VIDEO CODING WITH SPATIO-TEMPORAL TEXTURE SYNTHESIS AND EDGE-BASED INPAINTING

Chunbo Zhu[1]*, Xiaoyan Sun*[2]*, Feng Wu*[2]*, and Houqiang Li*[1]

[1]University of Science and Technology of China, Hefei, 230027, China
[2]Microsoft Research Asia, Beijing, 100080, China

ABSTRACT

This paper proposes a video coding scheme, in which textural and structural regions are selectively removed in the encoder, and restored in the decoder by spatio-temporal texture synthesis and edge-based inpainting. In the proposed scheme, two types of regions are classified based on two motion models: local motion and global motion. In local motion regions, conventional block-based motion estimation is employed for region removal and spatio-temporal texture synthesis is applied for recovery of the removed regions. In global motion regions, edge-based image inpainting is utilized to recover removed regions, and sprite generation is used as an auxiliary tool to keep temporal consistency. In the proposed scheme, both structures and textures are handled and some kinds of assistant information which can guide restoration are extracted and coded. This approach is block-based and thus is flexible and generic to be implemented into standard-compliant video coding schemes. It has been implemented into H.264/AVC and achieves up to 35% bitrate saving at similar visual quality levels compared with H.264/AVC without our approach.

Index Terms— Video coding, texture synthesis, inpainting, region removal, exemplar selection

1. INTRODUCTION

State-of-the-art video coding schemes such as H.264/AVC (H.264 for short hereafter) only exploit statistical redundancy among pixels using mean square error (MSE) criterion. However, it is generally believed that minimizing overall pixel-wise distortion does not guarantee good perceptual quality of reconstructed videos, especially in low bitrate scenarios.

Recently, developments of computer vision and graphics have shown remarkable progresses, some of which such as texture synthesis [1] and image/video inpainting [2], [3] can be applicable to exploit visual redundancy in images or videos. Although these techniques were originally designed for other purposes like error concealment [4] and object removal [5], they can lead to promising prospect in image/video compression. It has been reported that improvement is achieved by employing texture synthesis in compression even though in a straight-forward fashion [4]. Ndjiki-Nya *et al* have proposed in [6] a video coding scheme utilizing texture analysis/synthesis and global motion compensation. Moreover, some features of the original video sequences can be extracted and encoded in the bitstream as side information. Liu *et al* have proposed in [7] an image compression scheme in which texture synthesis and structure-aware image inpainting are used to exploit visual redundancies both in textural regions and structural regions. Guided by side information, e.g., edges, the scheme in [7] is able to efficiently restore missing image regions without obvious degradation of visual quality.

In this paper, we propose a video coding framework taking advantages of spatio-temporal texture synthesis [8] and edge-based inpainting [7]. The main idea is that: some regions in original video sequences can be selectively removed and skipped in the encoder and can be restored by spatio-temporal texture synthesis and edge-based inpainting in the decoder. Moreover, some parameters such as motion parameters, region types and edge information can be extracted to guide restoration.

The scheme proposed in this paper is carefully designed towards video compression. We would like to point out the novelties of our proposed scheme here. First, to keep temporal consistency, local motion and global motion are both considered in region removal process and recovering process. Second, different types of assistant information such as motion parameters and edge information are extracted to guide restoration in the decoder. Third, both structural regions and textural regions are processed in region removal and restoration. Furthermore, global motion regions are handled by registering all pixels into 2-D sprite images [9], with which rough region removal, region restoration and frame reconstruction are performed.

The remainder of this paper is organized as follows. In Section 2, the framework of our scheme is presented. In Section 3 and Section 4, we will describe the encoder and decoder algorithms in detail. In Section 5, experimental results are shown. This paper will be concluded in Section 6.

2. PROPOSED FRAMEWORK

In our proposed scheme, video sequences are processed group of pictures (GOP) by GOP. Each GOP consists of two key frames (I- or P-frames) and several bi-directional (B-) frames which are predicted from key frames. The last key frame of one GOP is also the start key frame of the next GOP. Key frames are encoded with H.264 encoder and B-frames are to be analyzed.

The proposed encoder is depicted in Fig. 1. In our proposed framework, removed regions in B-frames are skipped in the encoder and restored in the decoder. Un-removed regions which will be coded with conventional video coding scheme are so-called *exemplars*. The procedure of determining whether a region is removable or un-removable is called *region removal* or *exemplar selection*. For simplicity, this process is performed on block level. At the beginning of the encoder, all B-frames are divided into two types of non-overlapped 8x8-block-level regions: local motion regions (LMR) and global motion regions (GMR).

978-1-4244-2571-6/08/$25.00 ©2008 IEEE

In LMR, edge detection is performed to divide LMR into structural blocks and textural blocks. Structural blocks are exemplars coded with H.264 encoder, whereas textural blocks are further analyzed. Meanwhile, block-based motion estimation is performed to align all textural blocks into motion threads [10]. Each motion thread of textural blocks is pruned and treated as one candidate exemplar. Exemplar selection is then performed in the Exemplar Selection1 module so that some of the motion threads are chosen as exemplars and others are removed.

In GMR, we introduce sprite generation [9] as an auxiliary tool for exemplar selection. A sprite image (SPT) is generated by registering all GMR pixels in each GOP. The generated SPT is then classified into removable regions and un-removable regions by the means similar to that in [7]. Both the two types of regions contain structures and textures. Removable regions in the SPT are mapped into each B-frame and are snapped into grid of 8x8 blocks. Then, in the Exemplar Selection2 module, mapped removable blocks are removed and others are chosen as exemplars.

At the end of the encoder, four types of assistant information are losslessly coded by the Assistant Encoder, including region mask (*Mask1*), removing mask (*Mask2*), frame-level global motion estimation (GME) parameters and edge map of SPT for each GOP.

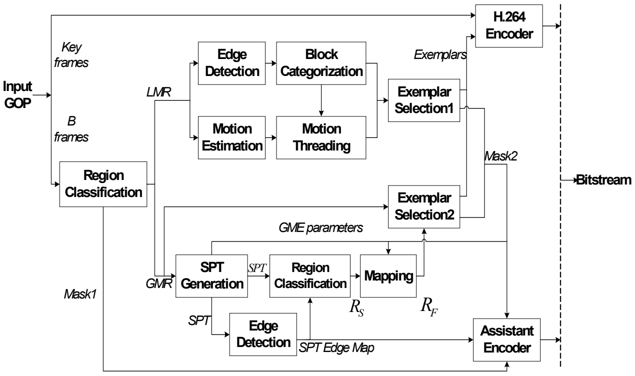

Fig. 1. Framework of proposed encoder.

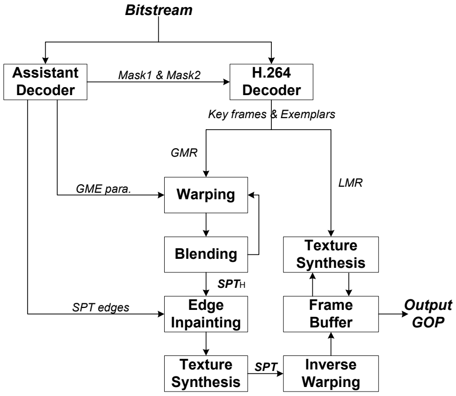

Fig. 2. Framework of proposed decoder.

The decoder framework of proposed scheme is depicted in Fig. 2. Assistant information is decoded by the Assistant Decoder. Guided by the decoded masks, key frames and exemplars in B-frames are decoded by standard-compliant H.264 decoder. Based on the decoded mask information, LMR and GMR are determined. Then the removed regions in LMR are restored by spatio-temporal texture synthesis which has been proposed in our previous work [8].

In GMR, a sprite image is first generated for each GOP using the available GMR regions and GME parameters decoded by Assistant Decoder. Note that, GME is not performed in sprite generation at the decoder side. All available pixels in GMR are warped using the GME parameters and are blended into the sprite image. Warping and blending are repeated until all GMR of one GOP have been registered. The sprite image (denoted as SPT_H) may contain "holes", because only exemplars in GMR can be used. Therefore, edge-based image inpainting [7] is employed to fill in the "holes", with the guidance by the SPT edge map. After SPT is completed, unknown GMR in all B-frames of the GOP are reconstructed by inverse warping and stored together with LMR regions in the frame buffer.

Once all LMR and GMR in a GOP have been completed, the GOP can be outputted.

3. ENCODER DESIGN

In our previous work [8], region removal has been done by aligning textural blocks with motion threads to preserve temporal consistency in synthesizable regions. However, for natural video scenes, different categories of motion may exist in different regions of one sequence. For example, in some regions, there are lots of heterogeneous textures without regular motion orientations or texture patterns, while in other regions, the motion of pixels or visual objects can be described by a simple global motion model. Therefore, we have to treat the two regions differently.

In region removal, two rules should be abided. First, any block which is selected to be removable must contain pixels or texture patterns that are repeated or similar in surrounding available areas. Second, temporal consistency must be preserved to avoid flickering artifacts. In this work, we have developed a way to keep temporal consistency, that is, to align all candidate samples (blocks or pixels) with motion trajectories and treat all samples in one motion trajectory as a whole candidate.

At the beginning of the proposed encoder, a simple region classification is performed to divide each input video frame into LMR and GMR. A rough segmentation tool proposed in [9] is employed to perform region classification. This method can roughly divide each video frame into two types of regions: one is GMR in which pixel motion can be described in terms of a parametric geometrical model:

$$x' = \frac{ax+by+c}{gx+hy+1}, \quad y' = \frac{dx+ey+f}{gx+hy+1} \qquad (1)$$

Where (*x, y*) and (*x',y'*) are coordinates of pixels in original frame and reference image (actually sprite image which will be introduced later) and *a~h* are eight global motion parameters. Other regions are LMR in which motion of pixels cannot be described with a fixed global model.

3.1. Region removal in LMR

It is generally accepted that pure textures can be satisfactorily generated even given a small sample, whereas structures with local motion are hard to be restored and they have strong impacts on human vision. Therefore, we divide LMR into structural and textural regions and select some of textural blocks to be removed. Region removal (exemplar selection) process in LMR has been proposed in our previous work [8]. In [8], conventional block-based motion estimation was used in exemplar selection to align candidate textural blocks with local motion trajectories, and then, some of the candidates were selected as exemplars according to their spatio-temporal variation. Please refer to [8] for more details.

3.2. Region removal in GMR

In LMR, region removal is performed locally with consideration of nonregular local motion. However, pixel motion in GMR can be detected by GME and can be described by a parametric geometrical model as shown in (1). For region removal in GMR, we employ sprite generation which is based on GME.

A sprite image is a panoramic image by registering all pixels with global motion in a number of images into one single big image and stitching these images together. There have been lots of sprite generation methods in literature. In this paper, we employ the sprite generation tool proposed in [9] by Yan *et al*. In our scheme, sprite generation is utilized as an auxiliary tool to help region removal in GMR and keep temporal consistency. The reason is that all pixels in contributing frames can be found in a sprite image and each pixel in the sprite image represents a number of pixels on a global motion trajectory.

For GMR, a sprite image is generated for each GOP with the sprite generation method mentioned above. Then region classification is performed on this sprite image. This region classification is similar to the exemplar selection in LMR. It is based on block-level and divides the sprite image into removable blocks and un-removable blocks. Specifically, the visible regions in sprite image are first divided into non-overlapped blocks for simplicity. Edge detection is also applied on the sprite image to categorize all visible blocks into structural blocks and textural blocks. Unlike the exemplar selection process in LMR, structural blocks and textural blocks are all candidates for exemplars. This is because motion of either structures or textures can be describe with the same global motion model. In this selection process, we use the exemplar selection method proposed in [7], for this process is performed on 2-D sprite image.

After all visible blocks in sprite image are classified into removable blocks and un-removable blocks, they are mapped into original frames. The mapping process can be seen as an inverse warping process as:

$$x = \frac{a'x' + b'y' + c'}{g'x' + h'y' + 1}, \quad y = \frac{d'x' + e'y' + f'}{g'x' + h'y' + 1} \qquad (2)$$

Where (x, y) and (x', y') are pixel coordinates in original frames and sprite image respectively. *a'~h'* can be calculated from GME parameters in (1). Removable blocks in sprite image are indicated by R_s in Fig. 1 and removable regions in original frames are indicated by R_F. Then in the Exemplar Selection2 module, blocks in R_F are matched with the grid of 8x8 blocks. Specifically, if one 8x8 block in an original frame contains more than half of the removable pixels, it is selected as a removable block and then skipped when encoding. Otherwise, it is selected as an exemplar and coded with H.264 encoder.

4. DECODER DESIGN

With the decoded exemplars and assistant information, we propose two restoration methods to recover removed regions in LMR and GMR, respectively.

4.1. Spatio-temporal texture synthesis in LMR

As only some of textural blocks have been removed, texture synthesis is applied to recover removed blocks in LMR. To keep temporal consistency, our previous work of spatio-temporal texture synthesis [8] is used for LMR restoration. Unknown textural regions are filled in with available pixels from spatio-temporal neighborhood. Therefore, temporal consistency can be preserved locally as much as possible. Please refer to [8] for more details.

4.2. Edge-based inpainting in GMR

Assisted by the decoded region map *Mask1* and removing map *Mask2*, H.264 decoder reconstructs key frames and exemplars in GMR of B-frames. For each GOP, all pixels in these available regions are registered into a sprite image. Specifically, GMR pixels in each frame are warped with the GME parameters that have been decoded by the Assistant Decoder, and then blended into the sprite image frame by frame. The warping and blending processes are the same as those in the encoder. Note that, no GME process is needed in the decoder as all the GME parameters have been transmitted to the decoder side already. After warping and blending are finished within a GOP, a sprite image which contains all available pixels in GMR of one GOP can be obtained.

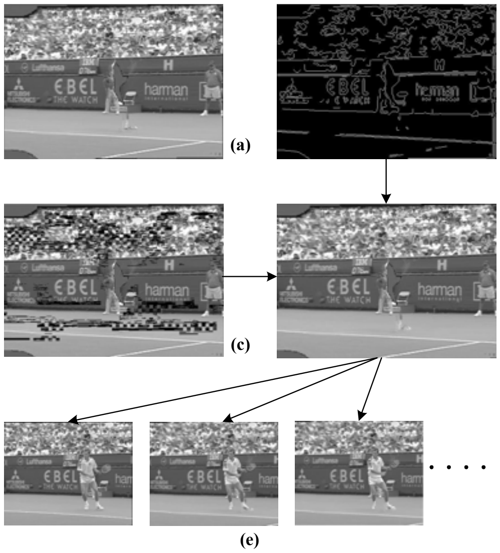

Fig. 3. Restoration process in GMR of "*Stefan*". (a) Original SPT of the first GOP generated at the encoder. (b) Extracted edge map of the SPT. (c) Generated SPT_H at the decoder. (d) Completed SPT after edge-based inpainting. (e) Reconstructed frames containing restored GMR, LMR and reconstructions of the H.264 decoder.

However, as only un-removed pixels contribute to this sprite image, it may contain some empty regions. This sprite image with "holes" is shown in Fig. 3 (c), in which dark regions in visible regions of *SPT* are "holes". These "holes" are then filled in with surrounding available pixels using the edge-based image inpainting algorithm proposed in [7]. Afterwards, the sprite image is completed as shown in Fig. 3 (d). This completed sprite image is then used to reconstruct all GMR in each frame in a GOP. The reconstruction process can be seen as an inverse warping process based on the model of (2). The parameters *a'~h'* can be calculated from decoded GME parameters. To reconstruct each frame, un-removed blocks are copied from outputs of H.264 decoder, whereas removed GMR blocks are copied from the output of Inverse Warping process and removed LMR blocks are copied from the output of spatio-temporal texture synthesis process. Finally, all reconstructed GMR and LMR are stored in Frame Buffer so as to be used to recover following frames. The outputs of the whole process are shown in Fig. 3 (e).

5. EXPERIMENTAL RESULTS

The proposed scheme has been implemented into the Joint Model version 10.2 (JM 10.2) [11] of H.264. We have tested the proposed scheme with video sequences "*Stefan*", "*Foreman*" and "*City*". In our experiments, we used YUV 4:2:0 sequence format with CIF resolution (352x288). All sequences were tested with 30Hz frame rate and IB..BPB..BP… sequence type. The starting intra frame and all P-frames were key frames coded with H.264 for both testing cases. B-frames were periodically inserted between key frames and were predicted from the nearest pair of key frames. GOP length was 16 for each sequence. Besides, rate distortion optimization (RDO) and CABAC were turned on.

Fig. 4 shows three sequentially reconstructed frames of the sequence "*Foreman*" with total bitrate saving of 24.34%. Compared with the reconstruction of H.264, our approach has a similar visual quality level. Fig. 5 shows bitrate saving vs. quantization parameter (QP) curves of three testing sequences. Up to 35% bitrate saving can be obtained for sequence "*Stefan*". For all sequences, bitrate saving decreases as QP increases, since coded bits for assistant information have been taken into account and remain constant over different QPs.

6. CONCLUDING REMARKS

In this paper, we present a video coding scheme in which some regions are removed in the encoder and restored in the decoder by spatio-temporal texture synthesis and edge-based image inpainting. To keep temporal consistency, different types of motion have been considered in region removal and restoration for both textural and structural regions. Besides, some parameters which can efficiently guide restoration are also extracted and coded. This approach has been implemented into conventional video coding scheme such as H.264 and has shown remarkable bitrate saving and similar visual quality compared with H.264.

7. ACKNOWLEDGEMENT

This work is supported by NSFC General Program under contract No. 60572067, 863 Program under contract No. 2006AA01Z317, and NSFC Key Program under contract No. 60632040.

8. REFERENCES

[1]. A. Efros, and T. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of International Conference on Computer Vision*, pp. 1033-1038, 1999.

[2]. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of ACM SIGGRAPH*, 2000, pp. 417-424.

[3]. M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proceedings of International Conference on Computer Vision and Pattern Recongnition*, Vol. 1, pp. 355-362, Hawaii, 2001.

[4]. S. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Transactions on Image Processing*, Vol.12, no. 3, pp. 296-303, Mar. 2003.

[5]. A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, Vol. 13, no. 9, Sept. 2004.

[6]. P. Ndjiki-Nya, T. Hinz, A. Smolic and T. Wiegand, "A generic and automatic content-based approach for improved H.264/MPEG-AVC video coding," *IEEE International Conference on Image Processing*, vol. 2, pp. 874-877, Sept. 2005.

[7]. D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, "Image compression with edge-based inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 10, Oct. 2007.

[8]. C. Zhu, X. Sun, F. Wu, and H. Li, "Video coding with spatio-temporal texture synthesis," *IEEE International Conference on Multimedia and Expo*, pp. 112-115, Jul. 2007.

[9]. Y. Lu, F. Wu, S. Li, and Y.-Q. Zhang, "Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 5, May 2003.

[10]. L. Luo, F. Wu, S. Li, Z. Xiong, and Z.Q. Zhuang, "Advanced motion threading for 3D wavelet video coding," *Signal Processing: Image Communication*, Vol. 19, *Issue 7, pp. 601-616, Aug. 2004.*

[11]. *MPEG-4 AVC/H.264 Joint Model* version 10.2.

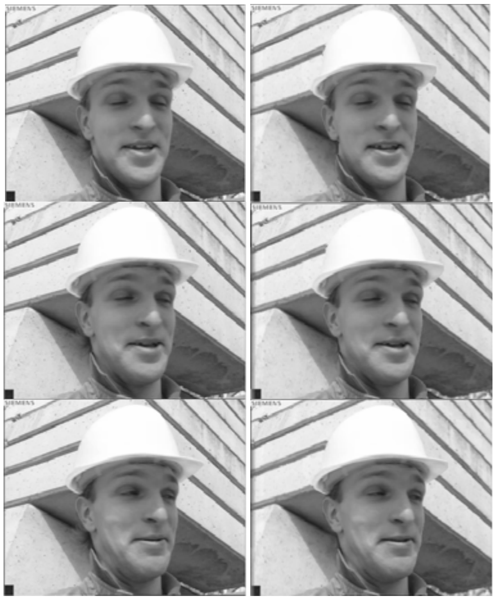

Fig. 4. Reconstructed frames of "*Foreman*". Left: H.264 with QP=16; right: proposed with QP=16 and 24.34% bitrate saving.

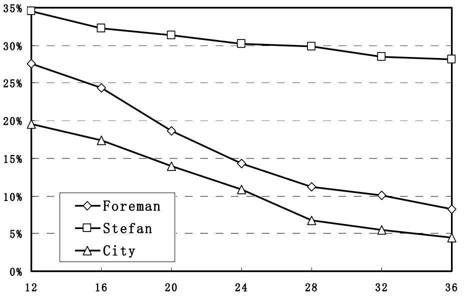

Fig. 5. Bitrate saving vs. QP curves for sequences "*Foreman*", "*Stefan*" and "*City*" with CIF resolution.