Image Coding Using Variable-Size Transforms and a Full Binary Tree Recursive Optimization

Anderson V. C. de Oliveira[1], *Nuno M. M. Rodrigues*[2,3], *Sérgio M. M. de Faria*[2,3]
Eduardo A. B. da Silva[4] *and Carla L. Pagliari*[1]

[1]PGED/DEE Instituto Militar de Engenharia; [2]Instituto de Telecomunicações, Portugal;
[3]ESTG, Instituto Politécnico de Leiria, Portugal; [4]PEE/COPPE/DEL, Univ. Federal do Rio de Janeiro;
e-mails: anderson.oliveira@ieee.org, nuno.rodrigues, sergio.faria@co.it.pt
eduardo@smt.ufrj.br, carla@ime.eb.br

ABSTRACT

In this work, we present the first results of a new algorithm for image compression, called *Variable Size Transform Coder* (VSTC). This algorithm uses spatial transforms and recursive optimization functions combined with a highly adaptive segmentation scheme, similar to the one proposed by the MMP algorithm (*Multi-dimensional Multiscale Parser*), that uses pattern matching. We replaced the pattern matching by spatial multiscale transforms, which include symmetric and asymmetric blocks, and we investigated and implemented some of the required tools for encoding these new blocks. The proposed method uses a much more dynamic adaptive-size transforms than the H.264/AVC standard and compares favorably with this encoder. It also consistently outperforms the pattern matching scheme of the MMP algorithm. Moreover, we have considerably reduced the computational complexity when compared to the MMP algorithm.

***Index Terms*—** Image coding, multiscale transforms, MMP, variable size transforms,

1. INTRODUCTION

For many years, organizations such as ISO and ITU have established standards for digital image/video compression, allowing the interoperability between different multimedia devices. All these standards have used spatial transforms to compact energy.

Other image compression algorithms have also been developed, presenting techniques that are very competitive, depending on the application and goals. In particular, MMP [1] is an algorithm that presents a different approach from the classic model used by the image/video compression standards. The main difference is that the encoding scheme is based on pattern matching rather than spatial transforms, usually employed by the standards. Furthermore, another important features of MMP are its recursive property and the use of multiscale blocks, allowing an adaptable encoding and thereby achieving better overall results when compared with standards such as H.264/AVC and JPEG2000 [2].

MMP has some properties similar to those of the H.264/AVC standard [3], as 16×16 initial blocks and the same intra-prediction framework, but using a highly adaptable optimization procedure. Under these conditions, MMP has proved to be a better compression algorithm than H.264/AVC standard in terms of rate-distortion performance for natural and artificial images, depth maps, stereoscopic and medical images ([2], [4] and [5]).

In this paper, we propose the usage of the optimization procedure from MMP, using spatial transforms rather than pattern matching, taking advantage of the scenario of multiscale blocks, investigating the required adaptations and tools for the new type of blocks. Section 2 presents an overview of the MMP algorithm. We describe the proposed algorithm in section 3, and the experimental results and conclusions are presented in sections 4 and 5 respectively.

2. OVERVIEW OF MMP ALGORITHM

The Multi-dimensional Multiscale Parser (MMP) is an algorithm proposed in [1] which uses an alternative approach for image coding. Instead of the traditional transform-quantization-encoding procedure, MMP exploits a paradigm based on pattern matching between an image segment and the recurrent elements of an adaptive dictionary.

First, a multiscale dictionary is initialized. It contains uniform blocks in 25 levels. Each level corresponding to a block size $2^M \times 2^N$, where M and N belong to the interval $[0, 4]$, and these levels are outlined by a binary segmentation tree.

The image is divided into 16×16 non-overlapping blocks, which are sequentially encoded. A hierarchical intra-prediction framework using the modes defined in the H.264/AVC is used. Then a recursive segmentation is performed from the largest block (16×16) to the smallest prediction block (4×4) in order to choose an intra mode (or a set of them) which represents the best intra-prediction for that block. The choice of the prediction mode and the block size $2^M \times 2^N$ is based on the Lagrangian cost $J = D+\lambda R$, where

This work was partially funded by CAPES/Pro-Defesa, Brazil, under research grant 23038.009094/2013-83

λ is an input parameter that represents the weight between the rate R and the distortion D during the optimization process. To carry out this process, MMP algorithm uses three distinct flags to identify the directions of the partition: a flag for *no partition*, a flag for *block partition in the vertical direction* and a flag for *block partition in the horizontal direction*.

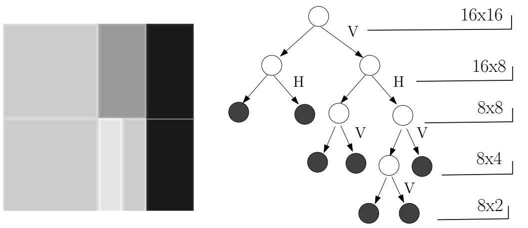


Fig. 1. Intra-prediction partition modes of a block 16×16.

MMP allows multiple intra-prediction modes on a 16×16 initial block, using both square and rectangular blocks. Figure 1 shows an example of intra-prediction segmentation of a 16×16 block. This partition can be represented by the binary tree represented in the figure, where the dark nodes represent a terminal node with an already defined intra-prediction mode.

While an intra-mode partition is tested, MMP also computes the corresponding residue block of each sub-block. For this purpose, MMP uses a similar segmentation scheme, represented by another binary tree in order to find the best residual partition setting. This new recursive partition optimization process goes from the actual residue block size until 1×1 blocks. To perform this step, MMP uses two other flags, different from those used for modes partition: one for *residual partition in the vertical direction* and another for *residual partition in the horizontal direction*. Therefore MMP uses five different flags to encode the segmentation indexes.

Matching is carried out between an $2^M \times 2^N$ residual block and a dictionary index that belongs to the same scale. The used indexes in a lower scale are concatenated, creating a new block pattern in the upper scale. Then, the new pattern is inserted in the dictionary. Moreover, this new block pattern is expanded and contracted by scale transformations in order to become available to approximate blocks of other scales. According to the original scale of each new dictionary block, the codevectors are organized into dictionary segments (dicsegs). Therefore, when MMP performs a matching, it sends two flags: one flag to identify the dictionary scale where the block is (dicseg), and another to identify the index of the dictionary representing the block. Figure 2 shows an example of MMP stream output. All these symbols are encoded by using a conventional arithmetic coder [6] with appropriate models for each type of flag. The recursive optimization using a full binary tree, associated with the procedures of dictionary index search, scale transformations and indexes' update leads to a great computational complexity of the MMP algorithm.

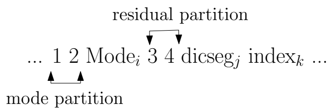


Fig. 2. MMP output example.

3. PROPOSED ALGORITHM

The idea is to use transforms with adaptive scales based on the MMP model. This approach substitutes the dictionary search and pattern matching, of the original MMP paradigm, by 2-D transforms, leading to a more efficient intra-prediction framework and transforms tools to encode images. This change involves a series of coding tools that need to be investigated, such as quantization, block scanning and entropy coding to be applied to the new blocks at each scale of the VSTC algorithm. Amongst a variety of alternative transforms, we use the well known 2-D DCT, which has shown to offer good energy compaction, and is used in most of the standards.

Let $\mathbf{X}$ be a square input block, so the transformed block $\mathbf{Y}$ is defined as $\mathbf{Y} = \mathbf{T}\mathbf{X}\mathbf{T}^T$, where $\mathbf{T}$ is a square orthogonal matrix called *transformation matrix*. Likewise, let $\mathbf{Y}$ be a transformed block, so the original block $\mathbf{X}$ can be determined by $\mathbf{X} = \mathbf{T}^T\mathbf{Y}\mathbf{T}$. Note that, in general, transform-based image compression schemes use square blocks. However, in VSTC algorithm, we use both square and rectangular blocks, so we propose an invertible transform process that operates on both square and rectangular matrices. Transform corresponds to a matrix multiplication in two stages: horizontal and vertical. For a rectangular block, we use two different square transformation matrices, one at each stage.

Let $\mathbf{X}$ be an input matrix of M rows and N columns, $\mathbf{T}_K$ a $K \times K$ transformation matrix, and $\mathbf{T}_K^T$ the transposed of $\mathbf{T}_K$ (equal to the inverse of $\mathbf{T}_K$), so the M×N transformed matrix $\mathbf{Y}$ is obtained as follows:

$$\text{if } M > N \qquad \begin{aligned} \mathbf{B}_H &= \mathbf{X} \cdot \mathbf{T}_N^T \\ \mathbf{Y} &= \mathbf{T}_M \cdot \mathbf{B}_H \end{aligned} \tag{1}$$

$$\text{if } M < N \qquad \begin{aligned} \mathbf{B}_V &= \mathbf{T}_M \cdot \mathbf{X} \\ \mathbf{Y} &= \mathbf{B}_V \cdot \mathbf{T}_N^T \end{aligned} \tag{2}$$

where $\mathbf{B}_H$ is the M×N matrix resulting from the horizontal multiplication, and $\mathbf{B}_V$ is the M×N matrix resulting from the vertical multiplication.

Multiscale transforms also have the properties of the square DCT, thereby equations 1 and 2 are invertible processes, so that the input block $\mathbf{X}$ can be recovered from the transformed block $\mathbf{Y}$ as follows:

$$\text{if } M > N \qquad \begin{aligned} \mathbf{B}_V &= \mathbf{T}_M^T \cdot \mathbf{Y} \\ \mathbf{X} &= \mathbf{B}_V \cdot \mathbf{T}_N \end{aligned} \tag{3}$$

$$\text{if } M < N \qquad \begin{aligned} \mathbf{B}_H &= \mathbf{Y} \cdot \mathbf{T}_N \\ \mathbf{X} &= \mathbf{T}_M^T \cdot \mathbf{B}_H \end{aligned} \tag{4}$$

3.1. Quantization parameter and rate control

MMP algorithm has a unique input parameter, the Lagrangian factor λ, that controls the weight between rate and distortion during the optimization functions by the cost equation $J = D + \lambda R$. VSTC algorithm uses an additional parameter: the quantization parameter, QP. This value is mapped into a quantization step Q_s, which defines the decision and reconstruction levels for a linear quantizer. So, the rate-distortion control of the VSTC algorithm depends on both the Lagrangian factor λ and the quantization parameter QP. In order to find the best relation between λ and QP in our algorithm, we experimentally evaluated a set of combinations of these quantities by using a set of test images. Based on these tests, we have defined the follow exponential growth relation by fitting the best relations resulting from the training:

$$\lambda = a \cdot 2^{\frac{QP-b}{c}} \tag{5}$$

with $a = 0.92$, $b = 13.74$ and $c = 3.428$. Then, VSTC algorithm has an unique input parameter QP, and λ is obtained by the equation 5.

3.2. Scanning process

The scanning process is a procedure intented to obtain an 1-D vector from the 2-D transformed block. It should group the coefficients in an order which facilitates the next stage: entropy coding. We have performed a set of experimental tests to determine the best scanning order for the transformed coefficients of the blocks at each scale. We have tested three scanning procedures: a straight scanning, a *z-form* scanning, which groups every four pixels into a z-form, and the well-known *zig-zag* scanning. The methods associated to each level are shown in Table 1. Figure 3 depicts the partition sequence (scale by level).

Scanning	Level
straight	1, 2, 4, 5, 9, 10, 16 and 17
Z-form	3, 6, 7, 11, 12, 18 and 19
Zig-zag	all other, except 0

Table 1. Block scanning by level.

By comparing Figure 3 with Table 1 , we can see that the straight scanning is applied to the row and column matrices, z-form is applied to the levels with side of two pixels in either direction, and *zig-zag* scanning is applied to the levels with both dimensions larger than two pixels.

3.3. Entropy coding

MMP algorithm uses the traditional arithmetic coder [6] to entropically encode all the symbols presented in Figure 2, by using different models for each of them. Note that all these symbols are single values and not groups of values as in transformed blocks. It was shown in [7] that *Context-Based Adaptive Binary Arithmetic Coding* (CABAC), proposed for the H.264/AVC, is more efficient to encoder data from residue blocks by efficiently exploiting the properties of the transformed coefficients. Then, we adapt the CABAC to be used for all the new scales, square and rectangular blocks, creating a different set of contexts to each of them in order to ensure an individual adaptation by each block during the coding process. We also experimentally observed that the models used by the MMP algorithm to encode the flags of segmentation and intra-prediction modes using the traditional arithmetic coder are more efficient than using the CABAC for these symbols, so we use both methods in the VSTC algorithm: CABAC to encode residual data, and the traditional arithmetic coder (AC) to encode the flags of segmentation and intra-prediction modes.

3.4. Rate-distortion optimization

The optimization routine is based on the one used in the MMP algorithm. Basically, it consists of the usage of two recursive optimization functions, one to optimize the prediction mode and another to optimize the residue. Both use multiscales to determine the best partition, and they operate as described in section 2. A residue block is obtained from the difference between the original block and the respective prediction block concerning to an available mode. Intra-prediction optimization can be performed from the level 24 (16×16) to the level 8 (4×4), following the sequence shown in Figure 3, and the resultant residue block can be partitioned from the actual level to level 0 (1×1). For each available mode, the algorithm recursively tests all the scales in Figure 3, which represents a full binary tree, with blocks of size from 16×16 to 1×1. Note that the rate-distortion optimization is only applied to the residue image. Figure 5 (a) shows the multiscale partition of an image segment, where the white lines represent partitions of the prediction and the black lines represent partitions of the residue. Figure 5 (b) shows the same partition overlapped to the corresponding original image segment.

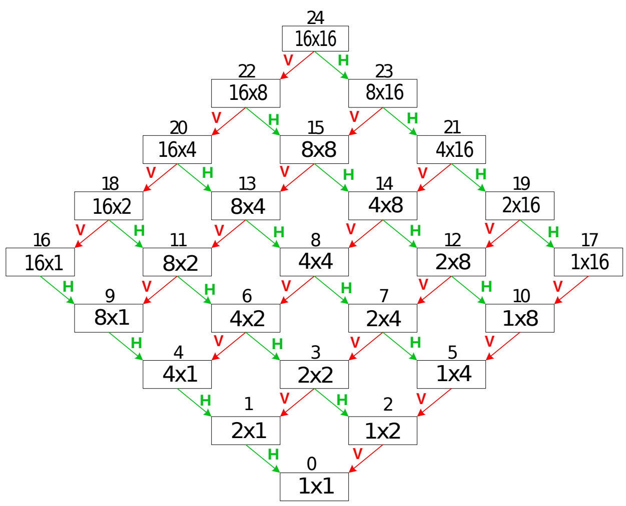
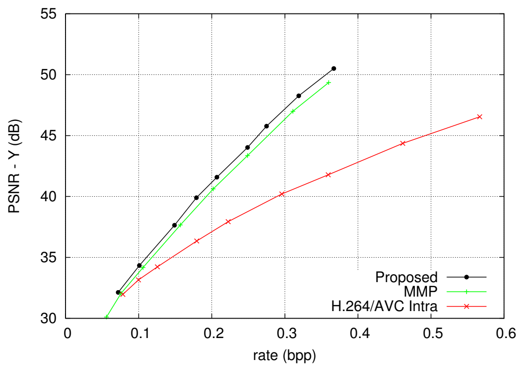
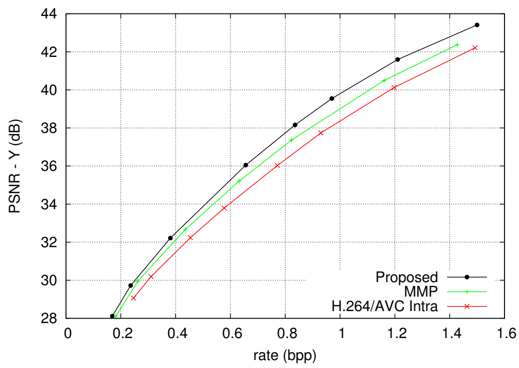


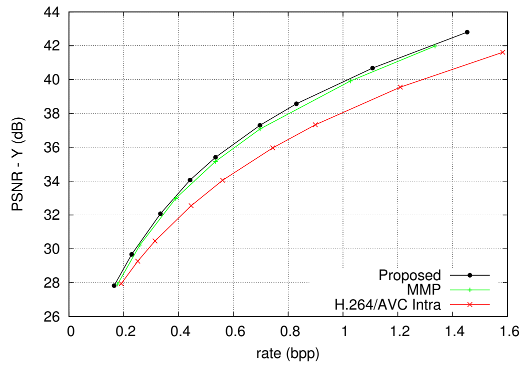
Fig. 3. Partition sequence, from 16×16 to 1×1.



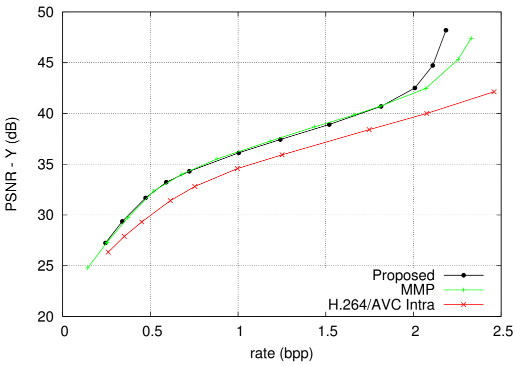
(a) Shapes (512×512)



(b) Cameraman (256×256)

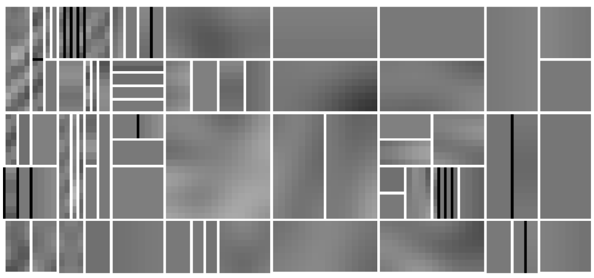


(c) Barbara (756×720)

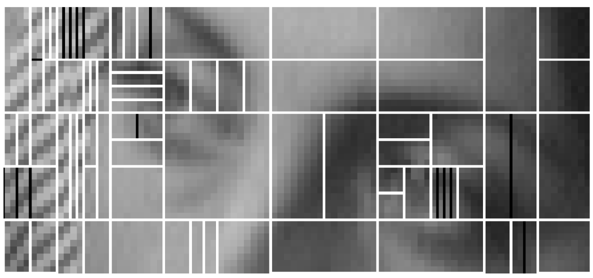


(d) pp1209 (512×512)

Fig. 4. Rate-distortion curves.



(a) Residue image.



(b) Original image.

Fig. 5. Multiscale partitions.

4. EXPERIMENTAL RESULTS

We compared the VSTC algorithm with both the MMP algorithm [1] and the H.264/AVC standard [3] (software JM18.0, using intra high profile). All these algorithms use 16×16 initial blocks and the same intra-prediction framework. The test images used in this paper are available in [8].

Figure 4 (a) shows the rate-distortion results for the synthetic image *Shapes*, which has well-defined edges, and Figure 4 (b) presents the results for the natural image *Cameraman*, which presents both uniform and textured areas. It is possible to observe that VSTC algorithm outperforms both MMP algorithm and H.264/AVC standard, mainly at higher bitrates, where we come to have more than 0.7 dB of gain in both images, when compared to MMP. Figure 4 (c) presents the result for the image *Barbara*, which presents several high-frequency areas due to the striped handkerchief, as can be seen in Figure 5 (b). VSTC algorithm outperforms both the MMP algorithm and H.264/AVC standards for all compression rates, with gains at around 0.2 dB compared to the nearest curve (MMP algorithm).

It was shown in [1] and [9] that MMP algorithm has a much better performance than standards such as H.264/AVC and JPEG2000 for hybrid images (where there are both text and grayscale figures) due to the fact of MMP does not use transforms and so the high frequencies present in the text borders are preserved. However, one can see in the Figure 4 (d) that the proposed algorithm is competitive with the MMP for the compound image *pp1209*, with consistent gains at the higher rates.

QP/Lambda(*)	Image	Processing time, compared to MMP:	
		Encoder	Decoder
22 / 5	pp1209	3.271%	0.208%
	Barbara	4.921%	0.948%
	Cameraman	9.502%	1.643%
	Shapes	13.704%	0.371%
38 / 100	pp1209	6.698%	0.404%
	Barbara	7.602%	1.157%
	Cameraman	18.188%	1.627%
	Shapes	32.747%	2.082%
45 / 500	pp1209	8.214%	0.557%
	Barbara	16.586%	1.102%
	Cameraman	27.589%	3.410%
	Shapes	47.182%	2.388%

(*) *for MMP, it was chosen the Lambda value which gives the point on the RD curve closest to the corresponding QP of the VSTC .*

Table 2. Processing time of the VSTC algorithm compared to the MMP algorithm.

Furthermore, VSTC algorithm has a much smaller computational complexity when compared to the MMP algorithm. Table 2 shows a relative time comparison between the proposed algorithm and MMP under the same conditions [1].

We can see that the VSTC algorithm spends about 3.271% of the encoding time of the MMP algorithm for image *pp1209* using $QP = 22$ (high rates), and only 0.208% of the decoding time. When we use $QP = 45$ (low rates), the VSTC algorithm spends about 8.214% of the encoding time of the MMP algorithm and only 0.557% of the decoding time.

Note that the rate-distortion optimization described in the section 3.4 is performed only on the encoder, so that the decoder does not need to test several possibilities of multiscale blocks. Even so, the MMP algorithm need to rebuild the dictionary in both the encoder and decoder, which includes the update processes. We observed a much smaller processing time of the proposed algorithm in the decoder than in the encoder for all tested images, as can be seen in Table 2. Furthermore, MMP algorithm presents a considerable difference in the processing time between low rates and high rates, since, at high rates, there is a increase in the number of dictionary index search, scale transformations and indexes update, required for the usage of small blocks. On the other hand, the VSTC algorithm does not presents a considerable difference of processing time between low and high rates as in MMP, then the time saving in the Table 2 is smaller for larger values of QP.

5. CONCLUSIONS

In this paper, we present a first version of a new encoder, that combines the multiscale nature of the MMP algorithm with spatial transforms. This led to an investigation over the entire encoder. Experimental results demonstrate that the proposed method is able to consistently outperforms both the MMP algorithm and H.264/AVC standard for image coding. The optimization scheme of the MMP algorithm can be efficiently used with multiscale transforms, so that the algorithm adapts to images with different characteristics. Moreover, one of the major advantages of the proposed method in relation to the MMP algorithm is the considerable time saving observed at the encoder and, mainly, at the decoder.

As part of a future work, some improvements to the algorithm are being researched, such as the usage of larger initial blocks (from 64×64), the usage of a more efficient intra-prediction framework and transforms tools, in order to enable this algorithm to efficiently encode high definition images, aiming to compete with the state of the art MPEG-H part 2 / H.265 (HEVC) [10].

[1] The test was performed using a processor Intel Core i7, clock 3.50GHz.

6. REFERENCES

[1] M. B. Carvalho, E. A. B. da Silva, and W. A. Finamore, "Multidimensional Signal Compression using Multiscale Recurrent Patterns," *Signal Processing, Special Edition in Image Coding Beyond Standards, Elsevier*, pp. 1559–1580, 2002.

[2] N. C. Francisco, N. M. M. Rodrigues, et al., "Multiscale Recurrent Pattern Image Coding With a Flexible Partition Scheme," *IEEE International Conference on Image Processing*, pp. 141–144, 2008.

[3] ITU-T, *Advanced video coding for generic audiovisual services. ITU-T Recommendation H.264 (MPEG-4 part 10) V9. ISO/IEC 14496-10*, 2014.

[4] D. B. Graziosi, N. M. M. Rodrigues, et al., "Compressing Depth Maps using Multiscale Recurrent Pattern Image Coding," *Electronics Letters*, vol. 46, no. 5, pp. 340–341, 2010.

[5] E. B. L. Filho, E. A. B. da Silva, et al., "Electrocardiographic Signal Compression using Multiscale Recurrent Patterns," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 12, pp. 2739–2753, 2005.

[6] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.

[7] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264 AVC Video Compression Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, 2003.

[8] "Test images.," Available at http://www02.smt.ufrj.br/ ∼eduardo/ eduardo_oficial/mmp/test_set.html.

[9] N. M. M. Rodrigues, E. A. B. da Silva, et al., "On Dictionary Adaptation for Recurrent Pattern Image Coding," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1640–1653, 2009.

[10] ITU-T and ISO/IEC JTC 1/SC 29 (MPEG), *High efficiency video coding*, Recommendation ITU-T H.265 and ISO/IEC 23008-2, 2013.