DYNAMIC TEXTURES MODELING USING TEMPORAL MIXING COEFFICIENTS REDUCTION

Michal Havlíček, Michal Haindl, Jiří Filip*

Institute of Information Theory & Automation
Czech Acadademy of Sciences
Pod Vodárenskou věží 4, Prague, 182 08, Czech Republic

ABSTRACT

Real world materials often change their appearance over time. If these variations are spatially and temporally homogeneous then the material visual appearance can be represented by a dynamic texture which is a natural extension of classic texture concept including the time as an extra dimension. In this article we present possible way to handle multispectral dynamic textures based on a combination of input data eigen analysis and subsequent processing of temporal mixing coefficients. The proposed method exhibits overall good performance, offers extremely fast synthesis which is not restricted in temporal dimension and simultaneously enables to compress significantly the original measured visual data.

Index Terms— dynamic texture, texture analysis, texture synthesis, data compression, computer graphics

1. INTRODUCTION AND RELATED WORK

Dynamic textures (DT) can be defined as spatially repetitive motion patterns exhibiting homogeneous temporal properties. Examples might be smoke, fire or liquids, also waving trees or straws or some moving mechanical objects. A sequence of either monospectral or multispectral images (frames) is the simplest representation of DT. Measured DT data are always represented by a finite length sequence, sometimes too short for an intended application. This property may limit possible use of DTs in virtual reality systems so temporally unconstrained synthesis of DT is an interesting and challenging research problem in several computer graphics, computer vision, and pattern recognition applications. DT synthesis can be also considered when the original dynamic texture measurements have to be compressed.

Already published articles dealing with DTs can be divided according to the application to: recognition, representation and synthesis [1]. The DT synthesis is the most difficult and there are only few papers on this topic available [2, 3, 4, 5, 6]. Some methods [2, 3] are limited by time consuming synthesis algorithm [5]. In addition method [2] requires some high level of temporal homogeneity of the input and this method is restricted to monospectral DTs. Method [4] is limited to finite length sequence generation [5].

Another possibility is to utilize so called video editing techniques [7, 8, 9], developed for general video sequences originally, which can be used for DT synthesis as DT can be considered as a special case of general video sequence. For example video texture generation based on searching for transition points for looping with additional blending and morphing [8]. Evident drawback is using blend and morphing to achieve continuity of the synthesized sequence which may introduce blur and other unfavourable visual artifacts. This issue was solved in [9]. Another possibility is tree structured vector quantization published in [7]. This method sometimes fail to reproduce global structures which may appear in the original data [9]. Video editing techniques are also very often time demanding [5]. We compare most of the above mentioned methods with our approach in Section 6.

The contribution of this paper is to propose straightforward multispectral DT modeling method with low computational demands enabling extremely fast synthesis of arbitrarily long DT sequence and in addition compression of original data. The method consists of input data dimensionality reduction using eigen analysis based on [5] and selective elimination of resulted temporal coefficients.

2. METHOD OVERVIEW

The scheme of the proposed method is shown in Fig.1. The whole modeling process can be divided into analysis and synthesis. The first step of the analysis is DT normalization: a per-pixel average frame from all frames in the sequence is computed and subtracted from each frame in this sequence. The results of following eigen analysis are eigen images and temporal mixing coefficients which are further processed during the temporal mixing coefficients reduction step. Average frame, eigen vectors and reduced temporal mixing coefficients are saved for synthesis purposes. Synthesis procedure consists of temporal mixing coefficients choice, normalized

*This research was supported by the grants GAČR 14-10911S, GAČR 14-0265S.

978-1-4799-7971-4/14/$31.00

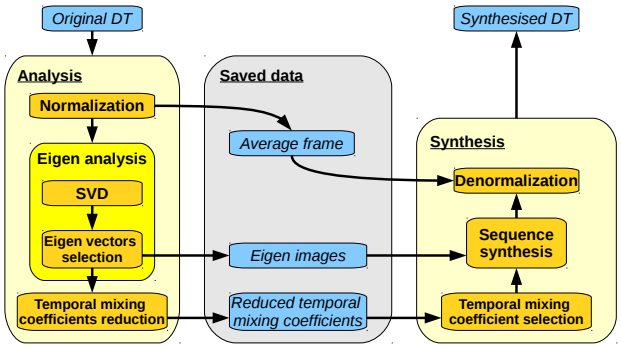

Fig. 1. Dynamic texture modeling method scheme.

frames synthesis driven by chosen coefficients and denormalization (inverse procedure to the normalization).

3. DYNAMIC TEXTURE EIGEN ANALYSIS

We used traditional PCA method for this task, similarly as in [5], because of its optimal performance beside alternative choices as for example non-linear techniques [10].

Values corresponding to pixels intensities of individual frames from the normalized sequence are arranged into column vectors forming $(n \times t)$ matrix C where n is a number of values equals frame width $\times$ frame height $\times$ number of spectral planes in the DT and t is a number of frames. Then a covariance $(t \times t)$ matrix A is computed as $A = C^T C$. The matrix A is decomposed using singular value decomposition so that $A = UDU^T$, where U is an orthogonal matrix of eigen vectors and D is a diagonal matrix of corresponding eigen numbers. Each eigenvalue is proportional to its significance for data reconstruction. Therefore only $k < t$ eigen vectors corresponding to eigenvalues representing the most of the information are saved. Unlike the approach [5] we use a threshold τ for selecting the vectors which are not used. The threshold is computed from the eigenvalues as:

$$\tau = \frac{1}{t} \sum_{i=1}^{t} D_{(i,i)} \quad . \tag{1}$$

If $D_{(i,i)} > \tau \, , i \in \{1, \ldots, t\}$ then i-th column of D and i-th column of U are used. All used columns of D and U form new matrices D^* and U^* respectively. The $(n \times k)$ matrix I of eigen images can be computed as: $I = CT$, where T is a $(t \times k)$ matrix with elements:

$$T_{(i,j)} = \frac{U^*_{(i,j)}}{\sqrt{D^*_{(j,j)}}} \quad . \tag{2}$$

Computed matrix I represents the reduced basis for the reconstruction of the original data therefore a matrix representing linear combination coefficient is needed. This role is played by a matrix of temporal mixing coefficients which is computed as: $M = I^T C$. The $(k \times t)$ matrix M, which in fact reflects the overall dynamics of the sequence, is a subject of further processing described in following section.

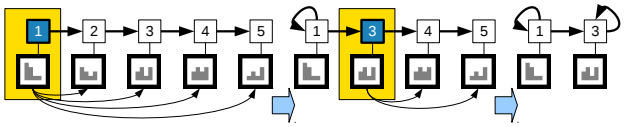

Fig. 2. The reduction algorithm schematic example.

4. TEMPORAL MIXING COEFFICIENT REDUCTION

The matrix of temporal mixing coefficients M is further processed to provide additional compression. M enables to apply non-deterministic synthesis algorithm guaranteeing potentially infinite DT sequence generation. For non-deterministic synthesis purposes analyzed DT is redefined in terms of graph theory so that the sequence is represented as a directed graph $\mathcal{G}$ where the individual frames play the role of vertices and the order of the frames plays the role of edges, i.e., the graph structure defines for each frame the set of frames which may immediately follow. Thus before the reduction an adjacency matrix A of $\mathcal{G}$ is formed as: $A_{(i,j)} = 1 \iff j = i + 1$, $A_{(i,j)} = 0 \; otherwise \; , i \in \{1, \ldots, t\} \; , j \in \{1, \ldots, t\}$. The idea of the reduction is to keep only those columns of M for which there is no other sufficiently similar column in M, in terms of certain metric, or generally distance. Let a $(t \times t)$ matrix Δ is composed of the elements:

$$\Delta_{(i,j)} = \sum_{l=1}^{k} (M_{(l,i)} - M_{(l,j)})^2 \quad , \tag{3}$$

i.e., Δ consists of all mutual distances of the columns of M. Apparently, Δ is symmetric, with zero diagonal, and therefore it is sufficient to take into account only those elements $\Delta_{(i,j)}$ for which $i < j$ holds and let $\Delta_{(i,j)} = 0$ otherwise. The distance (3) was chosen because of its proven reasonable properties for column comparison purpose and low computing demands.

An average distance δ is defined by the elements of Δ as follows:

$$\delta = \frac{1}{|Z|} \sum_{Z} \Delta_{(i,j)} \quad , \tag{4}$$

where Z is the set defined as $\{\Delta_{(i,j)} : i < j\}$. The average distance δ plays the role of the criterion determining the

similarity of the columns of M, in the sense of the distance (3). The matrix M is processed using Δ and δ.

Every column j which fulfils $D_{(i,j)} \leq \delta$ is removed and A is updated this way: $A_{(i,j)} = 0$, $A_{(i,i)} = A_{(i,i)} + 1$, and if $j < k$ holds then $A_{(j,j+1)} = 0$, $A_{(i,j+1)} = 1$. Reduction is step-wise applied to all columns of M. Remaining r columns of M , i.e., every column $i : \exists j \in \{1, \ldots, t\}$: $A_{(i,j)} = 1$, form new $(k \times r)$ matrix M^* which has to be stored and is later used for synthesis.

The reduction algorithm is illustrated on Fig.2. The sequence is represented by a graph structure, frames are the vertices, edges denote their order (thick arrows), and numbers correspond to the order of the frames in the original sequence. Vectors of the mixing coefficients corresponding to the individual frames are symbolically represented in the form of bar graphs (the height of each bar equals to the value of an individual coefficient, i.e., component of the vector). In this example reduction is performed in three steps. 1: vector of mixing coefficients of the first frame is compared with vectors of mixing coefficients of the frames of the rest of the sequence. 2: second frame was evaluated as too similar to the first so it was removed, vector of mixing coefficients of the third frame is compared with vectors of mixing coefficients of the remaining frames. 3: resulting graph structure to be stored.

5. DYNAMIC TEXTURE SYNTHESIS

DT synthesis produces a required length DT sequence. Synthetic frames have identical resolution with the original DT frames, i.e., the method is restricted to temporal enlargement. The only data needed for this task are: average frame computed during the normalization, matrix of eigen images I, matrix of reduced mixing coefficients M^* and adjacency matrix A.

During the synthesis a $(k \times t^\dagger)$ matrix of temporal mixing coefficients $M^\dagger$, where $t^\dagger$ is a length of the synthesized sequence (in general different from t) is created column wise according to the following algorithm. The first column of $M^\dagger$ is randomly chosen column of M^*. Let the last chosen column of M^* has index $i \in \{1, \ldots, r\}$ then the next column is randomly chosen column of M^* from those which fulfill $A_{(i,j)} = 1$, $j \in \{1, \ldots, r\}$. This provides required continuity of the synthesized sequence since the set from which the selection is performed consists of the frames such that there exists an edge between them and the last chosen one in $\mathcal{G}$ (see Section 4). If there does not exist any such column in M^* then the column closest to the i-th column of M^*, in sense of (3), is chosen. This can occur if the graph representing the DT after reduction step described in Section 4 is not connected. Above mentioned rule provides continuous sequence with no need to utilize any additional technique such as morphing which could introduce some unfavourable artifacts to the visual information.

The same columns should not be chosen several times in succession to avoid violation of texture dynamics. It was observed that up to three identical consecutive frames have negligible observable impact on the result. Synthesized normalized DT sequence $C^\dagger$ which is an $(n \times t^\dagger)$ matrix can be then computed simply as: $C^\dagger = IM^\dagger$ as explained in Section 3. The last step is an per-pixel addition of the average frame to each synthesized frame in the sequence.

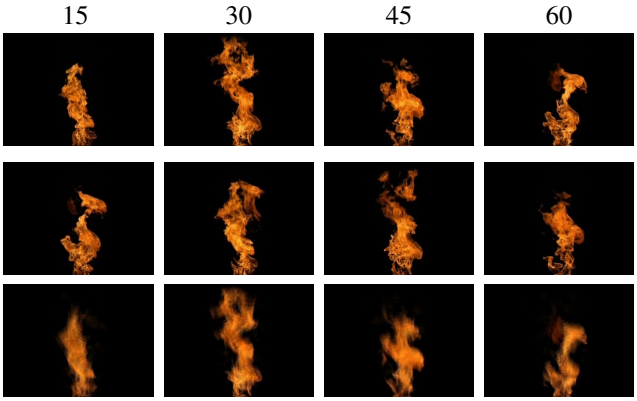

Fig. 3. Original DT flame frames (top row), their synthesis using the method [9]-temporal, and our method (bottom row).

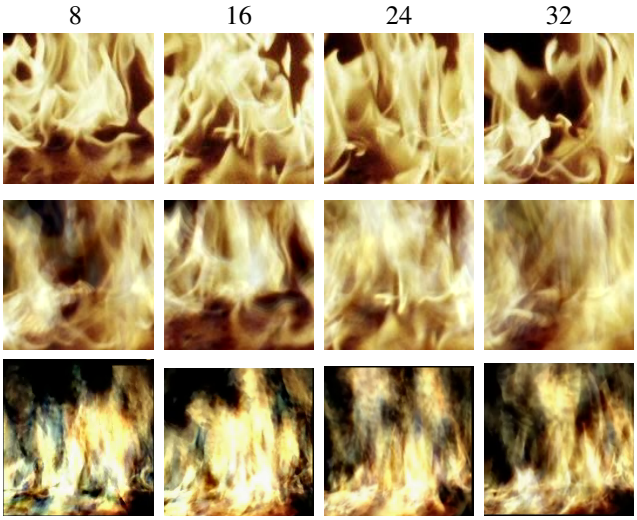

Fig. 4. The original fire DT frames (first row), their synthesis using the proposed method (second row), and the DT^{3DCAR} method [5], respectively.

6. RESULTS

We used dynamic texture data sets from DynTex texture database [11] as the first source of test data. Each dynamic

texture from this database is typically represented by 250 frames, which equals 10 seconds, long video sequence. We extracted its frames, converted, saved and used as 400×300 RGB colour images, so that $(n = 360000, t = 250)$. As test DTs were chosen: smoke, steam, streaming water, sea waves, river, candle light, detail of running escalator, sheet, waving flag, leaves, straws and branches.

Several examples of achieved results can be seen in Fig.4. From the shown results it is apparent that although there are some differences between original and synthesized sequences the overall dynamics is preserved.

DT	Gr. cut (temp.) [9]	Gr. cut (sp.-t.) [9]	Video text. [8]	Vect. quant. [7]	Text. mix. [4]	Dyn. text. [3]	pres. method
clouds		23.05			15.81		4.43
flame	9.61						3.12
fountain	29.74		6.76				2.86
grass	15.28		18.46				6.02
ocean	34.13	31.34	4.52	20.34			6.34
pond	14.3		13.78				7.5
river		40.4					11.92
smoke	37.0	15.47		23.85			1.79
sparkle	9.59						1.55
waterfall$_A$	18.78		21.45			55.84	5.08
waterfall$_B$		10.95			17.68		1.96

Table 1. Comparison of MAD quality modeling criterion values on the Graphcut texture database for six alternative DT synthesis methods.

For a comparison with alternative synthesis methods we also tested our method on video textures from Graphcut texture database [1] [9] (see (Fig.3)). This database consists of several very different textures including corresponding textures synthesized by alternative approaches.

It is really hard to compare visual quality of the results of those methods exactly as robust and reliable similarity comparison even between two static textures is still unsolved problem up to now. We decided to compare differences between original DT and synthesised DTs of individual methods. Let the original DT O is a L_O long sequence of $W_O \times H_O$ images with S_O spectral planes and the synthesized DT S is a L_S long sequence of $W_S \times H_S$ images with S_S spectral planes then Mean Absolute Difference of the original DT and the synthesized DT (MAD) is defined as:

$$MAD = \frac{1}{LSHW} \sum_{l=1}^{L} \sum_{k=1}^{S} \sum_{j=1}^{H} \sum_{i=1}^{W} |O_{(i,j,k,l)} - S_{(i,j,k,l)}| \ ,$$

where $L = \min\{L_O; S_S\}$, similarly S, H, W. From the results listed in Tab.1 it is apparent that our method outperforms, with one exception, the others, in this concept. Although there is not exist any DT in Graphcut database which would offer results obtained by all alternative methods, Tab.1 clearly demonstrates certain quality of our method.

Criterion (1) allows to adjust the compression level for each type of texture. Loss of the information can be expressed in the amount of energy (sum of used eigenvalues divided by the sum of all eigenvalues) which was preserved. In case of tested DTs: fire: 73%, clouds: 92%, flame: 78%, fountain: 82%, grass: 65%, ocean: 83%, pond: 77%, river: 80%, smoke: 93%, sparkle: 87%, waterfall: 62%, waterfall2: 90%. The processing time of the method on the 2GHz Pentium CPU needed to produce the same length DT as the original is the following (texture - frames$\times$width$\times$height, analysis / synthesis $[min : s]$): clouds,waterfall2 - $61 \times 128 \times 128$, 0:10 / 0:01; flame - $89 \times 320 \times 240$, 1:52 / 0:04; grass - $100 \times 224 \times 144$, 1:06 / 0:03; ocean,smoke - $32 \times 160 \times 112$, 0:04 / 0:01. Further comparison is discussed in following section.

7. DISCUSSION

The main advantage of this method is its simplicity, efficiency and performance, using optimal method for compression. Extremely fast synthesis can be even more efficiently performed by contemporary graphical hardware since only elementary instructions and matrix operations have to be realized. Our synthesis algorithm is less demanding than in case of most other methods. The synthesis is not restricted on number of frames to be generated, unlike [4], and it is not necessary to verify synthesized frames to prevent extremely long sequence to turn static as for example like in case [5]. However eigen analysis may cause observable loss of information (high frequencies which may occur in the original more precisely). In contrast to the sampling based DT modeling method called dynamic roller [6], our method cannot simultaneously enlarge the frames of the DT. On the other hand, this avoids possible spatial repetition of patterns which may appear in the synthesized DT produced by the dynamic roller.

8. CONCLUSION

We presented a novel method for fast synthesis of multispectral dynamic textures (DT). The main part of the approach is based on reduction of temporal coefficients resulted from DT dimensionality analysis step using the singular value decomposition which enables compress significantly the original data. This solution also enables extremely fast synthesis of arbitrary number of required multispectral DT frames, which can be even more efficiently performed by contemporary graphical hardware. From many presented results it is apparent that the visual properties of the original DTs stayed preserved in the synthesized ones. We also compared our method with several existing DT synthesis and video texture generation approaches. This method avoids some problems of

[1] http://www.cc.gatech.edu/cpl/projects/graphcuttextures/

the alternative methods. On the other hand, proposed synthesis algorithm does not extend DT in spatial domain. Overall, this method represents undemanding alternative to the existing approaches.

9. REFERENCES

[1] Dmitry Chetverikov and Renaud Peteri, "A brief survey of dynamic texture description and recognition," in *CORES 05*. 2005, Advances in Soft Computing, pp. 17–26, Springer.

[2] Martin Szummer and Rosalind W. Pickard, "Temporal texture modeling," in *Proc. IEEE Int. Con. Image Processing (ICIP)*. IEEE, September 1996, vol. 3, pp. 823–826, IEEE, MIT TR 381 report.

[3] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," in *IEEE International Conference on Computer Vision*, Vancouver, BC , Canada, 2001, IEEE, vol. 2, p. 439446, IEEE.

[4] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 120–135, Apr-Jun 2001.

[5] J. Filip, M. Haindl, and D. Chetverikov, "Fast synthesis of dynamic colour textures," in *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, Y.Y. Tang, S.P. Wang, D.S. Yeung, H. Yan, and G. Lorette, Eds., Los Alamitos, August 2006, vol. IV, pp. 25–28, IEEE Computer Society.

[6] M. Haindl and R. Richtr, "Dynamic texture enlargement," in *Spring Conference on Computer Graphics*, R. Durikovič and H. Rushmeier, Eds., New York, NY, USA, May 2013, SCCG '13, pp. 005:5–005:12, ACM.

[7] L. Wei and M. Levoy, "Texture synthesis using tree-structure vector quantization," in *ACM SIGGRAPH 2000*. 2000, pp. 479–488, ACM Press / Addison Wesley Longman.

[8] Arno Schodl, Richard Szeliski, David H. Salesin, and Irfan Essa, "Video textures," in *ACM SIGGRAPH 2000*, New Orleans, July 2000, ACM, pp. 489–498, ACM.

[9] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM T Graphic*, vol. 22, no. 3, pp. 277–286, July 2003.

[10] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," Tech. Rep. TiCC-TR 2009-005, Tilburg University, Tilburg, The Netherlands, 2009.

[11] Renaud Peteri, Sndor Fazekas, and Mark J. Huiskes, "Dyntex: A comprehensive database of dynamic textures," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1627 – 1632, 2010.