Segmentation-based Partitioning for Motion Compensated Prediction in Video Coding

Max Bläser, Cordula Heithausen, Mathias Wien
Institut für Nachrichtentechnik
RWTH Aachen University, GERMANY
{blaeser, heithausen, wien}@ient.rwth-aachen.de

***Abstract*—In video coding standards such as *Advanced Video Coding (AVC)* and its successor *High Efficiency Video Coding (HEVC)*, motion compensation is performed by partitioning each inter-predicted picture into square or rectangular regions. While HEVC introduces an efficient quad-tree based splitting of square-shaped coding blocks into prediction blocks by symmetric and asymmetric motion partitioning, the boundaries of natural moving objects can only be approximated by a fine block partitioning, resulting in a redundant representation of the motion and therefore a potential coding overhead. This paper studies the method of using a segmentation based partitioning of coding blocks into arbitrary shaped segments, where the segmentation is performed on coded reference pictures. Experimental results show that for cases where a reasonable segmentation can be obtained, bitrate reductions of around 2% can be achieved.**

I. Introduction

Motion vector fields for natural video sequences typically show the distinctive characteristics of a piecewise smooth signal with strong discontinuities at boundaries of moving objects. For the purpose of inter-picture prediction, modern video codecs such as HEVC search for an efficient representation of the dense motion by using a rectangular motion partitioning in combination with a translational motion model. Rectangular regions exhibiting homogeneous and coherent motion are represented by a single motion vector. The partitioning is chosen according to a rate-distortion optimization with the general notion, that a finer quantization of the motion vector field permits a better inter-prediction but at a higher bitrate for signaling the partitioning information and motion parameters [1].

Rectangular block partitioning has the fundamental limitation that it can only capture vertical and horizontal edges. In case of challenging video sequences with a high amount of motion, a very fine block partitioning along object boundaries can typically be observed by analysis of the partitioning structure. Although the fine partitioning is selected to be optimal in terms of the rate-distortion criterion, it is obvious that redundancy is introduced as motion vectors on either side of the object boundary will show similar motion [2]. This observation is exemplified in Figure 1, where the partitioned motion vector field is shown compared to the bit per pixel coding cost for motion vectors and partitioning side-information. The fine block partitioning can be avoided by applying a different block partitioning strategy. Different methods allowing a higher degree of freedom have been proposed in the past with varying coding gains and computational complexities. This paper studies the approach of using *Segmentation Based Partitioning (SBP)* for coding blocks as an additional partitioning option in extension of the HEVC codec. The block partitioning is estimated from the segmentation of reference pictures, which are used as additional side-information. By using blocks partitioned into arbitrarily shaped segments, less motion vectors need to be coded and the true motion of objects can be approximated more closely. The paper is organized as follows. Section II gives an overview over the current and proposed block partitioning strategies in modern video coding standards. Subsequently, Section III gives an overview of the SBP coding tool integrated into an extended version of HEVC. Section IV details aspects of SBP such as the reference picture segmentation process, segment-wise prediction and coding of the additional SBP related side-information. In Section V comparative simulation results are presented. Section VI concludes the paper.

II. Block Partitioning in HEVC and beyond

Methods of block partitioning vary depending on their flexibility and can generally be classified to fit one of the following categories:

- Rectangular partitioning with symmetric and asymmetric modes (Fig. 2a).
- Geometric partitioning, where a block is partitioned by an arbitrary straight line (Fig. 2b).
- Object-based partitioning, where the object boundary is approximated by a complex curve (Fig. 2c).

In HEVC and its predecessor AVC, motion compensation is performed via rectangular block partitioning. In HEVC, the largest possible block size has been increased to 64×64 pixels compared to AVC. A picture is partitioned into *Coding Tree Blocks (CTBs)*, which can be split into a quad-tree, where the nodes of the tree are square-shaped *Coding Blocks (CBs)*. Each CB can be further split into rectangular *Prediction Blocks (PBs)* by symmetric or asymmetric block partitioning. In total, eight different splitting configurations are possible for every inter-predicted CB [3].

Especially *Geometric Partitioning (GEO)* has been the subject of much research [4] [5] and has been proposed during the standardization of HEVC [6][7][8][9]. Coding of

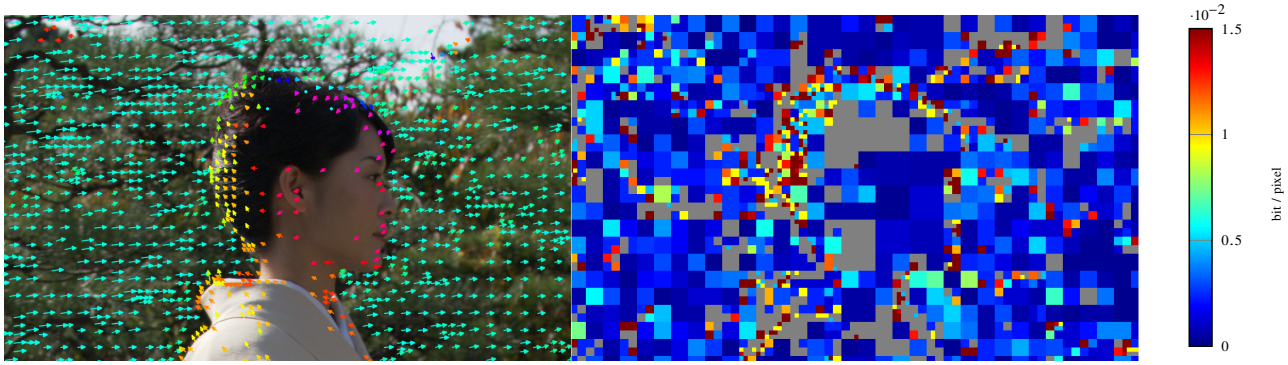


Fig. 1. Motion partitioning in case of a moving background and foreground. The left pictures exemplifies the motion vector field, the right picture shows the bit/pixel coding cost needed for the signaling of motion vectors and partitioning information.

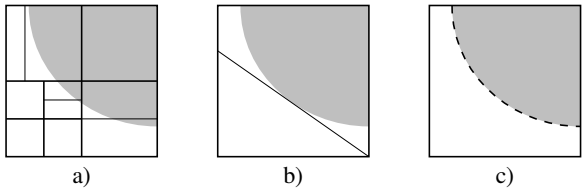


Fig. 2. Block partitioning: a) rectangular partitioning, b) geometric partitioning, c) object-based partitioning

the partitioning information is straightforward by specifying the position of a straight line within the block, which can be achieved in multiple ways: Escoda, Yin et. al introduced a polar coordinate representation [2], Muhit, Pickering et. al. utilize the line intercept points at the block boundary [10] and Paul and Murshed [11] chose to use a template based method of signaling. Variations of these methods have also been proposed during the standardization process of HEVC, where motion estimation complexity concerns have been addressed by a coarse quantization of line parameters, effectively reducing the number of allowed geometric partitioning modes or templates [12][13].

Object-based partitioning methods allow an even higher degree of freedom in terms of splitting a block. Here, a block is typically partitioned by an arbitrary shaped curve into two distinct segments. Thus, binary shape coding techniques as for example specified in MPEG-4 Visual can be used to transmit the partitioning, such as contour-based coding or context-based arithmetic coding [14].

A more recent approach in the 3D extension of HEVC (3D-HEVC), which exploits the benefit of arbitrary-shaped block partitioning, is made by Jäger with *Depth Based Block Partitioning (DBBP)* [15]. DBBP is a coding tool that utilizes the depth map as side-information in order to perform a segment-wise prediction of the texture. Explicit coding of the shape of the partition boundary is avoided by thresholding of the collocated depth map into a binary segmentation mask. Average bitrate reductions of -0.5% are reported for the texture.

An approach similar to SBP has been proposed by Chen, Lee et. al. [16] and later by Milani and Calvagno [17], both in the context of H.264/AVC, where thresholding of the luminance component is performed at the macroblock level in order to generate a segmentation mask.

III. Segmentation Based Partitioning in HEVC

Similar as in DBBP and in contrast to MPEG-4 Visual, the partition boundary is not explicitly coded using aforementioned shape coding techniques in the SBP approach. By means of image segmentation, the block partitioning can be derived with little additional signaling: while in DBBP the depth map values of the current block are thresholded to generate a binary segmentation mask, in SBP, the segmentation mask is obtained from the coded reference pictures. It is assumed that image segmentation allows the identification of coherent regions of pixels, which are affected by the same motion. If it is further assumed that object boundaries remain reasonably rigid over time and are only subject to translational movement, the partitioning of the current block can be estimated from segmented reference pictures using only an additional motion vector.

Segmentation methods such as single component thresholding as in [16] may produce reasonable segmentation results for small blocks of size 16×16 pixels in AVC, but it is deemed unsuitable for larger blocks such as the maximum specified CB size of 64×64 pixels as in HEVC and even larger blocks in proposals beyond HEVC.
For SBP, fast k-means clustering [18] in the YCbCr color space with additional spatial constraints is used, generating a segmentation mask on a block by block basis, where the block size can vary from 16×16 to 128×128 pixels. Cluster centroids are computed from the two chrominance values for each pixel. Thus, segmentation is performed based on color only. In order to unify the proposed partitioning method with existing HEVC coding tools, only two clusters and therefore two distinct segments per CB are considered. At the encoder side, motion estimation is performed on each segment independently. Segments can further be predicted from one

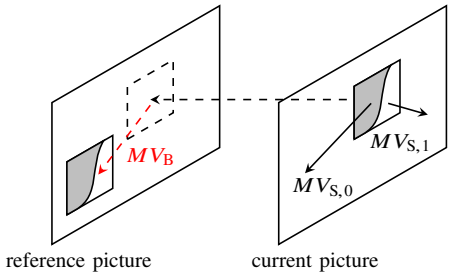


Fig. 3. Estimation of a matching partitioning through segmentation of a reference picture at a shifted position.

or two reference pictures with optional *weighted prediction* as in HEVC [3]. The predicted segments are merged together to form the motion compensated prediction block. Although deblocking as in HEVC is performed at block boundaries, no deblocking is applied to the samples on either side of the partitioning boundary within the block. Coding of segment motion vectors is achieved in the same way as for rectangular PBs, using *Advanced Motion Vector Prediction (AMVP)* and *Motion Vector Merging* [1] across block boundaries.

The segmentation mask of the current block must be known at the decoder to generate the motion compensated prediction block. In order to locate a rate-distortion optimal segmentation within one of the reference pictures, a boundary motion vector is needed in addition to the segment motion vectors. This boundary motion vector points to the position where a segmentation boundary can be found that best fits to the object boundary present in the current block. Figure 3 exemplifies the relation between the two segment motion vectors $MV_{S,0}$, $MV_{S,1}$ and the boundary motion vector MV_B, which is also exploited in the coding of the additional motion vector.

IV. Algorithm Description

In the following, the set of pixels $\mathbf{p}(x, y)$ of the current block for a given size of $N \times N$ is denoted as B and the two segment sub-sets as S^0 and S^1, such that $S^0 \cup S^1 = B$.
The motion vector MV_B points to a position within a given reference picture, where a segmentation matching the segmentation of the current block can be obtained. Therefore at the indicated position, image segmentation of the reference picture region having the same size as the current block is performed. The block diagram in Figure 4 shows the individual steps used in the segmentation process.

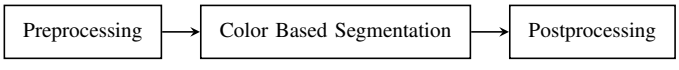


Fig. 4. Generation of segmentation mask for block partitioning through segmentation of reference pictures.

A. Reference Picture Segmentation

Preproccessing is applied in order to stabilize the segmentation result. The reference picture region is median-filtered with a kernel size of 5×5 in order to remove noise and small textures. As segmentation is performed on the chrominance, the reference block needs to be upsampled in case the input source is chroma subsampled.
For fast convergence of the clustering, a good initialization of the k-means algorithm is needed. Popular initialization methods such as the k-means++ variant [18], which rely on randomly selected starting points, can not be used safely in an encoder and decoder environment. Instead, a strictly deterministic initialization is used. Subsequently, the k-means algorithm is executed and for each block, an overall objective J_B is minimized,

$$J_B = \sum_{k=1}^{K} \sum_{\mathbf{p}_c \in S^k} d(\mathbf{p}_c, \mathbf{m}_k) \tag{1}$$

where $K = 2$ is the total number of clusters per block, $\mathbf{p}_c$ the two-dimensional pixel chrominance assigned to a cluster S_k, $\mathbf{m}_k$ the respective cluster centroid and d a distance norm. For large blocks of size greater than 32×32 pixels, a subsampling grid is applied to the color values $\mathbf{p}_c$ to lower the computational load when calculating the cluster centers. In order to further improve the segmentation result, which is especially useful for small blocks below 16×16 pixels, additional pixels from the surrounding neighborhood are also sampled. Figure 5 exemplifies a possible subsampling of the reference picture region and usage of additional samples from the neighborhood for a block of 8×8 pixels.

Through experimentation, it was determined that five iterations of k-means clustering suffice to achieve stable cluster centers. Subsequently, a binary segmentation mask M_B^i is generated. By comparing the distances d_i and d_j of the two respective cluster centers, every pixel $p(x, y)$ within the given reference block can be assigned unambiguously to either segment S^0 or S^1.

$$M_B^i(x, y) = \begin{cases} 1 \text{ if } d_i(x, y) < d_j(x, y) \\ 0 \text{ otherwise} \end{cases} \quad i, j \in \{0, 1\},\ i \neq j \tag{2}$$

$$S^i = B \circ M_B^i,\ i \in \{0, 1\} \tag{3}$$

After the binary valued segmentation mask has been generated, postprocessing is applied in order to remove any noise and close holes in the mask. For this purpose, simple morphological opening- and closing is applied using a square shaped kernel of size 3×3.

B. Segment-wise prediction

At the encoder, motion estimation is performed segment-wise to determine the segment motion vectors. Again, an objective function J_M is minimized,

$$J_M^i = D_{SAD}^i + \lambda R_{MV}^i \tag{4}$$

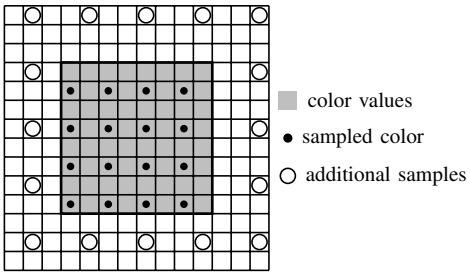


Fig. 5. Example of a subsampling grid applied to block using additional samples from the neighborhood.

where D^i_{SAD} is the *Sum of Absolute Differences (SAD)* measure of the current segment S^i luminance compared to a shifted reference block, R^i_{MV} is an estimate of the coding rate for the motion vectors and λ the Langrangian multiplier, which is a quality-dependent value. Block-matching is typically used to minimize J_{M} in an iterative process. As this is a computationally demanding part during encoding, fast block-based methods have been developed, utilizing vectorized hardware acceleration. These methods can easily be adapted for a segment-based motion estimation by taking the segmentation mask M^i_{B} into account when computing D^i_{SAD}:

$$D^i_{\mathrm{SAD}} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |B(x,y) - R(x+\Delta u, y+\Delta v)| \circ M^i_{\mathrm{B}} \qquad (5)$$

It must be noted that many block-matching algorithms use the *Sum of Absolute Transform Differences (SATD)* measure to find the best motion vectors. However, the usage of a masked SATD measure – in analogy to the masked SAD in Eq. 5 – is not easily possible. The masking of the residual $B - R$ would result in sharp edges and therefore high energy transform coefficients.

C. Coding of additional side-information

As the vector MV_{B} represents the motion of the boundary, MV_{B} is in many cases similar to one of the segment motion vectors $MV_{\mathrm{S},0}$ or $MV_{\mathrm{S},1}$, depending if the boundary moves in conjunction with the fore- or background. Figure 3 illustrates this relation for the case that the boundary moves with segment S^0. It must be noted that no explicit distinction is made in the proposed SBP approach, whether segment S^0 or S^1 is considered to be part of the foreground object or background respectively. The fact that MV_{B} is similar to $MV_{\mathrm{S},0}$ or $MV_{\mathrm{S},1}$ can directly be exploited for coding. In analogy to the PB-wise coding of motion vectors in HEVC, MV_{B} is predictively coded from the segment motion vectors. While motion vectors in HEVC are estimated and coded with quarter-pixel accuracy, the boundary motion vector is signaled in full-pixel accuracy.

Signaling of the SBP mode is achieved through a combination of flags at the CTB and CB level. A first flag coded at the root CTB indicates if the SBP mode is used in any of the CB nodes. If the flag is true, for each inter-predicted CB, a second flag is coded, which indicates if the block is a SBP block. If the CTB flag is false, no further CB flags are coded. For both flags as well as the additional motion vector, the efficient *Context-Based Adaptive Binary Arithmetic Coding (CABAC)* of HEVC is utilized.

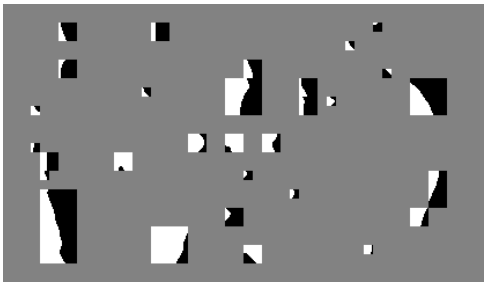
V. Experimental Results

For evaluation, the SBP coding tool has been implemented in the HM-14.0-KTA-1.0 HEVC Reference Software [19], which is a post-HEVC modification of the HEVC HM Reference Software, already containing additional coding tools such as OBMC, Sub-PU-TMVP and further lifting constraints on the maximum allowed block- and transform size. The evaluation of the proposed SBP mode is conducted in *Low-Delay B (LDB)* and *Low-Delay P (LDP)* configuration from the JCTVC CTC on a set of 12 sequences of different resolutions and of two different types: 3 sequences are cropped versions taken from the CTC of JCTVC-3V (3D-HEVC) and are computer-generated sequences for which also a depth map is available. The remaining sequences are mostly taken from the CTC of JCTVC and are sequences with different amounts of motion.
The computer-generated sequences are chosen in order to investigate the influence of the segmentation method on the rate-distortion performance: In a separate experiment, instead of using the color components of reference pictures, the depth maps of reference pictures are used for segmentation. Under the assumption that the depth map allows a perfect distinction between fore- and background, the coding results for this experiment can thus be seen as the upper limit, as an optimal segmentation of all moving objects is possible.

As a reference point, the KTA-1.0 software is run with the SBP coding tool turned off. The simulation results in table V show that for the computer-generated sequences, where segmentation of the depth map allows a perfect distinction between fore- and background, an average rate reduction of approximately -2.2% for LDB and LDP can be achieved.

For natural sequences and computer-generated sequences, where the color components are used to segment the reference pictures, the coding gains are significantly lower at approximately -0.4% for LDB and -0.6% for LDP configurations. This can be explained by the fact that the encoder will not choose the SBP mode in case of faulty and misaligned segmentation results. Small, isolated blocks, where a clear segmentation is easier to generate, as can be seen in Figure 6b, will further decrease any potential coding gain. These blocks can not use the efficient motion vector merging for the additional boundary motion vector as no neighboring SBP blocks are available. In general, sequences which contain heavily textured background such as the *Kimono* sequence (see Fig. 1) or fast and blurry motion such as the *BasketballDrive* sequences are difficult to segment correctly.

(a) Original



(b) Segmentation

Fig. 6. Original picture and segmentation masks for SBP blocks chosen by the encoder; "RaceHorses" test sequence, frame 193, coded at QP27.

TABLE I
SIMULATION RESULTS FOR LDB, LDP AND RA CONFIGURATION.

Sequence	**Resolution**	**BD-Rate Change (%)**			
		LDB		LDP	
		Depth	Color	Depth	Color
UndoDancerCrop	512×768	-2.53	-0.43	-2.37	-0.36
SharkCrop	768×512	-2.72	-0.91	-2.80	-1.07
GTFlyCrop	768×512	-1.39	-0.14	-1.27	-0.12
BlowingBubbles	416×240		-0.30		-0.17
RaceHorses	800×480		-0.95		-1.05
BasketballDrillText	832×480		-0.31		-0.59
ChinaSpeed	1024×768		-0.76		-0.69
SlideEditing	1280×720		-0.55		-1.17
BasketballDrive	1920×1080		-0.08		-0.18
Kimono	1920×1080		-0.14		+0.04
PeopleOnStreet	2560×1600		-0.08		-1.44
Drummer	3840×2160		-0.22		-0.60
Average		**-2.21**	**-0.39**	**-2.14**	**-0.62**

VI. CONCLUSION

This paper proposes a segmentation based partitioning for hybrid video coding, offering the encoder a block-based choice between traditional rectangular partitioning and object-boundary based partitioning of motion. While bitrate reductions of approximately 2% can be achieved in case of a perfect segmentation, they are lower for the currently used method of color based k-means clustering. The method can be improved in the future by using a more reliable method of segmentation, also taking the motion vector fields of already decoded pictures into account. With improvement of the segmentation method and more consistency of the mode usage over time, visual improvements can also be expected, as better approximation of object boundaries will also be beneficial for the transform coding of the prediction error and lead to sharper edges at low bitrates.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. 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.

[2] O. D. Escoda, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, April 2007, pp. I–657–I–660.

[3] M. Wien, *High Efficiency Video Coding – Coding Tools and Specification*. Berlin, Heidelberg: Springer, Sep. 2014.

[4] E. M. Hung, R. L. D. Queiroz, and D. Mukherjee, "On macroblock partition for motion compensation," in *Image Processing, 2006 IEEE International Conference on*, Oct 2006, pp. 1697–1700.

[5] L. Guo, P. Yin, Y. Zheng, X. Lu, Q. Xu, and J. Sole, "Simplified geometry-adaptive block partitioning for video coding," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 965–968.

[6] M. Karczewicz, P. Chen, R. Joshi, K. Wang, and W.-J. Chien, "Video coding technology proposal by qualcomm," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Dresden, Germany, 1st meeting, Tech. Rep. JCTVC-A121, Apr. 2010.

[7] L. Guo, P. Yin, and E. Francois, "TE3: simplified geometry block partitioning," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Geneva, Switzerland, 2nd meeting, Tech. Rep. JCTVC-B085, Jul. 2010.

[8] P. Bordes, P. Chen, I.-K. Kim, L. Guo, H. Yu, and X. Zheng, "CE2: unified solution of flexible motion partitioning," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Geneva, Switzerland, 5th meeting, Tech. Rep. JCTVC-E374, Mar. 2011.

[9] X. Zheng, H. Yu, S. Li, Y. He, and P. Bordes, "CE2: non-rectangular motion partitioning," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Torino, Itlay, 6th meeting, Tech. Rep. JCTVC-F415, Jul. 2011.

[10] A. A. Muhit, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Video coding using fast geometry-adaptive partitioning and an elastic motion model," *Journal of Visual Communication and Image Representation*, vol. 23, no. 1, pp. 31 – 41, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320311000770

[11] M. Paul and M. Murshed, "Video coding focusing on block partitioning and occlusion," *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 691–701, March 2010.

[12] P. Bordes, E. Francois, and D. Thoreau, "Fast encoding algorithms for geometry-adaptive block partitioning," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept 2011, pp. 1205–1208.

[13] Q. Wang, X. Ji, M. T. Sun, G. J. Sullivan, J. Li, and Q. Dai, "Complexity reduction and performance improvement for geometry partitioning in video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 338–352, Feb 2013.

[14] ISO/IEC, "Information Technology–Coding of Audio-Visual Objects–Part 2: Visual." 14496-2:2004 (MPEG 4), Apr. 1999.

[15] F. Jäger, "Segment-wise prediction in 3D video coding," in *Proc. of IEEE International Conference on Consumer Electronics ICCE '15*. Las Vegas, USA: IEEE, Piscataway, Jan. 2015.

[16] J. Chen, S. Lee, K.-H. Lee, and W.-J. Han, "Object boundary based motion partition for video coding," in *Picture Coding Symposium (PCS)*, 2007.

[17] S. Milani and G. Calvagno, "Segmentation-based motion compensation for enhanced video coding," in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 1649–1652.

[18] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, "The effectiveness of lloyd-type methods for the k-means problem," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, Oct 2006, pp. 165–176.

[19] Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, "Joint Exploration Test Model (JEM) 1.0." [Online]. Available: https://jvet.hhi.fraunhofer.de/