J Inf Process Syst, Vol.12, No.1, pp.46~56, March 2016
http://dx.doi.org/10.3745/JIPS.02.0009

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Content Based Dynamic Texture Analysis and Synthesis Based on SPIHT with GPU

Premanand P. Ghadekar* and Nilkanth B. Chopade**

Abstract
Dynamic textures are videos that exhibit a stationary property with respect to time (i.e., they have patterns that repeat themselves over a large number of frames). These patterns can easily be tracked by a linear dynamic system. In this paper, a model that identifies the underlying linear dynamic system using wavelet coefficients, rather than a raw sequence, is proposed. Content based threshold filtering based on Set Partitioning in a Hierarchical Tree (SPIHT) helps to get another representation of the same frames that only have low frequency components. The main idea of this paper is to apply SPIHT based threshold filtering on different bands of wavelet transform so as to have more significant information in fewer parameters for singular value decomposition (SVD). In this case, more flexibility is given for the component selection, as SVD is independently applied to the different bands of frames of a dynamic texture. To minimize the time complexity, the proposed model is implemented on a graphics processing unit (GPU). Test results show that the proposed dynamic system, along with a discrete wavelet and SPIHT, achieve a highly compact model with better visual quality, than the available LDS, Fourier descriptor model, and higher-order SVD (HOSVD).

Keywords
Discrete Wavelet Transform, Dynamic Texture, GPU, SPIHT, SVD

1. Introduction

Dynamic textures are videos that exhibit a stationary property with respect to time (i.e., they have the patterns that repeat themselves over a large number of frames). Examples of such dynamic textures are flame, pond, grass, etc. Doretto et al. [1] have proposed a dynamic texture model that exploits the temporal property of a dynamic texture as a linear dynamic system driven by an independent and identical process. The proposed model is especially attractive as it provides a mathematical model that can be used for the analysis and synthesis of a dynamic texture. This model is demonstrated with several examples in [1], and is capable of effectively representing dynamic textures with the model of a low dimensional space. This approach exploits the temporal correlation among the pixels, but it fails to exploit the spatial and chromatic correlation among the pixels. The model proposed by Abraham et al. [2] deals with the spatial and temporal correlation among the dynamic texture frames. The drawbacks of this model have higher computational and storage complexities. The models by Doretto et al. [1] and Abraham et al. [2] do not consider chromatic redundancy within the dynamic texture frames.

Manuscript received August 26, 2013; accepted December 24, 2013; onlinefirst October 28, 2014.
Corresponding Author: Premanand P. Ghadekar (ppghadekar@gmail.com)
* Dept. of Computer Engineering, VIT Pune 411037, India (ppghadekar@gmail.com)
**Dept. of E & TC Engineering, PCCOE, Pune 411033, India (nbchopade@gmail.com)

In this paper, a technique that exploits the spatial, temporal, and chromatic correlation of the different dynamic texture frame sequences is proposed. Instead of considering the pixel intensity for the linear dynamic system, a discrete wavelet transform is applied to the individual dynamic texture frames. This results in frames having wavelet coefficients in four different frequency bands, namely LL, LH, HL, and HH. The main idea of this paper is to apply content based threshold filtering based on Set Partitioning in a Hierarchical Tree (SPIHT) [3,4], so as to have more significant information in fewer parameters for singular value decomposition (SVD). Also, instead of arranging the whole frame into column vectors for SVD [1], the different bands of dynamic texture frames are arranged into different column matrices for SVD.

The paper is organized as follows: In Section 2, related work is given, which comprised of different papers that lists existing algorithms advantages and disadvantages. In Section 3, a brief overview of the dynamic texture analysis and synthesis model proposed by Doretto et al. [1] is given. In Section 4, our proposed method of incorporating wavelet coefficients and SPIHT is described. In Section 5, experiments comparing the proposed approach against the available approaches are presented. Finally, the conclusions are given in Section 6.

2. Related Work

A lot of work has been done on the dynamic texture synthesis and a number of methods have been proposed. Each method has its advantages and disadvantages that consider different measuring factors, such as the memory required to store the synthesis parameter, the visual quality of the synthesized dynamic texture, time complexity, etc. The understanding of different scenarios requires the study of various synthesis algorithms along with transform algorithms like Fourier, discrete cosine transform, and Wavelet. The proposed model requires a study of the various basic concepts that include different color formats, SPIHT, and conversions among them.

Stefano Soatto et al. [5] presented the characterization of a dynamic texture that poses the problem of analysis and synthesis on a firm analytical footing. The borrowed tool from system identification captures the essence of a dynamic texture. This can be done by using models. The proposed model has the predictive power to extrapolate the synthesis sequence to infinity. This paper does not deal with the spatial correlation among the pixels.

Filip et al. [6] proposed a novel hybrid method for colour dynamic texture modelling. The method is based on Eigen analysis of dynamic texture images and the subsequent pre-processing and modelling of temporal interpolation. Eigen coefficients use a causal auto-regressive model. This model demonstrates good performance for most of the tested dynamic textures, which depends mainly on the properties of the original sequence. This proposed algorithm does not exploit the spatial, temporal, and chromatic correlation among the pixels.

The model by Costantini et al. [7] is flexible and natural, as it causes dimension reduction in the spatial, temporal, and chromatic domain. The model is well suited to dynamic texture synthesis on devices limited by computational power and memory. Here, the analysis part is more expensive as compared to the existing algorithms.

The algorithm by Li et al. [8] is general and automatic, and it works well on different types of textures. Experimental results demonstrate that the approach can reconstruct dynamic texture sequences with promising visual quality and fewer coefficients. The scheme is incapable of searching for a global optimal solution, which would improve the visual synthesis quality.

3. Dynamic Texture Modelling System

It has been shown that dynamic textures can be represented as an output of the linear dynamic system [1]. These patterns can be easily tracked by a linear dynamic system. The approach described by [1] uses raw image vectors for the synthesis of a dynamic texture. The modeling of a dynamic texture includes analysis and synthesis. An analysis consists of; 1) finding an appropriate subspace that represents the trajectory motion, and 2) identifying the trajectory with the help of the dynamical system theory. The appropriate subspace can be found by retaining the first n singular values of the decomposition.

$$\bar{y}_1^{\tau} = [\overline{y_1} \dots \dots \overline{y_2}] = U_n S_n V_n^T \quad (1)$$

Assume that $C = U_n$ and $X = S_n V_n^T$, then a generic column vector is given by:

$$Y_k = Cx_k + e \quad (2)$$

where x_k is the k^{th} column vector of matrix X and e is a residual error. The equation above can be written as:

$$Y_k = c_1 x_{1k} + c_2 x_{2k} + \text{------} + c_n x_{nk} + e \quad (3)$$

where c_i with $i=1,\dots,n$ indicates the i^{th} column vector of matrix C.

Identification of system dynamics starts from matrix $X \epsilon R^{n \times \tau}$, which is the time evolution of a dynamic texture. The evolution of x_k is done using the MAR model.

$$x_k = Ax_{k-1} + BV_k \quad (4)$$

where $A \epsilon R^{n \times n}$ and $B \epsilon R^{n \times nv}$ are the matrix computed from matrix X and V_k is the Gaussian random noise. Finally, a dynamic texture can be synthesized using the following equations:

$$x_{k+1} = Ax_k + BV_k \quad (5)$$

$$Z_k = Cx_k + d \quad (6)$$

4. Dynamic Texture Model Along with Wavelet Coefficients and Spiht Based Thresholding

The approach described above uses raw image vectors for the analysis and synthesis of a dynamic texture. As mentioned earlier it does not deal with the spatial correlation among the pixels and thus, results into a large order system. In order to have a minimum order system a discrete wavelet transform is applied before the analysis and synthesis of a linear dynamic texture are carried out. The discrete wavelet transform results in different frequency coefficients. Most of the frequency coefficients in LH, HL, and HH frequency bands are not useful for the overall quality of a synthesized dynamic texture. So, the wavelet coefficients, which significantly contribute in the quality of synthesized frames, are considered in this model.

Roberto Costantini et al. [7] synthesized a dynamic texture using different color spaces instead of RGB color space and achieved a similar synthesis performance by using half of the model coefficients and less computational power. The proposed model considers all three redundancies (i.e., spatial, temporal, and chromatic). The proposed approach can be interpreted as a four stage modelling process, which is as follow: 1) chromatic correlation is exploited using different color spaces; 2) the spatial correlation of the pixels is exploited using a discrete wavelet transform; 3) content based threshold filtering based on SPIHT; and 4) time correlation among the different dynamic texture frames is exploited by learning about time evolution of an operator using the model proposed by [1]. Fig. 1 represents the overall flow of the proposed model. The idea of the proposed model is described in more detail next.

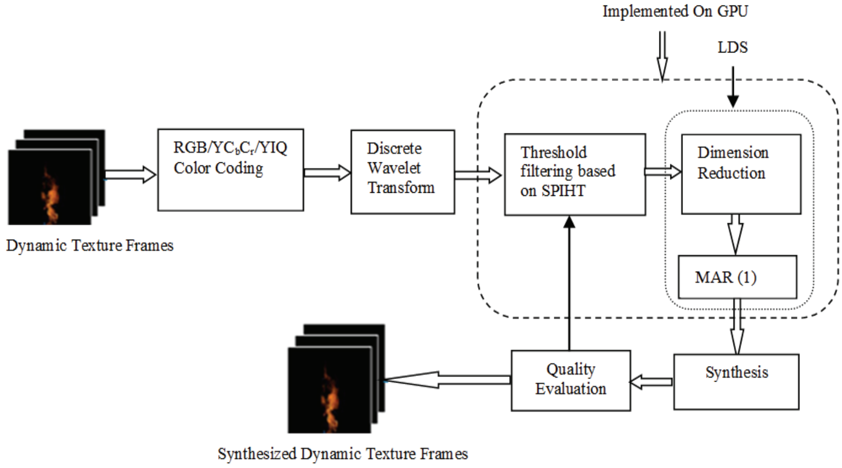

Fig. 1. The proposed dynamic texture analysis and synthesis model with Wavelet and SPIHT based threshold filtering.

In the proposed dynamic texture analysis and synthesis model, frames are extracted from a dynamic texture. Color space conversion is applied to these extracted frames [8] where RGB color components are converted to YC_bC_r color components [9] and where Y is luminance and C_b and C_r are chromatic blue and chromatic red color components, respectively. These color spaces help to analyze the luminance and chrominance components separately. A discrete wavelet transform is then applied to YC_bC_r color space frames and it leads to four different bands, namely LL, LH, HL, and HH [10]. The LL band contains low frequency or significant components and HH contains high frequency or insignificant components.

According to the SPIHT concept [11], firstly wavelet transform is applied then threshold is calculated based on the transform data. Generally, a set of fixed threshold values has an impact on the efficiency of model coefficients. The set of minimum selected threshold values leads to a lot of insignificant data coefficients. On the other hand, a set of maximum selected threshold values leads to a loss of most of the significant data coefficients [12]. So, the best threshold must be data dependent. In the SPIHT coding technique for threshold value section, the maximum value is selected from wavelet transform coefficients of an image. Then '*N*' (i.e., number of passes) is obtained by logarithms of the maximum selected transformed value with base 2. Mathematically, it is calculated as:

$$N = \lfloor log_2 \, Cmax \rfloor$$

where C_{max} is the maximum selected transform coefficient and N is the number of passes.

The threshold for each pass is calculated as:

$$Threshold = 2^N$$

For each successive pass, the value of N is decremented by 1 and the threshold value is calculated as shown above. Here, the number of passes is dependent on the available bandwidth and quality evaluation parameters.

In the proposed model, the maximum value is selected from all of the dynamic texture frames. To have uniformity in threshold selection, the minimum value is selected from all of the maximum selected values. The number of passes and threshold for each pass is calculated as shown above. The number of passes should be incremented until better quality is achieved.

The human visual system is more sensitive to variations in brightness (i.e., luminance) than color (i.e., chrominance) [13]. Therefore, these color components (i.e., C_b and C_r) are decimated by a factor of 2, while retaining the full luminance resolution. This helps in achieving a more compact model for the dynamic texture.

Doretto et al. [1] and Soatto et al. [5] have shown that a dynamic texture can be expressed as an output of the linear dynamic system. In the proposed system discrete wavelet transform based bands are then given as an input to the linear dynamic system (i.e., SVD). The same band of all dynamic texture frames are temporally correlated with each other. So, instead of applying SVD on dynamic texture frames by arranging all of the frequency bands of a frame into single column vectors, it is applied onto different frame bands independently. Fig. 2 shows an insight working of the proposed model. This proposed technique helps to attain more flexibility with component selection.

The LL band contains most of the significant information, while the LH, HL, and HH bands contain less significant information. Also, the SVD arranges information in descending order (i.e., from the most significant information to the least significant information). So, selecting a fewer number of components from the LH, HL, and HH bands of the discrete wavelet transform and a greater number of components from the LL band of the wavelet transform provides a more compact representation of a dynamic texture with better visual quality. Skipping more components from the LH, HL, and HH bands of the discrete wavelet transform does not affect the visual quality, as these are less important coefficients.

The compact model after LDS can be stored or sent to the receiver for synthesis. Mathematically, synthesis can be done as shown in Eqs. (5) and (6). Finally, the up sample of the C_b and C_r color components should be performed by adding a factor of 2 and applying an inverse discrete wavelet transform to get the synthesized frames. The proposed model was also implemented using the YIQ color model and results in the same peak signal-to-noise ratio (PSNR) and model size with some variations.

In the proposed system, threshold filtering is based on the SPIHT and LDS parameter learning blocks, which require more computational time. As such, these two blocks were implemented on the graphics processing unit (GPU).

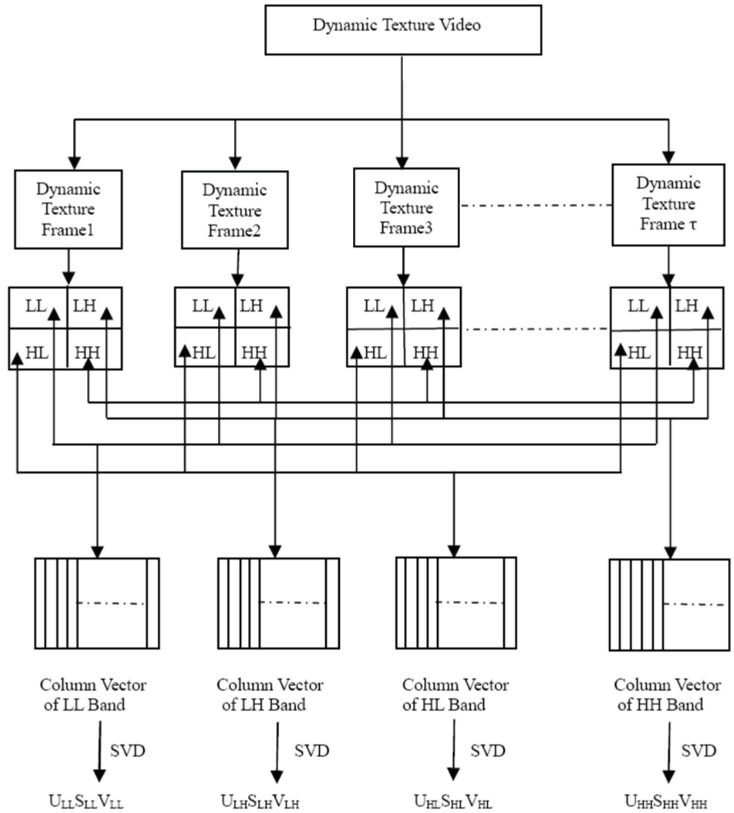

Fig. 2. Insight working of the proposed model.

The steps for the proposed algorithm are listed below.

1) Extract frames from dynamic texture.
2) Apply the YC_bC_r or YIQ color encoding technique.
3) Apply the discrete wavelet transform.
4) Select the maximum intensity (i.e., the C_{max} value) from each dynamic texture frame.
5) Take $N=\lfloor log_2\,Cmax \rfloor$ of all intensity values and select the minimum of all the values.
6) The threshold value is calculated as: $Threshold=2^N$
7) Apply filtering based on the calculated threshold value.
8) Down sample the C_b and C_r color components.
9) Apply the SVD on different bands of the wavelet transform individually.
10) The number of components selected is based on different bands of the wavelet transform.
11) Different blocks are synthesized using analyzed parameters.
12) Up sample the C_b and C_r color components.

Finally, apply an inverse discrete wavelet transform to get the reconstructed dynamic texture frames.

5. Results

The synthesis of a dynamic texture introduces some distortion in the reconstructed frames. Therefore, the quality evaluation parameter is an important issue in image processing. Several experiments have been performed using the proposed dynamic texture analysis and synthesis model. The execution of the proposed model was also done on GPU [14, 15]. GPU is designed to take advantage of parallel computation (i.e., the same program is executed on many data elements in parallel). This can be achieved with the help of thousands of GPU cores. As the same program is executed for each data element, there are fewer requirement for flow control. As the same program is executed on many data elements and has high arithmetic intensity, the memory access time is hidden in calculation instead of inside a big data cache.

CPU has very few cores and it is optimized for serial processing, while GPU has thousands of smaller and efficient cores that are designed for parallel processing. CPU + GPU is a powerful combination because a serial portion of the code is run on CPU, while a parallel portion of the code, which is independent of each other, runs on GPU [16]. In the proposed model, threshold filtering based on SPIHT and a dimension reduction technique is implemented on GPU to reduce time complexity.

Dynamic texture sequences, which were used for testing purposes, were taken from the MIT temporal texture database and are available at [17]. The raw sequences include the flowing water sequence (352 pixels × 288 pixels × 251 frames), steady sequence (352 pixels × 288 pixels × 251 frames), flush water sequence (352 pixels × 288 pixels × 251 frames), flame sequence (320 pixels × 240 pixels × 89 frames), grass sequence (224 pixels × 144 pixels × 100 frames), tides sequence (352 pixels × 288 pixels × 250 frames), river water sequence (352 pixels × 288 pixels × 251 frames), canal water sequence (352 pixels × 288 pixels × 251 frames), shower sequence (352 pixels × 288 pixels × 251 frames), bird sequence (320 pixels × 240 pixels × 135 frames), tex1 sequence (352 pixels × 288 pixels × 71 frames), and the pond sequence (368 pixels × 240 pixels × 150 frames). The available methods were compared with the proposed method in different aspects. Table 1 represents the results obtained from the proposed model on different dynamic textures.

Table 1. Results obtained from the proposed model

Dynamic texture	Original size (MB)	Compressed size[a] (MB)	PSNR[a] (dB)	Compression ratio[a] (%)
Flowing water	72.8	9.22	39.30	87.33
Steady	72.8	5.10	35.71	92.99
Flush water	74.8	8.41	35.29	88.75
Flame	19.5	4.23	29.81	77.74
River water	72.8	5.14	31.01	92.93
Canal water	72.8	1.83	30.33	97.41
Shower	72.8	2.15	33.14	97.04

[a] Proposed model.

Table 2 compares the proposed method with the LDS [1], and the Fourier descriptor method [2] for the flame sequence. While setting the same PSNR for all of the models, the number of model coefficients for the synthesis of a dynamic texture using the proposed model was significantly less than the LDS and Fourier descriptor model.

Table 2. Model coefficients for the flame sequence

No.	Model coefficients (by SVD)	PSNR (SVD)	Model coefficients (SVD+FFT)	PSNR (SVD+FFT)	Model coefficients[a] (SVD+Wavelet+SPIHT)	PSNR[a] (SVD+Wavelet+SPIHT)
1	9449960	31.38	9502183	31.61	2031832	31.54
2	10602405	32.20	10366068	32.26	2326352	32.66
3	11754850	33.03	11230009	33.01	2620872	33.79
4	12907295	34.23	12093966	34.63	2915392	34.90
5	14059740	35.53	12957883	34.90	3062652	35.45

[a]Proposed model.

Fig. 3 represents the model coefficients of a reconstructed flame sequence against an average PSNR. Fig. 3 shows the performance comparison between the proposed method, the LDS method [1], and the Fourier descriptor method [2] by using same average PSNR (dB) for all models. In Fig. 3, model coefficients and average PSNR (dB) of different models are taken from Table 2.

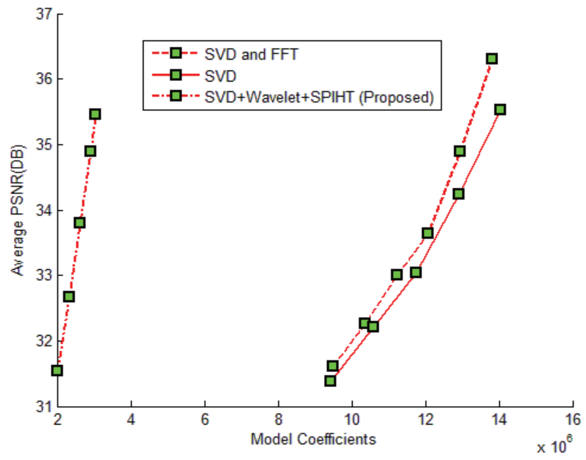

Fig. 3. Performance comparison between the proposed method, the LDS method, and the Fourier descriptor method.

Table 3 compares the model size of the proposed method with LDS+Wavelet, and higher-order SVD (HOSVD) [7] by setting the same average PSNR (dB) for a different dynamic texture sequence.

Table 4 shows the time required to synthesize a dynamic texture on CPU and GPU. The execution of a dynamic texture on CPU requires more time because of LDS and SPIHT based threshold filtering. So, these two blocks are implemented on the GPU by parallel processing and thus, require less time as compared to CPU. Fig. 4 shows a graph representing the time required for CPU and GPU for a different dynamic texture. Fig. 5 shows the frames sampled (frame no 1, 17, 49, and 88) from an original flame sequence and from different models [1, 2, 7]. The proposed model provides a better visual quality of reconstructed frames. The proposed model was implemented on the CPU and GPU configurations listed below.

CPU: Intel Core i5-2410M, CPU clock: 2.30 GHz, RAM memory: 4 GB.

GPU card: GeForce GT 525M, Cuda cores: 96, Processor clock: 1,200 MHz.

Table 3. Model size for a sequence of dynamic textures

Dynamic texture	Original size (MB)	Method	Model size (MB)	Compression ratio	PSNR (dB)
Flame	19.5	LDS+Wavelet	7.75	60.38	35.46
		HOSVD	7.2	63.07	
		Proposed	4.47	77.46	
Grass	9.24	LDS+Wavelet	7.87	14.78	32.26
		HOSVD	6.5	29.65	
		Proposed	4.46	51.69	
Tides	72	LDS+Wavelet	56.6	21.90	30.47
		HOSVD	33.5	53.79	
		Proposed	30	58.4	

Table 4. GPU time complexity

No.	Dynamic texture	Model components (no.)	Time complexity with CPU (sec)	Time complexity with CPU+GPU (sec)	Speedup (%)
1	Flame	50	603.222	344.098	1.75
2	Grass	60	270.91	116.44	1.62
3	Pond	60	628.14	328.02	1.91
4	Bird	80	877.63	283.23	3.09
5	Tex1	60	611.18	359.39	1.70

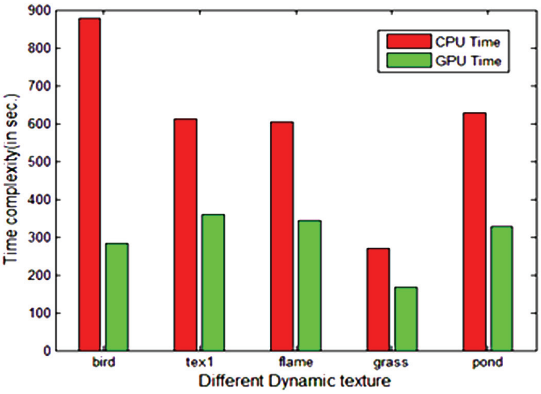

Fig. 4. CPU and GPU time for a different dynamic texture.

6. Conclusion

Dynamic texture analysis and the synthesis model using SVD along with wavelet transform and SPIHT has been proposed. The proposed scheme uses wavelet coefficients of frames in the sequence, instead of the frames themselves, to learn the dynamics of the texture. For the efficient use of wavelet transform, SPIHT based threshold filtering is applied to transformed wavelet coefficients. In doing so, a higher compression ratio is obtained than is available in the linear dynamic system, the linear dynamic system along with wavelet transform, and the HOSVD. SPIHT based threshold filtering helps to get a more compact representation of a dynamic texture. The results obtained from LDS, LDS along with Fourier transform, LDS along with wavelet transform, and LDS along with wavelet transform, and SPIHT in combination with their comparison is shown in Section 5. The experimental results demonstrate that the proposed system describes the dynamic texture in fewer parameters and the synthesized dynamic texture sequence visually resembles the original one. YC_bC_r, and YIQ color coding deals with the chromatic correlation among the pixels and thus, helps in obtaining a higher compact representation of a dynamic texture.

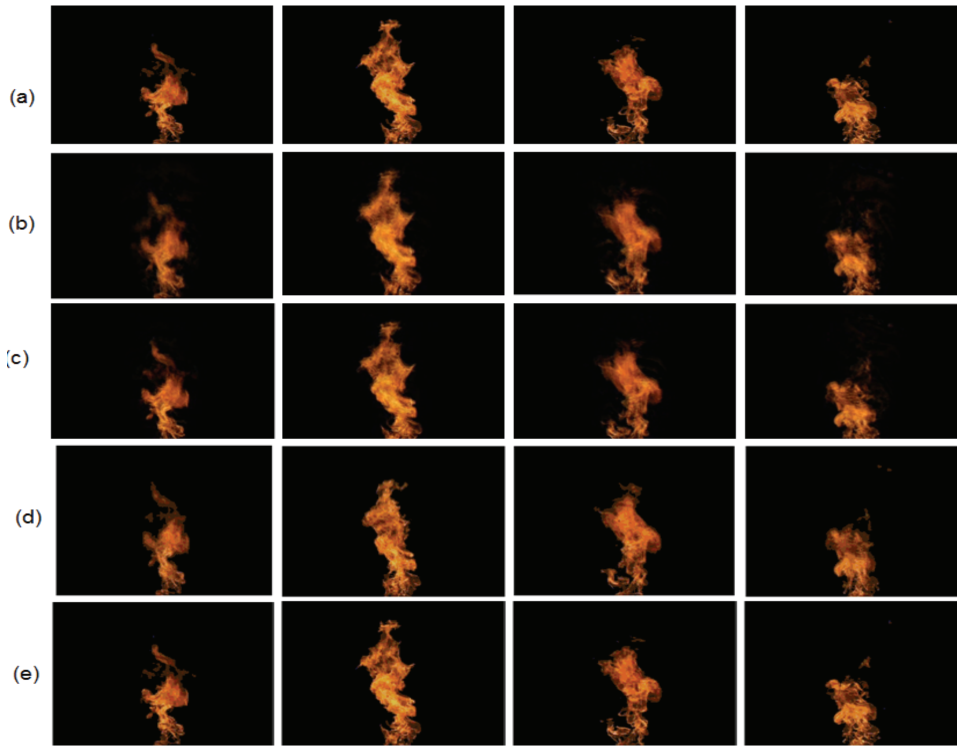

Fig. 5. (a) Samples from an original flame sequence, (b) sample frames reconstructed using LDS, (c) sample frames reconstructed using the Fourier descriptor method, (d) sample frames reconstructed using the HOSVD method, and (e) sample frames reconstructed using the proposed method

References

[1] G. Doretto, A. Chiuso, S. Soatto, and Y. N. Wu, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91-109, 2003.

[2] B. Abraham, O. I. Camps, and M. Sznaier, "Dynamic texture with Fourier descriptors," in *Proceedings of the 4th International Workshop on Texture Analysis and Synthesis*, Beijing, China, pp. 53-58, 2005.

[3] S. G. Mallat, "Multifrequency channel decomposition of images and wavelet models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 2091-2110, Dec. 1989.

[4] S. Jayaraman, S. Esakkirajan, and T. Veerakumar, *Digital Image Processing*. New Delhi: Tata McGraw Hill Education, 2009.

[5] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV2001)*, Vancouver, Canada, pp. 439-446, 2001.

[6] J. Filip, M. Haindl, and D. Chetverikov, "Fast synthesis of dynamic colour textures," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR2006)*, Hong Kong, China, pp. 25-28, 2006.

[7] R. Costantini, L. Sbaiz, and S. Susstrunk. "Higher order SVD analysis for dynamic texture synthesis," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 42-52, 2008.

[8] C. Li, J. Wang, L. Ye, and H. Wang, "A novel method of dynamic textures analysis and synthesis," in Proceedings of the IEEE International Joint Conference on Computational Sciences and Optimization (CSO2009), Hainan, China, pp. 328-332, 2009.

[9] A. Ford and A. Roberts, "Colour space conversions," Westminster University, London, 1998.

[10] P. N. Topiwala, *Wavelet Image and Video Compression*. Boston, MA: Kluwer Academic Publishers, 1998.

[11] J. R. Ding and J. F. Yang, "A simplified SPIHT algorithm," *Journal of the Chinese Institute of Engineers*, vol. 31, no. 4, pp. 715-719, 2008.

[12] W. A. Pearlman and A. Said, "Set partition coding: part I of set partition coding and image wavelet coding system," *Foundations and Trends in Signal Processing*, vol. 2, no. 2, pp. 95-180, 2008.

[13] Y. Itoh and T. Ono, "Up-sampling of YC_bC_r 4:2:0 image exploiting inter-color correlation in RGB domain," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2204-2210, Nov. 2009.

[14] W. Smith, "Matlab's parallel computation toolbox and the parallel interpolation of commodity futures curves," Mar. 2010; http://commoditymodels.files.wordpress.com/2010/03/matlab-parallel-computing-toolbox-and-interpolating-futures-curves.pdf.

[15] NVIDIA, "Accelerating MATLAB with CUDA using MEX files," WP-03495-001_v01, Sep. 2007; https://www.ljll.math.upmc.fr/groupes/gpgpu/tutorial/Accelerating_Matlab_with_CUDA.pdf.

[16] NVIDIA, "What is GPU computing?" http://www.nvidia.com/object/what-is-gpu-computing.html.

[17] The DynTex Database, http://projects.cwi.nl/dyntex/database.html.

Premanand P. Ghadekar http://orcid.org/0000-0003-3134-137X

He is pursuing Ph.D. in Electronics & Telecommunication from Department of Applied Electronics, SGBA University, Amravati. He received M.Tech degree in Electronics (Computer) from College of Engineering, Pune in the year 2008. He completed BE degree in Electronics and Telecommunication Engineering from Government College of Engineering, Amravati, in 2001. In 2003, he joined the Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India as a lecturer. He is presently working as an Assistant Professor in the same Department. His areas of research are Image Processing, Dynamic Texture Synthesis and Embedded System. He has contributed 4 papers in International conferences and 4 paper in International Journals. He is a life member of ISTE, member of IEEE. He has received research grants of two lakhs from BCUD Pune in March 2011.

Nilkanth B. Chopade

He received a Ph.D. in Electronics Engineering from SGBA University, Amravati in the year 2009. He received M.E. degree in Electronics Engineering from Government College of Engineering Amravati in the year of 1998. His areas of research are DSP, Image Processing and Wavelet application. He has joined SSGM College of Engineering Shegaon in the year of 1992 as a lecturer. He is currently working as Professor in the Department of E & TC Engineering, Pimpari Chinchawad College of Engineering, Pune, Maharashtra, India. He is Fellow of IE (India), IETE and member of BES (India) and ISTE. He has received research grants from AICTE, New Delhi, International Travel Grant from AICTE, New Delhi and DST, New Delhi.He has also received research grant from SGBA University, Amravati. He has contributed several numbers of research papers in International and National Journals and conferences.