Extending HEVC Using Texture Synthesis

Bastian Wandt, Thorsten Laude, Yiqun Liu, Bodo Rosenhahn, and Jörn Ostermann

Leibniz Universität Hannover, Institut für Informationsverarbeitung
Appelstr. 9a, 30167 Hannover, Germany

{wandt, laude, liuyiqun, rosenhahn, office}@tnt.uni-hannover.de

***Abstract*—The High Efficiency Video Coding (HEVC) standard provides superior coding efficiency compared to its predecessors. Nevertheless, the encoding of complex and thus hardly to predict textures either requires high bit rates or results in low quality of the reconstructed signal. To compensate for this limitation of HEVC, we propose a sophisticated texture synthesis framework which solves multiple lacks of previous texture synthesis approaches. By easing the bit rate cost for synthesizable regions and reallocating the freed bit rate resources to non-synthesizable regions, for high-value soccer content we are able to achieve average BD-rate gains of 21.9% for all-intra, 17.6% for low delay, and 16.3% for random access, respectively, while maintaining the same objective quality for the latter. Subjective tests for the synthesizable regions confirm the objectively measured convincing results. The general applicability of our method is confirmed for other types of content.**

I. Introduction

The steady improvement of video coding algorithms resulted in 2013 in the High Efficiency Video Coding (HEVC) [1] standard which was developed by the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG. Depending on the selected configuration, HEVC achieves a 40-60% bit rate reduction over the predecessor standard Advanced Video Coding (AVC) while maintaining the same visual quality [2]. Although the overall coding efficiency is superior, analyses reveal that HEVC performs differently good for varying signal characteristics. The predictability of the currently coded block based on previously coded blocks is of crucial importance for a high coding efficiency because the resulting prediction error accounts for a major part of the overall bit rate. While signal parts with low-complexity textures or foreground objects with distinct borders can be efficiently coded, this is not possible for signal parts with high-complexity and irregular textures. These textures are hardly predictable, neither by intra prediction nor by motion compensation. While this is not a major problem for videos which are encoded at low bit rates (since the texture is low pass filtered by the coarse quantization of the prediction error), the textures require a considerably high bit rate for high quality videos. Therefore, in this paper, we focus on high quality videos which are typical for high value content such as sports broadcasts. This content is of particular interest for the broadcasting industry.

The described limitation of HEVC can be traced back to the premise of the encoding system that a high pixel-wise fidelity of the reconstructed video is a suitable indicator for a well-encoded video. However, considering the properties of the human visual system and that the viewer never saw the originally encoded video, a high pixel-wise fidelity is not imperative. It has been demonstrated in multiple works (e.g. [3], [4]) that texture synthesis is an adequate procedure to cope with the low efficiency of conventional coding methods for these complex textures. Instead of aiming at pixel-wise fidelity, texture synthesis algorithms target a compelling subjective quality of the reconstructed video. The above mentioned methods achieve plausible results for sequences shown by the authors. However, most of the authors only show simple sequences which do not include challenges like lighting changes and frequency changes of textures which are to be expected for realistic sequences. Ndjiki-Nya et al. [5] were the first to consider motion of textures during the scene and define a simple motion model. However, they do not consider more complex camera motions like tilting and zooming. Dumitras and Haskell [4] synthesize very simple regions without noticeable lighting and frequency changes in low resolution pictures. Reconstructing lighting changes was addressed by several authors (e.g. [3] and [6]). They use information from neighboring pixels which allows for a plausible luminance reconstruction at the edges but cannot reconstruct lighting gradients reasonably well. Therefore, this approach is not well suited for larger areas. In all prior works a frequency change in a textured region is not considered explicitly. In contrast to that, we reconstruct the texture, motion, luminance gradients, and frequency components by using a small set of variables.

We segment the encoded video into synthesizable and non-synthesizable regions. Subsequently, we use texture synthesis to reconstruct the synthesizable regions. The remaining parts of the signal are encoded conventionally. Thereby, the bit rate costs for the synthesizable regions are drastically reduced and we achieve a high subjective quality for these regions. Furthermore, the released bit rate resources can be reallocated

978-1-5386-0462-5/17/ $31.00 VCIP 17, Dec. 10 - 13, 2017, St Petersburg, U.S.A.

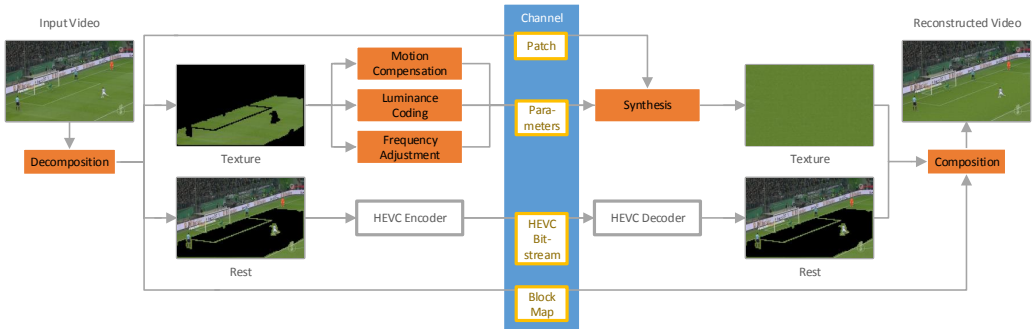

Fig. 1. Pipeline of the proposed texture synthesis system

to the conventionally encoded signal. Hence, the quality of these signal parts can be increased while maintaining the same overall bit rate.

Simply replacing a synthesizable region by a synthesized texture results in three major issues: 1) Without compensation of the camera motion, the synthesized texture is inconsistent between subsequent pictures. 2) Luminance information, perspective effects, and blurring are lost when reconstructing the texture from a single small patch. 3) Block artifacts between synthesized and non-synthesized regions result in poor subjective quality of the reconstructed video.

We solve these issues with specifically tailored texture synthesis solutions. In summary, **our contributions** are:

- motion compensation using **hyperplane fitting**,
- luminance reconstruction employing **polynomial fitting**,
- **frequency damping** by higher-order polynomial fitting to compensate perspective effects and motion blur,
- a **deblocking method** to reduce block artifacts between synthesized and non-synthesized regions by applying a *mincut* algorithm to neighboring blocks at region borders.

Furthermore, our method does not require any changes in the existing HEVC bit stream format or in the encoding or decoding process. Thereby, off-the-shelf HEVC encoders and decoders can be used to implement systems which use the proposed texture synthesis solution. The remainder of this paper is organized as follows: The proposed texture synthesis algorithm is presented in detail in Section II. Section III describes the evaluation of our method with objective and subjective tests and Section IV concludes the paper.

II. Decomposition and Texture Synthesis

The proposed method is based on the idea that a picture can be decomposed in textured and non-textured regions. We select a small picture patch containing all structural information of the textured region. Using patch-based texture synthesis algorithms such as [7] and [8] we are able to reconstruct the structural information of the region from this patch. Since lighting and blurring information is lost when simply replacing the region with a synthesized region, this information needs to be signaled. Fortunately, it can be coded very efficiently due to geometric properties of most texturized surfaces. The synthesis is only done once per scene and per tracked region of similar texture in this scene. This yields the necessity to deform the synthesized region which is done by calculating a linear transformation model. These steps are further elaborated on in the following sections and their consecutiveness is illustrated in Fig. 1 with a block diagram of our pipeline.

A. Region detection and tracking

A textured region is automatically detected by a k-means classification step. The feature vector for each pixel consists of the three color values and the picture coordinates. This five-dimensional vector not only enforces similar color but also spacial proximity. To track the regions between subsequent pictures, the detected regions are matched to the regions in the previous picture by using the Hungarian matching algorithm [9]. The Hungarian graphs edge weights are linear combinations of three distances between regions. Namely these are the distances of the centroids, differences in the number of pixels, and numbers of overlapping pixels.

B. Motion Compensation

Most textured surfaces lie on a plane in the underlying 3D scene. In consequence, camera motions such as pan, tilt, and zoom result in linear deformations of the textured area in the camera plane. We calculate a plane that approximates these deformations in x- and y-direction, respectively. The plane corresponding to the deformation u in x-direction can be described by the first order polynomial, $a_0+a_1x+a_2y = u$, where x and y are the picture coordinates and $a_{0,1,2}$ are the plane parameters. With v as deformation in y-direction the other plane can be defined analogously. Thereby, only 6 parameters are sufficient to reconstruct the deformation of the synthesized area in two consecutive pictures. The deformation is obtained by the dense optical flow between these two pictures using the algorithm of [10].

C. Luminance Coding

We assume that most textured regions are homogeneously lit. Hence, there exists a lighting gradient over the textured area. Reconstructing the scene lighting is essential to obtain a synthesized area that visually blends in pleasantly to the neighboring transform-coded blocks. By extracting the luminance information from the original picture we fit a higher-order polynomial (i.e. a plane) to the luminance map similar to Sec. II-B. The order of the polynomial depends on the lighting in the scene and determines the number of variables necessary to encode the luminance efficiently. Our experience shows that a first-order polynomial is sufficient in most cases.

D. Frequency Adjustment

Although a textured region has a similar structure in the whole region it also includes perspective effects and motion blur resulting in less high frequency energy. To compensate these effects we calculate the block-wise DCT transform of a block in the original picture and a block of the patch. We introduce a damping function for the AC components of the DCT coefficients of the patch. A new coefficient $\hat{c}_{i,j}$ at position i, j in the block is calculated from the coefficient $c_{i,j}$ of the synthesized region by

$$\hat{c}_{i,j} = c_{i,j} \cdot d^{(i+j)}, \tag{1}$$

where d is considered as the constant damping factor for this block. Calculating the damping factor results in a damping map over the textured region. Following the same argumentation as in Sec. II-C we fit another second-order polynomial to the damping map. It is easier to add blurring to a high-frequency block than to sharpen a low-frequency block. Therefore, we select the patch with the highest frequencies for the synthesis.

E. Encoder Integration and Signaling

There exist multiple possibilities to integrate the proposed texture synthesis approach into existing pipelines. As one possibility, one could fully integrate the texture synthesis into the HEVC encoder and decoder. However, this would require to change the encoder and decoder implementation as well as the bit stream format. We believe that this is undesirable for existing systems which are deployed for instance in the broadcasting industry. Thus, following the approach of Meuel et al. [11], we implement our algorithm solely as pre- and post-processing solutions. For this purpose, the sample values of the pixels in the synthesizable regions of the video are replaced by zeros. These *blacked* regions can be encoded very efficiently with HEVC because they contain only DC energy in contrast to the noisy green grass and are replaced by the synthesized textures at the decoder side. All information that is required by the decoder (to perform the synthesis and for the composition of conventionally coded regions and synthesized regions) is signaled in the bit stream. Three supplementary bit streams besides the main HEVC bit stream are encoded: The texture patch is signaled as separate HEVC bit stream once per scene. Per picture, there are 15 parameters for the synthesis which require in total 255 binary symbols. The decomposition of the picture into synthesizable and non-synthesizable regions is signaled by one binary symbol per 8×8 block. As the experimental evaluation in Sec. III will reveal, the signaling of the parameterization and of the binary map is negligible compared to the main HEVC bit stream. Thus, we straightforwardly employ gzip to compress these two descriptors.

F. Reconstruction at the Decoder

The decoder performs a patch-based texture synthesis known as *Image Quilting* [7] or *GraphCut Textures* [8]. It returns a picture slightly larger than the reconstructed region which is inserted in the blocks of the synthesizable region. This only needs to be done once per scene. In consequence, temporal coherence is ensured and the computational effort is reduced. It has not escaped our notice that this way the random access is limited to scene cuts. However, we believe that this is reasonable for the cutting techniques used by directors for the given application. Subsequently, the two planes corresponding to motion vector fields (cf. Sec. II-B) are reconstructed. The texture picture is transformed according to these planes. Since the texture picture consists of subsamples of the patch, its luminance is homogeneous. Therefore, we subtract the mean luminance of the patch from the reconstructed luminance surface (cf. Sec. II-C). Thus, the luminance difference is added to the reconstructed area. Reconstruction of the frequency can be done by applying the damping factor block-wise.

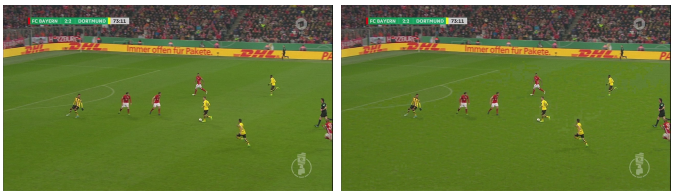

Fig. 2. Coded pictures using HEVC (left) and the proposed texture synthesis extension (right) for the sequence *Soccer 4*.

Performing these steps subsequently results in a visually pleasing textured region. However, the border between synthesized and non-synthesized blocks is still visible. To camouflage these borders and to therefore avoid visible blocking artifacts we apply a *mincut* algorithm. By overlapping a non-synthesized block with a synthesized block we can calculate a difference between them. We then finally calculate a minimal cut through this difference matrix in a similar way as during the texture synthesis with *Image Quilting* [7]. It is worth noting that the chosen synthesis algorithms are not suitable to dynamic textures which are considered as out-of-scope for our application.

III. Experimental Results

For the evaluation, the proposed texture synthesis algorithm was implemented based on the HEVC reference software HM-16.4. The encoder configurations all intra (AI), low delay (LD), and random access (RA) as defined in the HEVC *common test conditions* [12] were used for the evaluation. A set of four test sequences with a spatial resolution of 1280×720 pel and a temporal resolution of 50Hz as listed in Table I was used for the evaluation. As we are aiming at high value sports broadcasts, scenes from soccer matches were selected. A high quality is imperative for such content. For the evaluation of the HEVC range extensions at high qualities, the four quantization parameters 12, 17, 22, 27 were used [13]. Therefore, we adopt this quantization parameter tier.

The premise of texture synthesis approaches that a high pixel-wise fidelity is not imperative for the subjective quality of encoded textures allows for considerable bit rate reductions. This poses a tremendous challenge for the quantitative evaluation of texture synthesis approaches because metrics like PSNR do not apply. Our motivation is that the texture synthesis can free bit rate resources which can be reallocated to the non-synthesizable regions. Hence, we measure the overall bit rate

TABLE I
WEIGHTED AVERAGE BD-RATES AND MODE USAGE

	AI	LD	RA	**Mean**	Usage
Soccer 1	-19,95%	-13.24%	-12.21%	**-15.13%**	38.02%
Soccer 2	-17.36%	-12.15%	-10.44%	**-13.32%**	36.41%
Soccer 3	-20.25%	-11.77%	-12.90%	**-14.97%**	39.18%
Soccer 4	-30.15%	-33.18%	-30.05%	**-31.13%**	46.20%
Mean Soccer	**-21.93%**	**-17.59%**	**-16.4%**	**-18.64%**	39.95%
People on Street	-6.8%	-1.8%	-1.8%	**-3.5%**	11.6%
Basketball Drive	-16.8%	-13.8%	-13.5%	**-14.7%**	26.1%
Park Scene	-6.0%	-1.7%	0.3%	**-2.5%**	9.6%

TABLE II
RESULTS OF THE SUBJECTIVE TEST. NUMBER OF ACR AND CCR RATINGS FOR EACH SEQUENCE AND THE CALCULATED MOS VALUES.

	ACR						CCR							
	5	4	3	2	1	MOS	-3	-2	-1	0	1	2	3	MOS
Soccer 1	7	14	8	3	0	3.78	0	1	1	7	5	14	4	1.31
Soccer 2	2	12	15	3	0	3.41	0	0	1	3	7	15	6	1.69
Soccer 3	2	12	10	8	0	3.25	0	0	0	3	6	11	12	2.00
Soccer 4	0	2	7	19	4	2.22	0	0	0	0	1	8	23	2.69
Mean						**3.16**								**1.92**

(including the conventionally coded parts of the video and all the side information for the texture synthesis) and the PSNR on the non-synthesizable regions. By computing the Bjøntegaard-Delta (BD)-rate as defined in [14], we are able to conclude the extend of the desired bit rate reallocation to the non-synthesizable regions. As suggested in [15], weighted average BD-rates were calculated with weighting factors of 6/1/1 for the three color components Y/Cb/Cr. The resulting BD-rates are summarized in Table I. Mean coding gains of 18.64% are achieved for the high-value soccer content. Furthermore, evaluating the mode usage, it is observed that on average 39.95% of the pixels are synthesized. In average, the additional side information for the texture synthesis accounts for 1% of the overall bit rate. To demonstrate the applicability of our method for other types of content, results for multiple MPEG test sequences are summarized in Table I as well. The coding gains depend on the percentage of pixels which are synthesized. For instance, a mean BD-rate gain of 14.7% is achieved for the sequence *Basketball Drive* for which 26.1% of the pixels are synthesized.

To evaluate the subjective quality of the reconstructed scenes, we performed two experiments with 32 subjects following the standardized procedure of ITU-T Recommendation P.913 [16]. In the first experiment, the synthesized scenes were rated individually on the *Absolute Category Rating* (ACR) scale into the categories *bad (1)*, *poor*, *fair*, *good*, and *excellent (5)*. We calculate the *Mean Opinion Score* (MOS). In the second experiment, the subjects compared the synthesized sequence to a conventionally coded sequence at approximately the same bit rate. We used the *Comparison Category Rating* (CCR) with the seven levels *much worse (-3)*, *worse*, *slightly worse*, *the same (0)*, *slightly better*, *better*, and *much better (+3)*. The results for both subjective experiment are shown in Table II. The first experiment (ACR) reveals that most persons rate the synthesized sequences as *good* or *fair* with deviations in both directions. The second experiment (CCR) suggests that the subjects prefer the conventionally coded sequence over the synthesized sequence if they know both. However, this is not the case for the real-world application. Hence, it can be concluded that the subjects accept the texture synthesis as long as they do not know the original from a direct comparison which confirms the premise of our work. A visual example for the synthesis is provided in Fig. 2.

IV. CONCLUSION

In this paper, we proposed a sophisticated texture synthesis algorithm. With this algorithm, we were able to overcome the limitation of HEVC that the encoding of complex and thus hardly to predict textures requires high bit rates to achieve high quality. In summary, we eased the bit rate costs with average weighted BD-rate gains of 18.64% for high-value soccer content. Furthermore, we confirmed the high subjective quality of the reconstructed pictures with a comprehensive subjective test. These findings back up our assumption that it makes sense to have specialized coding tools for high-value content. The applicability of the method is confirmed for other types of content.

REFERENCES

[1] "ITU-T Recommendation H.265/ ISO/IEC 23008-2:2013 MPEG-H Part 2: High Efficiency Video Coding (HEVC)," 2013.

[2] J. De Cock, A. Mavlankar, A. Moorthy, and A. Aaron, "A large-scale video codec comparison of x264, x265 and libvpx for practical VOD applications," A. G. Tescher, Ed. International Society for Optics and Photonics, 9 2016, p. 997116.

[3] D. Liu, X. Sun, and F. Wu, "Edge-Based Inpainting and Texture Synthesis for Image Compression," in *IEEE International Conference on Multimedia and Expo (ICME)*, 7 2007, pp. 1443–1446.

[4] A. Dumitras and B. Haskell, "An Encoder-Decoder Texture Replacement Method With Application to Content-Based Movie Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 825–840, 6 2004.

[5] P. Ndjiki-nya, B. Makai, A. Smolic, H. Schwarz, and T. Wiegand, "Video Coding Using Texture Analysis And Synthesis," Tech. Rep., 2003.

[6] F. Racapé, S. Lefort, D. Thoreau, M. Babel, and O. Déforges, "Characterization and adaptive texture synthesis-based compression scheme," in *European Signal Processing Conference, EUSIPCO*, 8 2011, pp. 1–5.

[7] A. A. Efros and W. T. Freeman, "Image Quilting for Texture Synthesis and Transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 341–346.

[8] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Transactions on Graphics, SIGGRAPH 2003*, vol. 22, no. 3, pp. 277–286, 7 2003.

[9] H. W. Kuhn, "The Hungarian Method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[10] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2010, pp. 2432–2439.

[11] H. Meuel, M. Munderloh, F. Kluger, and J. Ostermann, "Codec independent region of interest video coding using a joint pre- and postprocessing framework," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 7 2016, pp. 1–6.

[12] F. Bossen, "JCT-VC L1100: Common HM test conditions and software reference configurations. 12th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. Geneva, CH," 2013.

[13] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 26, no. 1, pp. 4–19, 1 2016.

[14] G. Bjøntegaard, "VCEG-AI11: Improvements of the BD-PSNR model. ITU-T SG 16 Q 6. 35th Meeting, Berlin, Germany," 2008.

[15] G. J. Sullivan and J.-R. Ohm, "Meeting Report of the Fourth Meeting of the Joint Collaborative Team on Video Coding," *ITU-T/ISO/IEC JCT-VC Document JCTVC-D500*, 2011.

[16] "ITU-T Recommendation P.913 : Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment," 2016.