21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Improvements of Dynamic Texture Synthesis for Video Coding

Zhiqiang Hou, Ruimin Hu, Zhongyuan Wang, Zhen Han
National Engineering Research Center on Multimedia Software, Wuhan University, China
houzhiqiang1002@163.com

Abstract

We describe an algorithm for dynamic texture synthesis of video sequences of frames exhibiting certain stationary properties over time, such as sea-waves, whirlwind or moving crowds. The algorithm is based on taking into account the similarity among reference images in the video inter-frame coding. It allowed us to better express the time-varying relationship of the dynamic texture and to extend the algorithm described in [9].

1. Introduction

Texture is generally considered a special area of the image, which has a certain degree of randomness and self-similarity. Each pixel in the texture can be expressed by the adjacent pixels in time or space. The traditional methods in video coding are not sufficient to eliminate the correlation between the pixels in the texture image because the large number of high-frequency information, such as the block-based motion prediction and the Discrete Cosine Transform (DCT). For this problem, the method based on texture analysis and synthesis is proposed and gradually become one of the key technologies for video coding.

Many proposals have been made on how the texture can be used for video and image coding. In [1], one approach is presented where the textures are classified by geologic structures. In [2], texture is expressed by a MRF model. Texture image is segmented by Gabor filters in [3]. In [4], [5] and [6], texture image is segmented using the statistical characterization. For these methods, images are classified into detail-relevant and detail-irrelevant texture areas, and the texture synthesis effects subject to the constraints of texture segmentation accuracy. Therefore, these can only be effective for some particular textures.

Under the premise of preserving its advantages, texture synthesis can be further simplified by combining with the characteristic of video coding, so that, some complex operations such as the texture segmentation can be avoided [7] [8]. Specially, in order to improve the efficiency of inter prediction, Stojanovic et al. innovatively presented an algorithm for dynamic texture extrapolation [10], which is based on the model of dynamic texture proposed by Doretto et al. [9]. Moreover, in the following two years, two new approaches based on the model for video coding were proposed by Stojanovic [11], [12].

In this paper we propose to use an improved dynamic texture synthesis algorithm, in order to synthesize dynamic texture image that can be more in line with the characteristics of the video coding. Our algorithm can be viewed as an extension of Doretto's representational algorithm [9], where tools from pattern recognition are used to learn dynamic texture models.

2. Background

The definition of dynamic texture model was introduced in [9] for the purpose of dynamic texture synthesis for virtual reference frame. For a sequence $\{I(t)\}, t=1\ldots\tau, I(t)\in R^m$, let $y(t)=I(t)+w(t)$ be a noisy version of τ images, where $w(t)\in R^m$ is an independent and identically distributed (IID) sequence drawn from a known distribution. $\{I(t)\}$ is a dynamic texture sequence if there exists a set of n spatial filters, $\phi_\alpha : R \to R^m, \alpha = 1\ldots n$ and a stationary distribution $q(\cdot)$, defining $x(t)\in R^n$ such that $I(t)=\phi(x(t))$, we have $x(t)=\sum_{i=1}^{k} A_i x(t-i)+Bv(t)$, with $v(t)\in R^n$ an IID realization from the density $q(\cdot)$ for some choice of matrices $A_i \in R^{n\times n}, i=1\ldots k, B\in R^{n\times n}$.

978-4-9906441-1-6 ©2012 IAPR

The initial condition is $x(0)=x_0$, and in the simplest case, we take the filters as a dimensionality reduction step, and seek for a decomposition of the image in the simple form

$$I(t)=\sum_{i=1}^{n} x_i(t)\theta_i = Cx(t) \tag{1}$$

where $C=\{\theta_1,\theta_2,\cdots\theta_n\}\in R^{m\times n}$ and $\{\theta_i\}$ can be an orthonormal basis of L^2. Further, let $y_{means}\in R^n$ be a temporal mean of the sequence. So the sequence $y(t)$ can then be inferred as an autoregressive moving average (ARMA-process):

$$\begin{cases} x(t)=Ax(t-1)+Bv(t) \\ y(t)=Cx(t)+y_{means}+w(t) \end{cases} \tag{2}$$

The equation 2 shows the Doretto's solution for the dynamic texture synthesis, the synthesis frame $y(t)$ is consisted of the temporal mean y_{means} and the bivariate stochastic process driven by the noise $v(t)$ and $w(t)$. The model needs a large number of training sequences and shows the average movement trend of the reference pictures as a whole in time. Specially, when the length of the training sequence is short, the synthesis frame is only the repetition of the training sequence. So, this model is not suitable for the inter prediction, which using only few training pictures.

In [10], Stojanovic simplified the dynamic texture model as the form:

$$\begin{cases} x(t)=Ax(t-1) \\ y(t)=Cx(t) \end{cases} \tag{3}$$

According to the equation 3, the noise v and w is omitted. In addition, $n=\tau=5$ in this model, so the extrapolated frame is the repetition of the training sequence and y_{means} is zero. Under the modified model, Stojanovic presented one novel dynamic texture prediction algorithm (DTP) that combined the H.264 video coding system with dynamic texture by simply replaced the oldest frame in the reference picture buffer with the extrapolated frame. However, as we have shown in [13], the DTP algorithm can only be effective for some particular sequences. In the signal processing and system theory the image sequences can be thought of as the consequence of a bivariate stochastic process driven by the noise. Strictly speaking, if the noise process v and w are omitted, the solution of the dynamic texture model is incorrect.

Further, a new dynamic texture synthesis algorithm (DTS) is given by Stojanovic in [12], compared to DTP, the main difference is the number of training frames is higher. Namely, DTS is based on the model:

$$\begin{cases} x(t)=Ax(t-1) \\ y(t)=Cx(t)+y_{means} \end{cases} \tag{4}$$

The temporal mean y_{means} is not zero in the equation 4 because of the large number of training frames. So, besides the lack of noise process, another drawback of the DTS algorithm is not suitable to provide a real-time implementation for the decoder.

3. Method description

Through the above analysis, there exist two major unfavorable factors for Doretto's dynamic texture model to be integrated into the H.264 encoding and decoding system. Firstly, the noise process is varying in time, which would make the synthesized frames numerical inconsistent at the encoder and decoder. So, the noise v and w are omitted in DTP and DTS. Secondly, the model needs a large number of training frames, which would make the synthesized frames only have good effects on the linear motion, but have bad effects on the video sequences with non-linear motion and illumination changes.

In the case, we would extend the model and give an improved solution. The form of our model is

$$\begin{cases} x(t)=Ax(t-1)+Bv^{'}(t) \\ y(t)=Cx(t)+w^{'}(t) \end{cases} \tag{5}$$

and

$$y_{pow-means}=\begin{cases} y(1), & t=1 \\ a\dfrac{y(1)}{2^{t-1}}+\displaystyle\sum_{i=2}^{t}(a\dfrac{y(i)}{2^{t-i+1}}+b\cdot E), & t\geq 2 \end{cases} \tag{6}$$

where noise $v^{'}$ and $w^{'}$ are described by the pseudo-random number, and $y_{pow-means}$ is the temporal and weighted mean of the training frames. Parameters a , b in the equation 6 are the weighted factors. E is the unit matrix. Compared with the equation 2, 3 and 4, the improved dynamic texture synthesis (IDTS) has three main differences:

First, we use the pseudo-random number to describe the noise $v^{'}$ and $w^{'}$ in our Matlab and C++ code implementations, which could ensure the numerical consistency of our algorithm and the same synthesis effect at the encoder and decoder. Namely, the noise process is retained in our algorithm by using a new way.

Second, each of the training frames is given some corresponding weighted factors, which will be used to reflect the similarity between current frame and its reference frames. Therefore, we need not to limit the number of training frames, just adjust the factors.

The last and most important, our model can also be adapted to the non-linear motion and illumination changes by using the suitable weighted factors.

In the equation 2 and 4, y_{means} requires a large number of training frames and is defined as:

$$y_{means} = \frac{1}{t}\sum_{i=1}^{t} y(i) \tag{7}$$

We could find it is the main difficult for the dynamic texture model to be used in a video coding system, since a general requirement in video coding is to use only few reference frames. In addition, for the non-linear motion, there is no need to get the temporal mean, but want to get the movement status of recent one or two frames.

Table 1.
Improved dynamic texture synthesis algorithm

```
Algorithm IDTS( $Y^{n}_{n-\tau}$ )
Input: Decoded picture buffer matrix $Y^{n}_{n-\tau}$
Output: The value of $Y_{n+1}$
if $n > \tau \geq 2$ then
    $Y_{pow\_means}(1) \leftarrow \mathbf{pow2}(a \cdot Y(n-\tau), 1-n)$
    $k \leftarrow 1$
    for $i \leftarrow (n-\tau+k)$ to $n$
        $k \leftarrow k+1$
        $Y_{pow\_means}(k) \leftarrow \mathbf{pow2}(a \cdot Y(i), i-1-n) + b \cdot E$
    end for
    $U, S, V^{T} \leftarrow SVD(Y_{pow\_means})$
    $C \leftarrow \mathbf{left_k}(U)$
    $X^{n}_{n-\tau} \leftarrow \mathbf{upper_k}(S \cdot V^{T})$
    $A \leftarrow X^{n-1}_{n-\tau} \cdot \mathbf{pinv}(X^{n}_{n-(\tau-1)})$
    $Vhat \leftarrow X^{n-1}_{n-\tau} - A \cdot X^{n}_{n-(\tau-1)}$
    $Uv, Sv, Vv^{T} \leftarrow SVD(Vhat)$
    $B \leftarrow Uv^{n}_{n-(\tau-2)} \cdot Sv^{n}_{n-(\tau-2)} \big/ \sqrt{k-1}$
    $X_{n+1} \leftarrow A \cdot X_{n} + B \times \mathbf{randn}(size(B,2),1)$
    $Y_{n+1} \leftarrow C \cdot X_{n+1} + \mathbf{randn}(size(C,1),1)$
end if
return $Y_{n+1}$
```

For the $y_{pow-means}$ in our improved model, the parameters a , b show that the similarity between current frame and the reference images in linear motion scene, but the formula $1/2^{\varphi}\ (1 \leq \varphi \leq t-1)$ represents the similarity in the non-linear motion. Namely, the equation 6 includes two constraints ($t \geq 2$):

$$y(t) = a \cdot y(t-1) + b \cdot E \tag{8}$$

and

$$y(t) = \frac{y(t-1)}{2^{\varphi}} \quad \text{where: } 1 \leq \varphi \leq t-1 \tag{9}$$

Besides, there is one constraint for the factor $1/2^{\varphi}$:

$$\begin{aligned}
&\frac{1}{2^{t-1}} + \sum_{i=2}^{t} \frac{1}{2^{t-i+1}} = \\
&\frac{1}{2^{t-1}} + \left(\frac{1}{2^{t-1}} + \frac{1}{2^{t-2}} + \frac{1}{2^{t-3}} + \cdots + \frac{1}{2^{3}} + \frac{1}{2^{2}} + \frac{1}{2^{1}}\right) \\
&= \left(\frac{1}{2^{t-1}} + \frac{1}{2^{t-1}}\right) + \frac{1}{2^{t-2}} + \frac{1}{2^{t-3}} + \cdots + \frac{1}{2^{3}} + \frac{1}{2^{2}} + \frac{1}{2^{1}} \\
&= \left(\frac{1}{2^{t-2}} + \frac{1}{2^{t-2}}\right) + \frac{1}{2^{t-3}} + \cdots + \frac{1}{2^{3}} + \frac{1}{2^{2}} + \frac{1}{2^{1}} \\
&= \cdots = \left(\frac{1}{2} + \frac{1}{2}\right) = 1
\end{aligned} \tag{10}$$

Especially, when $a = 1$, $b = 0$ and $t < 3$, we can see that the value of y_{means} is equal to the value of $y_{pow-means}$. So, to some extent, $y_{pow-means}$ can be seen as the important compensation and improvement for y_{means} when only allows a very limited number of reference frames for the video coding.

It should be noted that the parameters a , b and the formula $1/2^{\varphi}$ are not used to represent the movement properties or the motion type, but to simulate the similarity. It is based on a reasonable assumption, for any reference frame as a whole, the farther away from the current frame, the lower in the similarity.

Like DTP or DTS, the IDTS algorithm also using a higher order SVD for decomposition and the precise description is given in Table 1. In Detail, **pow2** (f , p) computes $f \times 2^{p} = f \times (1/2^{-p})$ for corresponding elements of f and p , $\mathbf{left_k}$ (A) is the operation of taking left k columns of the matrix A , $\mathbf{upper_k}$ (A) is the operation of taking the upper k rows of A , **pinv**(A) computes the Moore-Penrose pseudoinverse of A , **randn**(m, n) is used to generate an m-by-n matrix containing pseudorandom values drawn from the standard normal distribution.

Table 2. Results for JM18.3 with DTP, DTS and our IDTS algorithm compared to original JM18.3

Sequences	DTP		DTS		IDTS				
	△PSNR (dB)	△Rate (%)	△PSNR (dB)	△Rate (%)	△PSNR (dB)	△Rate (%)	a	b	t
Container	0.39	- 7.93	0.45	- 8.90	0.48	- 9.02	1.21	0.1	7
Coastguard	- 0.04	1.00	0.05	- 0.16	0.05	- 0.19	1.02	0.3	6
Basketballpass	- 0.01	0.25	0.04	- 1.07	0.06	- 1.36	1.16	0.1	5
Blowingbubbles	- 0.04	0.86	0.01	- 0.13	0.03	- 0.27	1.72	0.6	11
Racehorses	- 0.02	0.39	0.02	- 0.25	0.08	- 0.93	1.30	0.0	5

4. Experiment results

The presented algorithm is integrated into the JM18.3 reference software [14]. For comparison, testing conditions and coding parameters for coding efficiency based on the description in [12] are used. The Container and Coastguard sequences are QCIF (176×144) and the others are 416×240, but only the Container sequence includes lots of linear motion textures.

Table 2 shows that average BDPSNR and BD-BitRate of DTP, DTS and our IDTS algorithm compared to the JM18.3. The experimental results show that the IDTS algorithm can achieve better performance than the former. The DTP algorithm is only effective for the Container sequence. The DTS algorithm is effective for all sequences by using a large number of reference frames. On the whole, the proposed IDTS algorithm can obtain better results only by using a few reference frames and can be suitable for a real-time implementation for the decoder. The optimal weighted factors a , b and the number of reference frames t are got by a lot of testing.

Acknowledgements

The research was supported by the National Basic Research Program of China (2009CB320906), the Major National Science and Technology Special Projects (2010ZX03004-003-03), the National Natural Science Foundation of China (60970160, 61070080, 61003184).

References

[1] V. Shankar, J.J. Rodriguez, M.E. Gettings. “Texture An-alysis for Automated Classification of Geologic Structur-es.” *IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 81–85, 2006.

[2] M. Haindl, V. Havlicek. “A compound MRF texture mo-del.” *International Conference on Pattern Recognition,* pp. 1792–1795, 2010.

[3] Huchuan Lu, Yunyun Liu, Zhipeng Sun, Yenwei Chen. “An active contours method based on intensity and reduced gabor features for texture segmentation.” *IEEE Inter-national Conference on Image Processing*, pp. 1369–1372, 2009.

[4] C.P. Loizou, V. Murray, M.S. Pattichis, M. Pantziaris, and C.S. Pattichis. “Multiscale Amplitude-Modulation Frequency-Modulation (AM-FM) Texture Analysis of Ultrasound Images of the Intima and Media Layers of the Carotid Artery.” *IEEE Transactions on Information Technology in Biomedicine*, pp. 178–188, 2011.

[5] J.T. Cobb, K.C. Slatton, G.J. Dobeck. “A parametric mo-del for characterizing seabed textures in synthetic apertu-re sonar images.” *IEEE Journal of Oceanic Engineering*, 35(2):250–266, 2010.

[6] Ben Othmen M., Sayadi M., Fnaiech F. “Interest of the multi-resolution analysis based on the co-occurrence mat-rix for texture classification.” *IEEE Mediterranean Elec-trotechnical Conference*, pp. 852–856, 2008.

[7] Byung Tae Oh, Yeping Su, Andrew Segall. “Synthesis-based texture coding for video compression with side information.” *IEEE International Conference on Image Processing*, pp. 1628-1631, 2008.

[8] LiYi Wei, Jianwei Han, Kun Zhou, Hujun Bao, “Inverse texture synthesis.” *ACM Transactions on Graphics*, 27(3):1–9, 2008.

[9] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto. “Dynamic textures.” *International Journal of Computer Vision*, 51(2):91–109, 2003.

[10] Aleksandar Stojanovic, Mathias Wien, Jens Rainer Ohm. “Dynamic texture synthesis for H.264/AVC inter coding.” *IEEE International Conference on Image Processing*, pp. 1608–1611, 2008.

[11] Aleksandar Stojanovic, Mathias Wien, Thiow Keng Tan. “Synthesis-in-the-loop for video texture coding.” *IEEE International Conference on Image Processing*. pp. 2293–2296, 2009.

[12] Aleksandar Stojanovic, Philipp Kosse. “Extended dynamic texture prediction for H.264/AVC inter coding.” *IEEE International Conference on Image Processing*. pp. 2045–2048, 2010.

[13] Hao Chen, Ruimin Hu, Dan Mao, Rui Zhong, Zhongyuan Wang. “Video coding using dynamic texture synthesis.” *IEEE International Conference on Multimedia and Expo.* pp. 203-208, 2010.

[14] JM18.3: http://iphome.hhi.de/suehring/tml.