Extending HEVC with a Texture Synthesis Framework using Detail-aware Image Decomposition

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

Institut für Informationsverarbeitung, Leibniz Universität Hannover
Appelstraße 9a
Hannover, Germany
{wandt,laude,rosenhahn,office}@tnt.uni-hannover.de

***Abstract*—In recent years, there has been a tremendous improvement in video coding algorithms. This improvement resulted in 2013 in the standardization of the first version of High Efficiency Video Coding (HEVC) which now forms the state-of-the-art with superior coding efficiency. Nevertheless, the development of video coding algorithms did not stop as HEVC still has its limitations. Especially for complex textures HEVC reveals one of its limitations. As these textures are hard to predict, very high bit rates are required to achieve a high quality. Texture synthesis was proposed as solution for this limitation in previous works. However, previous texture synthesis frameworks only prevailed if the decomposition into synthesizable and non-synthesizable regions was either known or very easy. In this paper, we address this scenario with a texture synthesis framework based on detail-aware image decomposition techniques. Our techniques are based on a multiple-steps coarse-to-fine approach in which an initial decomposition is refined with awareness for small details. The efficiency of our approach is evaluated objectively and subjectively: BD-rate gains of up to 28.81% over HEVC and up to 12.75% over the closest related work were achieved. Our subjective tests indicate an improved visual quality in addition to the bit rate savings.**

I. Introduction

In 2013, the joint effort of ITU-T VCEG and ISO/IEC MPEG resulted in the latest video coding standard High Efficiency Video Coding (HEVC) [1], [2] which is also known as H.265 and MPEG-H Part 2. Compared to its predecessor standard AVC/H.264, HEVC considerably increases the coding efficiency: depending on the selected configuration, HEVC achieves a 40-60% bit rate reduction while maintaining the same visual quality [3], [4]. However, the superior coding efficiency depends on the characteristics of the coded signal. Since the prediction error accounts for a major part of the overall bit rate, the predictability of the currently coded block based on previously coded blocks is of crucial importance to fully capitalize the benefits of the standard. This predictability exists for low-complexity textures or for foreground objects with distinct borders but not for high-complexity and irregular textures. Neither intra nor inter coding are capable of properly predicting these textures. The impact of this limitation of HEVC is especially severe for high quality video in which it is desired to retain the texture details. This application is of particular interest for the broadcasting industry. On the other hand, this is not a major problem for videos which are encoded at low bit rates because the details of the texture are low pass filtered by the coarse quantization of the prediction error. Therefore, in this paper, we focus on high value content (e.g. sport broadcasts) for which high quality is imperative to the consumer. The described limitation of HEVC is of systematic nature because it can be traced back to the premise that a high pixel-wise fidelity of the reconstructed video is an indicator for a high video quality. However, considering the properties of the human visual system and that the viewer of the broadcast never saw the original input video to the encoder, a high pixel-wise fidelity is not imperative.

Multiple works (e.g. [6], [7], [8]) show that texture synthesis is an adequate tool to improve the coding efficiency of complex textures compared to conventional coding methods. Texture synthesis algorithms target a compelling subjective quality of the reconstructed video instead of aiming at pixel-wise fidelity. As shown by multiple authors, texture synthesis methods achieve plausible reconstructions. However, most works 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. [9] were the first to consider motion of textures during the scene and defined a simple motion model. However, they did not consider more complex camera motions like tilting and zooming. Dumitras and Haskell [7] synthesized very simple regions without noticeable lighting and frequency changes in low resolution pictures. Reconstructing lighting changes was addressed by several authors (e.g. [6] and [10]). They use information from neighboring pixels which allows for a plausible luma 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, luma gradients, and frequency components by using a small set of variables. Additionally, we tackle the essential problem of all texture synthesis approaches of finding the optimal synthesizable region. We propose a sophisticated detail-aware texture detection to avoid wrongly classified regions that would lead to visually implausible reconstructions.

For conciseness, we briefly review the pipeline from [5] and highlight the novelty over our previous work. The pipeline of our approach is shown in Fig. 1. 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 sub-

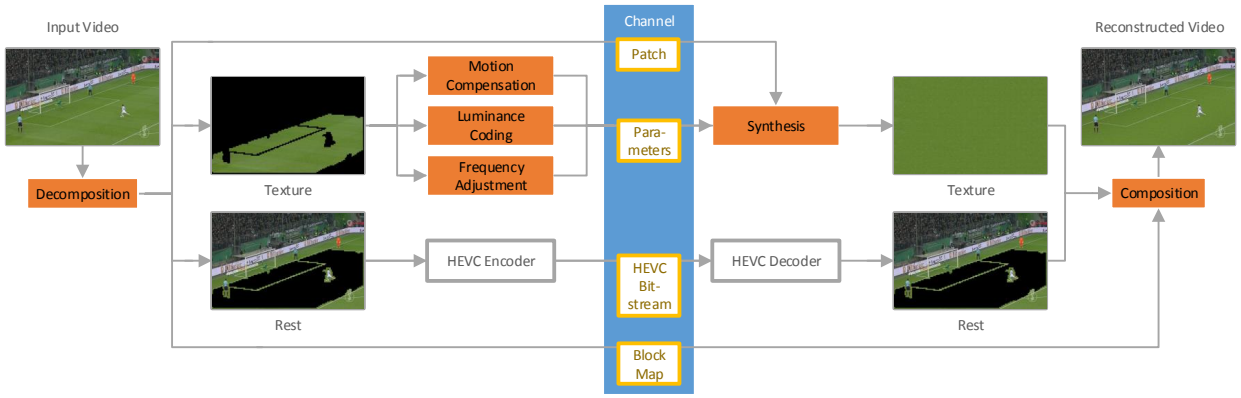

Fig. 1. Pipeline of the texture synthesis system [5].

jective quality for these regions. Furthermore, the released bit rate resources can be reallocated to the conventionally encoded signal. Hence, the quality of these signal parts can be increased while maintaining the same overall bit rate. In contrast to [5] we propose a detail-aware texture decomposition that detects even small differences in the detected synthesizable region, for instance barely visible markings and shadows in the region. Only with such a sophisticated decomposition, it is meaningful to deploy texture synthesis-based video coding systems to real-world content. This not only produces reconstructions of higher subjective quality but also reduces the bit rate as shown in Sec. III.

When a synthesizable region is simply replaced by a synthesized texture two major problems arise:

1) Without compensation of the camera motion, the synthesized texture is inconsistent between subsequent pictures.
2) luma information, perspective effects, and blurring are lost when reconstructing the texture from a single small patch.

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

- a **complete pipeline** for texture analysis and synthesis,
- a **sophisticated decompositioning technique** targeting high quality sports video broadcasting applications,
- **significantly improved** results in terms of bit rate savings and subjective quality, respectively.

Furthermore, the existing HEVC bit stream format and the encoding or decoding process do not need to be changed.

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

Our approach is based on the idea that a large textured region in a picture or video can be described by a small patch extracted from this region. This small patch needs to contain all relevant structural information of that region. However,

Fig. 2. Visualization of our clustering results. Left: original image. Middle: initial clusters after K-means. Right: Final blocks.

when reconstructing the whole region from this patch important global details such as luma and frequency changes are lost. Therefore, it is necessary to signal this information separately. By exploiting geometric properties of textured region it can be coded efficiently. The steps to detect, analyze and plausibly reconstruct the textured region are further elaborated in the following sections. Fig. 1 shows a block diagram of our pipeline. The implementation of the algorithms described in this paper includes the whole pipeline shown in Fig. 1 and tackles the most important problem of all texture synthesis approaches, namely the automatic decomposition of an input image into synthesizable and non-synthesizable regions. A meaningful decomposition directly results in better subjective quality.

A. Region detection and tracking

Most state-of-the-art approaches state that a picture can be decomposed into synthesizable and non-synthesizable regions. However, a detailed description of the decomposition is rarely given or the decomposition is assumed to be known. This section describes our clustering approach which not only detects synthesizable regions but also detects and preserves details in the region.

Firstly a textured region is roughly estimated by a K-means classification step. A feature vector

$$\boldsymbol{f} = (R, G, B, u, v) \tag{1}$$

for each pixel is built containing the three color values R, G, B and the picture coordinates u and v. The K-means clustering of all the five-dimensional vectors finds clusters consisting of pixels with similar color and in close spacial proximity. The result of this initial step is shown in the middle of Fig. 2. Due

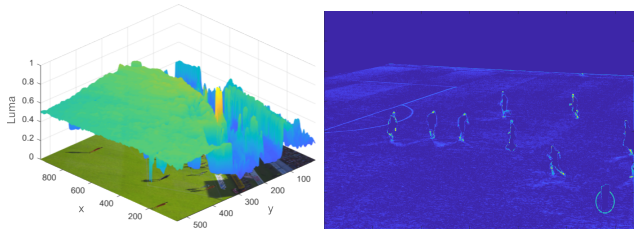

Fig. 3. 3D visualization of the luma channel scaled to [0, 1] (left) and the resulting distance map (right).

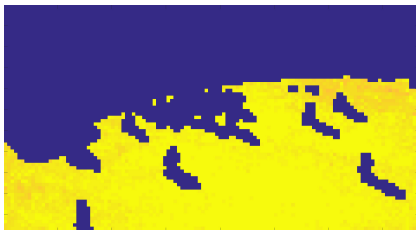

Fig. 4. The damping map that visualizes the damping coefficient distribution. Yellow correspond to a high damping coefficient that preserves high frequencies while orange corresponds to a low damping coefficient.

to the spacial proximity of the soccer field lines and the players shadows to the grass area these are marked as synthesizable. Obviously, the lines and shadows should be excluded from the synthesizable region. We propose a luma gradient distance metric that is used to detect these small anomalies in the initial region. We assume that a textured region is homogeneously lit. That means if all structural information is removed from the region the luma changes smoothly. This behaviour is modeled by fitting a plane to the luma values of the pixels in the previously marked region. It is described by

$$a_0 + a_1 x + a_2 y = L(x, y), \tag{2}$$

where x, y are the pixel coordinates, $a_{0,\dots,2}$ are the polynomials coefficients and $L(x, y)$ is the corresponding luma value at position (x, y). This gives a linear equation system which is solved for the coefficients $a_{0,\dots,2}$. Fig. 3 (left) shows a 3D visualization of the luma channel. The synthesizable region appears to be smooth. For each pixel in the region we calculate distances to the fitted surface. The distance map is shown in Fig. 3 (right). In the map, lines and shadows can be easily seen and a simple threshold is defined to distinguish between correctly and wrongly clustered pixels. This results in multiple small *holes* in the region. Considering spatial resolutions of typically 720p, 1080p or 4K in broadcasting applications, we assume that relevant details have to have a certain size. Thus, holes smaller than 64 pixels (not necessarily a squared 8×8 block) are closed.

Detected regions between two subsequent pictures are matched to the regions in the previous picture using the Hungarian matching algorithm [11]. The Hungarian graphs edge weights are linear combinations of three distances between regions. These distances are

1) difference of the centroids
2) difference of the number of pixels and
3) number of overlapping pixels.

As a final step the clustering is discretized into 8x8 blocks. An 8x8 block is marked as synthesizable if every pixel in the block is marked as synthesizable.

B. Motion Compensation

Textured surfaces mostly lie on a plane in the underlying 3D scene. That means pan, tilt and zoom camera motions result in linear deformations of the textured area in the camera plane. These deformation are approximated by fitting a plane to the deformation in x- and y-direction, respectively. Using the first order polynomial, $e_0 + e_1 x + e_2 y = u$, where x and y are the picture coordinates and $e_{0,1,2}$ are the plane parameters, we describe the plane corresponding to the deformation u in x-direction. The plane in y direction can be described analogously. Using this simple plane fitting only 6 parameters are sufficient to reconstruct the deformation of the synthesized area in two consecutive pictures. Although the plane fitting is sufficient for most cases, the model can be extended to handle perspective camera effects or camera motion with respect to curved surfaces. The difference to the above model is a higher order polynomial similar to Eq. (2). The deformation is obtained by the dense optical flow between these two pictures using the algorithm of [12].

C. Luma Coding

To achieve a visually pleasant blending of synthesized blocks to the neighboring transform-coded blocks it is essential to reconstruct the scene lighting. Using the assumption of Sec. II-A we describe the luma as a plane. Although a higher order polynomial is conceivable, our experience shows that a first-order polynomial is sufficient in most cases to achieve a sufficient blending. This reduces the number of variables to 3 to completely describe the luma gradient in one picture.

D. Frequency Adjustment

Perspective effects and motion blur can change the high frequency energy in a textured region. For pleasing visual results it is crucial to compensate these frequency changes in the reconstructed texture. Taking a textured patch from the highest frequency area in the textured region, we define a block-wise frequency damping function in the DCT domain. The objective is to match the AC coefficients of the blocks to the original ones. For each 8x8 block we calculate a damping factor d that dampens each DCT coefficient $c_{i,j}$ at position (i, j) to the new coefficient $\hat{c}_{i,j}$ by

$$\hat{c}_{i,j} = c_{i,j} \cdot d^{(i+j)}. \tag{3}$$

This gives a *damping map* as shown in Fig. 4. Similar to Sec. II-C we fit the second order polynomial

$$b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2 = D(x, y) \tag{4}$$

to the damping map D.

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. [13], 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 the *Deflate* algorithm [14] which combines the Lempel-Ziv-Storer-Szymanski algorithm with Huffman entropy coding to compress these two descriptors. These two compressed descriptors account for less than 1% of the overall bit rate.

F. Reconstruction at the Decoder

The decoder reconstructs the textured region by performing the patch-based texture synthesis of [15]. Once per sequence a region slightly larger than the reconstructed area is synthesized. This allows for the reconstruction of appearing and reappearing textures during the sequence. It also reduces computational effort and ensures temporal consistency. To perform the motion compensation the two planes corresponding to motion vector fields (cf. Sec. II-B) are used to transform the texture picture. The luma information is reconstructed by adding the mean corrected luma values of the texture image to the luma values of the reconstructed surface. Frequency reconstruction is applied by block-wise frequency damping with respect to the frequency surface. These steps lead to visually pleasing textured regions.

III. Experimental Results

In this section, we discuss our experimental results to demonstrate the efficiency of our new algorithm. For this purpose, we implement our coding system based on the HEVC reference implementation *HM-16.14*. As part of the HEVC common test conditions (CTC) [16], three encoder configurations are defined: all intra (AI), low delay (LD), and random access (RA). These configurations are used for the evaluation in combination with the four test sequences (1280×720 pel spatial resolution, 50Hz temporal resolution) which were proposed in [5]. Since we are aiming at high quality sport broadcasts, we adopt the HEVC Rext high tier of quantization parameters (12, 17, 22, 27) [17]. For the random access configurations, this covers the bit rate range between 1.6 Mbit/s and 27.3 Mbit/s which we consider as appropriate for the use case.

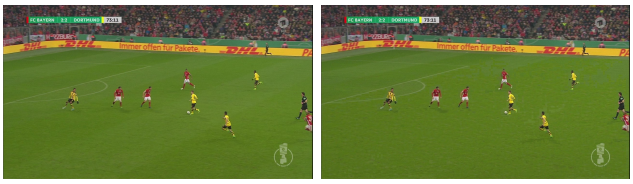

Fig. 5. Coded pictures using HEVC (left) and the proposed texture synthesis extension (right) for the sequence *Soccer 4*.

TABLE I
Luma BD-rates for the proposed algorithm in comparison with HEVC and with our previous work [5]; ratio of synthezised pixels. Negative values for the BD-rates indicate increased coding efficiency. It is observed that considerable gains are achieved both over HEVC and over the closest related work.

Configuration	Sequence	BD rate ours vs. HEVC	BD rate ours vs. [5]	Ratio of synth. pixels
All Intra	Soccer 1	-28.81	-11.4	53.6%
	Soccer 2	-15.60	-1.02	31.2%
	Soccer 3	-13.26	2.84	29.7%
	Soccer 4	-15.79	4.86	31.4%
Low Delay	Soccer 1	-27.02	-12.75	53.6%
	Soccer 2	-7.40	-1.10	31.2%
	Soccer 3	-6.51	-1.35	29.7%
	Soccer 4	-6.68	-0.30	31.4%
Random Access	Soccer 1	-23.85	-12.46	53.6%
	Soccer 2	-7.22	-1.14	31.2%
	Soccer 3	-5.45	-2.01	29.7%
	Soccer 4	-7.70	-0.57	31.4%

All texture synthesis-based video coding algorithms have in common that the objective evaluation of such algorithms poses a tremendous challenge because objective metrics like PSNR do not apply for synthesized videos. To overcome this challenge and adopting the evaluation procedure in [5], we perform both an objective and a subjective evaluation.

For the objective evaluation, we use the total bit rates (including the HEVC bit stream, the signaling of the hyper-parameters for the synthesis, and the signaling of the binary map for the decomposition of the video into synthesized and non-synthesized parts) and the luma ROI-PSNR [13] of the non-synthesized parts of the video. With these values, the luma Bjøntegaard-Delta (BD)-rate as defined in [18] can be computed. In our case, the BD-rate indicates the amount of bit rate which can be saved with our new algorithm while maintaining the same (or increasing the) subjective quality as demonstrated by the subjective tests below. It is observed that we achieve BD-rate gains between 5.5% and 28.8%

TABLE II
RESULTS OF THE SUBJECTIVE TEST. NUMBER OF CCR RATINGS FOR EACH SEQUENCE AND THE CALCULATED MOS VALUES COMPARED TO THE MOS VALUES OF OUR PREVIOUS WORK (COLUMN *MOS [5]*). THE LAST COLUMN INDICATES THE GAIN OVER OUR PREVIOUS WORK (NEGATIVE VALUES INDICATE BETTER QUALITY OF OUR NEW WORK).

	-3	-2	-1	0	1	2	3	MOS	MOS [5]	Δ-MOS
Soccer 1		1		1	1	7	8	**2.06**	2.69	-0.63
Soccer 2			3	7	4	4		**0.50**	1.31	-0.81
Soccer 3		1	2	4	7	4		**0.61**	1.69	-1.08
Soccer 4	1		1	1	4	7	4	**1.44**	2.00	-0.56
Mean								**1.15**	1.92	-0.77

over HEVC with an average gain of 13.7%. Furthermore, we increase the coding efficiency of our algorithm over our previous work [5] with an average BD-rate gain of 3%. It is noteworthy that especially for the sequence Soccer 1, for which the decomposition was considerably unsatisfying in our previous work, very high bit rate savings are achieved. Furthermore, it is observed that the gain over [5] is higher for random access and low delay than for all intra. This is because the motion compensated prediction benefits more from the avoided flickering due to the better decomposition. In average, the side information accounts for 1.49% of the total bit rate.

The subjective quality of the reconstructed scenes is evaluated following the standardized procedure of ITU-T Recommendation P.913 [19]. An experiment with 4 different sequences and a group of 18 persons was performed. As rating 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)*. We calculate the *Mean Opinion Score* (MOS) for each sequence. The four test sequences are coded conventionally using HM-16.14 and by the proposed algorithm at comparable bit rates. Table II shows the results of the subjective test. A negative MOS means the test subjects prefer our method over the other. The results of the experiment suggest that the subjects prefer the conventionally coded sequence over the synthesized sequence if they know both. However, in real world scenarios where the conventionally coded sequences are unknown test subjects accept the quality of the synthesized sequences as shown in our previous work [5]. Comparing our new results with the results of the previous work, our method improves the subjective quality of each sequence significantly. In average we gain 0.77 for the MOS score.

To conclude, our method outperforms both, HEVC and the current state-of-the-art, in terms of coding efficiency and produces visually better reconstructions than comparable methods. A visual example for the synthesis is provided in Fig. 5. There is no considerable increase in complexity.

IV. CONCLUSION

In this paper, we proposed a sophisticated texture synthesis algorithm that is well-suited for high quality sports broadcasts. By detecting small details, for instance lines and shadows, in a synthesizable region we are able to achieve visually pleasing reconstructions. 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. We reduced the bit rate costs with average BD-rate gains of 13.7% over HEVC and of 3.03% over the state-of-the-art. Since the subjective quality of the reconstructed sequences is hard to evaluate quantitatively, we performed a subjective test. The results are compared to our previous work. It was shown that we improved the state-of-the-art in both, bit rate savings and subjective quality. These findings back up our assumption that it makes sense to have specialized coding tools for high-value content.

REFERENCES

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

[2] M. Wien, *High Efficiency Video Coding - Coding Tools and Specification*, 1st ed. Berlin Heidelberg: Springer, 2015.

[3] 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, sep 2016, p. 997116.

[4] P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, "Subjective quality evaluation of the upcoming HEVC video compression standard," in *SPIE Optical Engineering + Applications*, oct 2012, p. 84990V.

[5] B. Wandt, T. Laude, Y. Liu, B. Rosenhahn, and J. Ostermann, "Extending HEVC Using Texture Synthesis," in *IEEE Visual Communications and Image Processing (VCIP)*, 2017.

[6] 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)*, jul 2007, pp. 1443–1446.

[7] 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, jun 2004.

[8] F. Zhang and D. R. Bull, "A parametric framework for video compression using region-based texture models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, Nov 2011.

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

[10] 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*, aug 2011, pp. 1–5.

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

[12] 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, jun 2010, pp. 2432–2439.

[13] 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, jul 2016, pp. 1–6.

[14] IETF, "RFC 1951: DEFLATE Compressed Data Format Specification version 1.3," 1996.

[15] 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.

[16] 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.

[17] 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, jan 2016.

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

[19] "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.