IMPROVED VIDEO CODING THROUGH TEXTURE ANALYSIS AND SYNTHESIS

Patrick Ndjiki-Nya, Christoph Stüber and Thomas Wiegand

Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institut
Image Processing Department
Einsteinufer 37, 10587 Berlin, Germany
{ndjiki/stueber/wiegand}@hhi.de

ABSTRACT

A new video coding approach based on texture analysis and synthesis is presented. The underlying assumption of our approach is that the textures in a video scene can be labeled subjectively relevant or irrelevant. Relevant textures are defined as containing subjectively meaningful details, while irrelevant textures can be seen as image content with less important subjective details. We apply this idea to video coding using a texture analyzer and a texture synthesizer. The texture analyzer (encoder side) identifies the texture regions with unimportant subjective details and generates side information for the texture synthesizer (decoder side), which in turn inserts synthetic textures at the specified locations. In this paper, it is shown that bit-rate savings of up to 19.4% can be achieved, using a semi-automatic MPEG-7-aided texture analyzer, compared to a standard conforming H.264/AVC video codec. The subjective video quality is thereby comparable to the H.264/AVC codec without the presented approach.

1. INTRODUCTION

It is known that textures featuring a high amount of visible details require high bit-rates when coding them using mean squared error (MSE) as the distortion criterion. Typical representatives of this texture class, called detail-irrelevant in the following, may be grass, trees, flowers, corn field, water, etc. Paradoxically, the MSE-exact regeneration of such textures is not necessary if they are shown with restricted spatial accuracy. That is, the coding efficiency can be significantly improved, if processing of detail-irrelevant textures available in a given sequence is done in consideration of the above-mentioned hint.

We implement this approach by introducing a texture analyzer at the encoder side and a texture synthesizer at the decoder side. The texture analyzer identifies detail-irrelevant texture regions and creates corresponding coarse masks. The encoder then signals these masks as side information to the texture synthesizer, located in the decoder. The texture synthesizer replaces the marked textures by inserting corresponding synthetic ones.

Similar wavelet-based analysis-synthesis video coding approaches were introduced by Yoon and Adelson [1] and by Dumitraş and Haskell [2]. The algorithms presented in [1],[2] are optimized for textures with absent or very slow global motion, whereas no such constraint is required for our system.

The combination of multiple reference frames and affine motion-compensated prediction was introduced by Steinbach et al. [3]. In [3], a segmentation-free solution with more than two reference frames is presented. A suitable reference frame for motion compensation is selected using a MSE-based cost function as the distortion criterion, whereas in this work MPEG-7 similarity measures [4],[5] are employed.

Smolic et al. introduced an online sprite coding scheme [6] that is better suited for real-time applications than MPEG-4's static sprite coding [7]. A major drawback of the approach in [6] is the requirement for a very precise background segmentation.

Some of the ideas of [3] and [6] are utilized within our framework for video coding, using analysis and synthesis, in this paper. The remainder of the paper is organized as follows. In Section 2 we introduce the texture synthesizer, while in Section 3 the texture analyzer is presented. Finally, in Section 4 the experimental results are shown.

2. TEXTURE SYNTHESIZER

In this paper, two texture synthesizers are presented. The first texture synthesizer (TS I) is designed for rigid objects, while the second (TS II) is optimized for non-rigid textures, i.e. textures with local motion activity. For TS I and TS II, it is assumed that the frame-to-frame displacement of the objects can be described using the perspective motion model [4],[5].

The texture synthesizer I warps the texture from a given key frame towards each synthesizable texture region identified by the texture analyzer as illustrated in Figure 1. A motion parameter set and a control parameter are required by the texture synthesizer for each synthesizable texture region. The motion parameters describe the global motion, of the considered texture region, between key and current frame, while the control parameter specifies the key frame to use to synthesize the current texture region.

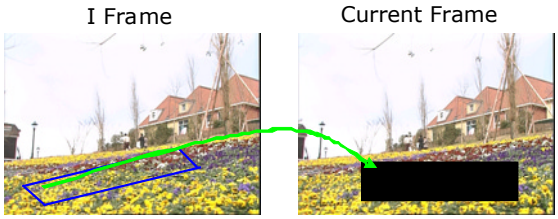

Figure 1 - Texture synthesizer I filling texture region identified by texture analyzer using given key frame

TS II was designed for non-rigid textures as already said above. Textures are modeled through Markov Random Field methods [8]. That is, each texture sample is predictable from a small set of spatially neighboring samples and is independent of the rest of the texture. TS II warps a given texture from all available key frames (2 at most) towards the corresponding detail-irrelevant texture region in the current partially synthesizable frame. Unlike TS I, the warped textures are not inserted into the marked area in the current frame.

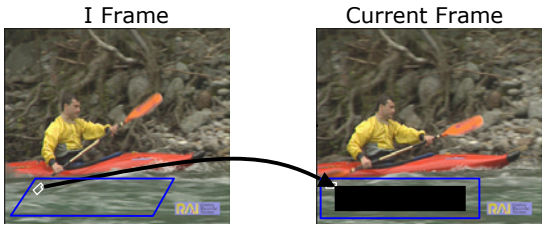

Figure 2 - Texture synthesizer II, warping of texture from reference frame towards region to be filled

The local motion activity in the considered detail-irrelevant texture region is modeled by matching the causal neighborhood (typically 3x3 window) of a given detail-irrelevant sample in the current frame with the corresponding neighborhoods of warped samples in a restricted area (cp. Figure 2). The warped sample with the most similar neighborhood is inserted at the location of the considered detail-irrelevant sample in the current frame. TS II requires a control parameter and motion parameters for synthesis of each detail-irrelevant texture region. Unlike TS I the control parameter can indicate more than one valid key frame for synthesis of the given detail-irrelevant texture region. The number of motion parameter sets required corresponds to the number of available key frames.

The determination of the motion and control parameters is explained into more detail in the following section.

3. TEXTURE ANALYZER

3.1. Spatial segmentation

3.1.1. Segmentation strategy

The texture analyzer performs a split and merge segmentation of each frame of a given video sequence.

The splitting step consists in analyzing a frame using a multi-resolution quadtree [9]. A block at level ℓ-1 of the quadtree is considered to have homogeneous content if its four sub-blocks at level ℓ have "similar" statistical properties. Inhomogeneous blocks are split further, while homogeneous blocks remain unchanged. The splitting stops, when the smallest allowed block size is reached, and the non-homogeneous areas of the considered frame are marked as not classified.

In the merging step, homogeneous blocks identified in the splitting step are compared pairwise and similar blocks are merged into a single cluster forming a homogeneous block itself. The merging stops if the obtained clusters are stable, i.e. if they are pairwise dissimilar. The final number of clusters is typically considerably reduced by the merging step.

3.1.2. Similarity assessment

The similarity assessment between two textures is done based on MPEG-7's "SCalable Color" (SCC) descriptor [4],[5]. The latter is selected among several MPEG-7 color descriptors for first investigations, because it a priori suits our application.

The SCC descriptor is basically a color histogram in the HSV color space. HSV is a three-dimensional color space with the components Hue, Saturation and Value (luminance). The resolution (number of colors or bins) of the SCC descriptor can be varied from 16 to 256 colors. We use the highest resolution in order to achieve best possible segmentation results given the SCC descriptor.

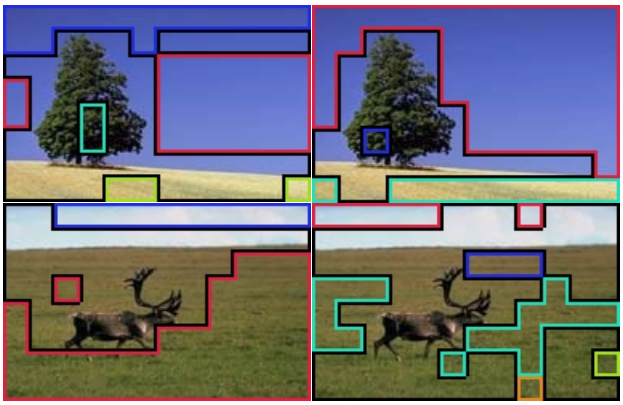

Figure 3 - Segmentation results obtained for SCC (left column) and SCC-RO (right column) given EMD as the similarity measure and two test images

The MPEG-7 standard conforming SCC histogram is best for textures with varying hue values and constant luminance and saturation values [10] given an adequate metric. The reference SCC descriptor was modified to achieve better segmentation results for images with varying saturation or luminance values of the same hue value. The modifications consist in re-ordering the bins of the MPEG-7 standard conforming SCC histogram, i.e. the dimension of the SCC histogram is not altered. The re-ordering (SCC-RO) basically yields storing all variations of a given hue in neighboring bins [10].

Two textures are considered to be similar if the distance between the corresponding feature vectors lies below a given threshold. The similarity threshold is optimized manually for the key frames of a given sequence. The optimal threshold is then used for all frames of the video. The texture analyzer presented here can therefore be seen as a semi-automatic segmentation algorithm. The Earth Mover's Distance (EMD) [11] is used as the metric. EMD is robust against noise, scaling and shift because it mainly compares the shapes of the histograms. This makes EMD eligible for compensating lighting variations, when used in combination with the SCC descriptor.

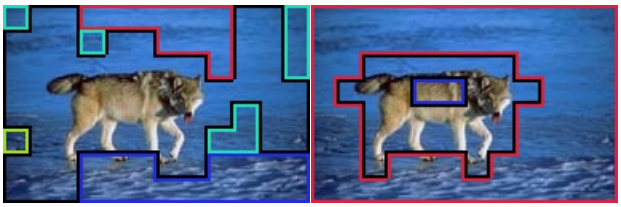

Figure 4 – Typical segmentation results obtained for the similarity measures L_1 (left) and EMD (right) given SCC-RO as the image content descriptor

Figure 3 and Figure 4 depict some segmentation results illustrating the improvements achieved by using the texture analyzer features described above. Note that detail-relevant textures are surrounded by a black border, while the others have a non-black border. SCC and SCC-RO can be seen as complementary descriptors, i.e. each of the two descriptors addresses different impacts of lighting variations. The EMD similarity measure is better than e.g. the L_1 metric for segmentation applications due to the above-mentioned properties of the former measure (Figure 4, [10]).

3.2. Temporal validation of spatial segments

3.2.1. Texture catalog

The splitting and merging steps (cp. Section 3.1.1.) segment each frame of a given sequence independently of the other frames of the same sequence. This yields inconsistent temporal texture identification. Thus a mapping of textures identified in a frame to textures identified in previous frames of the same sequence is required. However, in our approach it is important that the temporal consistency of identified textures is provided for a group-of-frames (GoF). A GoF encompasses two key frames (first and last frame of the GoF) and several partially synthesized frames between the key frames. Key frames are either I or P frames and coded using MSE as distortion criterion.

Temporal consistency of detected synthesizable textures is ensured by setting up a "texture catalog". Each identified texture is mapped to one of the indexed textures if similar. In case the current texture is not available in the texture catalog, the latter is updated with the SCC or SCC-RO feature vector of the considered texture.

3.2.2. Warping of segmented areas

The reliability of the color-based identification of synthesizable parts of a GoF is increased by matching the detail-irrelevant texture regions in the partially synthesized frames with the corresponding texture regions in the key frames. This mapping is achieved by warping the identified texture regions in the current frame towards the corresponding textures in the first or the last frame of the GoF. That is, within our framework, a detail-irrelevant texture region in a given partially synthesizable frame can only be synthesized if the same texture is available in one of the key frames (cp. Figure 1 and Figure 2). Warping is done using the planar perspective model as defined by the Parametric Motion Descriptor in MPEG-7 [4],[5]. The parametric motion of each identified texture region in relation to the first and last frame of the GoF is estimated as described in [12]. For TS I, the key frame that leads to the best synthesis result is used and a control parameter is set accordingly for the considered detail-irrelevant texture region. In the case of TS II, a motion parameter set is transmitted for each key frame containing the considered texture region. That is, two, one or no parameter sets are transmitted for a given detail-irrelevant texture region. The control parameter corresponding to the considered texture region is set accordingly.

The temporal validation of detail-irrelevant textures is optimized, in terms of maximization of the synthesizable frame area, compared to the corresponding approach presented in [13]. The proposed optimization relies on the observation that the selected similarity threshold is often too conservative for some frames of the considered video sequence, as it represents a trade-off between the optimized thresholds of several key frames (cp. Section 3.1.2.). That is, the identified area of a given detail-irrelevant texture can be smaller than it would be for an optimized similarity threshold. In such cases, the texture surrounding the identified detail-irrelevant texture region is marked detail-relevant by the texture analyzer although the former is of the same class as the identified detail-irrelevant texture. Thus we define an unreliable synthesis mode, where samples surrounding the identified detail-irrelevant texture region (in the key frame) are used for synthesis of the corresponding region in the partially synthesized frame if necessary.

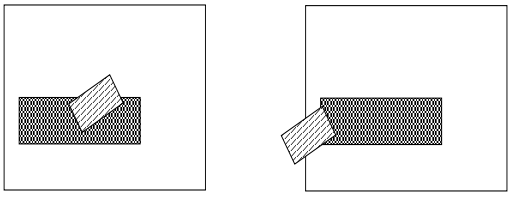

Figure 5 – Turning detail-irrelevant samples identified by the color-based texture analysis into detail-relevant samples in the course of temporal validation of spatial segments

In the reliable synthesis mode, only samples of identified detail-irrelevant texture regions are used for synthesis, i.e. detail-irrelevant samples that are warped towards a given

key frame and lie outside the corresponding texture region in the key frame are marked detail-relevant in the partially synthesized frame. As shown in Figure 5 (left) only the intersection between the detail-irrelevant texture identified in the key frame (dotted area) and the corresponding warped texture region identified in a given synthesizable frame (hatched area) is used for synthesis in the reliable mode. As a result of this, the unreliable mode yields larger detail-irrelevant texture regions than the reliable mode. Note that identified detail-irrelevant samples, for which the warped version lies outside the frame area of the corresponding key frame (cp. Figure 5 right), are turned into detail-relevant in the partially synthesized frame irrespective of the selected mode.

4. EXPERIMENTAL RESULTS

We have integrated the texture analyzer and synthesizer into an H.264/AVC codec [14]. The test sequences, Husky, Stefan, Flowergarden, Concrete and Canoe are used to demonstrate that an approximate representation of some textures can be done without subjectively noticeable loss of quality. Note that in some few frames of Husky and Stefan test sequences segmentation errors leading to false synthesis are eliminated manually.

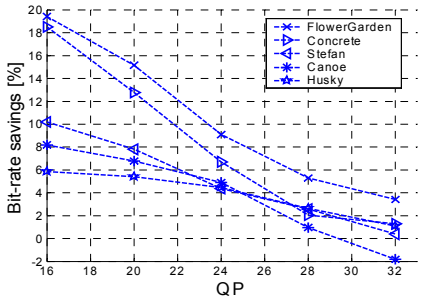

Figure 6 - Bit-rate savings w.r.t. quantization accuracy

The following set-up is used for the H.264/AVC codec: Three B frames, one reference frame for each P frame, CABAC (entropy coding method), rate distortion optimization, 30Hz progressive video at CIF resolution. The quantization parameter QP was set to 16, 20, 24, 28 and 32.

Bit-rate savings of up to 19.4% are measured for the Flowergarden sequence as shown in Figure 6. It can be seen that the higher the video quality and thus the required bit-rate, the bigger the bit-rate savings. This is due to the fact that the volume of the side information remains constant over the different QP settings. Substantial bit-rate savings are also achieved for the other test sequences as shown in Figure 6, except for Canoe at QP 32, where the side information yields a bit-rate higher than the bit-rate required by the reference video codec to transmit the video sequence.

The visual quality at the selected QP settings is in all cases comparable to the quality of the decoded sequences using the standard codec. Sequences for subjective evaluation can be down-loaded from http://bs.hhi.de/~ndjiki/SE.htm.

5. CONCLUSIONS AND FUTURE WORK

A video coding approach using a texture analyzer at the encoder side and a texture synthesizer at the decoder side was presented. This system is tested by integrating our modules into an H.264/AVC codec. Bit-rate savings of up to 19.4% are achieved given similar subjective quality as the reference H264/AVC video codec.

As a future work item, an analysis-synthesis loop for online consistency check of synthesized frames will be developed to ensure good video quality at the decoder output and to enable frame-adaptive parameter settings for analysis and synthesis.

6. REFERENCES

[1] S.-Y. Yoon and E. H. Adelson, “Subband texture synthesis for image coding”, *Proc. SPIE on HVEI III*, Vol. 3299, pp. 489-497, San Jose, USA, January 1998.

[2] A. Dumitraş and B. G. Haskell, “An Encoder-Decoder Texture Replacement Method with Application to Content-Based Movie Coding”, *To be published in IEEE Trans. on CSVT*.

[3] E. Steinbach , T. Wiegand, and B. Girod, “Using Multiple Global Motion Models for Improved Block-Based Video Coding”, *Proc. ICIP*, Vol. 2, pp. 56-60, Kobe, Japan, October 1999.

[4] ISO/IEC JTC1/SC29/WG11/N4358, „Text of ISO/IEC 15938-3/FDIS Information technology – Multimedia content description interface – Part 3 Visual“, Sydney, Australia, July 2001.

[5] ISO/IEC JTC1/SC29/WG11/N4362, „MPEG-7 Visual Part of eXperimentation Model Version 11.0“, Sydney, Australia, July 2001.

[6] A. Smolic, T. Sikora and J.-R. Ohm, “Long-Term Global Motion Estimation and its Application for Sprite Coding, Content Description and Segmentation”, *IEEE Trans. on CSVT*, Vol. 9, No. 8, pp. 1227-1242, December 1999.

[7] ISO/IEC JTC1/SC29/WG11/N3515, “MPEG-4 Video VM Version 17.0”, Beijing, China, July 2000.

[8] L.-Y. Wei and M. Levoy, „Fast Texture Synthesis using Tree-structured Vector Quantization”, *Proc. of SIGGRAPH 2000*, Conference on Computer Graphics and Interactive Techniques, New Orleans, USA, July 2000.

[9] J. Malki et al., „Region Queries without Segmentation for Image Retrieval by Content”, *VISUAL'99*, pp.115-22, 1999.

[10] P. Ndjiki-Nya, O. Novychny and T. Wiegand, “Video Content Analysis Using MPEG-7 Descriptors”, *Proc. CVMP 2004*, London, Great Britain, March 2004.

[11] Y. Rubner, et al., “A Metric for Distributions with Applications to Image Databases”, *ICCV'98*, pp.207-214, 1998.

[12] A. Smolic and J.-R. Ohm, “Robust Global Motion Estimation Using a Simplified M-Estimator Approach”, *Proc. ICIP2000*, Vancouver, Canada, September 2000.

[13] P. Ndjiki-Nya, et al., “Improved H.264 Coding Using Texture Analysis and Synthesis”, *Proc. ICIP 2003*, Vol. 3, pp. 849-852, Barcelona, Spain, September 2003.

[14] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC: "Advanced Video Coding for Generic Audiovisual Services", 2003.