ANALYSIS/SYNTHESIS CODING OF DYNAMIC TEXTURES BASED ON MOTION DISTRIBUTION STATISTICS

Olena Chubach, Patrick Garus, Mathias Wien, Jens-Rainer Ohm

Lehrstuhl und Institut für Nachrichtentechnik, RWTH Aachen University
Melatener Str. 23
Aachen, 52074, Germany

ABSTRACT

This paper presents improvements to a dynamic texture synthesis approach which is based on motion distribution statistics, able to produce high visual quality of synthesised dynamic textures. The aim is to recreate synthetically highly textured regions like water, leaves and smoke, instead of processing them with a conventional codec such as HEVC. The method involves two steps: analysis, where motion distribution statistics are computed, and synthesis, where the texture region is synthesized. Dense optical flow is utilized for estimating the random motion of dynamic textures. The performance of our dynamic texture analysis and synthesis approach is tested on cropped sequences, containing water, leaves and smoke. Simulation results show potential bitrate savings up to 50% on texture sequences at comparable visual quality.

Index Terms**—** Dynamic texture coding, texture synthesis, optical flow, motion distribution, perceptual coding.

1. INTRODUCTION

In the state-of-the-art video coding standard HEVC [1, 2] encoder decisions are typically based on the mean squared error (MSE) criteria. However, recently, there is an increasing interest in developing content-based perceptually optimised video compression schemes [3, 4, 5, 6], where the conventional coding is combined with other approaches, such as texture synthesis. The latter may be exploited for recreating highly textured parts of the scene, which are challenging for encoding conventionally but perceptually irrelevant for a human. Therefore, the common constraint of pixel fidelity may be relaxed for such content. This allows to omit encoding prediction residuals and motion vector coding of dynamic textures, which leads to substantial reduction of bits to be coded.

Perceptually optimised video compression schemes face the problem of identifying, which regions may be synthesised and which may not. In order to focus our research on analysis and synthesis rather than identification and classification problems, in our study only homogeneous dynamic texture regions are considered, and as a consequence, cropped versions of sequences containing only dynamic textures are used for experiments.

Our scheme for analysis and synthesis of dynamic texture regions has been first introduced in [7]. Relative to the latter, the current paper is focused on the compression efficiency of the suggested approach. An improved coding structure and compression of required parameters are described.

The remainder of the paper is organised as follows. In Section 2, motion-based characteristics of dynamic textures are introduced. In Section 3, the suggested approach of dynamic texture analysis and synthesis is described. The experimental results are presented in the Section 4 and Section 5 concludes the paper.

2. MOTION-BASED CHARACTERISATION OF DYNAMIC TEXTURES

Doretto et al. [8] define dynamic textures (DT) as sequences of images of moving scenes that exhibit certain stationarity properties in time. Let us extend this definition and consider dynamic textures as image sequences that reveal certain spatiotemporal regularity, yet have an undetermined spatial and/or temporal extent. This implies that effective characterisation of DT requires analysing them in both spatial and temporal directions.

There are various studies [9, 10, 11] on characterising dynamic textures based on different statistics, including motion distribution. Yet, most of them are still focused on the texture recognition and classification, rather than analysis and synthesis. Considering non-rigid stochastic nature of motion in DT, modeling of the occlusions and disocclusions of different segments of DT caused by motion becomes a serious challenge. Methods based on optical flow (OF) are able to provide sufficient information for motion characterisation of DT, since they capture temporal variations in DT and therefore allow to consider the evolution of the motion over time. Frame-to-frame motion estimation has been extensively studied and various computationally efficient algorithms of OF estimation have been developed over the last decades.

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

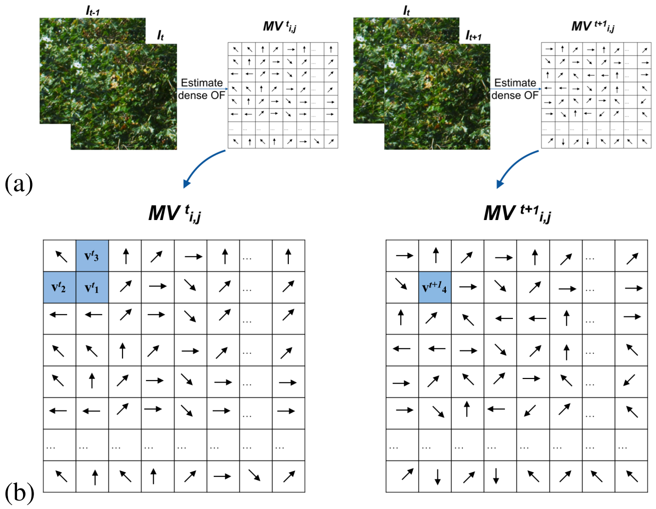

Fig. 1: MCM computation procedure. (a) Dense OF fields are estimated between adjacent frames. (b) All the combinations of four motion vectors $(v_1^t, v_2^t, v_3^t, v_4^{t+1})$ are collected to the MCM

A technique that suggests representing DT by a set of first order motion features which are computed along the space and time dimensions, was presented in [13]. Therein, a motion co-occurrence matrix (MCM) is introduced, where the motion information from spatial and temporal neighbors is considered providing an efficient representation of motion distribution in DT. This concept is employed in the current work, hence it is briefly reviewed in the following.

The MCM (**M**) is a tabulation of how often different combinations of four motion vectors $(\mathbf{v}_1^t, \mathbf{v}_2^t, \mathbf{v}_3^t, \mathbf{v}_4^{t+1})$ occur, where $(\mathbf{v}_1^t, \mathbf{v}_2^t, \mathbf{v}_3^t)$ are associated with MV_t and $\mathbf{v}_4$ - associated with MV_{t+1}. Here $MV_t(i,j)$ stands for motion vector frame (MVF), that is the dense OF field estimated between two adjacent frames I_t and I_{t-1}, describing at which position in I_{t-1} one gets a prediction for position (i,j) in frame I_t.

Let us denote the group of T frames considered for synthesis as sGOP. In the suggested approach, the motion co-occurrence matrix **M** is computed as follows. First, the dense OF is estimated between all adjacent frames of the sGOP, thus $T-1$ motion vector frames are obtained. An example of computing the MCM is illustrated in Fig.1. Here, the first two MVFs (in this case the MVF computed between frames I_{t-1} and I_t and between I_t and I_{t+1}) are considered and all the combinations of four motion vectors $(\mathbf{v}_1^t, \mathbf{v}_2^t, \mathbf{v}_3^t, \mathbf{v}_4^{t+1})$ are collected to the MCM. After that, the MVFs between frames I_t and I_{t+1} and between I_{t+1} and I_{t+2} are considered and the procedure described above is performed again. In the end, the computed MCM contains all the combinations of four motion vectors that were found in $T-1$ motion vector frames of the considered sGOP.

The information from the MCM may be utilised for generating synthetic dense motion vector fields, which would have similar characteristics as computed dense OF fields. Assuming that an initial MVF is available, e.g. estimated using dense OF method, the fourth motion vector $\mathbf{v}_4^{t+1}$ can be derived from the MCM based on information about $(\mathbf{v}_1^t, \mathbf{v}_2^t, \mathbf{v}_3^t)$. This process is described in more detail in Section 3.1

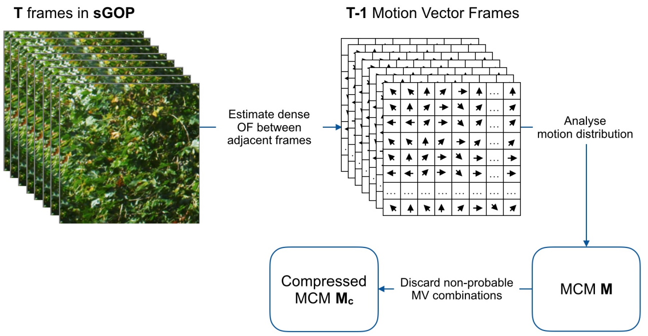
T frames in sGOP
T-1 Motion Vector Frames
Estimate dense OF between adjacent frames
Analyse motion distribution
MCM M
Discard non-probable MV combinations
Compressed MCM Mc

Fig. 2: Analysis step. Here dense OF is estimated between the adjacent frames of the sequence, and the motion co-occurrence matrix MCM is computed and compressed.

3. PROPOSED FRAMEWORK

The considered DT analysis and synthesis technique aims at analysing the input video of DT and then synthesising skipped pictures, using pictures available from past and future and additional parameters. Although it was previously presented in [7], various aspects of this method are discussed in this paper. Therefore, it is briefly described in the following section.

3.1. Analysis and synthesis

The method implies two steps: analysis and synthesis. During the analysis step, dense OF is estimated between the adjacent frames of the sequence, the motion distribution statistics are collected and the motion co-occurrence matrix MCM is computed, as described in the previous section. After computing the MCM, it is compressed and the compressed motion co-occurrence matrix **Mc** is signaled to the decoder side for synthesis. The schematic representation of the analysis step described above is presented in Fig.2.

In order to reduce the amount of additional information required for synthesis, it was decided to use a dedicated coding structure, such that reference frames at the beginning and at the end of every sGOP are reconstructed first and with better quality. Thereby, transmission of the initial OF needed for initialising the synthesis step is not required as it may be estimated directly by applying dense OF algorithm to the adjacent frames. The remaining frames must be skipped during encoding/decoding and are synthesised. The example of modified coding structure for the case of sGOP size 8 is illustrated in Fig. 3. In this case, the first sGOP consists of frames 1 to 8 and the second sGOP consists of frames 9 to 16. However, other options, with different sGOP sizes and decoding structures are possible, depending on video content.

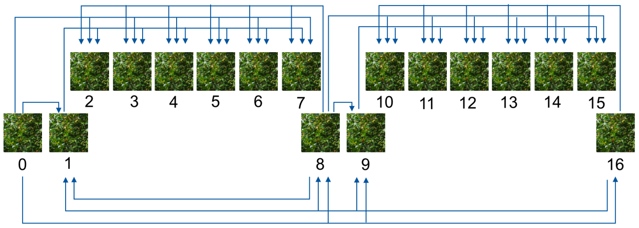

Fig. 3: Modified coding structure. Here reference frames 1, 8, 9 and 16 are reconstructed first and with better quality; the remaining 6 frames are considered to be synthesised. OF is estimated between frames 0/1, 8/9, 16/17 etc.

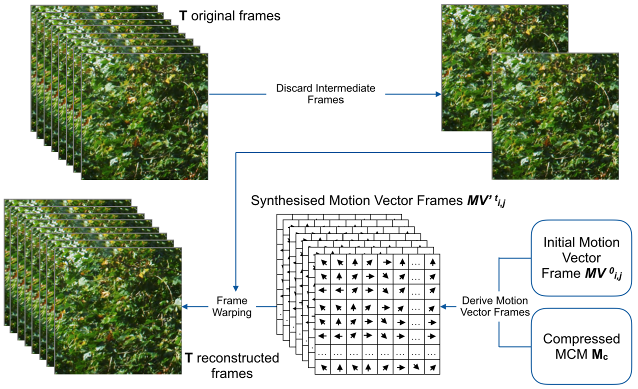
T original frames
Discard Intermediate Frames
Synthesised Motion Vector Frames MV'^t_{i,j}
Initial Motion Vector Frame MV^0_{i,j}
Frame Warping
Derive Motion Vector Frames
Compressed MCM Mc
T reconstructed frames

Fig. 4: Synthesis step. Every synthetic MVF in the sGOP is predicted considering previous MVF and **Mc**. Afterwards, the synthetic MVFs are utilised for generating intermediate frames.

The schematic description of the synthesis step is presented in Fig.4. In case of the shown coding structure, all intermediate frames of the considered sGOP, except for the first and the last frame of the sGOP, can be discarded. Moreover, the initial MVF does not have to be transmitted, as it may be estimated by applying a dense OF method between the last picture of the previous sGOP and the first picture of the current sGOP. In order to decide during synthesis on the entire motion vector and not on each component separately, every motion vector is mapped from a two-component vector to an entity k. The mapping is based on the considered motion vector range for the corresponding sGOP, which indicates the maximum MV length that shall not be exceeded by any motion vector component in the sGOP. To discard outliers and still allow for bigger motion vectors within the sequence, the optimum range is computed for each sGOP separately.

The first synthetic MVF in the sGOP is predicted based on the estimated initial motion vector field and the corresponding compressed motion co-occurrence matrix **Mc**. Any motion vector combination from **Mc** which has corresponding motion vectors $(\mathbf{v}_1^t, \mathbf{v}_2^t, \mathbf{v}_3^t)$ can be considered for synthesis. Among them, the motion vector is chosen which provides the best consistency with the luma value in the corresponding reference frame when it is compared to the best matching area in the next reference frame. When all the motion vectors are predicted, Gaussian smoothing with $\sigma = 1.3$ is applied to the synthesised MVF to make it more consistent and to remove possible outliers. At the next step, the synthesised MVF is utilised for estimating the next motion vector frame of the sGOP in the same way. After that, synthesized MVFs are utilised for generating intermediate frames using conventional pixel-based motion vector compensation. In case if motion vectors indicate a sub-sample position, intensity values are interpolated. In the suggested approach, the HEVC quarter-pixel interpolation filters have been applied.

For video compression, future frames may be available and temporal consistency of the content has to be preserved. In order to further improve the quality of synthesis in the suggested approach, the synthesis procedure is performed twice: in forward direction using frames from the past, and in backward direction using frames from the future. An analogous approach is suggested in [14]. Finally, both synthesised results are merged using the coefficients

$$\lambda(t) = \frac{\operatorname{atan}\left(\frac{9}{D-1} * (t - \frac{D}{2})\right)}{\pi} + 0.5, \tag{1}$$

where D corresponds to the distance between the reference frames in the sGOP. Then, the final values are computed as

$$\hat{I}_t = (1 - \lambda(t))\hat{I}_t^f + \lambda(t)\hat{I}_t^b, \tag{2}$$

where $\hat{I}_t^f$ and $\hat{I}_t^b$ are corresponding forward and backward synthesised frames at position t.

3.2. Coding of side information

The side information which has to be compressed and transmitted for synthesis contains MCMs and the corresponding motion vector range for every sGOP. As described earlier, every motion vector is mapped to an entity k based on the motion vector range, that is why the range is required for reconstruction and has to be transmitted to the decoder side.

The four obtained entities define a key combination s_k. The number of occurrences of every s_k is counted and then the probability mass function is computed. Afterwards, rarely found s_k values are discarded in order to remove motion vectors that were incorrectly estimated (e.g. due to the failure of the OF) and therefore might result in bad predictions.

The remaining set of s_k combinations is sorted and differences between adjacent s_k entries are computed. This leads to a result containing small numbers and many zeros. Every unique number is then considered as a symbol of the alphabet with a corresponding cumulative frequency. The message consisting of the modified MCMs and the interval ranges is encoded by an arithmetic coder. Finally, the symbols and their cumulative frequencies are compressed with Exp-Golomb code.

The described procedure provides significant reduction of bitrate needed for coding additional parameters required for synthesis, when compared to the results presented in [7].

4. EXPERIMENTS AND RESULTS

The proposed method was tested on a set of 11 sequences, featuring static camera and containing cropped water, leaves and smoke, listed in Table 1. All sequences (except for PetiBato) used for experiments were taken from the HomTex database [15]. The size of each sequence is 256x256 pixels, every sequence consists of 250 frames, with a frame-rate of 60fps. For our experiments, the Farneback algorithm [16] was employed for computing dense OF and only 50% of the most probable motion vector combinations in the MCM were utilised. Preliminary experiments indicated that this is the best tradeoff between the quality of synthesis and the amount of information to be sent.

Although the new coding structure may not be useful when all frames would be encoded, it indeed can reduce the bitrate when discarding intermediate frames that are to be synthesized. For example, for the case of the sGOP size 8 (see Fig.3) 6 out of every 8 frames can be discarded. This yields to significant bitrate savings, and, therefore, allows to code the additional parameters required for synthesis, and also encode the remaining frames with higher quality.

Due to the fact that considering more frames for analysis makes MCM more stable to outliers, in our experiments one MCM was computed for every 2 sGOPs (which corresponds to 16 frames per MCM, see Fig.3).

Analysis and synthesis were performed on the frames encoded with HEVC Test Model (HM-16.6) using modified configuration described in Section 2 and $QP = 22$. Frames that correspond to the first and the last key frame of the sGOP were preserved for synthesis. The total amount of information required for synthesis is shown in the third column of Table 1, where the first term indicates the rate required for reference frames, and the second term expresses the rate needed for the synthesis parameters. The rate reduction that corresponds to the synthesis rate is indicated in the fourth column.

Table 1: List of sequences and results

Sequence	HEVC rate, ($QP22$), [kb]	Synth. rate Modif. RA, ($QP22$) [kb]	Rate reduction %
BallUnderWater	50.1	19.5 + **0.23**	-60.7
BricksBushes Static-Bushes1	1744.5	766.1 + **5.7**	-55.7
BricksBushes Static-Bushes2	1579.1	717.3 + **2.9**	-54.4
LampLeaves-Bushes1	1578.9	728.3 + **11.9**	-53.1
LampLeaves-Bushes2	1146.4	507.4 + **54.1**	-51.0
LampLeaves-Bushes3	1294.6	545.8 + **135.5**	-47.4
LampLeaves-Bushes Background	552.4	281.2 + **7.5**	-47.8
Petibato-cropped	735.1	391.5 + **153.3**	-25.9
SmokeClear-middle	97.2	40.2 + **0.23**	-58.5
SmokeClear-side	145.04	62.02 + **0.23**	-57.1
TreeWills-cropped	970.6	584.3 + **0.44**	-39.8
Average			**-50.1**

5. CONCLUSIONS AND DISCUSSION

The suggested approach to synthesis-based coding of dynamic textures shows promising subjective results in conjunction with bitrate savings. Depending on the configuration, the synthesized content can reach similar subjective quality when compared to HEVC compressed sequences at the corresponding bitrates. Therefore, it may be applied for content which has high rate demand in conventional coding.

So far it was investigated only with homogeneous dynamic textures, but should be applicable for local areas as well. Integration into a complete coding scheme is planned for future research.

6. ACKNOWLEDGEMENTS

This work has been conducted in context of the PROVISION Initial Training Network. This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 608231.

The authors would like to thank C. Feldmann for very interesting and fruitful discussions on HEVC configurations.

7. REFERENCES

[1] Mathias Wien, *High Efficiency Video Coding – Coding Tools and Specification*, Springer, Berlin, Heidelberg, 2015.

[2] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] Patrick Ndjiki-Nya, Tobias Hinz, and Thomas Wiegand,

“Generic and robust video coding with texture analysis and synthesis,” in *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1447–1450.

[4] Johannes Balle, Aleksandar Stojanovic, and Jens-Rainer Ohm, “Models for static and dynamic texture synthesis in image and video compression,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1353–1365, 2011.

[5] Fan Zhang and David 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, 2011.

[6] Fabien Racapé, Olivier Déforges, Marie Babel, and Dominique Thoreau, “Spatiotemporal texture synthesis and region-based motion compensation for video compression,” *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 993–1005, 2013.

[7] Olena Chubach, Patrick Garus, and Mathias Wien, “Motion-based analysis and synthesis of dynamic textures,” in *Proc. of International Picture Coding Symposium PCS ’16*, Nuremberg, Germany, Dec. 2016, IEEE, Piscataway.

[8] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto, “Dynamic textures,” *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[9] Sándor Fazekas and Dmitry Chetverikov, “Analysis and performance evaluation of optical flow features for dynamic texture recognition,” *Signal Processing: Image Communication*, vol. 22, no. 7, pp. 680–691, 2007.

[10] Renaud Péteri and Dmitry Chetverikov, “Dynamic texture recognition using normal flow and texture regularity,” in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2005, pp. 223–230.

[11] V Andrearczyk and Paul F Whelan, “Dynamic texture classification using combined co-occurrence matrices of optical flow,” in *IRISH MACHINE VISION & IMAGE PROCESSING Conference proceedings 2015*, 2015.

[12] Dmitry Chetverikov and Sándor Fazekas, “On motion periodicity of dynamic textures.,” in *BMVC*, 2006, vol. 1, pp. 167–176.

[13] Ashfaqur Rahman, Manzur Murshed, et al., “Dynamic texture synthesis using motion distribution statistics,” *Journal of Research and Practice in Information Technology*, vol. 40, no. 2, pp. 129, 2008.

[14] Fabien Racape, Dimitar Doshkov, Martin Köppel, and Patrick Ndjiki-Nya, “2d+ t autoregressive framework for video texture completion,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4657–4661.

[15] Mariana Afonso, Angeliki Katsenou, Fan Zhang, Dimitris Agrafiotis, and David Bull, “Video texture analysis based on hevc encoding statistics,” in *Proc. of International Picture Coding Symposium PCS ’16*, Nuremberg, Germany, Dec. 2016, IEEE, Piscataway.

[16] Gunnar Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Scandinavian conference on Image analysis*. Springer, 2003, pp. 363–370.