APPLICATION OF LARGE MACROBLOCKS IN H.264/AVC TO WAVELET-BASED SCALABLE VIDEO TRANSCODING

Eduardo Peixoto, Toni Zgaljic and Ebroul Izquierdo

School of Engineering and Computer Science, Queen Mary, University of London
Mile End Road, E1 4NS, London, UK
phone: + (44) 20 7882 5354 , fax: + (44) 20 7882 7997, eduardopeixoto@ieee.org

ABSTRACT

In this paper an efficient transcoder from H.264/AVC to a wavelet-based scalable video (W-SVC) codec is proposed. It exploits the advantage of using large sizes of prediction blocks in the W-SVC codec, and it is flexible in the sense that it is able to cope with any prediction structure in H.264/AVC stream, such as *IPP* or *IBBP* configuration with multiple reference frames. The reference frame mismatch between the source and target codecs is solved by a novel framework for motion vector approximation and refinement. The transcoder performance benefits from the use of block sizes larger than 16×16, especially for higher resolution content. Experimental results show a very good performance in terms of decoded video quality and system complexity.

1. INTRODUCTION

In scalable video coding (SVC), the extraction of a lower resolution, frame-rate and/or quality is possible by simple parsing of the compressed bitstream, therefore allowing an efficient real-time adaptation of the video content. However, video stored on the server is often encoded using conventional, non-scalable, codecs, such as H.264/AVC [1]. In this case, an efficient low-complexity video adaptation cannot be achieved. To address this issue, a non-scalable bitstream can be converted into a scalable stream by transcoding. Thus, transcoding would be required only once and, afterwards, the transcoded video could be adapted many times.

The most straightforward approach to transcode between a source and a target formats is to cascade the required decoder and the target encoder, known as the cascaded pixel-domain transcoder [2, 3]. This results in high quality of the transcoded video, however it introduces a high transcoding complexity. To reduce complexity, one could use a fast motion estimation (ME) algorithm, instead of a exhaustive (full) ME. This significantly reduces complexity, but it may have a negative impact on the transcoded video quality. Another approach is to process the motion information gathered from the source stream by the target encoder, avoiding a costly ME and resulting in low complexity transcoding. The transcoder proposed here is based on the latter approach. The former approach, when no intermediate processing is performed, is denoted here as the trivial transcoder.

Transcoding between hybrid-based video coding structures has been extensively studied before [2, 3], even targeting hybrid-based scalable codecs [4]. However, to the best of authors' knowledge, in existing approaches, the macroblock (MB) size in the target encoder is always set to 16×16 pixels (or smaller). This is a reasonable approach, since all standard video codecs support MB sizes of up to 16×16 pixels. Recently, it has been shown that using larger MB sizes can significantly improve the compression [5]. In fact, large MB sizes (of up to 64×64 pixels) form the basis of the emerging High-Efficiency Video Coding (HEVC) standard [6].

The aim of this work is to develop a transcoder from H.264/AVC to a scalable format, supporting MB sizes larger than 16×16. Although a hybrid based technology was chosen for standardization of SVC within MPEG [7], it is not suitable for the proposed transcoding framework since its MB size is limited to 16×16 pixels. Furthermore, several recent W-SVC systems have shown a very good performance in different types of application scenario [8, 9, 10], while still being able to deliver some attractive features not supported by the standard, such as Fine Grain Scalability (FGS). The employed target codec [8, 11], denoted as W-SVC, supports quality (with FGS), spatial and temporal scalability and any of their combinations. Its main features are: hierarchical variable size block matching motion estimation (with MB size of up to 128×128 pixels), flexible selection of filters for both spatial and temporal wavelet transforms at each level of spatio-temporal decomposition, user-defined flexible decomposition path, support for conventional frame-based coding and object-based coding, bit-plane coding based on Embedded Zero Block Coding (EZBC), binary arithmetic coding and low-complexity post compression rate-distortion optimization for bit-stream allocation [12].

In this paper, we build on our previous work [13, 14]. The main improvement is the possibility to use larger MB sizes in the W-SVC codec. Since motion information coming from the source stream is available only for 16×16 MBs, motion vectors (MVs) for larger block sizes always need to be approximated. Thus, the techniques for intermediate motion information processing have been modified to support these cases. The proposed transcoder is able to cope with different H.264/AVC coding structures, such as *IPP* and *IBBP*, while maintaining high performance in both cases. The proposed transcoder is generic in the sense that it can be used with most of the well-known wavelet-based coding architectures [9, 10]. Furthermore, the techniques for MV approximation and refinement can be used with any cascaded pixel domain transcoder, regardless if they are hybrid or wavelet codecs. Therefore, the transcoder is rather based on an open and easily adaptable model since the key strategies proposed for the adaptation of motion information are independent from the W-SVC codec and its underpinning algorithms.

2. THE W-SVC CODEC

In the W-SVC codec, the ME process is not performed together with motion compensation, as it is the case in the H.264/AVC and other DPCM/DCT codecs. Instead, MVs are

This research was partially supported by the European Commission under contract FP7-247688 3DLife and FP7-248474 SARACEN.

used in the temporal filtering process, resulting in so-called Motion Compensated Temporal Filtering (MCTF). The aim is to reduce the correlation between consecutive frames and provide the basis for temporal scalability, without drift errors. In the W-SVC codec, several wavelet filters can be used in the temporal filtering step, but in this paper only the $5/3$ is used. Since the filtering is dyadic, and also to allow for temporal scalability, two reference frames (RFs) are chosen in each decomposition stage (one previous and one future frame). This is depicted in Fig. 1, which shows a 3 level dyadic temporal decomposition of 9 frames of the input video. The markings $\mathbf{L_x}$ and $\mathbf{H_x}$ in the figure represent low-pass and high-pass temporal sub-bands, respectively, at the x-th decomposition scale, and the arrows point from the frames being predicted to the RFs in the process of MCTF.

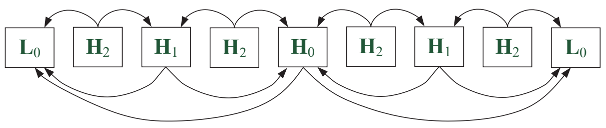

Figure 1: Hierarchical structure with 3 levels.

From the W-SVC RF structure on Fig. 1, it is clear that, depending on the H.264/AVC coding configuration, some H.264/AVC MVs may not be directly reused in the transcoder, due to a reference frame mismatch. Only those MVs that point to the same RF can be directly reused, the others need to be approximated first.

In the W-SVC ME, the MB size itself and the maximum number of partitioning levels allowed (i.e., the depth of the quadtree) are encoder parameters. The MB is partitioned in a recursive way: if the MB size is $n \times n$ pixels, and up to k partitioning levels are allowed, then at the first level the MB will have 4 partitions of $\frac{n}{2} \times \frac{n}{2}$ pixels. Each of these blocks can be further partitioned (provided $k > 1$), independently. The minimum block size will be $\frac{n}{2^k} \times \frac{n}{2^k}$. Allowing larger block sizes results in a better overall performance, even for small resolutions. Fig. 2 shows the results for City sequence, using 4*CIF* resolution and different block sizes. Therefore, allowing larger partitions to be used in the transcoder can raise its performance, reducing the transcoder loss.

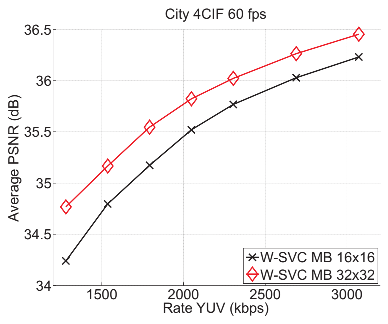

Figure 2: W-SVC performance using different MB sizes.

3. THE TRANSCODER

The transcoder decisions are based on the W-SVC MB, which can be set as $n \times n$. In this paper, we use $n \in \{16, 32, 64\}$. In order to maximize the use of the H.264/AVC MVs, the minimum size is chosen as 4×4 for any MB size. For each MB, the transcoder attempts to reuse the H.264/AVC partitioning scheme to avoid testing partitions that are unlikely to be chosen by the W-SVC rate-distortion optimization. Here, we define the testing of a partition as the approximation (or reuse) and refinement of MVs for each direction (forward and backward) and the mode selection (forward, backward or bidirectional prediction). When testing a given partition, the transcoder attempts to directly reuse the H.264/AVC MVs (or, if that is not possible, use them to approximate new MVs) in the W-SVC codec. This way, complexity is reduced by testing less partitions than the trivial transcoder, and also by using a better starting point in the fast ME for each partition, utilised in the refinement step.

3.1 Selecting Partitions for Testing

In the H.264/AVC codec, the MB partitions and MVs are optimized to a single rate-distortion target. However, the W-SVC codec performs encoding tailored to a wide range of bitrates and spatio-temporal resolutions, and a single set of MVs is used for all decoding points. This results in a favouring of larger partitions in the W-SVC codec if compared to H.264/AVC [13]. In the transcoder, if the MB was encoded in inter mode in H.264/AVC, then only those partitions of the same or larger size than that of the H.264/AVC will be tested, regardless whether MVs can be directly reused. An example of the partitioning decision when using a W-SVC MB size of 32×32 in the transcoder is shown in Fig. 3.

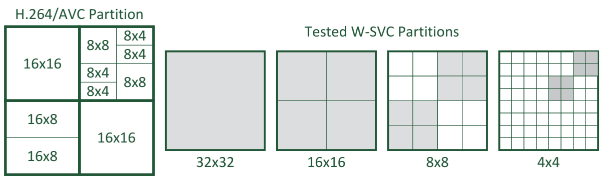

Figure 3: Example of partitioning decision with a MB size of 32×32. The shaded partitions will be tested.

3.2 Framework for MV Approximation and Refinement

For each partition tested, the proposed transcoder produces two MV candidate lists, one for each direction (forward and backward). For each candidate in the two lists the cost is computed in a conventional way, considering the residual and the rate of the chosen MV. The best candidate is selected for each direction and then a further refinement step can be applied. In all cases, the refinement considered is a hexagon search [15], starting at the best candidate for each direction. When testing the W-SVC partition, all H.264/AVC partitions overlapping with the same frame area bounded by the tested partition are observed. The MVs corresponding to these observed partitions that can be directly re-used are added to the appropriate candidate list. Otherwise, several strategies are used to populate the candidate list(s): spatial prediction, scaling, inversion and composition.

3.2.1 Spatial Approximation

The approximated MV is formed as the weighted average of MVs of blocks above and on the left of the currently observed block. The weights are defined according to the sizes of its corresponding blocks relative to the current block. The advantage of this method is that it uses MVs already refined by the W-SVC.

3.2.2 *MV Scaling*

When none H.264/AVC MV can be reused due to the RF mismatch, scaling is applied to produce a candidate for the appropriate direction. The H.264/AVC MV that correspond to the largest area within the tested W-SVC partition is scaled and added to the candidate list. If there is more than one MV in this situation, all of them are used. The scale factor is directly proportional to the distance between the H.264/AVC and the W-SVC RFs to the current frame.

3.2.3 *MV Inversion*

This method is used to create backward MV candidates from already found forward MV candidates and vice versa. All reused MVs and all candidates generated by the other methods are inverted and added to the opposite-direction candidate list. This method can be useful for instance in *IPP* coding configuration when only forward MVs are available.

3.2.4 *MV Composition*

For simplicity, a similar notation as found in Lee et. al. is used [16]. Let B_n^k represent the block at the position k in frame n. The MV for a particular partition B_n^k, which uses RF $n-\alpha$, is similarly denoted as $mv_{n\to n-\alpha}^k$. The aim of MV composition is to compose a MV for a W-SVC partition B_n^k, between the current frame n and the RF $n-\beta$, considering MVs available from the decoded H.264/AVC stream. The basis of this method was presented in our previous work [14], but it has been slightly altered here.

The proposed MV Composition method was designed specifically to cope with different coding configurations, with multiple reference frames, multiple block sizes and bidirectional motion estimation. It is based on the popular FDVS [17] and TVC [18] MV Composition methods. In our method, MV Composition can be performed in two phases.

In the first phase, the aim is to compose a MV from the current frame n to the target RF $n-\beta$. The algorithm is based on an ordered MV list, which starts empty. At each step of composition, the list is refreshed. In the first step, all H.264/AVC MVs within the W-SVC partition $P_0 = B_n^k$ are considered. These MVs are grouped according to their RFs. Then, for each group, the algorithm takes the MV corresponding to the largest area (as opposed to the average MV [14]). If, within the same group, more than one MV correspond to the largest area, a simple average is used between those MVs. The MVs with RFs between n and $n-\beta$ are then added to the list, ordered such that the elements towards the end of the list have RFs that are closer to $n-\beta$.

In the next step, the algorithm pops the last MV on the list and adds it into the current composition group *CG*. The composition group is the group of MVs that are effectively used to compose the MV from frame n to frame $n-\beta$, and it also starts empty. The new position of the partition in the next frame is calculated as $P_i = P_0 + \sum_{i\in CG} MV_i$. To continue the composition, the position P_1 is aligned to the grid. Since the minimum block size for a H.264/AVC MV is 4×4, the position is always aligned in 4×4 coordinates, no matter the size of the partition for which Composition is being performed (as opposed to aligning to the same partition size [14]). Then, the algorithm repeats itself, until the target RF $n-\beta$ is reached. If, at some intermediate step, the MVs cannot be used (either because they point outside the desired RF range, or the block has been coded in intra mode), the algorithm will remove all MVs from *CG* that were added in any subsequent step to the one in which the currently last element of the ordered list has been added. Then it will pop another MV from the ordered list, using it in the same way as before. The algorithm continues in such a manner until the reference frame $n-\beta$ has been reached or until the list is empty.

The second phase is executed only if the composition is unsuccessful in the first phase. In the first phase, the algorithm keeps a record of composed MVs for frames beyond the target RF. It takes the composed MV for the frame closest to the targeted RF $mv_{n\to n-\beta-\gamma}^k$. Then, it tries to compose a MV from frame $n-\beta$ to the frame $n-\beta-\gamma$. This second composition is the same as in the first phase, except that it uses TVC instead of FDVS (thus, the considered position remains the same throughout all considered reference frames, $P_i = P_0$). The result for the second phase is MV $mv_{n-\beta\to n-\beta-\gamma}^k$. The final composed MV is then computed as: $mv_{n\to n-\beta}^k = mv_{n\to n-\beta-\gamma}^k - mv_{n-\beta\to n-\beta-\gamma}^k$.

Finally, the last modification of the previous algorithm [14] regards how these methods are applied to find a forward and backward candidates. These are depicted in Fig. 4.

For the forward candidate, the algorithm is directly applied to the forward H.264/AVC MV to generate a candidate for the forward direction. The example in Fig 4(a) depicts a case when the first phase fails, and the second phase has to be applied. Thus, the final candidate MV is given as:

$$mv_{n\to n-\beta}^k = mv_{n\to n-2}^k + mv_{n-2\to n-\beta-y}^j - mv_{n-\beta\to n-\beta-y}^k$$

For the backward case, we are now considering the popular H.264/AVC coding configuration when *B*-frames are not used as references during encoding. Thus, if a backward MV is available for the current *B*-frame, this MV will necessarily point to a *P*-frame, which will not have a backward MV, and thus the composition will fail. In Fig. 4(b), this is depicted as $mv_{n\to n+1}^k$. For this reason, even if a backward MV is present, the algorithm will use the same procedure as it is used when no backward MV is present (as in *IPP* configuration). It will try to compose a MV from frame $n+\beta$ to frame n (also in two phases, if necessary), and then it will invert the result to use as the backward candidate. Thus, the final candidate for the backward direction in the example is given as:

$$mv_{n\to n+\beta}^k = (-1)\cdot\left(mv_{n+\beta\to n+1}^k + mv_{n+1\to n-x}^k - mv_{n\to n-x}^k\right)$$

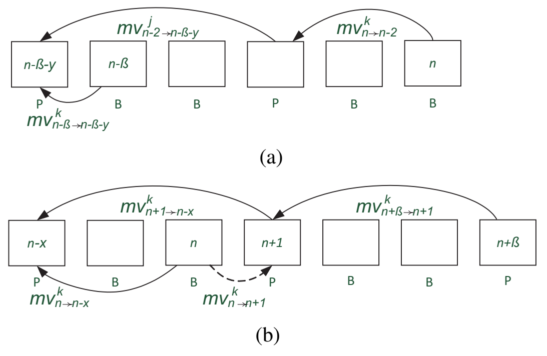

Figure 4: Example of MV Composition for: (a) forward direction; and (b) backward direction.

3.3 The Reduced Complexity Module

The goal of the techniques described so far is to achieve maximum quality (PSNR), while reducing the transcoder's complexity. The Reduced Complexity (RC) Module allows for a further reduction in complexity, at the cost of a small impact on quality. Two techniques are used in the RC Module: using the H.264/AVC CBP and the similarity of MVs.

3.3.1 Using the H.264/AVC CBP

If the H.264/AVC MVs for a partition can be reused, the refinement for this partition may be avoided based on the presence of residual data [13]. If there is no residue (here, if the syntax parameter coded block pattern (CBP) is zero), and if the entire partition uses the same H.264/AVC MV, this MV and mode are used directly for this partition. Other partitions may also be tested even when this happens, and then the usual approach of approximation and refinement is used.

3.3.2 MV Similarity

The transcoder may also avoid testing some partitions based on the similarity of the H.264/AVC MVs. Since the W-SVC favours larger partitions [13], if H.264/AVC MVs are considered similar, then smaller partitions are not tested in the transcoder. MVs are considered similar if the difference for both components between each MV compared to the mean MV is below 0.5, in integer-pixel scale. However, when the distance between the current frame and the RF becomes too large, the correlation between the H.264/AVC and the W-SVC motion information drops. Thus, this strategy is only used if this distance is up to 4 frames.

4. EXPERIMENTAL RESULTS

For the *IPP* structure at *CIF* resolution, the first 300 frames of the original sequence were used. At the 4*CIF* and 1280×704 resolutions, the first 600 and 598 frames were used, for the *IPP* and *IBBP* structures, respectively. The PSNR shown is the average among luminance frames, and it always refers to the original sequence as the reference, while the rate considers also the chrominance components.

The proposed transcoder is compared to two reference trivial transcoders: (i) using full ME and (ii) using Hexagon Search [15], referred as RT-FS and RT-HS, respectively. The two options of the proposed transcoder are referred as PT and PT-RC, where the former refers to the use of the techniques from Sec. 3 through Sec. 3.2.4 and the latter additionally uses the RC Module (Sec. 3.3). The search window used is 32 for RT-FS, and 60 for all other cases.

Table 1 presents both the results for PSNR and complexity. Selected results are shown in Fig. 5. The complexity is measured as the number of SAD Calculations. This has proven to be a good indication of the complexity, with the advantage that it is more easily reproducible [14]. The complexity of RT-FS is usually very high, so RT-HS provides a more viable option for low complexity transcoding.

It can be seen that both the PT and PT-RC have a very close performance to RT-FS. In the worst case, the average loss is 0.03 dB and 0.12, for PT and PT-RC, respectively, and in the best case it outperforms RT-FS by 0.21 and 0.20 dB, for PT and PT-RC, respectively. The transcoder can outperform RT-FS because it uses a larger search window, and it tests different partitions. The trivial transcoder tests first the partitions of size $n \times n$ and $\frac{n}{2} \times \frac{n}{2}$ and, only if the latter has a lower cost, it tests the next level.

The PT is generally less complex than RT-HS, by up to 50%, except when using 64×64 MBs. The RC Module has a very small impact on quality, of -0.14 dB in the worst case, but it reduces the complexity by up to 38%. With the exception of Crew 4*CIF IBBP* sequence using 64×64 MBs, PT-RC is always less complex than RT-HS.

Fig 6 shows the transcoder results with different MB sizes, using the MB size of 16×16 as basis for comparison. It can be seen that even for the small *CIF* resolution, using larger MB sizes yields a gain of up to 0.4 dB. As expected, for the 4*CIF* and 1280×704 resolutions, the gain is even higher, of up to 1.25 dB and 1.4 dB, respectively. For both resolutions, most of the gain is already present when using the 32×32 MB, however a small gain (up to 0.25 dB) can still be achieved using MB size of 64×64).

5. CONCLUSION

A flexible and efficient transcoder from H.264/AVC to a Wavelet-Based SVC codec was presented. The proposed transcoder offers a similar performance to that of the trivial transcoder using full motion estimation, but it keeps the complexity up to 40% lower than that of the trivial transcoder using a fast search algorithm. The use of larger MB sizes allows the transcoder to significantly reduce the transcoding loss, by up to 1.4 dB, while still keeping the complexity low.

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", in IEEE Trans. on Circuits and Systems for Video Technology, vol.13, pp. 560-576, Jul. 2003.

[2] A. Vetro, C. Christopoulos and H. Sun, "Video Transcoding Architectures and Techniques: An Overview", in IEEE Signal Proc. Mag., vol.20, Mar. 2003.

[3] J. Xin, C.-W. Lin and M.-T. Sun, "Digital Video Transcoding", in Proceedings of the IEEE, vol.93, pp. 84-97, Jan. 2005.

[4] J.D. Cock, S. Notebaert, P. Lambert and R.V. de Walle, "Architectures for Fast Transcoding of H.264/AVC to Quality-Scalable SVC Streams", in IEEE Trans. on Multimedia, vol. 11, n. 7, pp 1209-1224, Nov. 2009.

[5] D. Marpe, H. Schwarz, S. Bobe, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Shring, M. Winken, and T. Wiegand, "Video Compression Using Nested Quadtree Structures, Leaf Merging and Improved Techniques for Motion Representation and Entropy Coding", in IEEE Trans. on Circ. and Syst. for Video Tech., vol.20, Dec. 2010.

[6] JCT-VC, "WD1: Working Draft 1 of High-Efficiency Video Coding", JCTVC-C403, Oct. 2010.

[7] Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, V.8, Nov. 2007.

[8] N. Sprljan, M. Mrak, T. Zgaljic and E. Izquierdo, "Software proposal for Wavelet Video Coding Exploration group", in ISO/IEC JTC1/SC29/WG11/MPEG2005, M12941, 75-th MPEG Meeting, Jan. 2006.

[9] S. Issa and O. Khalifa, "Performance analysis of dirac video codec with H.264/AVC", in Computer and Communication Engineering (ICCCE), 2010 Int. Conf. on, pp. 1-6, May. 2010.

[10] A. Naman and D. Taubman, "Predictor selection using quantization intervals in jpeg2000-based scalable interactive video (JSIV)", in Image Processing (ICIP), 2010 17th IEEE International Conference on, pp. 2897-2900, Sep. 2010.

[11] N. Sprljan "A flexible scalable video coding framework with adaptive spatio-temporal decompositions", Ph.D. dissertation, Queen Mary, University of London, London, August 2006.

[12] T. Zgaljic, "Entropy coding schemes for scalable video coding supporting flexible bit-stream organisation and adaptation", Ph.D. dissertation, Queen Mary, University of London, London, June 2008.

[13] E. Peixoto, T. Zgaljic and E. Izquierdo, "Transcoding from H.264/AVC to a Wavelet-Based Scalable Video Codec", in Proc. Int. Conf. on Image Proc. (ICIP), pp. 2845 - 2848, Sept. 2010.

[14] E. Peixoto, T. Zgaljic and E. Izquierdo, "H.264/AVC to wavelet-based scalable video transcoding supporting multiple coding configurations", in Picture Coding Symposium, 2010 (PCS 2010), Dec. 2010.

[15] C. Zhu, X. Lin, L. P. Chau, K. P. Lim, H. A. Ang and C. Y. Ong "A novel hexagon-based search algorithm for fast block motion estimation", in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., pp. 1593-1596, Jun. 2001.

[16] T.-K. Lee, C.-H. Fu, Y.-L. Chan, and W.-C. Siu "A new motion vector composition algorithm for fast-forward video playback in H.264", in Circuits and Systems (ISCAS), Proc. of 2010 IEEE Int. Symp. on, June 2010, pp. 36493652.

[17] J. Youn and M. Sun, "Motion vector refinement for high-performance transcoding", in IEEE Trans. on Multimedia, vol.1, pp. 30-40, Mar. 1999.

[18] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats", in IEEE Trans. on Multimedia, vol.2, pp. 101-110, Jun. 2000.

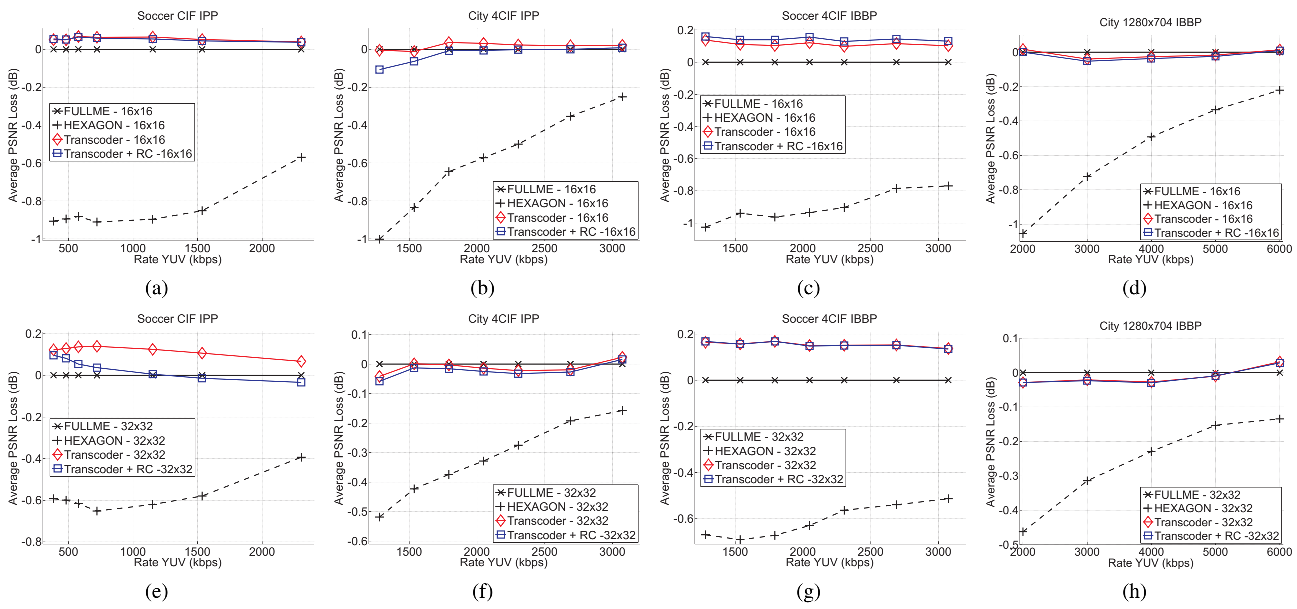

Figure 5: Transcoder results using MB 16×16 for: (a) Soccer *CIF IPP*; (b) City 4*CIF IPP*; (c) Soccer 4*CIF IBBP*; (d) City 1280×704 *IBBP*; and using MB 32×32 for: (e) Soccer *CIF IPP*; (f) City 4*CIF IPP*; and (g) Soccer 4*CIF IBBP*; and (h) City 1280×704 *IBBP*.

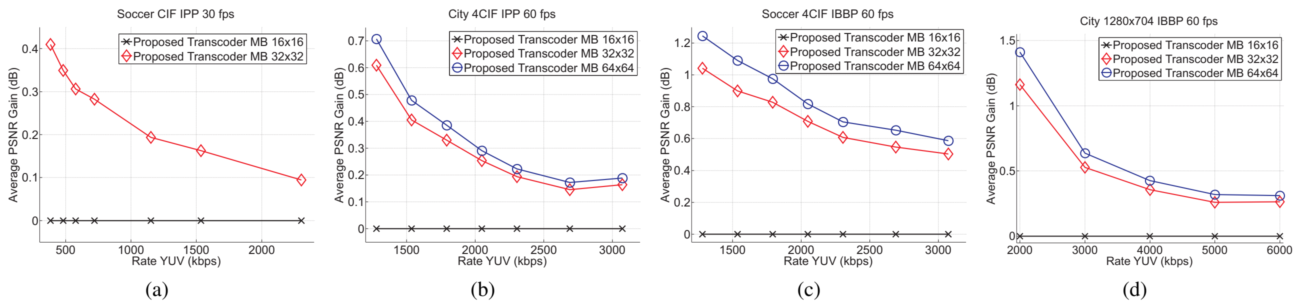

Figure 6: Results using different MB sizes for: (a) Soccer *CIF IPP*; (b) City 4*CIF IPP*; (c) Soccer 4*CIF IBBP*; and (d) City 1280×704 *IBBP*.

Table 1: Transcoder results. The average PSNR Loss is compared to RT-FS, and the complexity is compared to RT-HS. The bitrates used for *CIF*, 4*CIF* and 1280×704 resolutions were: $\{384, 480, 576, 720, 1152, 1536, 2304\}$ kbps, $\{1280, 1536, 1792, 2048, 2304, 2688, 3072\}$ kbps and $\{2000, 3000, 4000, 5000, 6000\}$ kbps, respectively.

		Average PSNR Loss (dB)			SAD Calculations		
MB Size		16×16	32×32	64×64	16×16	32×32	64×64
City *CIF IPP*	RT-HS	-0.69	-0.40	N/A	100.00%	100.00%	N/A
	PT	0.01	0.04	N/A	58.92%	89.14%	N/A
	PT-RC	-0.02	-0.10	N/A	41.64%	55.66%	N/A
Soccer *CIF IPP*	RT-HS	-0.85	-0.58	N/A	100.00%	100.00%	N/A
	PT	0.06	0.12	N/A	71.29%	96.75%	N/A
	PT-RC	0.05	0.03	N/A	54.12%	70.78%	N/A
City 4*CIF IPP*	RT-HS	-0.59	-0.32	-0.19	100.00%	100.00%	100.00%
	PT	0.02	-0.01	0.03	52.85%	87.24%	121.21%
	PT-RC	-0.03	-0.02	0.00	37.50%	55.25%	78.45%
Soccer 4*CIF IBBP*	RT-HS	-0.90	-0.61	-0.43	100.00%	100.00%	100.00%
	PT	0.11	0.15	0.21	56.93%	89.13%	112.99%
	PT-RC	0.14	0.15	0.20	53.14%	75.14%	98.07%
Crew 4*CIF IBBP*	RT-HS	-0.73	-0.52	-0.35	100.00%	100.00%	100.00%
	PT	0.01	0.00	0.02	65.12%	97.67%	119.48%
	PT-RC	-0.01	0.00	0.02	62.20%	90.66%	113.13%
City 1280×704 *IBBP*	RT-HS	-0.56	-0.26	-0.17	100.00%	100.00%	100.00%
	PT	-0.01	-0.01	0.02	49.13%	82.33%	117.58%
	PT-RC	-0.02	-0.01	0.02	45.98%	64.94%	95.71%