

Compressed Domain Video Compositing with HEVC

Robert Skupin, Yago Sanchez, Thomas Schierl

Multimedia Communications Group

Fraunhofer Heinrich-Hertz-Institute

Einsteinufer 37, 10587 Berlin

{robert.skupin;yago.sanchez;thomas.schierl@hhi.fraunhofer.de}

Abstract—Video compositing such as blending of user interfaces or advertisements on top of video content is used in many applications. Compositing is usually carried out in the pixel domain either after decoding on the end device or based on transcoding before or during transport, e.g. on cloud resources. This paper proposes a novel method to create a composition of several coded input videos in the compressed domain, i.e. without performing entropy coding at runtime. The method entails merging of input video bitstreams into a single output video bitstream and insertion of pre-encoded inter-predicted composition pictures. Such a lightweight approach is computationally much less demanding than transcoding-based compositing and can be beneficial for service scalability. This paper explores the coding performance of the proposed method through experiments with the composition of a transparent ticker overlay on top of video sequences. The proposed method is reported to outperform transcoding-based pixel domain compositing in rate distortion and quality.

I. INTRODUCTION

Video compositing is used in numerous applications in which a composition of multiple video sources is presented to the user. Common examples are picture-in-picture (PiP) compositing and transparent blending of overlays with video content, e.g. for advertisements or user interfaces. Producing such compositions in the pixel-domain requires parallel decoding of input video bitstreams that is computationally complex and may even be infeasible on devices with a single hardware decoder or otherwise limited resources. For instance, in current IPTV system designs, capable set top boxes carry out compositing and are a major service cost factor due to their complexity, distribution and limited lifetime. Reducing these cost factors motivates ongoing efforts to virtualize set top box functionality, e.g. shifting user interface generation to cloud resources. Mere video decoders, so-called zero clients, are the only hardware to remain at the customer premises in such an approach [1]. The state of the art in such a system design is compositing based on transcoding, i.e. in its simplest form: decoding, pixel-domain compositing, and re-encoding before or during transport. To reduce the workload from a full de- and encoding cycle, operation in the transform coefficient domain instead of the pixel-domain was first proposed for PiP compositing in [2]. Since then, numerous techniques to fuse or cut short the individual compositing steps and apply them to current video codecs were proposed with a good overview given in [3]. However, transcoding based approaches for general compositing are still computationally complex which compromises system scalability. Depending on the transcoding

approach, such compositing may also impact rate distortion (RD) performance.

In this paper, the authors propose a novel method for compressed domain compositing of coded video bitstreams that takes a different approach to compositing and that can be a key component for the scalability of IPTV and other systems. Compositions are produced by merging coded input video bitstreams into a single common output bitstream and inserting pre-encoded, inter-predicted pictures that use the merged input bitstreams as reference.

The proposed method relies on a number of features of H.265/High Efficiency Video Coding (HEVC) [4] (the successor of H.264/AVC). HEVC features the spatial segmentation of video pictures into a grid of so-called tiles [5]. In contrast to the comparable H.264/AVC feature, flexible macroblock ordering, tiles are included in all currently existing HEVC profiles. These tiles are coded independently within a picture and divide it along the borders of coding tree units that serve as basis for block-based coding. Tiles are especially interesting for another form of compressed domain video processing, namely video stitching as presented in [6] for conferencing scenarios with decoder modifications. Extensions thereof were proposed in [7] and [8] to allow application in standard compliant HEVC decoders and are applied in this work as well. HEVC further improves on the implicit reference picture management of H.264/AVC by introduction of the robust reference picture sets (RPSs) concept [9]. Reference pictures in HEVC are explicitly signaled per picture slice, which allows for significantly easier loss detection but also for easier reference manipulation with respect to H.264/AVC, a key component of the proposed method. Another interesting new concept in HEVC that is used in the proposed method are non-output pictures. These pictures are decoded and can be used for reference by other pictures of the bitstream. However, these pictures are not output for presentation to the user.

The paper is structured as follows. In Sect. II, the proposed method for compressed domain video compositing is presented while Sect. III delivers system level considerations. Section IV presents experiments to evaluate the RD performance of the proposed method in comparison to a simple transcoding-based approach followed by a conclusion in Sect. V.

II. COMPRESSED DOMAIN VIDEO COMPOSITING

The general outline of the proposed compressed domain video compositing method is as follows: first, pictures of the input videos are merged into a unified output video bitstream

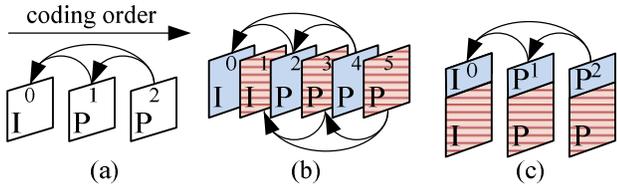


Figure 1: (a) IPP coded input bitstream and merging via (b) temporal multiplex (TM) and (c) spatial multiplex (SM).

through multiplexing in the compressed domain as detailed in subsection II.A. Second, pictures generating a composition of the input videos are inserted into the output bitstream as detailed in subsection II.B.

A. Merging Input Bitstreams

The process of merging n input bitstreams generates a single output bitstream that contains all pictures of the input bitstreams. Bitstream merging is either conducted via temporal multiplex (TM) or spatial multiplex (SM) as explained in more detail in this section. The input bitstream pictures are referred to as source pictures (SPs) in this work and are not intended for display in the original form but for creation of a composition to be output by a standard compliant HEVC decoder. Alignment of coding parameters between input bitstreams is advantageous as the parameter sets have to be merged and values of different picture parameter sets within a coded video sequence are partly constrained, e.g. with respect to picture dimensions. The following assumes that input bitstreams are encoded in a synchronous fashion, i.e. with similar group of pictures (GOP) size and referencing structure.

The two proposed multiplexing approaches are explained given the exemplary low delay IPP coding structure shown in Figure 1 (a) indicating prediction dependencies as solid arrows and denoting the picture order count (POC) in the top of each picture. The first merging approach TM is shown in Figure 1 (b) for n equal to two input bitstreams (solid blue and striped in red). TM requires pictures to have equal picture size. The output bitstream then consists of pictures of the i -th input bitstream alternatingly with $i=[0, n-1]$. POC values of the output bitstream (POC_{out}) are adjusted or stretched with respect to the POC values of the i -th input bitstream (POC_{in}^i) to provide for the increased number of pictures according to

$$POC_{out} = n \cdot POC_{in}^i + i. \quad (1)$$

The RPS of the output bitstream must be adjusted accordingly. For each RPS in the input bitstream, n RPSs have to be signaled in the output bitstream, as the RPS of a picture from a given input bitstream has to include references of the other input bitstreams. As an example, consider the pictures of the output bitstream shown in Figure 1 (b) with POC_{out} equal to 2 and 3. These pictures share the same value of POC_{in}^i equal to 1 and also the same RPS in the two input bitstreams. However, when decoding the picture with POC_{out} equal to 2 of the output bitstream, the decoder has to keep the pictures with POC_{out} equal to 0 and 1 in the decoded picture buffer (DPB) while for decoding of the picture with POC_{out} equal to 3, the DPB must additionally keep the just decoded picture with POC_{out} equal to 2 for use of the following pictures in coding order. Hence, the pictures with a value of POC_{out} equal to 2 and 3 require individual RPS in the output bitstream. The

TM merging approach thereby increases the required space in the DPB and the amount of pictures per second by factor n , which is of relevance for the HEVC level definitions as detailed in Sect. III.

The second merging approach SM avoids the above-mentioned increase of RPS amount and DPB size through compressed domain stitching as illustrated in Figure 1 (c). In bitstream stitching, only a single picture dimension of the input video bitstreams is required to be sized equally along which pictures can be stitched. POC values and RPS can remain unchanged in the merged bitstreams. Slice segment data of the input bitstreams is copied to slices or tiles of the output bitstream, depending on the stitching layout. The required adjustments to high-level syntax are lightweight. For example, slice addresses in the merged slice headers need to be adjusted to the new tile or slice positions within the merged picture, and slice delta quantization parameters (QPs) might need adjustment to reflect a possible change of the initial QP as signaled in the merged parameter set.

Furthermore, to allow for such compressed domain stitching without prediction mismatches between the multiple individual encoders and the single decoder of the merged output bitstream, the encoders have to be constrained. Below is a short summary of the constraints for HEVC coded bitstreams as detailed in [8].

1) *MV Constraints*: MVs should not point to samples outside the picture borders or sub-pel sample positions, for which the encoder-side invoked sub-pel interpolation filter kernel overlaps with the picture borders.

2) *Prediction Units*: The rightmost prediction units within a picture shall not use the MV prediction candidate that corresponds to a temporal motion vector prediction (TMVP) candidate or the spatial MV candidate at the position of a non-existent TMVP candidate.

3) *In-loop filters*: Slice segment and tile borders (if present) shall not be crossed by in-loop filters such as the deblocking and SAO filter.

While POC values and RPS can remain unchanged in the merged bitstream, this approach increases the number of samples per picture, which is relevant for the HEVC level definitions as detailed in Sect. III.

In both approaches, TM and SM, the output flag in slice segment headers of SPs is disabled and therefore SPs are not output. The input bitstream referencing structures present a crucial point of the merging process. When the input bitstreams are not encoded in a synchronous fashion, merging may require heavy adjustments to RPS structures up to the point of infeasibility due to level constraints regarding the DPB size or an incompatible order of coded pictures.

B. Composition Pictures

The second step of the proposed method is the addition of pre-encoded composition pictures (CPs) to the output bitstream. These pictures may be encoded beforehand and added to a merged output bitstream at runtime as detailed in the following. Picture area samples of a set of SPs are copied to the associated CP picture area via block-based inter-prediction.

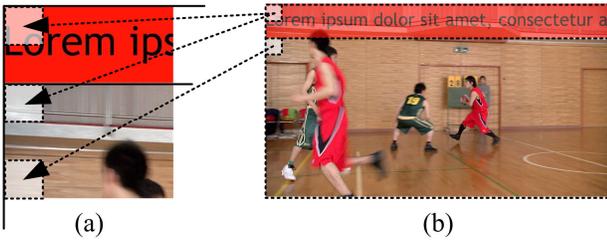


Figure 2: (a) spatially multiplexed SP and (b) associated CP with two slice segments and exemplary MVs.

Figure 2 illustrates the construction of a CP (b) via inter-prediction from a spatially multiplexed SP (a). The exemplary CP consists of a bi-predictive slice segment at the picture top and a uni-predictive slice segment at the bottom. Inter-prediction MVs are depicted as dashed arrows for the first prediction block of each CP slice segment. The depicted setup leads to a transparent overlay of the first input video on top of the second input video. As can be seen from the resulting composition given in Figure 2 (b), CP MVs are invariant throughout the CP slice segment to seamlessly copy the intended SP picture area samples, e.g. samples that are collocated to the given CP sample positions (e.g. as for the top dashed arrow) or samples with an optional invariant spatial offset to the collocated CP sample positions (e.g. as for the two bottom dashed arrows).

An efficient signaling scheme when encoding CP slice segment data in this scenario is to use a large block size, e.g. 64×64 samples, signaling of the desired MV once for the first prediction block of the CP slice segment and skip mode for the coded tree blocks following in coding order. Blending of input bitstream picture samples as illustrated in the top CP slice segment in Figure 2 (b) can be realized via (weighted) bi-prediction from the respective areas of the set of SP. Weighting factors can be signaled in the CP slice segment header and allow for gradual fading over time. Loop filters are disabled for CPs, as it is likely undesired to filter already loop-filtered and subsequently copied SP samples. Apart from a varying number of references or MVs as in the example of Figure 2, other factors may motivate spatial segmentation of the CP. As tiles influence the coding order of coding tree blocks, they can be used to groups together coding tree blocks with invariant MV for the efficient signaling scheme described above.

As shown in Figure 3 for both SP multiplexing approaches, the slice segments of CPs are inserted into the output bitstream after the associated set of SPs in coding order. Therefore, to accommodate the CPs, equation (1) for TM is altered to

$$POC_{out} = (n + 1) \cdot POC_{in}^i + i. \quad (2)$$

For the SM merging approach, POC values of SPs also have to be stretched according to

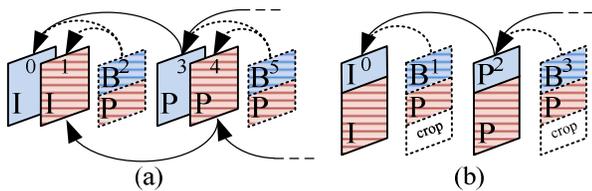


Figure 3: CP insertion for (a) TM and (b) SM.

$$POC_{out} = 2 \cdot POC_{in}^i. \quad (3)$$

The POC value of a given CP can then be derived as $POC_{CP} = POC_{out} + 1$ using the largest value of POC_{out} of the preceding set of SPs in bitstream order.

Note that in HEVC, POC value differences are used for scaling of spatial and temporal MV candidates under certain conditions. The POC value difference td between the picture of the candidate block and its reference picture is used to derive

$$tx = ((2^{14} + |td| \gg 11) / td), \quad (4)$$

which in turn is used together with the POC value difference tb between the current picture and the reference picture of the current block to derive the scaling factor through a clipping operation

$$SF = \text{Clip3}(-4096, 4095, (tb \cdot tx + 32) \gg 6). \quad (5)$$

The product $tb \cdot tx$ in (5) must remain unaffected by POC stretching for MV scaling to function properly. Since tb scales proportionally with POC stretching, e.g. $tb_{out} = tb_{in} \cdot m$ with $m = (n+1)$ following (2) in case of TM, the value of tx after POC stretching is required to fulfill $tx_{out} = tx_{in} / m$ with $\text{mod}(tx_{in}, m) = 0$. A straightforward solution is to constraint tx and m to be equal to powers of two. However, MV scaling is not applied if tb and td are equal or one of the references is a long-term reference. Therefore, instead of the above constraint, the encoder of an input bistream may use the same picture as reference in case of using neighbor MV predictors, disable TVMP for certain pictures or, as worst case, even encode a prediction unit as intra if necessary.

Also additional RPSs are required to signal references of the CPs, namely the one (for SM) or n (for TM) active reference pictures plus all reference pictures signaled in the RPS of the associated set of SPs in coding order. It is likely that picture dimensions of at least one of the input videos are also the desired output picture dimensions, e.g. when adding a user interface overlay. For TM SPs as illustrated in Figure 3 (a), the CP picture dimensions match the input video picture dimensions. However, smaller output picture dimensions may be desired or, as in the SM approach illustrated in Figure 3 (b), unused picture areas of the CP might be required to be hidden from the user. For such cases, the HEVC picture cropping procedure allows output of the desired picture area of CPs only.

III. SYSTEM LEVEL CONSIDERATIONS

Applicability of the proposed method has to be seen in context of the system level as given in subsection III.A as well as random access handling as discussed in subsection III.B.

A. Implementation aspects

From system perspective, carrying out video compositing such as user interface insertion on service operator side, e.g. on cloud resources, instead of the consumer side reduces the requirements for the customer equipment to mere video decoding. In such a system design, transcoding scales poorly and can potentially impair RD performance. The proposed compositing method on the other hand presents a lightweight alternative that can support large-scale service operation without the downsides of transcoding. Since CPs slice segment

data only depends on high-level parameters such as picture size and desired composition, a data set matching the targeted bitstream parameters can be pre-encoded beforehand for subsequent insertion. The remaining workload of the proposed method consists of simple parameter set and slice segment header adjustments for correct reference picture management. However, without many of the time-consuming steps of regular video encoding such as motion estimation or loop filtering [10], CP generation is of comparatively low complexity and thus could even be carried out on-demand at runtime.

The proposed method however increases the luma sample rate and in the case of stitched SPs the luma picture size too, both of which are of relevance for the HEVC level definitions as pointed out before. As these constraints are tailored to serve a handful of common operation points, increased processing demands may lead into subsequent levels. For instance, consider 1280x720@25Hz video encoded with a GOP size of 4 and hierarchical bi-prediction, which corresponds to a level 3.1 bitstream. Using the proposed method to overlay the latter with a 1280x128@25Hz video, leads to an output bitstream with level 4 for SM SPs an even level 4.1 for TM SPs. Albeit operation points exists for which no level increase occurs, a way of mitigation is input frame rate reduction, i.e. the resulting luma sample rate, of given input bitstreams if applicable, e.g. overlay video.

B. Random Access Handling

The concept of random access, i.e. decoding from specific points mid-bitstream onwards, is specified in HEVC as intra random access points (IRAPs) and extends the respective functionality of H.264/AVC. So-called leading pictures may follow an IRAP picture in coding order, but precede it in output order, i.e. leading pictures have a lower POC value than the associated IRAP pictures, as opposed to the so-called trailing pictures, that follow the associated IRAP picture in coding and output order. Two important restrictions are given: the leading pictures must precede all trailing picture of the associated IRAP picture in bitstream order and should not be referenced by trailing pictures.

For the SM merging approach, these two restrictions of leading pictures do not pose a problem. For the TM merging approach, however, these two restrictions cannot be fulfilled simultaneously. For example, let $IRAP_i$ be the IRAP picture of the i -th of two input bitstreams with equal POC_{in}^i and leading pictures. If $IRAP_1$ is chosen to serve as an IRAP picture of the output bitstream and $IRAP_2$ is converted into a non-IRAP picture, $IRAP_2$ can neither be a trailing picture as leading pictures follow, nor can it be a leading picture as following trailing pictures use it as reference. If $IRAP_2$ is chosen to serve as an IRAP picture of the output bitstream, the same problem applies to $IRAP_1$.

Therefore, when IRAP pictures have associated leading pictures, the adequate procedure for TM is as follows. In a point-to-point per-user scenario, TM requires conversion of IRAP pictures and associated leading pictures to trailing pictures. Thereby, this procedure removes full random access functionality but maintains error resiliency. The user tune-in or access point is known, for which random access can be provided by using the current $IRAP_l$ as output bitstream IRAP

picture, converting $IRAP_i$ with $i > 1$ to trailing pictures, removal of all associated leading picture and converting all future IRAPs and leading pictures to trailing pictures. However, in a random access broadcast scenario with leading pictures, the described procedure requires respective signaling to and the ability for such bitstream operations at receiver side.

The same restrictions determine that CPs using an IRAP picture with associated leading pictures as reference are required to be leading pictures and precede the IRAP picture in output order. Otherwise, CPs can be added as trailing pictures.

IV. EXPERIMENTS

The first experiment was carried out to measure the BD-rate overhead [11] of the stitching constraints [8] when applied only to a single picture border (top) compared to unconstrained encoding. Such constrained bitstreams can subsequently be stitched to bitstreams that are likewise constrained on the opposite picture border (bottom). Unconstrained encoding was carried out with HM 14.0 [12] while a modified version was used for constrained encodings. Experiments were carried out following [13] using the “Low-delay P – Main” (LD) and “Random access – Main” (RA) configuration with four QPs (22, 27, 32, 37) and the category B test sequences with a resolution of 1920x1080 pixels and a duration of 10 seconds at varying frame rates from 24fps to 60fps, referred to as content video in the following. However, to prevent MV scaling mismatches as discussed in subsection II.B, the configurations were adjusted to use reference pictures for which the POC difference td is a power of two. Table I reports the BD-rate overhead of constrained encodings with respect to unconstrained encoding for the individual sequences. An average BD-rate overhead of encoder constraints on the top picture border of 0.96% for LD and 1.13% for RA is reported.

The second experiment evaluates the proposed method based on the composition depicted in Figure 2 (b) against a transcoding-based approach. The constrained and unconstrained encodings of content videos created in the previous experiment were used and a ticker overlay video with a resolution of 1920x192 pixels for the SM approach with flat red background and moving black text was encoded with encoder constraints on the bottom picture border. For the TM approach, the overlay video was extended with a flat red video to match the content video resolution of 1920x1080 pixels and subsequently encoded without constraints. RD-performance of overlay encodings is evaluated by measuring PSNR over the envisioned output picture area of 1920x192 pixels containing the text. While the constrained overlay encodings achieve BD-rate improvements of -9.74% for LD and -4.31% for RA over

TABLE I. BD-RATE OVERHEAD OF CONSTRAINED CONTENT VIDEO ENCODINGS.

Sequence		BD-rate	
Name	Frame rate	LD	RA
BQTerrace	60fps	1.55%	1.60%
BasketballDrive	50fps	0.50%	0.81%
Cactus	50fps	0.96%	0.97%
Kimono	24fps	0.90%	1.04%
ParkScene	24fps	0.87%	1.22%
Average		0.96%	1.13%

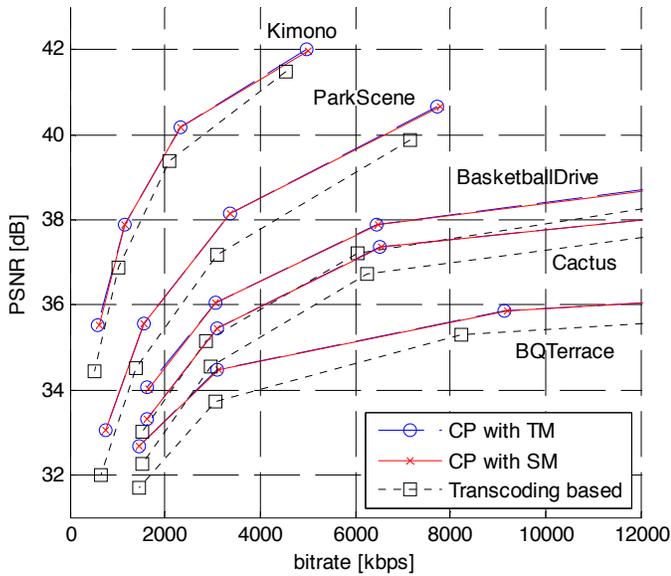


Figure 4: RD-curves of compositing by proposed method (TM and SM) and transcoding based approach, all with RA.

the unconstrained encodings in this evaluation, the bitrates of the overlay videos are in general several orders of magnitude smaller than the content video bitrates and are therefore insignificant.

For all PSNR measurements of compositions in this experiment, a coding noise free composition is created via pixel-wise blending of the uncompressed content and overlay videos. For evaluating the proposed method, CPs with two slice segments were added to the respective TM and SM bitstreams. The top slice of the CP bi-predicts from the content and overlay videos and the bottom slice uni-predictively mirrors the remaining picture area of the content video as indicated in Figure 2 (b). The transcoding-based approach for pixel domain video compositing is simulated by decoding each unconstrained content video bitstream, pixel-domain mixing with an uncompressed overlay video and subsequent unconstrained re-encoding at the same QP as the initial coded content video with the genuine configurations of [13].

Figure 4 reports a section of RD-curves for the RA encoded composition bitstreams using the proposed method with TM and SM as well as the transcoding-based approach for pixel-domain compositing. It can be seen from Figure 4 that for any content video bitstream, the proposed method achieves a higher quality in terms of PSNR with an average quality benefit of 1.11dB PSNR for TM and 1.10dB PSNR for SM over the

TABLE II. BD-RATE SAVINGS OF THE PROPOSED METHOD RELATIVE TO TRANSCODING BASED COMPOSITING.

Sequence name	BD-rate			
	LD		RA	
	TM	SM	TM	SM
BQTerrace	-34.64%	-32.19%	-29.28%	-28.33%
BasketballDrive	-15.99%	-15.65%	-21.22%	-20.73%
Cactus	-17.70%	-17.05%	-21.77%	-21.20%
Kimono	-11.03%	-10.50%	-14.46%	-13.84%
ParkScene	-18.74%	-18.32%	-18.79%	-18.12%
Average	-19.62%	-18.74%	-21.10%	-20.44%

transcoding-based approach in the LD configuration and 0.83dB PSNR for TM and 0.82dB PSNR for SM in the RA configuration. BD-rate measurements as reported in detail in Table II further indicate a better RD-performance of the proposed method with average BD-rate improvements of around -20% across all configurations over the transcoding-based approach and up to around -30% for higher input frame rates.

V. CONCLUSION

This paper presents a novel method for compressed domain video compositing in HEVC for which input video bitstreams are merged via temporal or spatial multiplexing and additional pre-encoded composition pictures are added to form a composition via inter-prediction. Method details and system aspects were discussed. Experiments show that for the given test sequences, the BD-rate overhead of encoder constraints for stitching HEVC bitstreams applied to a single picture border (top) are around 1%. Experiments evaluating the proposed method in a ticker overlay scenario with respect to a transcoding-based compositing approach show on average a BD-rate reduction of around -20% while archiving PSNR gains of around 1dB.

REFERENCES

- [1] Mikityuk, Alexandra, Jean-Pierre Seifert, and Oliver Friedrich. "Paradigm shift in IPTV service generation: Comparison between locally- and Cloud-rendered IPTV UI." *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, 2014.
- [2] Chang, Shih-Fu, and David G. Messerschmitt. "Manipulation and compositing of MC-DCT compressed video." *Selected Areas in Communications, IEEE Journal on* 13.1 (1995): 1-11.
- [3] Vetro, Anthony, Charilaos Christopoulos, and Huifang Sun. "Video transcoding architectures and techniques: an overview." *Signal Processing Magazine, IEEE* 20.2 (2003): 18-29.
- [4] Sullivan, G. J., Ohm, J., Han, W. J., & Wiegand, T. "Overview of the high efficiency video coding (HEVC) standard." *Circuits and Systems for Video Technology, IEEE Transactions on* 22.12 (2012): 1649-1668.
- [5] Chi, Chi Ching, et al. "Parallel scalability and efficiency of HEVC parallelization approaches." *Circuits and Systems for Video Technology, IEEE Transactions on* 22.12 (2012): 1827-1838.
- [6] Amon, Peter, et al. "Compressed domain stitching of HEVC bitstreams for video conferencing applications." *Packet Video Workshop (PV), 2012 19th International*. IEEE, 2012.
- [7] Feldmann, Christian, et al. "Efficient bitstream-Reassembling for Video Conferencing Applications using Tiles in HEVC." *MMEDIA 2013, The Fifth International Conferences on Advances in Multimedia*. 2013.
- [8] Sanchez, Yago, et al. "Low complexity cloud-video-mixing using HEVC." *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, 2014.
- [9] Sjoberg, Rickard, et al. "Overview of HEVC high-level syntax and reference picture management." *Circuits and Systems for Video Technology, IEEE Transactions on* 22.12 (2012): 1858-1870.
- [10] Bossen, Frank, et al. "HEVC complexity and implementation analysis." *Circuits and Systems for Video Technology, IEEE Transactions on* 22.12 (2012): 1685-1696.
- [11] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *Proc. VCEG-M33 Meeting*, 2001 pp.1-4
- [12] ITU-T and ISO/IEC JTC 1, Reference Software for High Efficiency Video Coding, *ITU-T Rec. H.265.2 and ISO/IEC FDIS 23008-5*, 2014
- [13] Bossen, Frank. "Common test conditions and software reference configurations." *m28412*, Joint Collaborative Team on Video Coding of ISO/IEC and ITU-T, *JCTVC-L1100*, 12th Meeting: Geneva, CH, 14-23 January 2013