Phase Based Modelling of Dynamic Textures

Bernard Ghanem and Narendra Ahuja
Beckman Institute for Advanced Science and Technology
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
bghanem2,ahuja@vision.ai.uiuc.edu

Abstract

This paper presents a model of spatiotemporal variations in a dynamic texture (DT) sequence. Most recent work on DT modelling represents images in a DT sequence as the responses of a linear dynamical system (LDS) to noise. Despite its merits, this model has limitations because it attempts to model temporal variations in pixel intensities which do not take advantage of global motion coherence. We propose a model that relates texture dynamics to the variation of the Fourier phase, which captures the relationships among the motions of all pixels (i.e. global motion) within the texture, as well as the appearance of the texture. Unlike LDS, our model does not require segmentation or cropping during the training stage, which allows it to handle DT sequences containing a static background. We test the performance of this model on recognition and synthesis of DT's. Experiments with a dataset that we have compiled demonstrate that our phase based model outperforms LDS.

1. Introduction

A dynamic texture sequence (DT) captures a random spatiotemporal phenomenon. The randomness reflects in the spatial and temporal changes in the image signal. This may be caused by a variety of physical processes, e.g., involving objects that are small (smoke particles) or large (snowflakes), or rigid (grass, flag) or nonrigid (cloud, fire), moving in 2D or 3D, etc. Even though the overall global motion of a DT may be perceived by humans as being simple and coherent, the underlying local motion is governed by a complex stochastic model. For example, a scene of "translating" clouds conveys visually identifiable global dynamics; however, the implosion and explosion of the cloud segments during the motion result in very complicated local dynamics. Irrespective of the nature of the physical phenomena, the usual objective of DT modeling in computer vision and graphics is to capture the nondeterministic, spatial and temporal variation in images. DT modeling is motivated by a range of applications including DT synthesis, background subtraction in dynamic environments, and multi-layer motion separation. The challenges of DT modeling arise from the need to capture the large number of objects involved, their complex motions, and their intricate interactions. A good model must accurately and efficiently capture both the appearance and global dynamics of DT.

1.1. Related Work

The majority of methods that model DT fall into three broad categories which we briefly review next. **(1)** Motion field methods [15, 4] are based on motion analysis algorithms, such as those that compute and model optical flow. They are convenient, since frame-to-frame estimation of the motion field has been extensively studied and computationally efficient algorithms have been developed. However, these methods are best suited to estimate local and smooth motion fields. The non-smoothness, discontinuities, and noise inherent to rapidly varying, non-stationary DT's (e.g. fire) pose a challenge to optical flow algorithms. Object tracking methods [8] also tend to be infeasible here due to the large number of extremely small and non-rigid moving objects with little shape stability, complex motion characteristics, and inter-object interactions.

(2) Physical modeling methods [11] attempt to capture the attributes of the physical process from first principles. These methods are primarily used to synthesize specific textures such as ocean water, smoke, etc. Being closely tied to specific physical processes, they are difficult to generalize to other DT's. They are also computationally expensive since they must model physical phenomena.

(3) The third category consists of methods that obtain statistical models of spatiotemporal interdependence among images. They include the spatiotemporal auto-regressive (STAR) model by Szummer et al. [23] and multi-resolution analysis (MRA) trees by Bar-Joseph et al. [3]. These methods suffer from the following shortcomings: (i) DT representation limits the amount of data involved (e.g. only a finite-length sequence can be synthesized from the original DT), (ii) constraints are imposed on the types of motion that can be modeled (e.g. neighborhood causality is imposed in both the spatial and temporal domains), or (iii) they are applied directly to pixel intensities instead of more succinct representations, thus, making them computationally more challenging and sometimes infeasible. Within this class of DT models, we mention the notable work of Doretto et al. [22] that derives a stable linear dynamical system (LDS) model for DT's. Consecutive frames of a DT sequence are linearly related and viewed as the responses of the LDS to random noise input. This model has been applied to DT synthesis [22], recognition [19], and segmentation [10]. LDS has been expanded to accommodate a mixture of modeled DT's in [6] and its computational complexity has been improved in [2]. Modifications that have been made to this method include incorporating a lower dimensional representation by using high energy Fourier descriptors or state space variables instead of the estimated model parameters [2]. However, its modelling of the intensity values of a DT as a stable, linear ARMA (1) process leads to three main disadvantages: (i) the assumption of second-order probabilistic stationarity, which does not hold for numerous sequences (e.g. fire as in Figure 1), (ii) the sub-optimal relationship between the order of the LDS model and the extent of temporal modelling possible (i.e. an LDS of order n does not capture the most temporal variation in a DT among all models of order n), and (iii) significant computational expense, since the model is applied directly to pixel intensities without appropriately mitigating spatial redundancy.

Our method can be categorized as a spatiotemporal, image-based model that uses the Fourier phase content of the DT sequence to model both its appearance and global dynamics. In what follows, we justify our choice of using phase (Section 2), present the details of our phase based model (Section 3), apply it to DT synthesis and recognition, and provide experimental results that compare its performance to that of LDS (Section 4).

2. Motivation

In this section, we will establish that a model for the appearance and dynamics of a DT can be attained by representing its Fourier phase content alone. Following are the advantages of using the frequency domain representation that alleviate certain problems encountered in the spatial domain and motivate our proposed approach. **(1)** Spatially global features are captured locally in the frequency domain, since the change of the amplitude or phase of a certain frequency results in a global spatial variation. This makes frequency space modelling more appropriate for modelling global patterns such as those associated with DT appearance and dynamics. **(2)** Frequency analysis has been shown to be robust to unavoidable perturbations in images such as illumination changes [20] and additive noise [5]. **(3)** Computational complexity can be reduced by exploiting the inherent conjugate symmetry of the Fourier transform and the usually seen concentration of spectral image energy at low frequencies. **(4)** Furthermore, computationally efficient algorithms and specialized hardware are available for the computation of the Fourier transform (e.g. FFT).

In what follows, we justify why the phase content of DT is a useful dual representation of its appearance and temporal variations, and leads to a compact spatiotemporal model. **(1)** In [13], Hayes proved that it is possible to reconstruct multi-dimensional signals from their phase content alone, provided that these signals do not have symmetric factors in their Z-transforms. In fact, if a hybrid image is constructed from the phase spectrum of a given image and the amplitude spectrum of any other, we use the iterative algorithm, described in [16], to reconstruct the original image from the hybrid image. This process is called phase-only reconstruction. Figure 1 shows an example of this algorithm applied to ocean and fire images. In the rest of this paper, we assume that DT sequences enjoy this phase-only reconstruction property. This assumption is justified, since symmetric Z-transform factors seldom occur in practice.

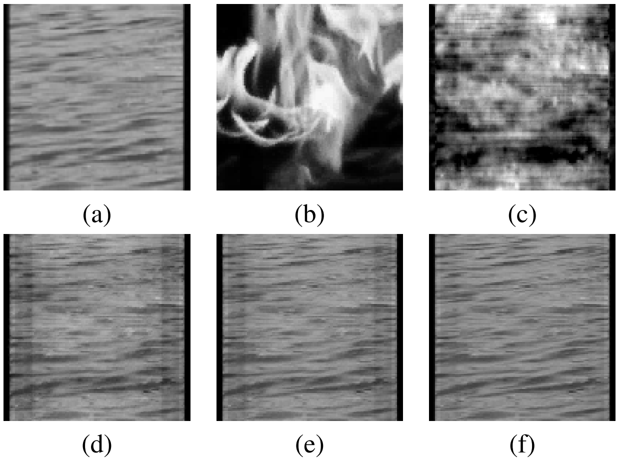


Figure 1. The phase spectrum of an ocean image (a) combined with the amplitude spectrum of a fire image (b) forms image (c). Images in (d), (e), and (f) the results of 50, 100, and 250 iterations of the reconstruction algorithm in [16], respectively.

(2) Complex stochastic motion, which characterizes a DT, leads to complex stochastic variations in its phase content. We have empirically shown (see next section), for a number of commonly encountered DT's, that the temporal variations of phase values do indeed capture most of the DT's dynamical characteristics and hence its global motion. This further validates, in addition to the phase-only-construction property, the value of phase for DT modelling.

Figure 2 shows that many more principal components are required to represent 80% of the variation in the phase of a DT than to represent the same amount of variation in its amplitude. This implies that a DT's phase varies significantly more than its amplitude over time, and so the DT's dynamical properties are better captured in the Fourier phase space. As a result of (1) and (2) above, we conclude that modelling a DT's global spatiotemporal features can be efficiently performed in Fourier phase space.

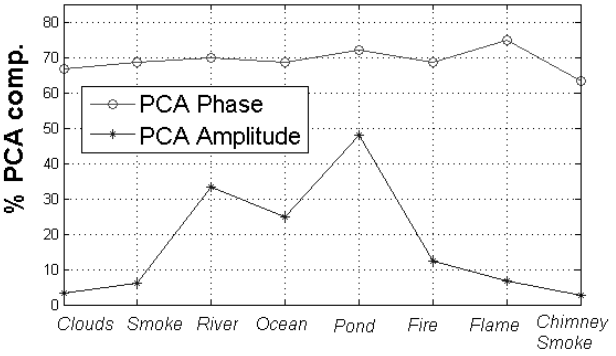


Figure 2. PCA components for DT phase and amplitude.

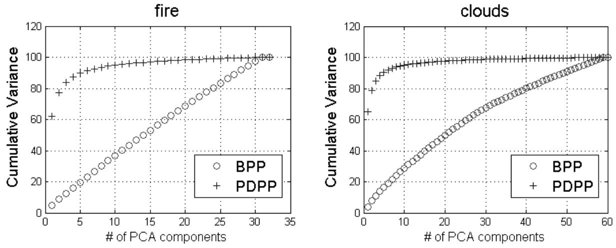
Contributions The contributions of our model are two fold: **(1)** it exploits the globally spatial features inherent to the Fourier phase domain to form a computationally efficient spatiotemporal model of a DT's appearance and global dynamics and **(2)** its training is insensitive to the static background that might accompany the DT itself and hence does not require any segmentation or specialized cropping.

3. Proposed Model

Our proposed model captures the phase portion of a DT and, for efficiency, represents it in PCA space. We call this the Basic Phase PCA (BPP) model. To further increase the efficiency, we propose a model which captures the phase changes in DT over time, instead of the absolute phase values of each individual frame, as in BPP. We call this the Principal Difference Phase PCA (PDPP) model. PDPP represents the phase changes in terms of the principal angle of the difference between phase spectra of consecutive frames. We transform the BPP phase into PDPP format as follows. Each extracted phase spectrum is vectorized and replaced by the sum of the previous phase spectrum and the principal angle of their difference,as illustrated in Equation 1.

$$\begin{aligned} &\Phi_i^{r+1} \longleftarrow \Phi_i^r + \Gamma(\Phi_i^{r+1} - \Phi_i^r) \qquad (1) \\ &\Gamma(x) = x + 2\pi k \ \in \ ]-\pi, \pi], \text{ for some } k \in \mathbb{Z} \end{aligned}$$

In fact, this transformation expands the domain of the original BPP space by 2π in each dimension. Hence, the PDPP space can be spanned by fewer principal components, giving rise to a more compact spatiotemporal model. Figure 3 provides empirical evidence that PDPP can capture significantly more variation in DT phase than BPP, for the same number of principal components. Accordingly, a DT is treated as a sequence of features embedded in a low dimensional PDPP space. We use two different methods to represent the temporal variations in these features: either in a holistic manner, for all components, using a probabilistic framework, or modelling each component separately using a deterministic framework. In the rest of this section, we give a detailed description of both frameworks, and how the model can be applied to two major tasks involving DT's: DT synthesis and recognition.



(a) (b)

Figure 3. BPP vs. PDPP for two DT's

3.1. PDPP Algorithm

Each DT sequence is a set of F frames (MxN in size). In order to mitigate spectral leakage, we preprocess the DT frames with a Hanning filter, whose spatial extent is set to be the image size. We extract the phase sub-spectra, whose energies exceed a predefined fraction of the total image energy. This fraction is increased to increase the compactness of the model. Only half the phase spectrum is required due to Fourier conjugate symmetry. Next, we transform these sub-spectra into principal difference format. Equation 2 illustrates how these spectra are embedded in a lower dimensional PCA space to form PDPP features ($\{\vec{x}^r\}_{r=1}^F$).

By reducing the number of principal components used in the representation to $L < F$, a more compact model is obtained. More importantly, a DT sequence containing a static background does not need to be cropped to show the dynamic texture alone. This follows from the fact that a static background will result in an additive term in the Fourier domain, which varies minimally with time. Since the PDPP features are formed from zero mean spectra and its basis spans the directions along which temporal variance is maximized (property of PCA), the effect of background on the model is highly reduced. Moreover, unlike LDS, no prior assumptions are made on the statistics of the DT sequence to be modelled.

$$\vec{\Phi}^r = A_{PDPP}\vec{x}^r + \vec{\Phi}_m \;\; \forall r = 1, \ldots, F \tag{2}$$

We will now describe the two frameworks for learning the spatiotemporal manifold of a DT sequence in the PDPP feature space. The first framework (Section 3.2) captures the variations of all components in probabilistic terms, while the other (Section 3.3) captures the variations of each PDPP component separately and deterministically. Here, we note that forming images from phase is a nonlinear operation, since the phase appears in the complex exponent of the Fourier spectrum. So, for both frameworks, to achieve linearity we define the PDPP feature space to span the sinusoidal functions (i.e. cosine and sine) of the Fourier phase instead of the phase itself.

3.2. Non-Parametric Probabilistic PDPP Model

In this framework, a DT sequence, (S) is represented as a set of F PDPP features, $\{\vec{x}_i\}_{i=1}^F$, corresponding to F frames. We model the posterior distribution of S nonparametrically, using Parzen windows with a suitably scaled unitary function $\Phi(\vec{x})$ as illustrated in Equations 3 and 4. We assume conditional independence between the features $\{\vec{y}_i\}_{i=1}^L$. We choose to use the RBF (radial basis function) kernel due to the simple functional form of its gradient and hessian.

$$p_S(\{\vec{y}_i\}_{i=1}^L | \{\vec{x}_m\}_{m=1}^F) = \prod_{i=1}^{L} p_S(\vec{y}_i | \{\vec{x}_m\}_{m=1}^F) \tag{3}$$

$$p_S(\vec{y}_i | \{\vec{x}_m\}_{m=1}^F) = \sum_{m=1}^{F} \frac{1}{V_F} \Phi_{RBF}(\frac{\vec{y}_i - \vec{x}_m}{h_F}) \tag{4}$$

$$\Phi_{RBF}(\vec{x}) = \frac{1}{2} e^{-\frac{\|\vec{x}\|^2}{2}}; \;\; V_F = (h_F)^L = 1/\sqrt{F}$$

We now describe how we use this probabilistic formulation for DT synthesis and recognition, using techniques from Bayesian machine learning.

3.2.1 MAP-Based DT Synthesis

We first consider using a given DT sequence to synthesize novel DT sequences, which resemble the original in appearance and global dynamics. In other words, given a set of PDPP features $\{\vec{x}_i\}_{i=1}^F$ that represent a DT, we want to find a new feature vector $\vec{x}_{MAP}$, which preserves the chosen spatiotemporal properties, namely of this DT. We formulate this synthesis problem as a multi-dimensional signal estimation problem according to the probabilistic framework described earlier. $\vec{x}_{MAP}$ is computed as the feature vector that maximizes the weighted posterior probability defined in Equation 5.

$$\vec{x}_{MAP} = \arg\max_{\vec{y}} \; p(\vec{y} | \{\vec{x}_i\}_{i=1}^F, \vec{w}) \tag{5}$$

$$p(\vec{y} | \{\vec{x}_i\}_{i=1}^F, \vec{w}) = \sum_{i=1}^{F} \frac{w_i}{V_F} \Phi_{RBF}(\frac{\vec{y} - \vec{x}_i}{h_F})$$

$$\sum_{i=1}^{F} w_i = 1 \;\; ; \;\; w_i \geq 0 \;\; \forall i = 1, 2, \cdots, F$$

This optimization problem can be solved locally using Newton gradient descent to find $\vec{x}_{MAP}$, since it is an unconstrained, non-convex maximization problem. Using the RBF kernel simplifies the descent update stage. Different synthetic frames are produced when the frame MAP weights $(\{w_i\}_{i=1}^F)$ are varied. The impact of each original frame on the synthesis process is proportional to the magnitude of its corresponding weight. The larger the weight is, the more the synthetic frame resembles the corresponding original frame, in appearance, and dynamics relating it to the next frame. A DT sequence of arbitrary length can be synthesized by varying the MAP weights. This allows for extrapolation of appearance and dynamics of the original sequence without reproducing the original frames.

3.2.2 MAP-Based DT Recognition

We are given a set of C classes of DT, each of which contains DT sequences that have similar appearance and dynamical properties. A class c is represented as either a single PDPP model formed by concatenating all the DT instances in c or as a set of PDPP models (each for a different DT in c) that will be processed independently. Experimentation shows us that both methods result in similar recognition rates. For the sake of simplicity, let us assume that each trained model (c) is represented by a single DT sequence, $\{\vec{y}_i^c\}_{i=1}^F$.

Each given test sequence (S_T) of T frames is represented by a sequence of T features $\{\vec{x}_i^c\}_{i=1}^T$ for each class c. In other words, $\{\vec{x}_i^c\}_{i=1}^T$ are the projections of S_T onto the PDPP space that spans $\{\vec{x}_i^c\}_{i=1}^T$ and $\{\vec{y}_i^c\}_{i=1}^F$. Therefore, the task of recognizing S_T becomes the task of finding the class c^*, which maximizes the posterior probability of $\{\vec{x}_i^c\}_{i=1}^T$ over all classes. This is formulated as follows:

$$c^* = \arg\max_{c} \; p(\{\vec{x}_i^c\}_{i=1}^T | \{\vec{y}_k^c\}_{k=1}^F) \tag{6}$$

$$p(\{\vec{x}_i^c\}_{i=1}^T | \{\vec{y}_k^c\}_{k=1}^F) = \prod_{i=1}^{T} p(\vec{x}_i^c | \{\vec{y}_k^c\}_{k=1}^F)$$

$$p(\vec{x}_i^c | \{\vec{y}_k^c\}_{k=1}^F) = \sum_{k=1}^{F} \frac{1}{V_F} \Phi_{RBF}(\frac{\vec{x}_i^c - \vec{y}_k^c}{h_F})$$

3.3. Piecewise Smooth PDPP Model

Local correlations between neighboring frequencies in the Fourier phase domain are considerably smaller than those between neighboring pixels in the spatial domain. Making use of this property, we assume that the components of a PDPP feature can be modelled as being independent. In this regard, each component is represented by a temporally varying trajectory, which is inherently oscillatory. Consequently, we choose to model the trajectory of the m^{th} PDPP component as a piecewise smooth function, $\Phi(t|\vec{\theta}_m)$, (e.g. spline). Using the independence assumption among the components, the PDPP model of a DT can then be viewed as a sequence of samples from L independent models given in Equation 7. Although the component-wise independence assumption neglects underlying component correlations, it allows for a more compact and computationally efficient model, as compared to the probabilistic model described earlier.

$$x_m(t) \triangleq \Phi(t|\vec{\theta}_m) \;\; \forall t \geq 0;\; \forall m = 1, \ldots, L \qquad (7)$$
$$\vec{\theta}_m = \arg\min_{\vec{\theta}} \sum_{i=1}^{F} [\Phi(i|\vec{\theta}) - x_m^i]^2$$

For DT synthesis, we choose $\Phi(t|\vec{\theta}_m)$ to be a cubic spline (i.e. piecewise cubic polynomials), which is sampled at equal intervals. Continuity and smoothness constraints (e.g. consecutive cubic pieces must have equal 1^{st} and 2^{nd} order derivatives where they meet) must be incorporated in estimating $\vec{\theta}_m$. Figure 4 illustrates the trajectory of a PDPP feature component modelled as a cubic spline.

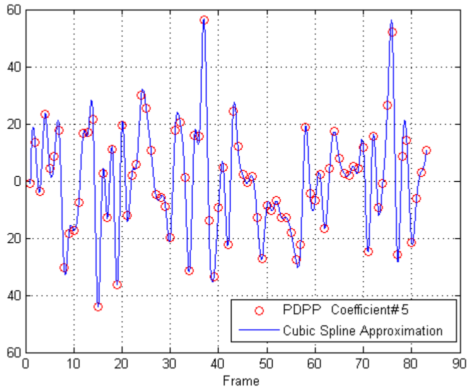


Figure 4. Temporal variation of a PDPP feature component modelled by a cubic spline.

4. Experimental Results

In this section, we will illustrate the performance of the PDPP model with respect to DT synthesis and recognition. The probabilistic and component-wise models are used in synthesis, while only the former is used in DT recognition. We will compare our results with those of LDS.

4.1. DT Synthesis

Numerous techniques have been proposed for DT synthesis. Some model the physical process underlying the DT (e.g. formation of ocean waves) [11]. Despite their high visual quality, the specificity of these models prevents them from being generalized to other DT's. As an alternative to physical models, purely image-based approaches have also been developed. In this category, we distinguish between two main groups: the first does not formulate a model of the DT but instead it reuses real frames from various locations in the sequence to extend the original sequence, while maintaining smooth frame-to-frame transition [21]. The other group of methods synthesizes frames based on a learned model of the DT [15, 3, 22, 23]. Among the few such model-based techniques that have been proposed, LDS has received the most attention in recent work. Despite its succinct representation, its main assumptions (e.g. second order stationarity, linearity in the spatial domain, and sub-optimal temporal modelling) limit the visual quality of synthesized DT sequences, especially non-stationary ones (e.g. fire). For such DT sequences, the visual quality of the synthetics frames deteriorates over time.

To evaluate PDPP based synthesis, we compiled a database of DT sequences from various online sources including the recent DynTex database [17]. MATLAB implementations for both of our PDPP methods were developed. For the MAP-based method, only two consecutive MAP weights were set to nonzero values. For the cubic spline method, we used equal length sampling intervals. Figure 5 shows some images randomly sampled from synthetic sequences produced by both PDPP models. Samples frames and full videos of additional DT's are provided in the supplementary material.

Evaluation: We compare the quality of our MAP synthesis with that of LDS for the same DT sequence. The dimensionality of the LDS model was set to be the same as that of our PDPP model. In Figure 5, we show an example of DT synthesis for a flame sequence. The quality of the synthetic frames appears to be more "natural" for PDPP than for LDS, since PDPP better preserves DT boundaries compared to LDS, which tends to blur them. Note that PDPP maintains smooth frame-to-frame transition, while undergoing global dynamics that resemble the original sequence.

Following observations can be made about the synthesis quality of PDPP and LDS. **(1)** Frames synthesized by LDS tend to be blurred (e.g. (c)) caused by LDS' underlying linear, spatial model. PDPP does not suffer from this problem; however, some synthetic frames contain spatially pe-

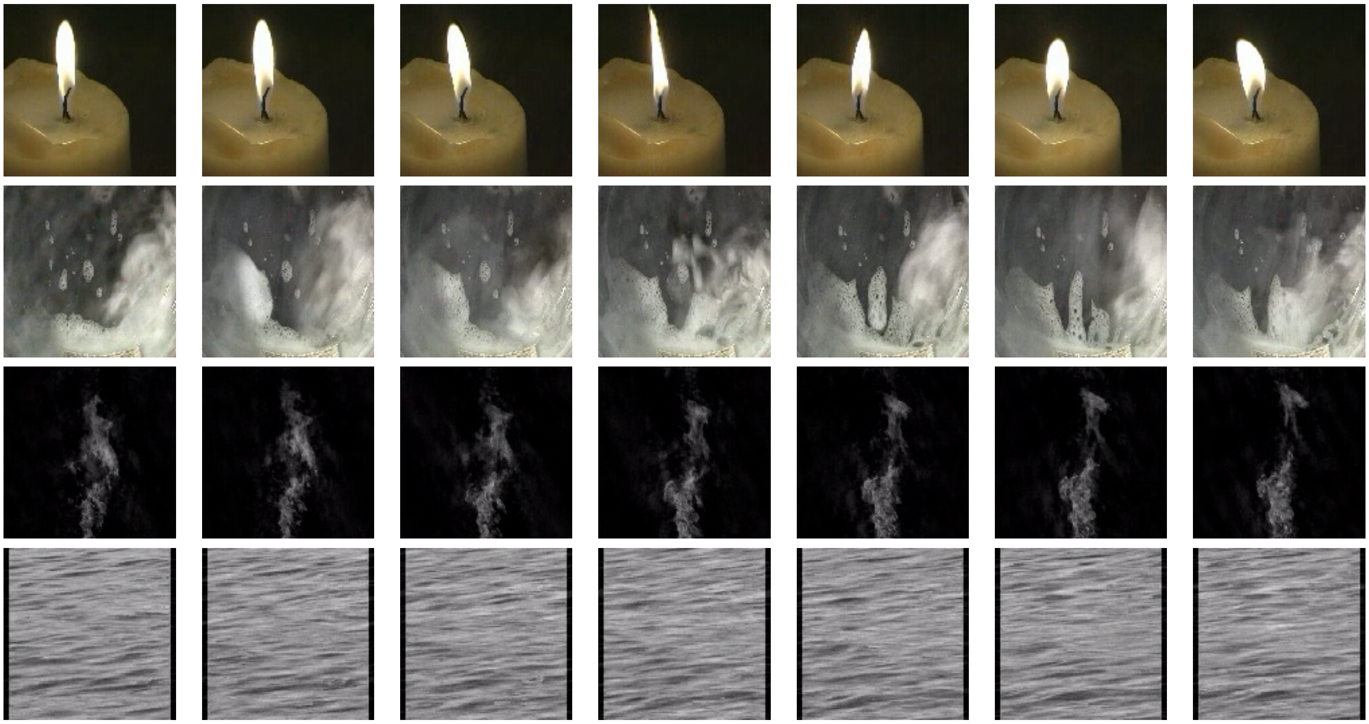
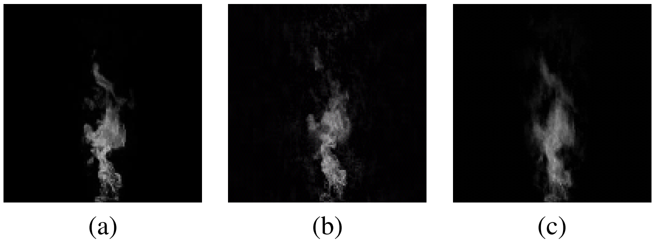


Figure 5. Sample synthetic images produced by the Cubic Spline (top two rows) and MAP (bottom two rows) synthesis methods. Videos of these sequences can be seen in the supplementary material.

riodic noise due to residual spectral leakage. **(2)** For some DT sequences, LDS produces synthetic frames whose visual quality degrades with time. For PDPP, the temporal quality degradation is considerably less. **(3)** As the order of the PDPP or LDS model decreases, the visual quality (i.e. appearance and global dynamics) of the synthesized frames degrades for both; however, the LDS frames tend to display significantly less temporal variation than PDPP. This follows from the fact that an $n < F$ dimensional PCA basis captures the maximum variance, among all n dimensional bases, of the same set of data points.



(a) (b) (c)

Figure 6. Images (a), (b), and (c) are respectively a frame from the original sequence, a MAP synthetic frame, and a synthetic LDS frame.

4.2. DT Recognition

DT recognition involves the use of both image appearance and temporal changes in appearance. For an overview of recent techniques developed for DT recognition, we refer to [7]. In [19], Doretto et al. use the LDS model parameters of each DT to recognize them. Fujita et al. use impulse responses of state variables as alternative features for recognition using the LDS model [12]. In [18], Peteri et al. propose a DT recognition algorithm based on six translation invariant features (i.e. normal optical flow and texture regularity to describe DT dynamics and appearance respectively). Recent work by Zhao et al. proposes local binary patterns (LBP) and volume local binary patterns (VLBP) as the underlying features to be used in recognizing DT sequences [25, 24]. The latter two methods are based on local descriptors, which do not incorporate the global dynamics that characterize a dynamic texture. All the above algorithms have been evaluated on subsequences or sub-blocks of the initial sequences on which the model was trained. It is unclear how they will perform on test DT sequences not already used in training. Since LDS is the model that has been used the most and seems to have the best performance among image-based models, in what follows, we compare the performance of our MAP-Based recognition method to that of LDS and evaluate its generalization performance.

Evaluation: We constructed a database of color DT sequences from the DynTex database and online sources. These sequences portray natural scenes including different bodies of water (rivers, oceans, waterfalls, etc. as in Figure 5), fire, foliage, clouds, and smoke. These DT's possess a variety of appearance and dynamics characteristics, which is a significantly richer and more challenging environment for testing our recognition algorithm as compared to the MIT temporal texture database [1]. In our database, similar looking textures (e.g. fire) may have different dynamics, while some different looking textures (e.g. smoke, water, and fire) may have similar dynamics. These DT's have different sizes and number of frames $(40 - 250)$. To expedite FFT, images in each sequence are resized to 128x128 pixels and converted to gray scale format. However, no cropping is performed. We formed two groups of DT subsequences for this purpose: the first group contains a subsequence of the frames of these formatted sequences for training the PDPP model, while the second group contains subsequences formed by randomly choosing T consecutive frames that were not used for training and do not overlap. The latter subsequences form our test set.

In our experiments, we vary either: F (number of frames used to train the PDPP model) or T (number of frames in the test subsequence). In each case, PDPP performance is compared to that of LDS, based on the implementation of [19]. Because of space restriction, we only show the results of one of these experiments, where 147 test subsequences were formed from $C = 17$ different DT classes from the database. F is equal to the first half of the C original sequences and $T = 20$ frames. Figure 7 shows portions of the confusion matrices for both PDPP and LDS. The columns represent the labelled test subsequences, while the rows represent the corresponding recognition results. The recognition is deemed correct if the recognized class is among the types shaded in gray. For example, of the 9 "flame" test subsequences, LDS recognizes 6 as "flame" and 2 as "fire$_1$", while PDPP recognizes them all as "flame". We obtained overall recognition rates of 95.2% and 46.3% for PDPP vs. LDS. This shows the greater discriminating power of PDPP. Note that if we were to accept only diagonal entries as correct recognition, then the performance disparity between PDPP and LDS would be greater.

Based on the results of these experiments, we draw the following conclusions. **(1)** The recognition performance of LDS improves as T becomes comparable to F. In our experiments, $T \ll F$, as compared to [19] where $T = F = 75$. This follows from the nature of the distance metric used (Martin distance [14]), which requires that the orders of the training and test LDS models be the same, so the test LDS model has to be expanded to the same size as the training model, as described in [9]. On the other hand, PDPP naturally accommodates any sized test sequence. **(2)** The recognition performance of both methods is directly proportional to F. However, the performance change is more significant for LDS, which means that LDS, in general, requires a larger training set than PDPP for comparable recognition rates. **(3)** The presence of a static background in the training and/or test sequences decreases the recognition rate for LDS considerably, since the LDS model does not distinguish between DT and background properties. This follows from its direct modelling of pixel intensities in the spatial domain. **(4)** A major drawback of the LDS model is its memory usage and computational complexity. In fact, all our experiments required $T \leq 30$ frames in order to run on a Pentium IV (2GB RAM) PC and keep the running time of the LDS algorithm less than 3 minutes per test sequence; otherwise, the recognition process ran out of memory. PDPP does not face this problem. For $T = 30$ frames, LDS runs in approximately 3 minutes, while our algorithm runs in about 30 seconds for the same test sequence.

Unlike the DT recognition methods we referred to earlier, we tested the generalization performance of PDPP with $C = 4$ classes (i.e. smoke, fire, water, and grass). The experiment was set up as follows: one sequence was used to learn the "smoke" class, two for "fire", five for "water", and one for "grass". A total of 141 test subsequences were formed from 7 new DT sequences, which did not participate in the training stage. Table 1 summarizes the recognition results. We conclude that the PDPP model achieves better generalization performance than LDS.

DT	LDS (%)	PDPP (%)
Smoke	100	100
Fire	87.5	100
Water	77	81
Grass	44.4	100
Weighted Average	**61**	**87**

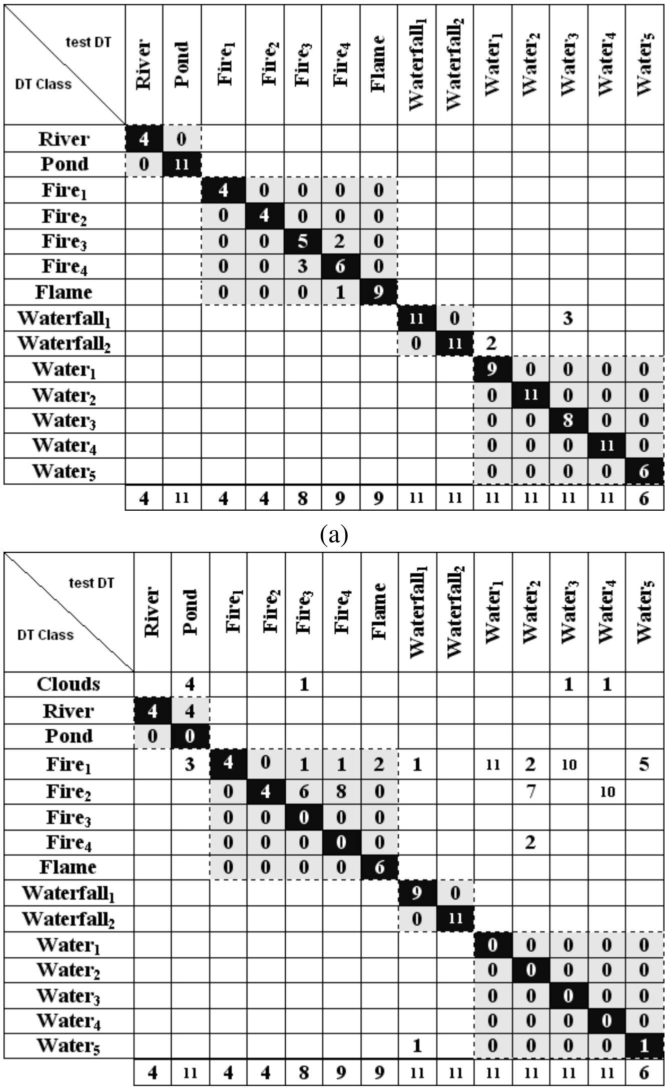
Table 1. Recognition rates for LDS vs. PDPP

5. Conclusion

In this paper, we have presented a novel spatiotemporal model (PDPP) for dynamic textures, which is based on the variation of phase content. PDPP compactly and efficiently represents both the appearance and dynamics of a DT, thus, establishing a framework for higher-level applications. We have validated the significance of our method by applying it to DT synthesis and recognition, while also comparing it to the LDS model. In the future, we plan to develop a PDPP-based algorithm for DT segmentation.

References

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



test DT / DT Class	River	Pond	Fire$_1$	Fire$_2$	Fire$_3$	Fire$_4$	Flame	Waterfall$_1$	Waterfall$_2$	Water$_1$	Water$_2$	Water$_3$	Water$_4$	Water$_5$
River	4	0												
Pond	0	11												
Fire$_1$			4	0	0	0	0							
Fire$_2$			0	4	0	0	0							
Fire$_3$			0	0	5	2	0							
Fire$_4$			0	0	3	6	0							
Flame			0	0	0	1	9							
Waterfall$_1$								11	0			3		
Waterfall$_2$								0	11	2				
Water$_1$										9	0	0	0	0
Water$_2$										0	11	0	0	0
Water$_3$										0	0	8	0	0
Water$_4$										0	0	0	11	0
Water$_5$										0	0	0	0	6
	4	11	4	4	8	9	9	11	11	11	11	11	11	6

(a)

test DT / DT Class	River	Pond	Fire$_1$	Fire$_2$	Fire$_3$	Fire$_4$	Flame	Waterfall$_1$	Waterfall$_2$	Water$_1$	Water$_2$	Water$_3$	Water$_4$	Water$_5$
Clouds		4			1							1	1	
River	4	4												
Pond	0	0												
Fire$_1$		3	4	0	1	1	2	1		11	2	10		5
Fire$_2$			0	4	6	8	0				7		10	
Fire$_3$			0	0	0	0	0							
Fire$_4$			0	0	0	0	0				2			
Flame			0	0	0	0	6							
Waterfall$_1$								9	0					
Waterfall$_2$								0	11					
Water$_1$										0	0	0	0	0
Water$_2$										0	0	0	0	0
Water$_3$										0	0	0	0	0
Water$_4$										0	0	0	0	0
Water$_5$								1		0	0	0	0	1
	4	11	4	4	8	9	9	11	11	11	11	11	11	6

(b)

Figure 7. (a) and (b) are portions of the confusion matrices for recognition of column DT as row DT by PDPP and LDS respectively. Blank entries correspond to no decisions.

[2] B. Abraham, O. I. Camps, and M. Sznaier. Dynamic texture with fourier descriptors. In *Proc. of International Workshop on Texture Analysis and Synthesis*, pages 53–58, 2005.

[3] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans. on Visualization and Computer Graphics*, pages 120–135, 2001.

[4] P. Bouthemy and R. Fablet. Motion characterization from temporal cooccurrences of local motion-based measures for video indexing. In *Proc. of ICPR*, volume 1, pages 905–908, 1998.

[5] A. Briassouli and N. Ahuja. Spatial and fourier error minimization for motion estimation and segmentation. In *Proc. of ICPR*, volume 1, pages 94–97, 2006.

[6] A. B. Chan and N. Vasconcelos. Mixture of dynamic textures. In *Proc. of ICCV*, volume 1, pages 641–647, 2005.

[7] D. Chetverikov and R. Peteri. A brief survey of dynamic texture description and recognition. In *Proc. of the International Conference on Computer Recognition Systems*, 2005.

[8] D. Comanicui, V. Ramesh, and P. Meer. Kernel-based object tracking. In *IEEE Trans. in Pattern Analysis and Machine Intelligence*, volume 25, pages 564–575, 2003.

[9] K. DeCock and B. De Moor. Subspace angles between ARMA models. In *Systems and Control Letters*, volume 46, pages 265–270, 2002.

[10] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. In *Proc. of ICCV*, volume 2, pages 1236–1242, 2003.

[11] A. Fournier and W. Reeves. A simple model of ocean waves. In *Proc. of ACM SIGGRAPH*, pages 75–84, 1986.

[12] K. Fujita and S. K. Nayar. Recognition of dynamic textures using impulse responses of state variables. *Proc. Of Third International Workshop on Texture Analysis and Synthesis*, pages 31–36, 2003.

[13] M. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 30(2), 1982.

[14] R. J. Martin. A metric for arma processes. *IEEE Trans. on Signal Processing*, 48:1164–1170, 2000.

[15] R. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. In *Proc. of ICPR*, pages 56–78, 1992.

[16] A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69:529–541, 1981.

[17] R. Peteri, M. Huiskes, and S. Fazekas. Dyntex: www.cwi.nl/projects/dyntex/ at the Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. 2006.

[18] R. Pteri and D. Chetverikov. Dynamic texture recognition using normal flow and texture regularity. In *Proc. of the Iberian Conference on Pattern Recognition and Image Analysis*, 2005.

[19] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *Proc. of CVPR*, volume 2, pages 58–63, 2001.

[20] M. B. Savvides, V. Kumar, and P. Khosla. Eigenphases and eigenfaces. In *Proc. of ICPR*, volume 3, pages 810–813, 2004.

[21] A. Schodl, R. Szeliski, D. Salesin, and I. Essa. Video textures. In *Proc. of ACM SIGGRAPH Conference*, volume 25, pages 489–498, 2000.

[22] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. *International Journal of Computer Vision*, 51:91–109, 2003.

[23] M. Szummer and R. W. Picard. Temporal texture modeling. In *Proc. of ICIP*, volume 3, 1996.

[24] G. Zhao and M. Pietikainen. Dynamic texture recognition using volume local binary patterns. In *Proc. of ECCV 2006 Workshop on Dynamical Vision*, pages 12–23, 2006.

[25] G. Zhao and M. Pietikainen. Local binary pattern descriptors for dynamic texture recognition. In *Proc. of ICPR*, volume 2, pages 211–214, 2006.