PHASE PCA FOR DYNAMIC TEXTURE VIDEO COMPRESSION

Bernard Ghanem, 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

Temporal or dynamic textures (DT's) are video sequences that are spatially repetitive and temporally stationary. DT's are temporal analogs of the well known spatial still image texture. Examples of DT's include moving water, foliage, smoke, clouds, etc. We present a new DT model that can efficiently compress DT sequences. The proposed method (Phase PCA) models the varying phase content of a DT, which is employed as the major determinant of both its dynamics and appearance. Consequently, Phase PCA combines both temporal and spatial properties in a compact spectral framework. Making use of the benefits inherent to working in the frequency domain, this model provides a significant improvement in DT compression, which can be used to improve the performance of MPEG-I encoding. We will present experimental evidence that validates this method for a variety of complex sequences, while also comparing it to the most recent DT representational model that is based on linear dynamical systems (LDS).

Index Terms— Dynamic texture, Phase, LDS, PCA

1. INTRODUCTION

Modelling of complex motion patterns in images remains unsolved in computer vision, since it poses numerous problems especially those related to reliable motion field estimation. These problems become even more complex when considering non-rigid stochastic motions (e.g. DT's). 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. So, it is evident that efficient DT compression poses a serious challenge.

Methods ([1]) relying on optical flow are convenient, since the frame-to-frame estimation of the motion field has been extensively studied and computationally efficient algorithms have been developed. However, these methods only capture temporal characteristics of the DT and are prone to error especially due to noise sensitivity and motion discontinuity. In fact, motion field estimation becomes a significantly harder task due to the non-rigid nature and complex motion prevalent in DT's.

In comparison to the previous techniques, fewer spatiotemporal models have been developed for DT's. These include the pioneering work by Nelson and Polana (1992) [2], the spatio-temporal auto-regressive (STAR) by Szummer and Picard [3], and multi-resolution analysis (MRA) trees by Bar-Joseph et al. (2001) [4]. These methods impose restrictions on the textures that can be modelled or are applied directly on pixel intensities instead of more compact representations (i.e. pixel groupings), thus, precluding feasible compression. The most recent representational DT model was developed by Doretto et al. (2003) [5], in which a linear time invariant dynamical model (LDS) is derived for the DT. This model has been applied to DT compression and synthesis [5], recognition [6], and segmentation [7]. However, the main disadvantages of this particular method include its assumption of second-order probabilistic stationarity for the DT and its computational expense, since it is applied directly to pixel intensities.

Our method uses the phase content of the DT sequence to model both its appearance and dynamics. In what follows, we justify our choice of using phase, present the details of our compression model, and provide experimental results that compare its performance to that of LDS and a standard video compression scheme, MPEG-I.

2. MOTIVATION

In general, there are some distinct advantages inherent to using the image frequency domain over the spatial domain. Most importantly, the frequency domain captures spatially global features. This is more suited for high-level applications including DT compression and recognition. Also, frequency analysis has been shown to be robust to some variations (e.g. illumination changes [8]). Furthermore, computationally efficient algorithms and specific hardware are available for the computation of frequency transforms (e.g. FFT).

Let us motivate why we propose using the phase content of a DT to dually represent its appearance and its temporal variations. First, Monson Hayes ([9]) proved that multi-dimensional signals can be reconstructed from their phase content only. Certain signals that have symmetric factors in their Z-transform are excluded. In fact, if a hybrid image is constructed out of the amplitude spectrum of one image and the phase spectrum of another, an iterative algorithm exists that will reproduce the original image, from which the phase spectrum was extracted. Figure 1 shows a result of the phase-only reconstruction algorithm applied to ocean and fire DT's. In what follows, we will assume that DT sequences have this phase-only reconstruction property.

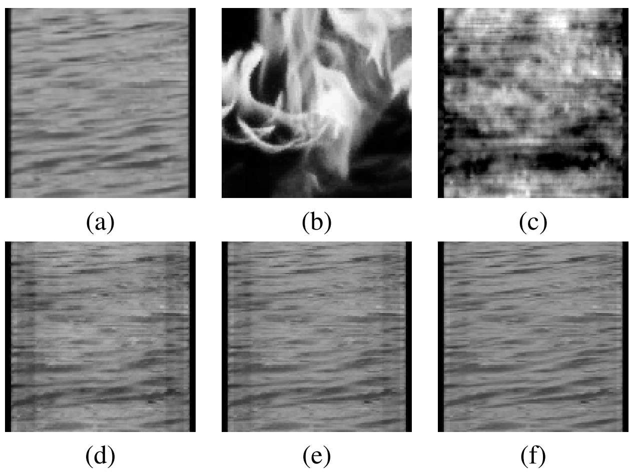


Fig. 1. (a) is the phase spectrum image. (b) is the amplitude spectrum image. (c) is the hybrid image. (d), (e), and (f) are the images after 50, 100, and 250 iterations respectively.

Next, we empirically justify the fact that the variations in DT phase capture most of the DT's temporal characteristics and not the variations in DT amplitude. Figure 2 shows that many more principal components are required to represent 80% of the variation in DT phase than to represent the same amount in DT amplitude. Hence, we conclude that it is relevant to represent only the phase spectrum of DT's for the purpose of compression.

3. PHASE PCA COMPRESSION MODEL

For DT compression, we perform PCA on the DT feature vectors, which are the vectorized half spectra of DT phase for the frames in the DT sequence. Only half the phase spectrum is required due to the conjugate symmetry property of the FFT. Here, we note that the DT frames are preprocessed with an appropriately sized Hanning window to mitigate spectral leakage. We will use the notation in (1) and (2) to denote the PCA basis (**A**), the feature vectors ($\vec{\Phi}$), their projections in the PCA space ($\vec{x}$), and the mean feature vector ($\vec{\Phi}_m$). For an image of size MxN, the length of the feature vector is K = $\frac{MN}{2}$, so that the size of **A** is KxL, where L is the number of principal components that have been selected to represent the data. For complete representation, L = F, which is the number of frames in the DT sequence.

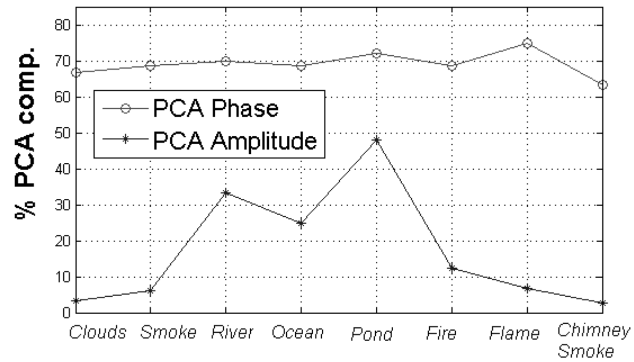


Fig. 2. PCA components for DT phase and amplitude.

$$\vec{\Phi} = A\vec{x} + \vec{\Phi}_m \tag{1}$$

$$\vec{x} = A^T(\vec{\Phi} - \vec{\Phi}_m) = [x_1 x_2 \cdots x_L]^T \tag{2}$$

Due to the symmetry of DT phase, considerable compression is possible for each individual frame. We note here that additional compression can be achieved by neglecting frequencies with low energy, mainly in the high frequency bands and by using the dimension reduction option inherent to PCA. However, we must ensure that the visual quality of the compressed frames be "close" as possible to that of the uncompressed ones.

Below, we first present the compression rates that can be achieved by both Phase PCA and LDS in terms of the number of principal components used for DT representation. Then, we describe how the compression performance of our method can be enhanced by forming a more compact PCA space.

3.1. Basic Phase PCA (BPP) Compression

In this section, we will present the overall compression rate that can be achieved by reducing the dimensionality of the phase PCA space. So, with this generic layout, we can compute the expected overall compression rate (R_{comp}) for an arbitrary DT sequence as in (3). In fact, since L $\leq$ F and $\frac{MN}{2}$ $\gg$ F, then the removal of any PCA component will lead to significant data compression.

$$R_{comp} = 1 - \frac{size(\mathbf{A}) + size(\vec{\Phi}_m) + \text{\# of PCA coefficients}}{MNF}$$

$$= 1 - \left[\frac{L+1}{2F} + \frac{L}{MN}\right] \approx 1 - \frac{L+1}{2F} \quad (L \ll MN) \tag{3}$$

The main factor dictating the extent of the data compression is $\frac{L}{F}$, the fraction of the PCA components used in the representation. Also, note that even with a complete representation (L=F), the compression rate is about 50%. This is due to the fact that only half the phase spectrum is used to represent the DT, so the amplitude spectrum must be initially determined by the iterative process mentioned in the context of phase-only reconstruction. In fact, sending a single magnitude response is possible with a slight decrease in the compression rate.

Using LDS, we require two matrices (**A** and **C**) and the initial state $\vec{x}_0$ in order to reconstruct the DT. The dimension of **A** is L'xL' and that of **C** is 2KxL', where L' represents the order of the LDS and K = $\frac{MN}{2}$ as defined before. So, the overall compression rate in the case of a DT with F frames is estimated as in (4). Note that if both methods render the same compression rate, then the LDS method requires about half the number of principal components used by BPP.

$$\begin{aligned} R_{comp} &= 1 - \frac{size(\mathbf{A}) + size(\mathbf{C}) + size(\vec{x}_0)}{MNF} \\ &= 1 - \frac{L'L' + 2KL' + L'}{MNF} \approx 1 - \frac{L'}{F} \end{aligned} \tag{4}$$

3.2. Principal Difference Phase PCA Compression

In this section, we describe a different, yet equivalent DT phase spectrum, which forms a framework that allows for further DT compression. This is achieved when the difference in phase spectra between every two consecutive frames is forced to be a principal angle, since the dynamics of the DT sequence is embedded in the frame-to-frame change of the phase and not the absolute value of each individual frame.

This can be better understood by considering the example of a simple rigid body translating in a video sequence with constant displacement. In this case, the phase difference between every two consecutive frames is the same. Principal Difference Phase PCA (PDPP) dictates that each of the ordered DT phase spectra be replaced by the sum of the previous spectrum and the principal angle of the difference between the original spectrum and the previous one. If this is done, the feature vectors will be collinear in the high-dimensional space and a more compact PCA representation can be acquired. Figure 3 shows a 2D version of this situation, where A, B, and C represent the original feature vectors while A, B', and C' represent the transformed ones. Obviously, we can represent A, B', and C' in a lower dimensional space (i.e. the line connecting these collinear points) than the one required to span A, B, and C.

In Figure 4, we show that for the same number of PCA components (i.e. compression rate), PDPP can represent significantly more variation in DT phase than the BPP method described in **Section (3.1)**.

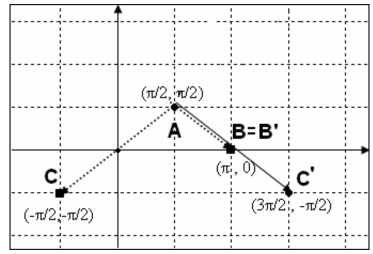
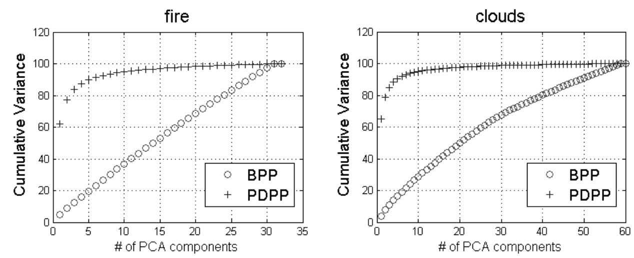


Fig. 3. BPP vs. PDPP in 2D



(a) (b)

Fig. 4. BPP vs. PDPP for two DT's

4. EXPERIMENTAL RESULTS

In this section, we present experimental results that validate the significance of our proposed method for DT compression and compare its performance to that of LDS. Figure 5 ((a)-(c)) compares the performance of LDS, BPP, and PDPP over a range of compression rates, that are proportional to the number of principal components used. We note here that the compression rate is computed from the number of principal components required by LDS. It is evident from these plots that BPP, in general, outperforms the LDS compression scheme, mainly due to the fact that only half the phase spectrum is modelled. Furthermore, PDPP renders a significant improvement over BPP even at very low compression rates. In (d), we show the temporal performance of each compression scheme at a compression rate of 63%. We notice that both BPP and PDPP tend to oscillate about a steady PSNR value, while LDS performance decreases with time. This is due to the fact that LDS produces a DT frame as a linear combination of the L' chosen principal components, which are computed from L' frames of the original DT and not all of them (i.e. F).

Next, we compare PDPP compression to that of the MPEG-I standard, as portrayed in Figure 6. Here, we define the compression rate for each case as the ratio of the size of the MPEG-I video to that of the original, uncompressed video.

From the above plots, we see that our method outperforms MPEG-I in the first three DT's, but does worse for the fire sequence. This improvement is primarily due to the more compact representation of the temporal characteristics of the DT inherent to PDPP. Since MPEG-I requires computation of motion fields and these estimates based on optical flow al-

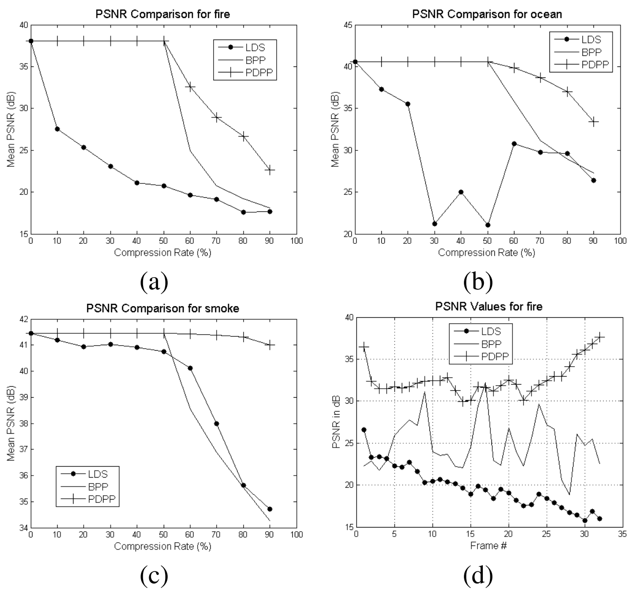


Fig. 5. LDS vs. BPP vs. PDPP in terms of compression rate

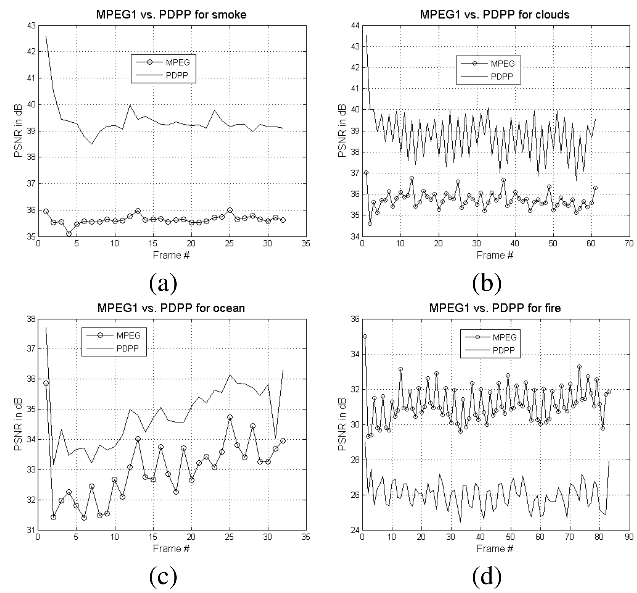


Fig. 6. MPEG-I vs. PDPP at compression rates of 95%, 19%, 33%, and 32% respectively for (a)-(d)

gorithms tend to degrade with the complexity of the motion, MPEG-I does not perform as well for the random-like motion of DT's. On the other hand, the fire sequence requires more PCA components to render the same compression rate mainly due to the nature of the DT and its background.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel compression model for DT's, which represents both appearance and temporal information based on DT phase content. This model was shown to outperform the most recently used DT model in the literature. Moreover, we compared our model with MPEG-I and showed that more compression is possible, when a more compact representation of the temporal properties of the DT is available. In the future, we would like to extend this model to incorporate only high energy frequencies and examine how information theoretic coding might improve its compression performance.

6. REFERENCES

[1] P. Bouthemy and R. Fablet., "Motion characterization from temporal cooccurrences of local motion-based measures for video indexing," in *Proc. of ICPR*, 1998, vol. 1, pp. 905–908.

[2] R.C. Nelson and R. Polana, "Qualitative recognition of motion using temporal texture," in *Proc. of ICPR*, 1992, pp. 56–78.

[3] M. Szummer and R. W. Picard, "Temporal texture modeling," in *Proc. of ICIP*, 1996, vol. 3.

[4] 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*, pp. 120–135, 2001.

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

[6] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *Proc. of ICPR*, 2001, vol. 2, pp. 58–63.

[7] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *Proc. of ICCV*, 2003, vol. 2, pp. 1236–1242.

[8] M. B. Savvides, V. Kumar, and P.K. Khosla, "Eigenphases and eigenfaces," in *Proc. of ICPR*, 2004, vol. 3, pp. 810–813.

[9] 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*, vol. 30, no. 2, 1982.