B. Abraham, O. Camps and M. Sznaier: Dynamic texture with Fourier descriptors.
In *Texture 2005: Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, pp. 53–58, 2005.

Dynamic Texture with Fourier Descriptors

Bobby Abraham[1], Octavia I. Camps[1,2] and Mario Sznaier[1]

[1] *Dept. of Electrical Engineering*
[2] *Dept. of Computer Science and Engineering*
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

Dynamic textures encompass the class of video sequences that exhibit some stationary properties in time. Examples of such textures include ripples on water, smoke out of a chimney, waving flags etc. Recently, it has been shown that such textures can be expressed as the output of a linear dynamic system that can be identified from a sample image sequence. In this paper, we propose to identify the underlying linear dynamic system using a set of Fourier descriptors of the frames rather than the raw sequence. It is shown that this approach enhances the performance time of the learning process and only requires a much smaller learning data set. The proposed algorithm improves upon published algorithms by exploiting the inter pixel dependency, while producing similar visual quality for the synthesized sequences, as shown through several experiments conducted on different dynamic texture sequences.

I. INTRODUCTION

Dynamic textures encompass the class of video sequences that exhibit some stationary properties in time [4]. Recently, Doretto et al [4] have proposed to exploit this temporal property to characterize dynamic textures as the output of a dynamical system driven by an independent and identically distributed process. This characterization is especially attractive since the underlying system provides a mathematical representation that can be used for analysis, recognition and synthesis of dynamic textures. Furthermore, the system can be easily identified from a set of training frames by using techniques from the well established field of system identification [3],[6]. In particular, in [4] Doretto et al proposed a modified version of the subspace identification method N4SID [6] that finds a system realization based on a singular value decomposition (SVD) of a matrix that has as its columns the sequence frames, stacked as one dimensional vectors. This approach, as demonstrated with several examples in [4], produces models of a fairly small order, that effectively represent the given dynamic textures. However, while this approach exploits the *temporal* correlation among the frames in the sequence, it does not take full advantage of the *spatial* correlation which is usually present in each frame in a dynamic texture. Prior work in dynamical texture modeling by [5] uses a spatio-temporal autoregressive model (STAR) to model the dynamics of the texture. STAR is a three-dimensional extension of autoregressive models. which expresses each pixel as a linear combination of the surrounding pixels lagged both in space and time. However, this spatial causality constraint in the STAR model severely curtails its ability to capture complex dynamical textures.

In this paper, we propose a simple technique that can be used to exploit both *temporal* and *spatial* correlations in the given sequence. The main idea of the proposed approach is to work with the Fourier transform of the individual frames since each frequency component of a frame is a function of *all* its pixels, while the temporal stationarity properties of the sequence in the spatial domain are preserved. Thus, as shown in this paper, given a dynamic texture sequence, it is possible to learn more efficiently, more compact systems that describe a carefully chosen set of Fourier descriptors from the image sequence. Then, dynamic textures can be synthesized by applying a Discrete Fourier Inverse transform to the output of the identified system.

The paper is organized as follows. In Section II, a brief overview of the learning and synthesis algorithm for dynamic textures described in [4] is given. In Section III, the proposed method incorporating Fourier descriptors is described. In Section IV, experiments comparing the proposed approach against the original method of [4] are presented. Finally conclusions are given in Section V.

II. DYNAMIC TEXTURE MODELING SYSTEM

For the sake of completeness, we briefly summarize the dynamic texture modeling algorithm presented in [4]. There, a dynamic texture sequence is modeled as the output of a linear dynamic system, which is associated with an auto-regressive moving average process (ARMA). Such a system can be mathematically written as

$$\begin{cases} x[k+1] = Ax[k] + Bv[k] \\ y[k] = Cx[k] + w[k] \end{cases} \quad (1)$$

where $x[k]$ is the state of the system at the k^{th} instant, with initial state $x[0]$, A is the state transition matrix, $y[k]$ is the

This work was supported in part by AFOSR grant FA9550-05-1-0437 and NSF grants IIS–0117387, ECS–0221562 and ITR–0312558.

observation or output of the system at the k^{th} instant, $v[k]$ is the driving input to the system at the k^{th} instant, which is assumed to be Gaussian white noise ($v[k] \sim N(0, Q)$) and $w[k]$ is the measurement noise, at instant k. We are not interested in $w[k]$ since it is the measurement noise and is of little significance for the synthesis process. Equation (1) above represents a first order ARMA system driven by Gaussian white noise. The parameters of the system are estimated from an input dynamic texture sequence.

Consider now a texture sequence with τ image frames in it, with each image having $m = N \times M$ pixels in it. The observation vectors for the learning system at time instant k denoted as $y(k)$, is the k^{th} image frame. Thus $y(k) \in \Re^m$. All such observation vectors are stacked into a matrix $Y^\tau = [y(1), \ldots, y(\tau)] \in \Re^{m \times \tau}$. This is the matrix of input sequences and is used as the training set. The dimension of the state, n, has to be determined. One way suggested in [4] is to empirically determine the state dimension by viewing the singular values of the Y^τ matrix and selecting n as the number of singular values above a certain threshold. From this it can be seen that

$$Y^\tau = CX^\tau + W^\tau; C \in \Re^{m \times \tau}; X \in \Re^{n \times \tau} \qquad (2)$$

It is well known that the state space models are not unique, since any similarity transformation can transform the basis of the state space to a new set of basis vectors, resulting in a new set of states and the associated matrices. Therefore a canonical model for the system is chosen by making the columns of the C matrix orthonormal

$$C^T C = I_n$$

This canonical form is estimated from the singular value decomposition (SVD) of the input observation matrix Y^τ. If $Y^\tau = U \Sigma V^T$, then the matrix C is estimated as $\widehat{C} = U$ and the collection of state matrix X^τ is estimated as $\widehat{X}^\tau = \Sigma V^\tau$. The A matrix can be estimated from the matrix $\widehat{X}^\tau$ as

$$[\widehat{x}(1), \ldots, \widehat{x}(\tau)] = \widehat{A}[\widehat{x}(0), \ldots, \widehat{x}(\tau - 1)]$$

Therefore A is estimated as

$$\widehat{A} = [\widehat{x}(0), \ldots, \widehat{x}(\tau - 1)]^\dagger . [\widehat{x}(1), \ldots, \widehat{x}(\tau)]$$

where † represents the Moore-Penrose pseudo-inverse. The input noise covariance matrix is estimated as

$$\widehat{v}(k) = \widehat{x}(k) - \widehat{A}\widehat{x}(k-1)$$
$$\widehat{Q} = \frac{1}{\tau - 1} \sum_{t=1}^{\tau} \widehat{v}(i)\widehat{v}^t(i)$$

The above solution as noted in the original paper is a sub-optimal closed form solution to the dynamical system, since it approximates the space spanned by the observation vectors. An asymptotically stable solution can be obtained, which is the N4sid algorithm [6], but as noted is computationally expensive and the gain of quality is marginal.

Once the system parameters are estimated, the dynamic texture can be synthesized using it, along with proper initial conditions. Ideally, an infinitely long sequence can be generated using this dynamic system. The matlab pseudo-code for both estimating the system parameters and for generating the image sequences are given in [4].

III. Dynamic Texture using Fourier Coefficients

In the approach described above, the parameters of the dynamic system were learned by using the raw image vectors themselves. As it was noted in the introduction this approach does not exploit the spatial correlations on the individual frames and thus can result in unnecessarily large order systems. Previous attempts to address this issue, such as the STAR approach [5] have problems with the non-causality spatial relation, in particular when the textures are complex.

More recently, Sznaier et al, [7] circumvented the spatial non-causality while modeling static texture by considering the $n \times m$ image as one period of an infinite 2D signal with period (n, m). Thus, at any given location (i, j) in the image, the intensity values $I(r, s)$ at other pixels are also available at position $(r - qn, s - qm)$, and the integer q can always be chosen so that $r - qn < i, s - qm < j$. From this observation they proceeded to model the given image as the output of an operator S admitting a state space representation of the form (1) but with the additional constraint that $A^n = I$, that is, the impulse response of the operator is periodic. Moreover, Sznaier et al [7] showed that the above operator can be approximated by an operator that has a Hankel matrix with its most significant singular values equal to the square of the largest magnitudes of the DFT of the given image. These results are related to the ideas proposed in [1] for dynamic shapes where the evolution of curves over time is modeled using an ARMA model of the Fourier descriptors of the curve. However, it should be noted that in [1] the authors retain the first N_w frequencies while in [7] the authors retain the N_w most *significant* frequencies.

Inspired by these results, and by noting that the magnitude at each frequency of the DFT of an image is a function of all its pixels in the spatial domain, we propose to model the dynamic textures in the frequency domain. This approach can be interpreted as a two stage modeling process: 1) the *spatial correlation* of the pixel is exploited using the results in [7] to model each (static) textured frame of the sequence by preserving its frequencies with the highest magnitude and 2) the *time correlation* among frames is exploited by learning the time evolution of the operators obtained in the previous step, using for example

the algorithm used in [4] and summarized in the previous section. These ideas are described in more detail next.

Consider a dynamic texture sequence that contains τ image frames in it, each of which is $N \times M$ in size. Initially a two-dimensional Fourier transform is performed on each of the image frames to get the 2D Fourier coefficients.

$$F(p,q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j(2\pi/M)pm} e^{-j(2\pi/N)qn}$$

for $p = 0, 1, \ldots, M-1$ and $q = 0, 1, \ldots, N-1$. This results in $N \times M$ Fourier coefficients for each frame. Since the Fourier coefficients occur in conjugate pairs, we only need to keep half of it, while the other half can be discarded. Using the ideas in [7] the Fourier coefficients thus obtained are thresholded and only those coefficients which are significant and above a certain threshold value are retained for learning. In other words, each frame of the learning sequence could be filtered such that only the most significant frequencies are retained.

However, it should be obvious that this approach presents us with a potential problem: it is not necessary for the same frequency component to be significant at all the frames. For instance a certain frequency component may be present in the thresholded set for the first 10 frames, absent in the next five frames and again present in the remaining $\tau - 15$ frames. As we are trying to learn the temporal variations present in a frequency component, it is necessary that the same frequency components be present in the entire set of the learning frames. In order to overcome this problem, we first threshold the Fourier coefficients in all the training frames, but then find the union of all the significant frequencies from all the frames. Once the union set of all significant frequencies is obtained, we must learn the temporal variations of the magnitudes at these frequencies. Thus, each individual frame is revisited and the Fourier coefficients at the frequencies in the set are selected. This process ensures that the same frequency components are thresholded across all the learning frames. Furthermore, in order to be able to recreate the images later, the information regarding which frequency component was present in the thresholded set, has to be stored. This is easily done by creating a binary two-dimensional Index array, equal to the size of the image, which has a positive entry for all frequencies in the thresholded set and a negative entry for all other frequencies.

Once the thresholded Fourier coefficients are obtained, we proceed to learn their temporal evolution by using the algorithm from the previous section as follows. Let r be the number of frequency components that are in the thresholded set. Note that typically, $r << N \times M$, the total number of Fourier coefficients for an image. Since the Fourier coefficients are complex, we separate the real and imaginary parts and form a vector $f(n) \in \Re^{2r}$ for the n^{th} frame , with the real part followed by the imaginary part. Thus we create a set of vectors $f(n)$ for $n = 1, 2, \ldots, \tau$ and using these vectors, we create a matrix $F^\tau \in \Re^{2r \times \tau}$ defined as

$$F^\tau = [f(1) f(2) \ldots f(\tau)] \tag{3}$$

The F^τ matrix in equation (3) is equivalent to the Y^τ matrix in equation (2). Using F^τ as the input data matrix, the dynamic system learning algorithm in [4], is used to estimate the parameters of the system.

The estimated linear dynamical system is then used to generate a set of vectors. Each of these generated vectors correspond to the Fourier coefficients of the images for the newly generated dynamic sequence. Using the Index array, that was stored in the learning phase, we then create a two-dimensional array of Fourier coefficients from each of these generated vectors. In these arrays, all Fourier coefficients corresponding to frequency components for which there is a negative index entry in the index array, have a value of zero.

The images are then obtained by taking the inverse Fourier transform of the two-dimensional arrays of Fourier coefficients as

$$f(m,n) = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} F(p,q) e^{j(2\pi/M)pm} e^{j(2\pi/N)qn}$$

for $m = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N-1$. The sequence of recreated images form the synthetically

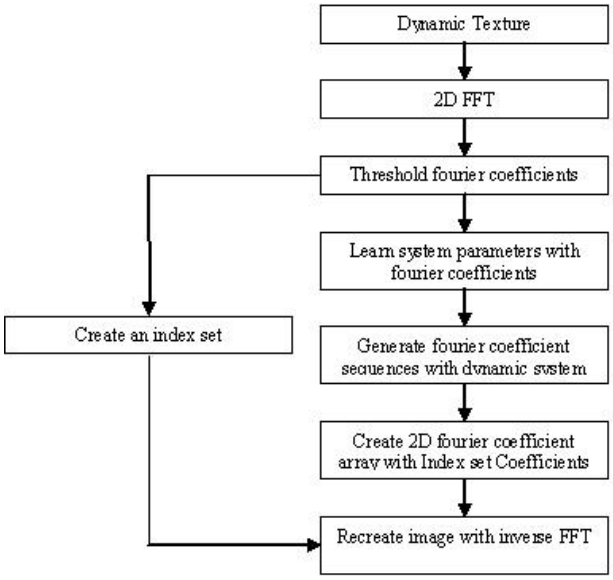

Fig. 1. Dynamic Texture using Fourier coefficients. We use Fourier descriptors to learn the temporal variation in each frequency component that is of interest. The images are then recreated using inverse Fourier transform

	size of frame	# of pixels	$2 \times r$	reduction ratio(%)
fountain	89×150	13350	2868	21.48
escalator	115×170	19550	3622	18.53
plastic	147×190	27930	4038	14.46
water	115×170	19550	5528	28.28
smoke	115×170	19550	2820	14.42

TABLE I
PERCENTAGE REDUCTION USING FOURIER DESCRIPTORS.

	Proposed scheme (time in secs.)	Original Scheme (time in secs.)
fountain	4.056	9.594
escalator	8.011	19.348
plastic	8.252	25.757
water	9.134	19.368
smoke	7.591	22.582

TABLE II
COMPUTATIONAL TIME FOR THE SYNTHESIS OF DYNAMIC SEQUENCES.

generated dynamic texture sequence. Figure 1 shows the flow of the proposed algorithm.

IV. EXPERIMENTS

Several experiments were performed using the dynamic system with our proposed learning scheme. The dynamic texture sequences used for these experiments were obtained from the MIT temporal texture database[1]. Figure 2 shows the results of synthesizing fountain ($\tau = 100, m = 89 \times 150(13350), 2r = 2868, n = 41$), escalator ($\tau = 120, m = 115 \times 170(19550), 2r = 3622, n = 15$), plastic ($\tau = 119, m = 147 \times 190(27930), 2r = 4038, n = 31$), water ($\tau = 120, m = 115 \times 170(19550), 2r = 5528, n = 47$) and smoke ($\tau = 150, m = 115 \times 170(19550), 2r = 2820, n = 13$), where τ is the number of training frames in the sequence, m is the dimension of each frame, r is number of Fourier coefficients retained after thresholding and n is the state dimension chosen for the dynamic system in [4]. The learning was performed using these original frames and 250 frames of each sequence were synthesized. The state dimension, n, was set as the number of singular values whose magnitudes were at least one-tenth of the magnitude of the highest singular value.

The experiments show that using Fourier descriptors in place of raw images, makes it possible to achieve a reduction in size of the input data matrix, used for learning. Table I shows the percentage reductions that were observed for different sequences. Generally we see that a reduction of about 15 to 30 percent can be achieved by retaining only the significant Fourier coefficients and to synthesize visually similar sequences. The exact value of the reduction obtainable depends on the actual sequence itself and varies based on the amount of dynamic motion involved in the sequence. These values reiterate the fact that each Fourier descriptor has a contribution of the motion from all parts of the image and hence we need to retain only a small subset of them. Using a smaller data matrix for learning reduces the amount of memory needed for computations and this becomes a significant issue when the sizes of the images in the texture sequence are large.

[1]Available at ftp://whitechapel.media.mit.edu/pub/szummer/temporal-texture

The size of the input data matrix also has a direct impact on the time required for the learning process. As noted in [4], the synthesis of each frame can be done at real time. Therefore the usage of Fourier descriptors also lead to significant difference in the time needed for the execution of the learning process of the system. A comparison of the time required for the learning process using the two schemes, for different sequences is shown in Table II. The original scheme is the algorithm described in [4] and the proposed scheme is the algorithm described in this paper. We implemented the matlab pseudo-code given in [4] and the experiments were performed in matlab (version 6.5.1) on a PC with 996 MHz Pentium III processor and 512 MB of RAM. The learning phase of the our scheme includes calculating the Fourier coefficients of all the frames (using FFT2 routine in matlab) and thresholding them, along with the learning time required for the dynamical system. Our scheme achieves a lower performance time due to the fact that FFT routines are very fast and that we present a much smaller data matrix to be learned by the dynamical system. Since the data matrix is small the system takes much lesser time to estimate the parameters.

We have also done the analysis of error in reconstruction and error in prediction as given in [4]. The error in reconstruction is calculated as follows. A dynamic model for the sequence is first obtained by fixing the number of states, n, of the system. Then using the states estimated as part of the learning process, the original sequence is reconstructed. Then, the error in reconstruction is calculated as the norm of the difference between the original images and the reconstructed images. Figure 3(a) shows reconstruction error plots the sequences shown in Figure 2 as a function of the number of states. In order to calculate the error in prediction, the system is trained with τ number of frames from the training sequence. Using the learned system, the $\tau + 1$ frame is predicted and the norm of the difference between this predicted frame and the corresponding frame in the original sequence gives the error in prediction. This is plotted against the number of training frames in Figure 3(b). It is observed that both, the error of reconstruction and the error of prediction, for the proposed and the

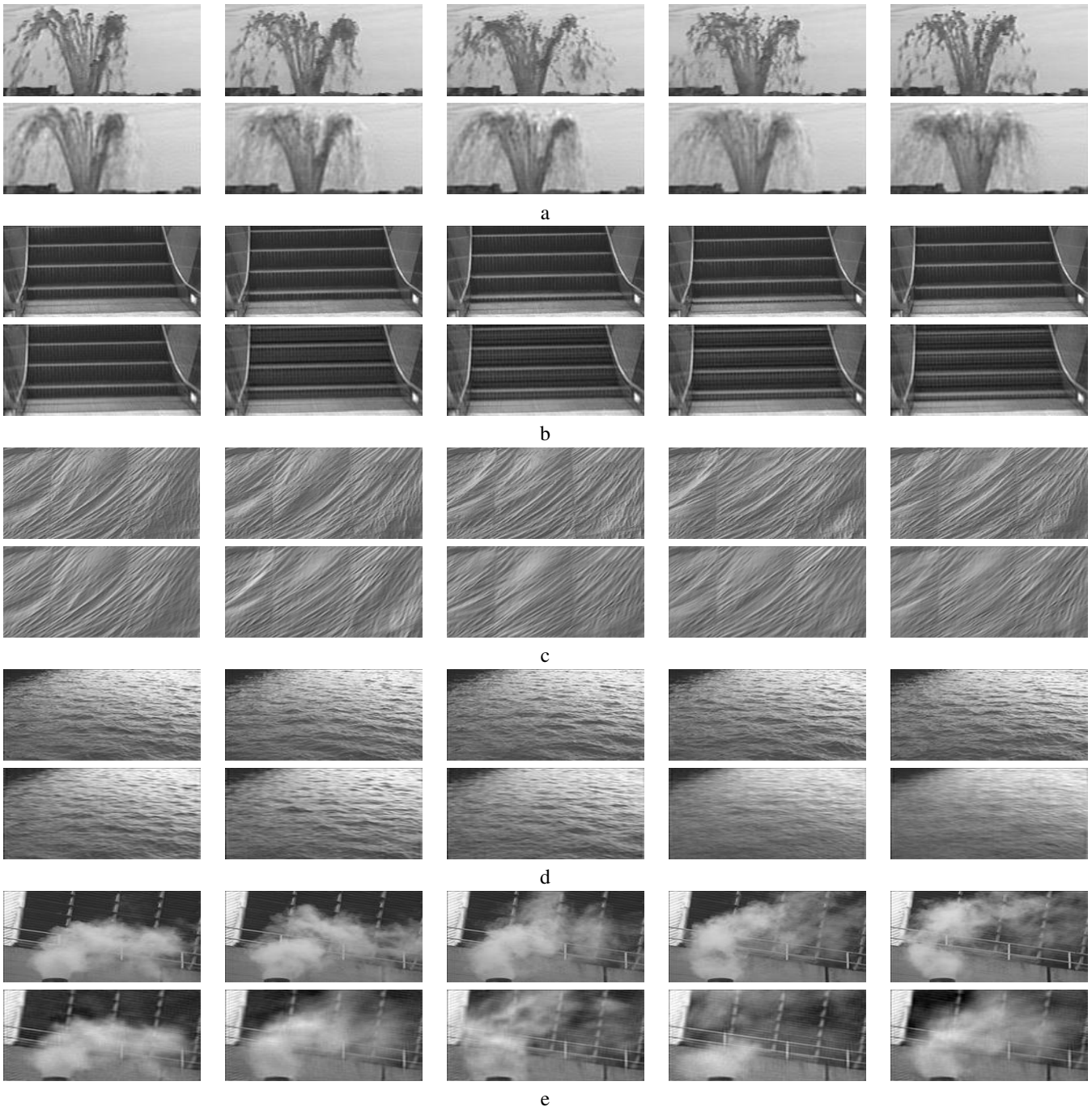

Fig. 2. Dynamic texture sequences of (a) fountain, (b) escalator, (c) plastic, (d) water and (e) smoke, generated using Fourier descriptors as input to the dynamic system in [4]. The top row are samples from the original sequence and the bottom row are samples from the synthesized sequence. (The original data was obtained from the MIT temporal texture database).

original methods are very similar.

V. Conclusion

In this paper, we propose a scheme to learn and synthesize dynamic texture sequences, based on the algorithm provided by [4] and [7]. The proposed scheme uses Fourier descriptors of frames in the sequence, in place of the frames themselves to learn the dynamics of the texture sequence. In doing so, we have shown that similar results as the ones obtained using the original scheme can be

achieved while reducing the computational complexity and memory requirements during learning. Since each of the Fourier transform coefficients contains contributions from all the pixels of the image, the proposed scheme enables us to harness better, the inter pixel dependency that exists in each frame. In fact it is this harnessing ability that reduces the load of the system and improves the performance time. Further work would focus on extending the same scheme to color images and also try to investigate the usage of this scheme for dynamic texture recognition.

REFERENCES

[1] Liu, C. and Ahuja, N. A model for dynamic shape and its applications", Proc. of Conference on Computer Vision and Pattern Recognition, 2004, Vol. 2, pp 129-134.

[2] Golub, G. and Van Loan C. 1989. Matrix Computations, 2nd edn. Johns Hopkins University Press: Baltimore MD.

[3] Ljung, L. 1987. System Identification. - Theory for the user. Prentice Hall: Englewood Cliffs, NJ.

[4] Doretto, G., Chiuso A., Wu Y.N. and Soatto, S. Dynamic Textures. International Journal of Computer Vision, Feb. 2003, Vol 51(2), pp 91-109.

[5] Szummer, M. and Picard, R.W. Temporal Texture Modeling. Proc. of International Conference on Image Processing, 1996, Vol. 3, pp 823-826.

[6] Van Overschee, P. and De Moor, B. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, 1996, 30:75-93.

[7] Sznaier, M., Camps, O.I., and Mazzaro, C. Horizon Model Reduction of a class of Neutrally Stable Systems with Application to Texture Synthesis and Recognition. Proceedings of the CDC, 2004.

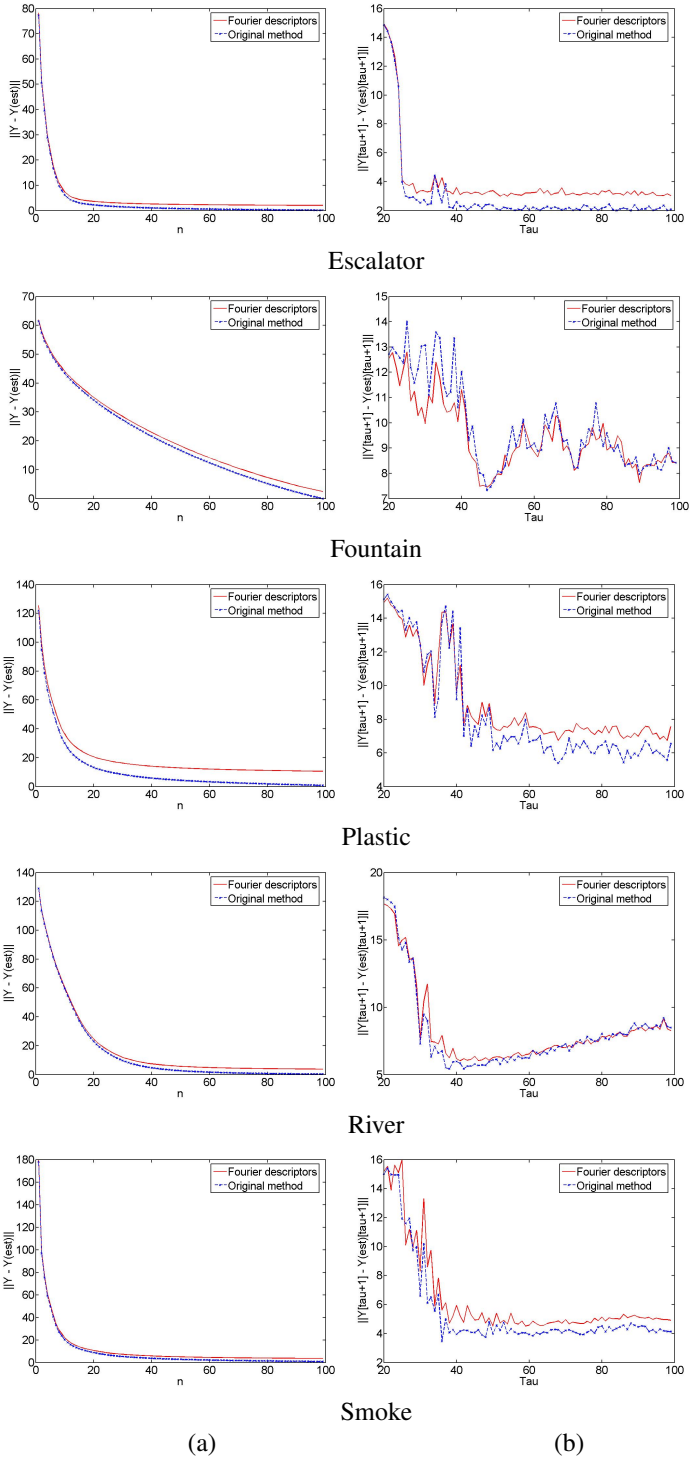

Fig. 3. Error of (a) reconstruction against number of states and (b) Error of prediction against number of frames using in training.