Recognition of Dynamic Textures using Impulse Responses of State Variables

Koki Fujita and Shree K. Nayar

***Abstract*— Dynamic textures are image sequences which contain moving scenes such as a flowing river, drifting smoke, waving foliage, etc. Such image sequences have dynamical properties that are related to motion in the physical world. In this paper, we propose a novel analytical tool for analyzing dynamic textures. The key idea is to exploit the properties of the impulse responses of the state variables computed using the previous algorithm for dynamic texture representation [1]. It turns out that the fundamental dynamical properties of a dynamic texture are captured very efficiently by these impulse responses. We have used our approach to develop an algorithm for the recognition of *local* dynamic textures. This algorithm is significantly more efficient than previously proposed techniques that use distances between computed dynamic texture models in a non-linear space. We test the recognition accuracy of our algorithm using a variety of real-world dynamic textures.**

***Index Terms*— Classification, texture models, dynamic textures, impulse responses.**

I. Introduction

Dynamic texures are image sequences of moving scenes such as a flowing river, drifting smoke, waving foliage, etc. Such sequences contain the dynamical properties of the physical motion in the real scene. Soatto et al. [1] have proposed a novel framework for the analysis and synthesis of such dynamic scenes. This framework is based on a system identification theory which estimates the parameters of a stable dynamical model. Unlike other methods for analyzing and recognizing spatio-temporal textures, this technique is an image-based one which does not require one to develop a physical model of the changing three-dimensional scene. As pointed out in the original work [1], the estimated dynamical model is a closed-form sub-optimal solution for the simplest first-order ARMA model with white Gaussian input. The uniqueness or asymptotic properties of the estimated model have not been validated. Nevertheless, stable dynamical models are obtained for a class of real-world phenomena. Estimated models have been used to synthesize realistic image sequences.

In this paper, we propose a novel recognition scheme for dynamic textures. This approach has several advantages over previous work on dynamic texture recognition [2]. First, the previous algorithm assumes that the dynamic texture occupies the complete spatial extent of the image. In contrast, our approach is a local one in that it can recognize multiple dynamic textures in different regions of the image. For example, a sequence may include a building with a smoking chimney and a fluttering flag, located by a flowing river. Our algorithm automatically recognizes and labels the local textures as being different. Second, the previous recognition algorithm compares computed dynamical models using distances in a non-linear space [2]. In contrast, we use a simple but effective feature of the dynamic texture for matching purposes. This feature is based on the impulse responses of the estimated dynamical model. From the viewpoint of system identification [3], the impulse responses capture the inherent dynamical properties that we are looking for. At the same time, they are very efficient to compute and compare. For these reasons, our recognition scheme is very simple and practical.

It is known from system identification theory [3] that the estimated state spaces of two different systems (textures) are not constrained to have the same bases. As a result, we can expect the corresponding impulse responses also to have different bases. Therefore, we apply principal component analysis to the impulse responses of different textures to compute an optimal linear space within which they can all be represented and compared. Within this space, the impulse responses are represented as trajectories (like the parametric eigenspace representation [6]). The closest trajectory to a novel one is the one that has the most number of points on it that are closest to points on the novel trajectory. Although we use exhaustive search to find the closest points in our current implementation, this can be made more efficient by using an algorithm such as the one described in [7]. We conclude our paper with several experiments conducted using video clips of a wide range of natural phenomena.

II. Background: Dynamic Textures

The dynamic texture technique proposed in the original work by Soatto et al. [1] is based on system identification theory. System identification basically aims at obtaining a mathematical model that describes the dynamical properties of a physical system, e.g. an aircraft flight control system, an air-conditioning system, a human speech system, etc. Soatto et al. [1] showed that system identification is also applicable to spatio-temporal textures because such textures do possess dynamical properties that are correlated with the dynamical properties of the physical phenomena taking place in the scene.

The texture is assumed to be a stationary second-order process with arbitrary covariance driven by white Gaussian noise (i.e. a first order ARMA model). Such a process can be modeled as:

$$\begin{cases} x(k+1) = Ax(k) + v(k), & v(k) \sim \mathcal{N}(0, Q); x(0) = x_0, \\ y(k) = Cx(k) + w(k), & w(k) \sim \mathcal{N}(0, R), \end{cases} \tag{1}$$

K. Fujita is with the Department of Aeronautics and Astronautics, Kyushu University (e-mail: fuji@aero.kyushu-u.ac.jp).

S. K. Nayar is with the Department of Computer Science, Columbia University (e-mail: nayar@cs.columbia.edu).

where, $k = 0, 1, 2, \cdots$ is the discrete time instant, $y(k) \in \mathbb{R}^m$ are measured pixel brightness values in the k-th image frame, m equals the number of pixels in an image frame, $x(k) \in \mathbb{R}^n$ is an n-dimensional state vector, $v(k) \in \mathbb{R}^n$ and $w(k) \in \mathbb{R}^m$ are white Gaussian noise. The above dynamical model is characterized by the parameter matrices $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times n}$, $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$.

In the original work on dynamic textures, the matrices A, C and Q are estimated as follows. First, equation (1) is rewritten using $Y_1^\tau \doteq [y(1), \cdots, y(\tau)] \in \mathbb{R}^{m\times\tau}$, $X_1^\tau \doteq [x(1), \cdots, x(\tau)] \in \mathbb{R}^{n\times\tau}$ and $W_1^\tau \doteq [w(1), \cdots, w(\tau)] \in \mathbb{R}^{m\times\tau}$:

$$Y_1^\tau = CX_1^\tau + W_1^\tau. \tag{2}$$

C and X_1^τ can be estimated by solving the optimization problem [8] given by $\hat{C}(\tau), \hat{X}(\tau) = \arg\min_{C, X_1^\tau} \|W_1^\tau\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. To solve this problem, we can use singular value decomposition (SVD). We express $Y_1^\tau = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times n}; U^TU = I_n$, $V \in \mathbb{R}^{\tau\times n}; V^TV = I_n$. Then, C and X_1^τ are determined as:

$$\hat{C}(\tau) = U, \tag{3}$$
$$\hat{X}(\tau) = \Sigma V^T. \tag{4}$$

A can be similarly estimated by solving the problem: $\hat{A}(\tau) = \arg\min_A \|X_1^\tau - AX_0^{\tau-1}\|_F$, where $X_0^{\tau-1} \doteq [x(0), \cdots, x(\tau-1)] \in \mathbb{R}^{n\times\tau}$. The solution is determined as:

$$\hat{A}(\tau) = \hat{X}_1^\tau \cdot (\hat{X}_0^{\tau-1})^\dagger, \tag{5}$$

where † denotes the Moore-Penrose pseudo-inverse. Finally, the input noise covariance Q is estimated as:

$$\hat{Q}(\tau) = \frac{1}{\tau}\sum_{k=1}^{\tau} \hat{v}(k)\hat{v}^T(k), \tag{6}$$

where, $\hat{v}(k) \doteq \hat{x}(k+1) - \hat{A}(\tau)\hat{x}(k)$.

As pointed out in the original paper, the above expressions represent a sub-optimal closed-form solution to the dynamics of the texture. This solution is sub-optimal because it uses an approximation to the actual subspace spanned by the measured image sequence $y(k)$, $k = 1, 2, \cdots$. An exact solution requires the use of the detailed structure of the subspace spanned by the sequence; such an approach is used in popular system identification algorithms such as N4SID [4] and MOESP [5]. However, in the context of dynamic textures, such an exact solution would prove computationally intensive. Despite its sub-optimality, for many real-world textures, the method of [1] successfully estimates the parameters of the dynamical model. These parameters can be used to synthesize dynamic textures of at least finite duration with a high degree of realism.

III. Local dynamic textures

In the previous work on dynamic texture recognition [2], the texture is assumed to occupy the complete spatial extent of the image. In contrast, our approach assumes that there may exist multiple dynamic textures in different regions of the image. We refer to these as local dynamic textures (see Fig. 1). For example, a sequence may include a building with a smoking chimney and a fluttering flag, located by a flowing river. We are interested in automatically recognizing and labeling such multiple textures.

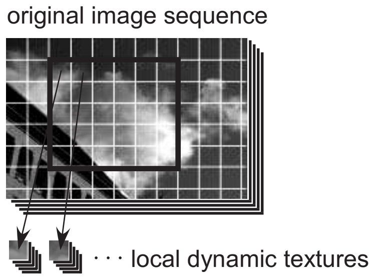


Fig. 1. Local dynamic textures are defined as the dynamical properties of small spatial regions of an image sequence. In this figure, the image is partitioned into several blocks, each block with its own dynamical model.

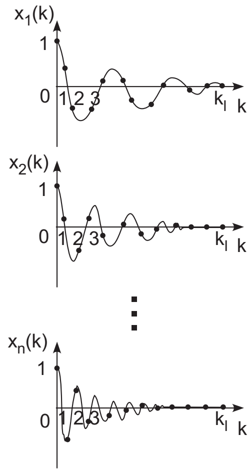


Fig. 2. Impulse responses of state variables of the estimated dynamical model. For an n-dimensional state space, n impulse response sequences are obtained. In the case of discrete time sequences each sequence consists of $k_l + 1$ (k_l is the terminal time index) discrete points.

IV. Impulse responses of state variables of estimated dynamical model

Once the parameter matrices of the dynamical model in equation (1) are estimated, they can be used to classify dynamic textures. In [2], the distances between the model parameters $(\hat{A}, \hat{C})$ in a non-linear space are used for recognition. We show in this section that a simpler approach can be taken for recognition that captures the essential dynamical properties of the texture.

From the perspective of system identification, two estimated dynamical models can be validated by comparing the state variables generated using the estimated models. The state variables are effectively represented by their impulse responses. The impulse responses of the state variables in Eq. (1) are computed using the estimated model parameters $\hat{A}$ as:

$$\hat{\boldsymbol{x}}(k+1) = \hat{A}\hat{\boldsymbol{x}}(k), \quad \hat{\boldsymbol{x}}(0) = [1, 1, \cdots, 1]^T, \quad k = 0, 1, 2, \cdots, \tag{7}$$

where, $\hat{\boldsymbol{x}}(k) \in \mathbb{R}^n$, $\hat{A} \in \mathbb{R}^{n\times n}$.

These impulse responses are n-dimensional sequences which contain the discrete points at the time instants k for each

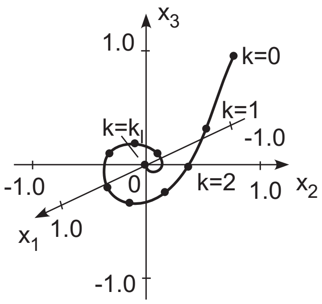


Fig. 3. Mapping the impulse responses to an optimal linear space. The transformed discrete points form a trajectory in the n-dimensional linear space. This figure shows the three dominant dimensions of the linear space. If the dynamical system is stable, the trajectory finally converges to the origin.

dimension, as shown in Fig. 2. These responses are unique with respect to the dynamical properties of original image sequence.

V. Mapping the impulse responses to an optimal linear space

In spite of their uniqueness property, the impulse responses of two different textures are not constrained to have the same bases. Along any given dimension, the impulse responses of two image sequences of the same texture can be different from each other. Thus, we apply principal component analysis to the impulse responses of different textures to compute an optimal linear space within which they can all be represented and compared in a consistent manner.

Once this is done, the impulse responses can be viewed as trajectories (as in the parametric eigenspace representation [6]) in an n-dimensional linear space. An illustration of such a trajectory in the three dominant dimensions is shown in Fig. 3. Since this trajectory is unique for each dynamic texture, textures can be classified by checking how close a novel trajectory is to a set of stored trajectories that belong to different classes.

VI. Recognition Scheme

The proposed recognition scheme is divided into two stages. One is a learning stage and the other is a recognition stage, as shown in Fig. 4. In the learning stage, first, the original image sequences are divided into local block sequences. These blocks are labeled accordingly to the types of textures they contain. Next, the dynamic texture algorithm is applied to each block sequence to obtain the model parameters, $\hat{A}, \hat{C}$ and $\hat{Q}$. Then, from the $\hat{A}$ matrix for each block sequence, n-dimensional impulse responses are computed. Finally, the impulse responses of all the blocks that belong to the same texture (same label) are used to compute a linear space and then they are mapped to this space to obtain trajectories. At the end of this process we obtain a set of linear spaces, each of which has a set of trajectories that belong to the same texture.

In the recognition stage, the model parameter matrix $\hat{A}$ for a given novel block sequence is used to compute n-dimensional impulse responses. These impulse responses are mapped to

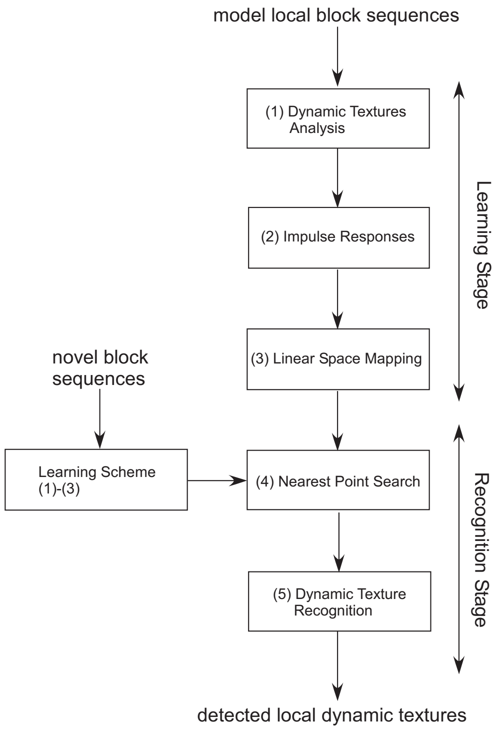


Fig. 4. Dynamic texture recognition scheme. The proposed recognition system is divided into two stages; a learning stage and a recognition stage. In the first stage, the point sets (trajectories) in the optimal linear space corresponding to each block (local texture) of the image sequence are computed. In the next stage, the dynamic texture recognition is applied to each block of a novel image sequence.

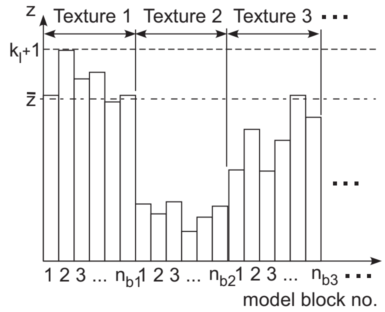


Fig. 5. An example histogram of the number of closest points between trajectories. Since two different types of dynamic textures can have many points close to each other, the similarity between two trajectories is evaluated only if the number of closest points exceeds the threshold $\bar{z}$.

trajectories in each of the linear spaces (corresponding to different textures) that were computed in the learning stage.

Finally, to recognize the dynamic texture in a novel block sequence, a nearest point search is conducted. We use an exhaustive search in our current implementation. The closest trajectory is taken to be the one that has the most number of points that are closest to points on the novel trajectory (see Fig. 5). Generally, since the trajectories of dynamic textures in the optimal linear space often intersect each other, two different types of dynamic textures can have many points close to each other. Therefore, we use a measure to evaluate the quality of the match between two textures as follows:

$$J_i = \sum_{j=1}^{n_{b_i}} (z_{ij} - \bar{z}), \quad \bar{z} = 0.8 \times (k_l + 1), \tag{8}$$

where, i is the type of texture (ex. 'snow', 'tree', etc.), n_{b_i} is the number of model block sequences of the i-th type texture, z_{ij} is the most number of closest points for j-th model block sequences of the i-th type of texture, $\bar{z}$ is the threshold used to evaluate the similarity. The texture is assigned the label that maximizes the above measure.

VII. Experiments

A series of experiments were conducted to validate the effectiveness of the proposed scheme. As shown in Fig. 6, five different types of image sequences are used in the experiments. Each image has 180×120 pixels and 256 gray levels. Each sequence has 300 images. The first 100 frames are used in the learning stage and the remaining 200 frames are used for recognition. Each image sequence is divided into local block sequences and a number of the block sequences that contain the same kind of texture are selected for learning. In this experiment, each local block frame is of size 16×16 pixels. From each image sequence, 30 among a total of 77 block sequences are selected for learning (see Fig.6). The dimension of the state space is varied to be 10, 30 and 50.

In the recognition stage, 30 local block sequences for each type of the textures are used to test the recognition scheme. The novel block sequences are selected from the last 200 frames in the original sequences and also from the same regions as in the learning stage. With the exhaustive search algorithm, a point on the trajectory of the novel block sequence is considered to be closest to a point on the trajectory of a model block sequence if it is within the distance ϵ. In our experiments, we used $\epsilon = 0.15$.

Fig. 7 shows recognition rates plotted as a function of the number of frames used in the recognition stage. The failure rates in the recognition are also shown in the figure. In this case, 'failure' means the inability to recognize any type of texture because the threshold value in Eq. (8) is too large to determine the texture type. Two of the impulse responses for model block sequences and their corresponding trajectories in the optimal linear space are shown in Fig. 8 and Fig. 9, respectively. In all of the experiments, the number of the time instants in the impulse response sequence was set to 20.

As seen in the figures, the relation between recognition rate and the number of the frames is different for each chosen dimensionality of the state space, especially between $n = 10$ and the other cases. For the relatively higher dimensional dynamical models, the recognition rates don't necessarily improve with the number of frames. These results can be explained as follows. Spatio-temporal textures of natural phenomena usually include non-stationary properties rather than stationary ones. In the current learning technique (which assumes a stationary dynamical model), unique parameters (with asymptotic properties) cannot be obtained for a high-dimensional dynamical model from a small number of the image frames.

On the other hand, a relatively low-dimensional dynamical model is more effective as shown in Fig. 7(a). The recognition rate for most of the texture types ('snow', 'tree' and 'river') improves with the number of the frames. The reason why the other types of textures ('smoke' and 'flag') cannot obtain high recognition rates is that they don't have intrinsically stationary properties. Note that, as expected, the failure rates decrease with the number of the frames in most of the cases.

Fig. 10 shows the results of labeling (recognition) for an image sequence obtained by tiling block sequences with a variety of textures. The block sequences used for tiling were selected from the last 200 frames of the five different image sequences shown in Fig. 6. As seen from the figure, the proposed scheme does well in recognizing the local dynamic textures.

Finally, the computational time for the recognition of novel block sequences is shown in Table I. This table is for comparing the proposed scheme with the original scheme described in the previous work [2]. The dimension of the state space was set to 10, the number of the frames used for the recognition was 200. These computational times are for a MATLAB implementation running on a Pentium III PC with a 700MHz CPU. The results of the original scheme are based on the Martin's distance, which produced the highest recognition rate among a few different metrics we tried. While the recognition rate of the original method is comparable to that of ours, it is twice as expensive in terms of computations.

TABLE I
Computational time for recognition.

	proposed scheme		original scheme	
	time (sec./block)	recognition rate (%)	time (sec./block)	recognition rate (%)
snow	5.01	100.0	10.17	100.0
tree	5.05	93.3	10.26	3.3
smoke	5.13	20.0	10.38	86.7
flag	5.21	56.7	10.64	0.0
river	5.17	96.7	10.36	100.0
average	5.11	73.3	10.36	58.0

VIII. Conclusion

In this paper, a new approach to spatio-temporal texture recognition based on the dynamic textures method was proposed. As a departure from previous work, we utilized the impulse responses of the state variables computed from the estimated dynamical model parameters. These responses are easy to compute and yet very effective for matching purposes. Our recognition algorithm automatically recognizes and labels local textures. Our experiments using a variety of real-world dynamic textures demonstrated the effectiveness of the proposed scheme for sequences with high stationary dynamics. Future work will focus on recognition of textures that have strong non-stationary dynamics.

Acknowledgment

The authors would like to thank G. Doretto for providing the original MATLAB source code for dynamic textures.

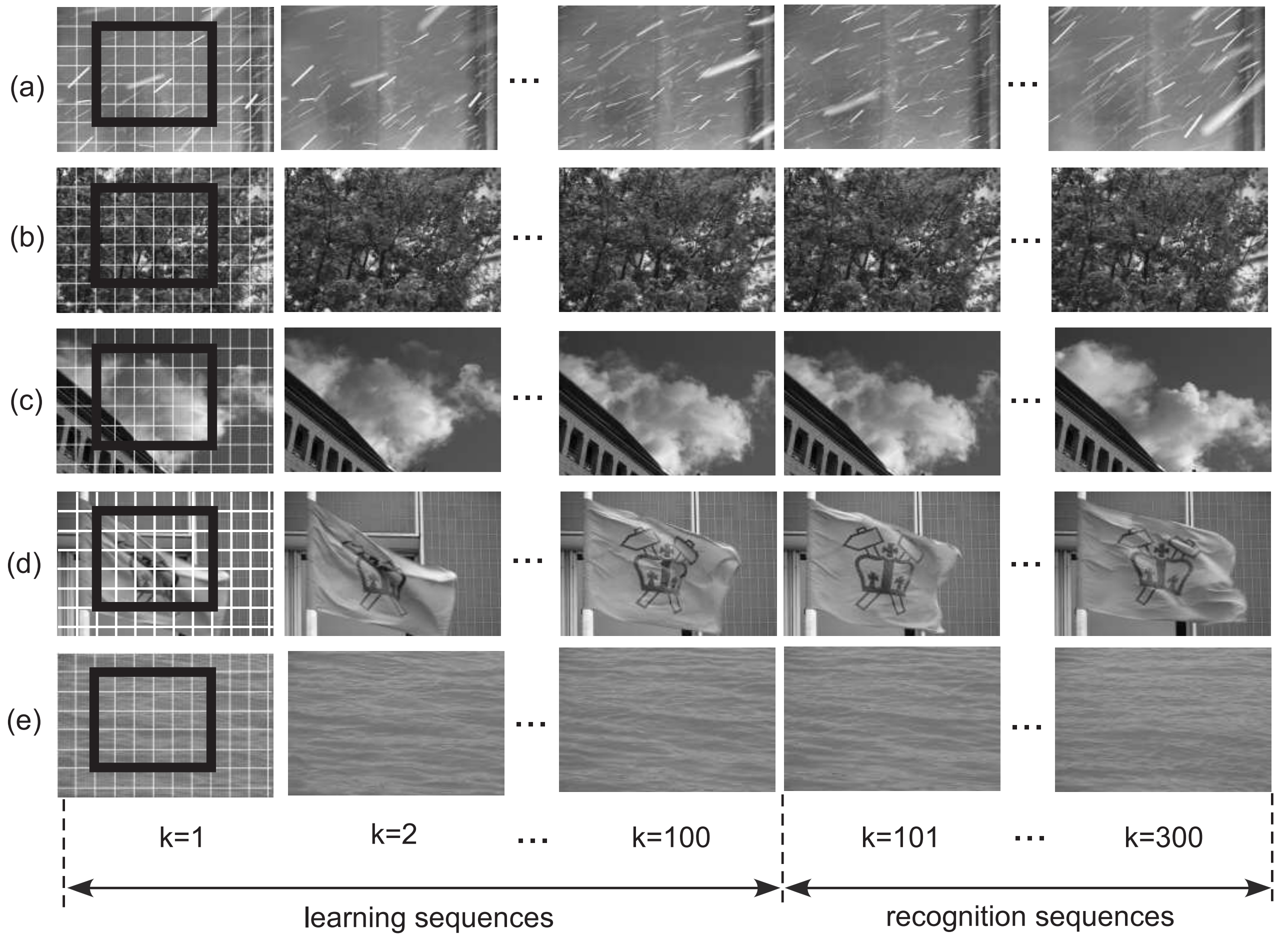
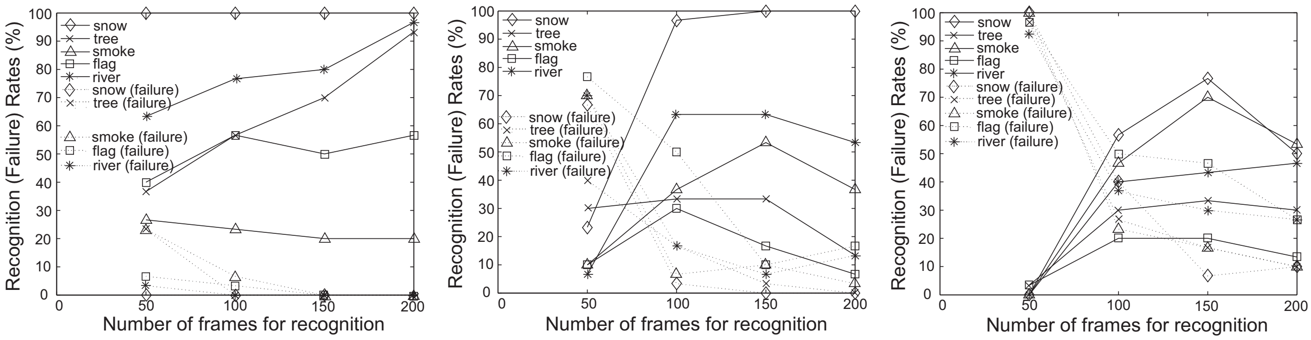


Fig. 6. Dynamic texture sequences used in the experiments. (a) 'snow' (falling snow) sequence, (b) 'tree' (waving foliage) sequence, (c) 'smoke' (drifting smoke) sequence, (d) 'flag' (fluttering flag) sequence, (e) 'river' (flowing river) sequence. Each image has 180×120 pixels and 256 gray levels. Each sequence has 300 images. The frame rate used to capture these phenomena is 30 fps. The first 100 frames of each sequence are used in the learning stage and the remaining 200 are used in the recognition (testing) stage. From the learning sequences, the blocks with dominant textures (within the black square) were used for learning.

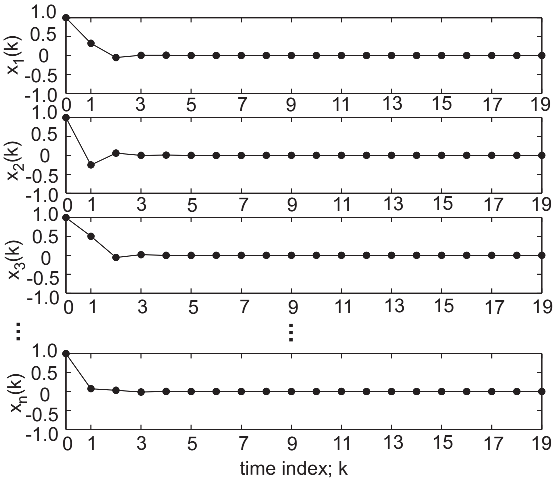


(a) n=10

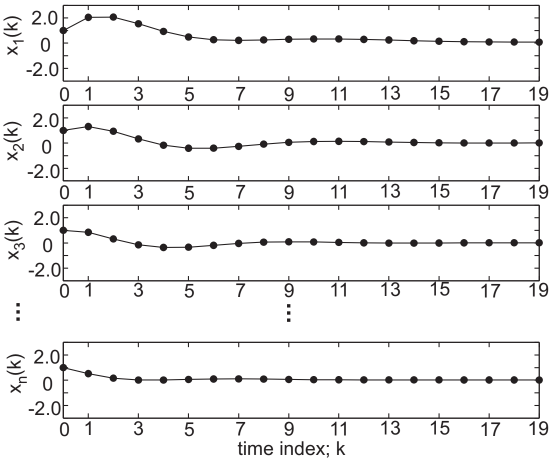
(b) n=30

(c) n=50

Fig. 7. Recognition and failure rates for local dynamic textures: (a) recognition (failure) rates for each type of texture (dimension $n = 10$), (b) recognition and failure rates for each type of texture (dimension $n = 30$), and (c) recognition and failure rates for each type of texture (dimension $n = 50$).

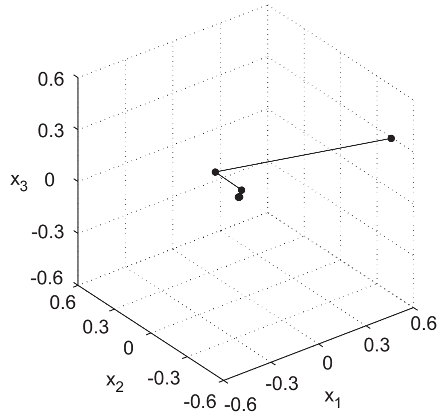


(a) 'snow' block sequence

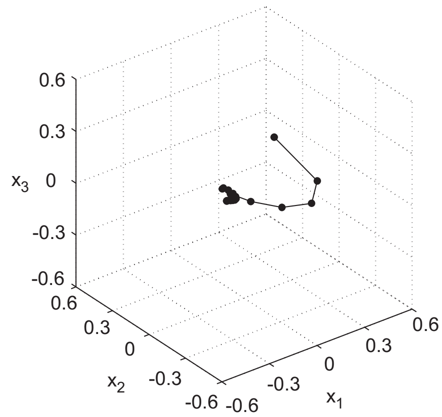


(b) 'flag' block sequence

Fig. 8. Impulse responses computed for (a) a 'snow' block sequence (the 15th block) and (b) a 'flag' block sequence (the 15th block). In this case, $n = 10$.



(a) 'snow' block sequence



(b) 'flag' block sequence

Fig. 9. Trajectories of the two impulse responses in Fig.8. Here, the three dominant dimensions are shown.

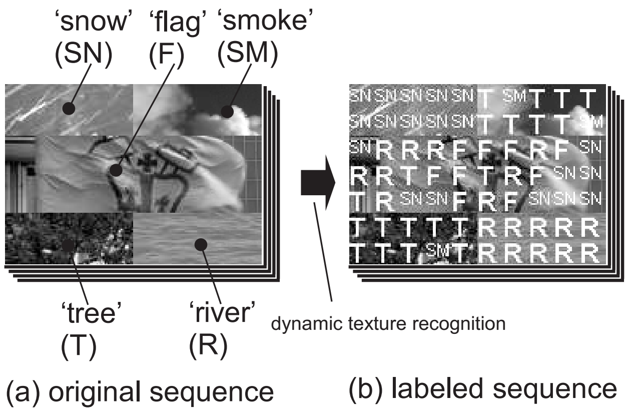


Fig. 10. Recognition results (labels) obtained for a dynamic texture sequence that was constructed by tiling block sequences of various texture types ('snow', 'tree', 'smoke', 'flag' and 'river'). (a) The original sequence and (b) the sequence labeled by the recognition algorithm.

REFERENCES

[1] S. Soatto, G. Doretto and Y. N. Wu, Dynamic Textures, International Journal of Computer Vision, 51, No. 2, 2003, pp. 91-109.

[2] P. Saisan, G. Doretto, Y. N. Wu and S. Soatto, Dynamic Texture Recognition, Proc. of Conference on Computer Vision and Pattern Recognition (CVPR '01), vol. 2, December 2001, pp. 58-63.

[3] L. Ljung, *System Identification-Theory for the User(2nd Edition)*, Prentice Hall, Englewood Cliffs, NJ, 1999.

[4] P. Van Overschee and B. De Moor, N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems, Automatica, 30, January 1994, pp. 75-93.

[5] Michel Verhaegen and Patrick Dewilde, Subspace model identification: Part1. The output-error state-space model identification class of algorithms, International Journal of Control, vol. 56, No. 5, pp. 1187-1210.

[6] H. Murase and S. K. Nayar, Visual Learning and Recognition of 3-D Objects from Appearance, International Journal of Computer Vision, 14, No. 1 , 1995, pp. 5-24.

[7] S. A. Nene and S. K. Nayar, A Simple Algorithm for Nearest Neighbor Search in High Dimensions, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, No. 9, 1997, pp. 989-1003.

[8] G. H. Golub, C. F. Van Loan, *Matrix Computation(3rd Edition)*, The John Hopkins University Press, Baltimore, 1996.