(\mathbf{X}) = p(\mathbf{x}_T, \dots, \mathbf{x}_1)$. From (2), this is exactly $p(\mathbf{x}_T | \mathbf{x}_{T-1})p(\mathbf{x}_{T-1} | \mathbf{x}_{T-2}) \cdots p(\mathbf{x}_2 | \mathbf{x}_1)p(\mathbf{x}_1)$ where each term in the product is Gaussian, so that the joint distribution is a Gaussian, whose covariance matrix may be computed from the model parameters \mathbf{A} and \mathbf{Q} . Thus any sequence \mathbf{X} drawn from the model is a draw from a Gaussian distribution whose parameters depend only on the model parameters. Comparing two AR models then amounts to comparing two Gaussian distributions, i.e. measuring their *divergence*. We consider two possible definitions: the Kullback-Leibler divergence and the Chernoff distance.

Given two probability distributions over X with pdfs f_1 and f_2 , the Kullback Leibler divergence is

$$I_{KL}(f_1, f_2) = E_1 \left[\frac{f_1(X_1)}{f_2(X_1)} \right]. \quad (7)$$

A generalization of this is the Chernoff distance, given by

$$I_{CH}(f_1, f_2) = -\ln E_2 \left[\left(\frac{f_2(X_1)}{f_1(X_1)} \right)^\alpha \right], \quad (8)$$

where $0 < \alpha < 1$ is a parameter. It was found in experiments that for our task, the success rate did not depend sensitively on α near the middle of the interval $(0, 1)$. Thus we often took $\alpha = 0.5$, yielding Bhattacharya's symmetric divergence.

In order to compute the KL divergence of our dynamic texture model, suppose we have two movies $(C_j, A_j, Q_j)_{j=1,2}$. Compute the spectral densities $F_j(\nu_k)$ as above. From this definition, we can compute the Kullback Leibler distance from (C_1, A_1, Q_1) to (C_2, A_2, Q_2) by [22, p459]

$$I_{KL}(F_1, F_2) = \sum_{0 < \nu_k < 1/2} \left[\text{trace} \{ F_1(\nu_k) F_2^{-1}(\nu_k) \} - \ln \frac{|F_1(\nu_k)|}{|F_2(\nu_k)|} - N \right]. \quad (9)$$

The Chernoff distance may also be expressed in terms of the spectral density as follows [22, p461].

$$I_{CH}(\alpha, F_1, F_2) = \frac{1}{2} \sum_{0 < \nu_k < 1/2} \left[\ln \frac{|\alpha F_1(\nu_k) + (1 - \alpha) F_2(\nu_k)|}{|F_2(\nu_k)|} - \alpha \ln \frac{|F_1(\nu_k)|}{|F_2(\nu_k)|} \right]. \quad (10)$$

Note that these distance measures are not invariant to transformations of the form described in §2.3, so Chan and Vasconcelos suggest resolving the ambiguity by projecting A_2 into the space of A_1 using the appearance matrices C_1, C_2 .

3.2 Distances based on the Cepstrum

The cepstrum of a time series may be thought of as being derived from the frequency domain representation in the same way that this comes from the time domain. Intuitively, peaks in the cepstrum correspond to “echoes” in the signal. The cepstrum coefficients are powerful features for characterizing speech and music signals, so it is of interest to see how they may apply to repetitive video signals. In this section, we give the conventional univariate definition of cepstrum and apply it to dynamic texture recognition via cepstral distance. We suggest three extensions of the cepstrum to the case of multivariate time series.

Univariate case For a general univariate time series (x_t) the cepstrum, written (\hat{x}_t) , is defined as [23] the inverse z-transform of the logarithm of the z-transform of (x_t) . In symbols:

$$(x_t) \rightarrow \sum_{t \in \mathbb{Z}} x_t z^{-t} = X(z) : \text{the z transform} \quad (11)$$

$$\rightarrow \log X(z) \quad (12)$$

$$= \sum_{t \in \mathbb{Z}} \hat{x}_t z^{-t} \quad (13)$$

$$\rightarrow (\hat{x}_t) : \text{the complex cepstrum.} \quad (14)$$

If the time series (x_t) above is drawn from an autoregressive model of order K the above definition gives rise to a characterization of the cepstrum in terms of poles. Assume (x_t) comes from such an $AR(K)$ model, that is:

$$x_t + a_1 x_{t-1} + \dots + a_K x_{t-K} = w_t; w_t \sim N(0, \sigma^2). \quad (15)$$

Define the model's poles as $p_i \in \mathbb{C}$ where the system function is:

$$H(z)^{-1} = 1 + a_1 z^{-1} + \dots + a_K z^{-K} = \prod_{i=1}^K (1 - p_i z^{-1}). \quad (16)$$

The cepstral coefficients \hat{x}_t are then given by

$$\begin{aligned} \hat{x}_t &= 0, \text{ for } t \leq 0 \\ &= \frac{1}{t} \sum_{i=1}^K p_i^t \text{ for } t > 0. \end{aligned} \quad (17)$$

The cepstral distance between two univariate time series (x_t) and (x'_t) , with cepstra (\hat{x}_t) and (\hat{x}'_t) , is then [23]

$$\sum_{t=0}^{\infty} |\hat{x}_t - \hat{x}'_t|^2. \quad (18)$$

Note the similarity to the Martin distance [4, 24] where the weighting of higher degree cepstral coefficients is increased linearly:

$$\sum_{t=0}^{\infty} t |\hat{x}_t - \hat{x}'_t|^2. \quad (19)$$

For practical computation, in our application, the sum may be terminated at about $t = 20$. Performance (i.e. success rate in the classification task of §4, figure 4) rises quickly for $t < 20$ and then plateaus.

Multivariate case There is no consensus definition in the literature of either cepstra or cepstral distance for multivariate time series, to the best of our knowledge. We present three such definitions, which are mutually incompatible, and use them to construct distances for classifying dynamic textures.

Summed univariate distances One simple extension of the univariate definitions is to fit univariate $AR(K)$ models to each component of the series (\mathbf{x}_t) independently. The distance is simply the sum of the per-component distances. Although this ignores correlations between the components, the fact that \mathbf{C} is obtained by projection onto a PCA space will have the effect of somewhat decorrelating the components, and thus this technique can provide good results, as we shall see.



Fig. 3. Crop regions. In order to test the invariance of recognition to shifts of the image, all comparisons are between cropped sub-sequences. This figure indicates the two crops of the test dataset used. Note that the appearance of the tree varies considerably (globally—local texture measures will be similar) between the two regions, so that motion is the main recognition cue, even for schemes which include some appearance modelling.

State matrix eigenvalues This definition is by analogy to (17). The state equation for a dynamic texture is $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{v}_t$; $\mathbf{v}_t \sim N(0, \mathbf{Q})$, that is a multivariate AR(1) process. Let the system function be

$$H(z) = (\mathbf{I} - \mathbf{A}z^{-1})^{-1}, \quad z \in \mathbb{C}. \quad (20)$$

Let the poles be solutions of $|\mathbf{I} - \mathbf{A}p_i^{-1}| = 0$, that is to say eigenvalues of \mathbf{A} . Now define the cepstrum by analogy with (17) as

$$\begin{aligned} \hat{x}_t &= 0, \quad \text{for } t \leq 0 \\ &= \frac{1}{t} \sum_{i=1}^N p_i^t \quad \text{for } t > 0. \end{aligned} \quad (21)$$

Note that the cepstral coefficients of a multivariate time series are scalars, according to this definition. The multivariate cepstral distance is again given here by (18).

Discrete Fourier transform Here we let the cepstrum of a multivariate time series $(x_t)_{t=1}^T$ be the inverse DFT of the logarithm of the DFT of (x_t) :

$$(\hat{x}_t)_{t=1}^T = \text{IDFT}(\ln(\text{DFT}((x_t)_{t=1}^T))). \quad (22)$$

Here the DFT of a sequence of vectors is taken componentwise. Thus, the cepstral coefficients of a multivariate time series are vectors. The cepstral distance is then

$$\sum_{t=1}^n \|\hat{x}_t - \hat{y}_t\|. \quad (23)$$

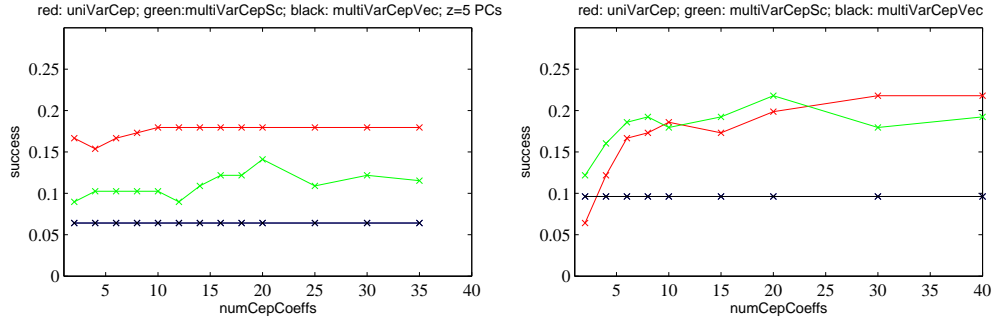


Fig. 4. Performance of cepstral distances for two values of N , the number of principal components. The abscissa is the upper bound on the summations in (18). Left: $N = 5$, Right: $N = 15$. The three distances tested are (red) summed univariate distances, (green) state matrix eigenvalues, (black) DFT. The DFT method is uniformly outperformed by the other two.

3.3 Distances based on feature extraction

In this section we measure discrepancy between dynamic textures by Euclidean distances between feature vectors. A feature vector is some vector function of the sequence parameters $(\mathbf{C}, \mathbf{A}, \mathbf{Q})$ which we hope characterizes a movie.

The choice of feature vectors is subject to two constraints, for the purposes of this paper. Firstly we restrict ourselves only to consider motion. So the state matrix \mathbf{A} and driving noise covariance matrix \mathbf{Q} are both allowed, but we may not examine the output matrix \mathbf{C} or the movie frames \mathbf{y}_t . Secondly, recall that we aim to measure distances between observationally equivalent classes of dynamic textures. Thus any property of \mathbf{A} we examine should be invariant under a change of basis $\mathbf{A} \rightarrow \mathbf{M}^{-1}\mathbf{A}\mathbf{M}$. Similarly, any property of \mathbf{Q} we use should be invariant under $\mathbf{Q} \rightarrow \mathbf{M}^T\mathbf{Q}\mathbf{M}$.

A typical feature vector consists of some eigenvalues of \mathbf{A} and some eigenvalues of \mathbf{Q} . From the above considerations, eigenvalues of \mathbf{A} seem valid choices for feature vectors, and we note that the set of eigenvalues of \mathbf{A} already appears in the definition of multivariate cepstrum above. Eigenvalues of \mathbf{Q} are invariant under $\mathbf{Q} \rightarrow \mathbf{M}^T\mathbf{Q}\mathbf{M}$ when \mathbf{M} is orthogonal, but not otherwise, in general. Nevertheless, experiments suggest there is some information to be gained from the eigenvalues of \mathbf{Q} .

Specifically, denote by α_i the eigenvalues of \mathbf{A} with $|\alpha_i| > |\alpha_{i+1}|$, and denote by σ_i^2 the eigenvalues of \mathbf{Q} , again in descending order. Generate the feature vector $\mathbf{v}(K) = [\alpha_1, \dots, \alpha_K, \sigma_1^2, \dots, \sigma_{N-K}^2]$. Figure 5 shows performance of this feature vector as a function of K .

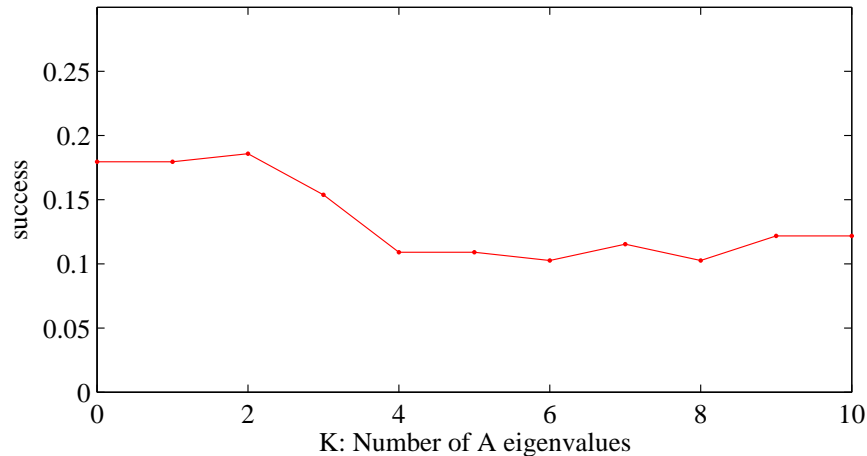


Fig. 5. Feature-based distance. Each sequence is characterized by a feature vector comprising K eigenvalues of \mathbf{Q} and $(N - K)$ eigenvalues of \mathbf{A} .

4 Experimental results

In order to compare the distance measures on experimental data, we tested classification performance on the UCLA test database [6]. The UCLA database comprises 50 sets of four sequences of a dynamic texture scene, for a total of 200 sequences. The movies are 75-frame sequences of size 110×160 , and were converted to grayscale before any computation. In each category, the four movies are captured from the same camera viewpoint, and thus recognition performance is dominated by the sequence appearance. Indeed simply using the mean frame of each sequence, and performing a 1-vs-all classification using a 1-nearest neighbor classifier (described in more detail below) yields a 60% classification rate. Existing dynamic texture recognition algorithms quote performance figures of 90% on this dataset.

In order to more rigorously test the performance of motion-based classification, we have cropped the test data to remove the effects of identical viewpoint. The sequences were cropped into a pair of 48×48 subsequences, denoted “L” and “R” for left and right crop windows (illustrated in figure 3). Comparisons between sequences were only ever performed between different crop locations. From the 51 categories in the UCLA database, we discarded 12 which violated the assumption of spatial stationarity (e.g. “candle”, “fire”, “fountain”, in all of which the “L” cropping viewed stationary background, while the “R” cropping viewed the motion). Retaining these sequences would be expected to yield similar results, but with a reduced success rate on all algorithms. There were thus 39 categories. The introduction of this cropping reduces the performance of state-of-the-art metrics from a quoted 90% to about 15%. Note that this is still

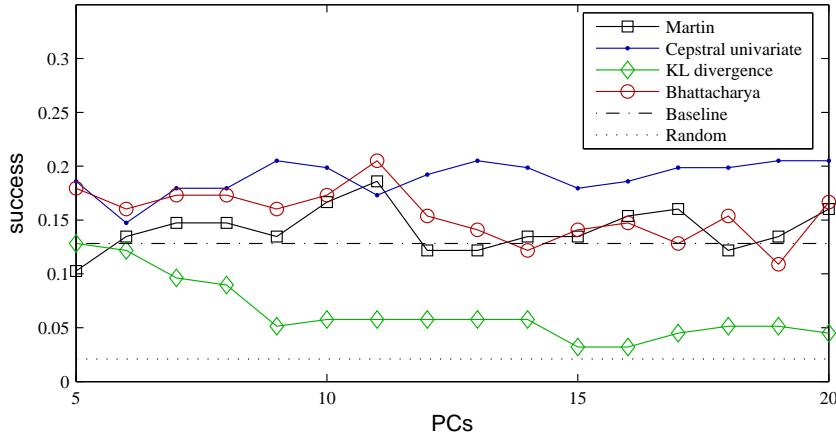


Fig. 6. Performance of distance metrics as a function of state-space dimension (i.e. number of principal components) N . The state of the art is represented by the “Martin” and “KL” schemes, which are generally outperformed by the new cepstral univariate scheme. The Bhattacharya metric performs comparably to the Martin distance. The “Baseline” metric simply compares the mean frames of the (greyscale) sequences. Note that all performance figures are low—best achieved performance is about 20%—reflecting the difficulty of the dataset when cropping is introduced.

well above the performance of random guessing, which is expected to be about 1%.

In all experiments we considered a nearest-neighbour classifier—classifiers with stronger priors on the density could be considered, such as a support vector machine using these distance metrics as a kernel [7], but the NN classifier makes the fewest assumptions about the parameter distribution, and generally performs competitively with a wide range of classifiers [25], providing a useful baseline.

The experimental procedure may be defined as follows. Index the $m = 36 \times 4$ test sequences by i , with the sequence category given by $c(i)$. For each sequence, fit models $\theta_{iL} = (\mathbf{C}_L, \mathbf{A}_L, \mathbf{Q}_L)$ and $\theta_{iR} = (\mathbf{C}_R, \mathbf{A}_R, \mathbf{Q}_R)$ to the left and right croppings. For a distance metric $d(\theta, \phi)$ between AR models, define the distance between sequences i and j as

$$d_{ij} = \min\{d(\theta_{iL}, \theta_{jR}), d(\theta_{iR}, \theta_{jL})\}. \quad (24)$$

One-NN classification performance is then computed as

$$\text{success} = \sum_i \delta(c(i), c(\operatorname{argmin}_{j \neq i} d_{ij}))$$

where $\delta(x, y) = 1/m$ for $x = y$, zero otherwise. Figure 6 summarizes the primary result. The tuning parameter common to all techniques is N , the number of

principal components used to characterize the sequence, and the figure plots performance against N . The graph shows that for a wide range of values, the leading performers were the Bhattacharya distribution comparison (Chernoff information with $\alpha = 0.5$) and the summed univariate cepstral distances of §3.2.

5 Conclusion

This paper has introduced a new and challenging recognition problem: shift-invariant dynamic texture recognition. We have shown that existing dynamic texture recognition algorithms, when applied to classification problems where there is a difference in camera viewpoint, show a significant drop in performance. Several new similarity measures have been proposed, and some have been shown to outperform the state of the art. In particular, the use of the cepstrum appears to be a natural tool for the comparison of AR models.

The investigation has concentrated on defining distance metrics between AR models, rather than modelling the distributions of model parameters in a learning framework. This allows us to test classification without requiring a large labelled training set, and provides insight into the behaviour of these model parameters which may be useful in feature selection for distributional approaches.

The reader will note that we are quoting absolute performance figures of the order of 20%, which may appear unusually low. We comment that the absolute performance figures are not relevant, providing that performance is significantly different from random, which is true here. The absolute performance figures can be increased by further pruning of the dataset, but relative performance of the algorithms is expected to remain unchanged. In a real-world system, of course, we would not expect to use cues based on motion alone—distinguishing grass from water is rendered artificially difficult if colour is removed from consideration. It is our contention however, that when testing metrics for motion-based recognition, it is valuable to exclude textural cues as much as possible.

The paper has concentrated on global modelling approaches in order to capture large-scale correlations in the motion sequences. However, the relatively small size of our crop windows may be thought of as positioning the technique between local and global strategies. It may be valuable to further explore this tradeoff, and build a multi-scale strategy.

References

1. Johansson, G.: Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics* **14** (1973) 201–211
2. Chetverikov, D., Peteri, R.: A brief survey of dynamic texture description and recognition. In: *Intl. Conf. on Computer Recognition Systems*. (2005) 223–230
3. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. *IJCV* **51** (2003) 91–109
4. Soatto, S., Doretto, G., Wu, Y.N.: Dynamic textures. In: *Proc. ICCV*. (2001) 439–446

5. Fitzgibbon, A.W.: Stochastic rigidity: Image registration for nowhere-static scenes. In: Proc. ICCV. Volume 1. (2001) 662–670
6. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: Proc. CVPR. Volume 2. (2001) 58–63
7. Chan, A.B., Vasconcelos, N.: Probabilistic kernels for the classification of autoregressive visual processes. In: Proc. CVPR. (2005) 846–851
8. Vidal, R., Ravichandran, A.: Optical flow estimation and segmentation of multiple moving dynamic textures. In: Proc. CVPR. (2005)
9. Doretto, G., Soatto, S.: Towards plenoptic dynamic textures. In: Proc. Intl. Workshop on Texture Analysis and Synthesis. (2003)
10. Polana, R., Nelson, R.: Detecting activities. In: Proc. CVPR. (1993) 2–7
11. Black, M.J.: Explaining optical flow events with parameterized spatio-temporal models. In: Proc. CVPR. (1999) 1326–1332
12. Niyogi, A.A.: Analyzing and recognizing walking figures in *xyt*. In: Proc. CVPR. (1994) 469–474
13. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proc. ICCV. (2003) 432–439
14. Shechtman, E., Irani, M.: Space-time behavior based correlation. In: Proc. CVPR. (2005) 405–412
15. Yacoob, Y., Black, M.J.: Parameterized modeling and recognition of activities. In: CVIU. (1999) 232–247
16. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. ICCV. (2003)
17. Schodl, A., Szeliski, R., Salesin, D.H., Essa, I.: Video textures. In: Proc. ACM SIGGRAPH. (2000) 489–498
18. Polana, R., Nelson, R.: Recognition of motion from temporal texture. In: Proc. CVPR. (1992) 129–134
19. Szummer, M.: Temporal texture modeling. Master’s thesis, MIT Media Lab, Cambridge MA (1995)
20. Fablet, R., Bouthemy, P.: Motion recognition using nonparametric image motion models. IEEE PAMI **25** (2003) 1619–1624
21. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: Proc. ICCV. (2003) 1236–1242
22. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and its Applications. Springer (2000)
23. Manolakis, D.G., Ingle, V.K., Kogon, S.M.: Statistical and Adaptive Signal Processing. McGraw-Hill, Boston (2000)
24. Martin, R.: A metric for ARMA processes. IEEE Trans. on Signal Processing **48** (2000) 1164–1170
25. Ripley, B.D.: Why do nearest-neighbour algorithms do so well? SIMCAT (Similarity and Categorization), Edinburgh (1997)