Intra Predictive Depth Map Coding Using Flexible Block Partitioning

Luís F. R. Lucas, *Member, IEEE*, Krzysztof Wegner, Nuno M. M. Rodrigues, *Member, IEEE*, Carla L. Pagliari, *Senior Member, IEEE*, Eduardo A. B. da Silva, *Senior Member, IEEE*, and Sérgio M. M. de Faria, *Senior Member, IEEE*

***Abstract*— A complete encoding solution for efficient intra-based depth map compression is proposed in this paper. The algorithm, denominated predictive depth coding (PDC), was specifically developed to efficiently represent the characteristics of depth maps, mostly composed by smooth areas delimited by sharp edges. At its core, PDC involves a directional intra prediction framework and a straightforward residue coding method, combined with an optimized flexible block partitioning scheme. In order to improve the algorithm in the presence of depth edges that cannot be efficiently predicted by the directional modes, a constrained depth modeling mode, based on explicit edge representation, was developed. For residue coding, a simple and low complexity approach was investigated, using constant and linear residue modeling, depending on the prediction mode. The performance of the proposed intra depth map coding approach was evaluated based on the quality of the synthesized views using the encoded depth maps and original texture views. The experimental tests based on all intra configuration demonstrated the superior rate-distortion performance of PDC, with average bitrate savings of 6%, when compared with the current state-of-the-art intra depth map coding solution present in the 3D extension of a high-efficiency video coding (3D-HEVC) standard. By using view synthesis optimization in both PDC and 3D-HEVC encoders, the average bitrate savings increase to 14.3%. This suggests that the proposed method, without using transform-based residue coding, is an efficient alternative to the current 3D-HEVC algorithm for intra depth map coding.**

***Index Terms*— Depth map coding, 3D-HEVC, intra predictive coding, directional prediction, flexible block partitioning, depth modelling, linear fitting, view synthesis.**

Manuscript received January 12, 2015; revised April 21, 2015; accepted June 20, 2015. Date of publication July 16, 2015; date of current version August 10, 2015. This work was supported by the Fundação para a Ciência e a Tecnologia, Portugal, under grant SFRH/BD/79553/2011, and by the CAPES/Pro-Defesa under grant number 23038.009094/2013-83. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yui-Lam Chan.

L. F. R. Lucas is with the Instituto de Telecomunicações, 2411-901 Leiria, Portugal, and also with PEE/COPPE/DEL/Poli, Universidade Federal do Rio de Janeiro, 21941-972 Rio de Janeiro, Brazil (e-mail: luis.lucas@smt.ufrj.br).

K. Wegner is with the Poznań University of Technology, 60-965 Poznań, Poland (e-mail: kwegner@multimedia.edu.pl).

N. M. M. Rodrigues and S. M. M. de Faria are with the Instituto de Telecomunicações, 2411-901 Leiria, Portugal, and also with the Instituto Politécnico de Leiria, 2411-901 Leiria, Portugal (e-mail: nuno.rodrigues@co.it.pt; sergio.faria@co.it.pt).

C. L. Pagliari is with the Department of Electrical Engineering, Instituto Militar de Engenharia, Rio de Janeiro 22290-270, Brazil (e-mail: carla@ime.eb.br).

E. A. B. da Silva is with PEE/COPPE/DEL/Poli, Universidade Federal do Rio de Janeiro, 21941-972 Rio de Janeiro, Brazil (e-mail: eduardo@smt.ufrj.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2456509

I. Introduction

In THE last years, 3D video formats have become popular in multimedia and entertainment industries. The high immersive visual experience provided by 3D systems, as well as the recent developments in 3D production and displaying technologies, have been the main motivations for its strong adoption in the cinema industry. The multiview autostereoscopic system is one of the these technologies which enables improved 3D perception with horizontal motion parallax and does not require the use of specific 3D glasses [1]. However, the technology for multiview 3D distribution has not yet reached a mature state.

Current approaches for multiview video coding involve the transmission of a large number of views, which consequently results in larger bitrates [2]. Such a solution is known as simulcast. This format encodes all the required views using a traditional single view video encoder, such as H.264/AVC [3] or the recent H.265/HEVC [4], [5]. Its main advantage is the minimum computation delay and backward compatibility, yet presenting the lowest compression efficiency. The Multiview Video Coding (MVC) is the state-of-the-art standard for stereo and multiview video representation, proposed as an extension to the H.264/AVC standard [3], [6], [7]. In order to take advantage of the most recent video coding technology, the MV-HEVC algorithm is under development. This is an HEVC-based multiview video coding extension [8], which is conceptually similar to the MVC extension of H.264/AVC. The main advantage of the HEVC standard is its superior rate-distortion (RD) performance for single view video coding, claiming bitrate savings of about 50% over its predecessor H.264/AVC for the same video quality. The multiview extensions provide a higher compression ratio than the simulcast solution. This is achieved by exploiting the inter-view correlation, mainly through the use of block-based disparity compensation techniques. However, even with the coding gains of multiview extensions, the bitrate required for multiview video tends to linearly increase with the number of views [2], being impractical for transmission when tens of views need to be encoded.

In order to reduce the bitrate requirements for the multiview video systems, new approaches based on depth information have been recently investigated [9]. By transmitting only few depth maps and few texture views, the decoder is able to render most of the necessary views for 3D display.

Such philosophy, known as video plus depth approach, allows high bitrate savings, because most of the views do not need to be explicitly encoded and transmitted. The process used for rendering additional views is known as depth image based rendering (DIBR) [10], [11].

In the scope of the video plus depth approach, the standardisation process comes out with three solutions using different base coding technologies. MVC+D is proposed as a simple solution for sending texture views along with depth maps, which does not introduce any change to the algorithm used in MVC. All changes are related to high level syntax elements providing a way to signal the presence of depth views [12]. A more advanced approach which provides higher compression efficiency, being backward compatible with AVC (allowing a fast and easy adoption in the market), is referred to as 3D-AVC [13]. A depth enhanced 3D video coding extension to the HEVC standard, known as 3D-HEVC [8], [14], [15] is under development, being built over the MV-HEVC algorithm. This is the state-of-the-art solution for 3D video transmission. In essence, all of the 3D video coding solutions use the standard hybrid coding solution for depth compression, incorporating few additional tools for depth coding.

In this context, specific methods for depth map compression have been recently proposed [8], [16]–[18]. Regarding intra coding techniques for depth map coding, algorithms based on Platelet [16] and Wedgelet [17], [19] modelling has been successfully used for efficient depth map representation. Bitrate savings of about 25% using Platelet-based coding of depth maps have been reported in comparison to H.264/AVC intra coding [16]. The use of Wedgelet block segmentation has been early investigated on HEVC standard, presenting bitrate reductions on depth map coding up to 11% [17], [19]. Currently, Wedgelets are used as part of the state-of-the-art depth intra coding techniques adopted in 3D-HEVC standard [8]. An interesting proposal on depth intra coding fully replaces the transform-based residue coding and directional intra prediction framework of 3D-HEVC standard by an advanced geometry-based intra prediction approach which includes plane fitting, Wedgelet modelling, inter-component prediction and constant offset residual coding [18]. In terms of coding performance, this algorithm is not superior to current 3D-HEVC encoder. However, it supports efficient triangular mesh extraction for scene surface representation.

In this paper, we propose an alternative coding solution based on intra techniques for efficient compression of depth maps. The developed algorithm, referred to as Predictive Depth Coding (PDC), outperforms the most recent version of 3D-HEVC reference software [14] for intra depth map coding, when the assessment metric is the objective quality of the views synthesised using original textures and coded depth maps. This work is built over a previous version of the algorithm presented in [20]. The experimental results show that the novel coding tools and the improvements proposed in this paper resulted in a state-of-the-art RD performance for intra depth map coding. Significant gains were achieved over the current depth map intra coding algorithm used in the most recent proposal of the 3D-HEVC encoder.

This paper is organised as follows. Section II presents an overview of the current intra-based tools used for depth map coding in the on-going 3D-HEVC standard. An explanation of the overall structure of PDC is presented in Section III. A detailed description of the coding techniques used in PDC is given in Section IV. Section V presents the encoder rate-distortion control mechanism for PDC. Experimental results are presented and discussed in Section VI, and the conclusions are drawn in Section VII.

II. Overview of Current Intra Techniques for Depth Map Coding in 3D-HEVC

In general, depth maps are characterised by approximately constant regions, separated by sharp edges at object boundaries. Due to this characteristic, current transform-based video coding standards present some issues for coding depth map signals. Most of these issues are caused by transform coding that generates a limited number of coefficients in flat areas, and quantisation that introduces strong ringing artefacts which blur sharp edges. As the depth maps are typically used to generate intermediate views, these artefacts highly affect rendering capabilities. These observations motivated the investigation of alternative coding techniques for depth maps.

For efficient depth map intra coding, 3D-HEVC algorithm employs transform coding with directional intra prediction, as typically used for texture image coding [8]. However, in order to better represent depth maps, 3D-HEVC introduces new coding tools. For depth map intra coding, 3D-HEVC disables all in-loop filters which were designed for natural image coding. The de-blocking filter (DBF) and the sample adaptive offset filter (SAO) are useless for depth signals, and add unnecessary computational complexity.

In order to better represent the sharp edges of depth maps, 3D-HEVC has three main additional intra coding tools [14], [21]: depth modelling modes (DMM), segment-wise DC coding (SDC) and single depth intra mode. Depth Lookup Table (DLT) was also proposed to reduce the bit depth of residual signal, for those depth maps with reduced depth range due to quantisation. The View Synthesis Optimisation (VSO) consists in using the distortion of synthesised views to directly evaluate the effect of the encoded depth block error in view synthesis during the RD optimisation process. The following sub-sections present a brief description of the most important depth intra coding methods used in 3D-HEVC algorithm [21].

A. Intra Prediction Framework

Intra prediction is a key tool of standard image and video compression algorithms, being equally important for depth map coding. For an efficient prediction of the sharp edges and flat areas present in depth maps, 3D-HEVC maintains the same intra prediction modes, as proposed to the HEVC standard. This prediction framework includes 35 modes defined for square prediction unit sizes, from 4×4 up to 32×32.

HEVC directional block prediction is based on 33 different angular modes (numbered from 2 to 34) [4]. This represents a significant extension to the 8 directional modes used

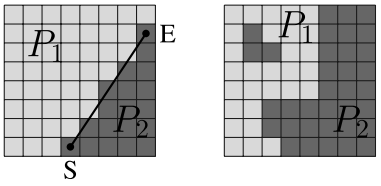


Fig. 1. Example of Wedgelet (left) and Contour (right) block partitions.

in AVC encoder, which is mainly motivated by the increased size of prediction units. To generate the predicted block, the reconstructed block boundary samples are projected in some directions, using bi-linear interpolation with $1/32$ sample accuracy.

Alternatively, intra planar and DC prediction modes can be used to predict smooth areas, which are frequent in depth maps. Intra planar assumes an amplitude surface with vertical and horizontal slopes, derived from the block neighbourhood reference samples, while intra DC uses a flat surface with a constant value estimated from the block neighbourhood.

B. Depth Modelling Modes

Depth modelling modes (DMM) consist in new intra prediction modes for efficient edge representation [8], [22]. These modes are available together with the original HEVC intra directional prediction modes, providing an alternative approximation for depth maps. The residual difference between the depth modelling approximation and the original depth map is encoded using transform coding, as for ordinary intra prediction modes, or explicitly modelled by constant partition values (CPVs). The main idea of the depth modelling modes is to divide the block into two disjointed regions, and approximate them by using constant values. Two types of partitions are defined, namely Contours and Wedgelets.

In Wedgelet partition, a straight line defined between two points located on different borders of the block is used to separate the block into two regions P_1 and P_2. This type of partition is illustrated in Figure 1 (left), using the straight line defined between points S and E. At the encoder side, the best matching Wedgelet partition is searched using the original depth signal. The Wedgelet which presents the lowest distortion, between the original depth and the pre-defined Wedgelet pattern, is transmitted. More details about Wedgelet partition search and signalling can be found at [21].

The Contour partition mode differs from Wedgelet partition in the sense that it is not a geometry guided block division, but texture guided block segmentation. This is an inter-component-predicted mode, which uses co-located texture block to generate block partition (see Figure 1 - right). A thresholding method, based on the mean value of the luminance of the texture block, is used to define two regions. Note that each region may contain multiple parts. Because it uses the texture information, Contour is an efficient method to transmit arbitrarily shaped regions. Otherwise, it would require a large amount of bits to explicitly represent the image contours.

Depth modelling modes also require the transmission of the depth model in each partition. Depth is modelled as a flat surface, characterised by a single constant partition value, called CPV. During encoding, CPVs are estimated as the mean depth value over each partition. For an even more efficient representation, estimated CPVs are predicted based on neighbour reconstructed pixels. The difference between the estimated and predicted CPVs, denominated delta CPVs, can be transmitted using two different approaches. In the first one, delta CPVs are transformed, quantized and entropy coded, while the other approach signals CPVs using SDC and DLT methods, which are detailed below.

C. Depth Lookup Table

It has been observed that most depth maps were originally quantised, not using the full available depth range of 2^8 values. Depth Lookup Table [21], [23] was proposed to reduce depth residue bitrate using a restricted set of valid depth values. 3D-HEVC constructs the DLT based on the depth values used in each group of pictures (GOP) of the input sequence, and it transmits the DLT information to the decoder.

DLT algorithm uses some auxiliary mapping tables to map the valid depth values to index values and vice-versa. From the construction procedure, detailed in [21], three tables are derived, namely the DLT $D(.)$, the Index Lookup Table $I(.)$ and the Depth Mapping Table $M(.)$. In order to derive the residual index i_{resi} to be transmitted, the original depth, d_{orig}, and the predicted depth, d_{pred}, are converted to the respective indexes and subtracted:

$$i_{\text{resi}} = I(d_{\text{orig}}) - I(d_{\text{pred}}). \tag{1}$$

At the decoder, the reconstructed mean depth value is firstly derived as

$$\hat{d}_{\text{orig}} = I^{-1}(I(d_{\text{pred}}) + i_{\text{resi}}), \tag{2}$$

and the mean residual signal is obtained using

$$\hat{d}_{\text{resi}} = \hat{d}_{\text{orig}} - d_{\text{pred}}. \tag{3}$$

The reconstructed samples $\hat{P}_{\text{x,y}}$ are computed by adding the mean residual value $\hat{d}_{\text{resi}}$ on each prediction sample $P_{\text{x,y}}$.

D. Segment-Wise DC Coding

The Segment-wise DC Coding is an alternative residual coding method, which does not require transform and quantization methods [21]. SDC can be applied to all depth intra prediction modes and it only can be used at Prediction Units (PUs) of size $2N \times 2N$. For directional intra prediction, one segment is defined, while for DMM two segments are defined. For each segment, SDC uses the mean value of original depth values and a predicted depth value to form a residual value that is encoded using DLT.

E. Single Depth Intra Mode

The main motivation for single depth intra mode is the fact that most areas on depth maps are smooth and present similar depth values. The principle of this mode is to reconstruct the depth block by using a single depth sample value which is obtained from a sample candidate list. Specific sample

positions of current block neighbouring are used in a predefined order to derive the sample candidate list. No residue information is transmitted when this mode is used.

F. View Synthesis Optimisation

As pointed out previously, depth maps are not directly observed by the viewer, since they present the structural information of the scene, which is mainly used for view synthesis. In order to better encode depth maps for this purpose, the View Synthesis Optimisation (VSO) method is used in 3D-HEVC. The distortion measure consists in the distortion between the synthesised views, using the original and the encoded maps [22]. This method is motivated by the fact that the small coding errors in depth data may lead to significant distortions in the synthesised views and vice-versa. To measure the synthesised view distortion, two VSO metrics can be applied in the RD optimisation process.

The first VSO metric, that is barely used, is known as the synthesised view distortion change (SVDC). It directly performs view synthesis using the encoded data and measures the distortion of the synthesised views. The distortion is computed by using the sum of squared errors (SSE) between the synthesised view, based on encoded data, and the one based on the original reference. Commonly, up to six views are used to evaluate the view synthesis distortion. The second metric, that is more often used, estimates the synthesised view distortion without actually performing view rendering. It was designed to be fast, but still provides a reliable model for view synthesis distortion. Details about these two metrics are available at [21].

III. Overview of Predictive Depth Coding

An initial version of the algorithm for the Predictive Depth Coding (PDC) approach was presented in [20] using intra coding. Such initial proposal only consisted on a flexible block partitioning scheme combined with an intra predictive framework, similar to the one used by HEVC standard, and constant residue modelling. Each generated residue block could be further partitioned, according to the flexible partitioning scheme, by a piecewise constant approximation.

Experiments, using the algorithm in [20] for depth map intra coding and the VSRS-3.5 software [11] for view synthesis, showed significant gains in terms of synthesis objective quality (PSNR), when compared to the use of the Platelet algorithm [16], specifically developed for depth map coding. Furthermore, comparisons with the state-of-the-art standards for natural image and video coding, as the AVC and HEVC encoders, showed a clear advantage of this preliminary algorithm.

The novel PDC algorithm, presented in this paper, was mostly motivated by the last developments of 3D-HEVC algorithm for depth map coding, based on HEVC technology. These advances allow the 3D-HEVC algorithm to achieve state-of-the-art RD performance for depth map coding, being more efficient than alternative algorithms presented in literature, including the previous PDC-based algorithm in [20]. In this context, new coding techniques and improvements to the PDC algorithm were investigated in this work in order to develop a viable alternative algorithm for intra depth map coding.

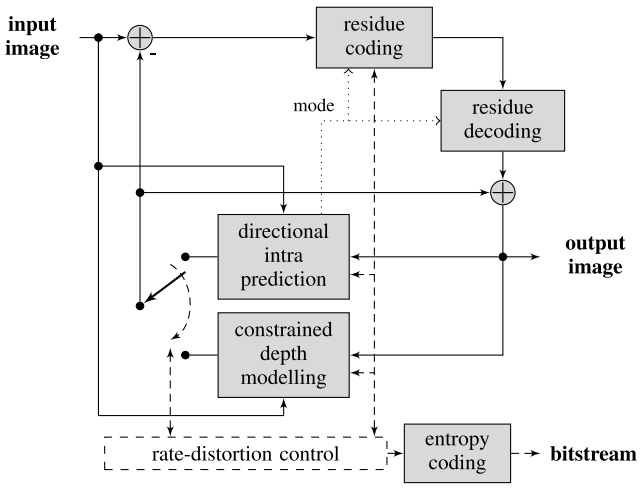


Fig. 2. Block diagram of the proposed intra PDC algorithm.

The block diagram of the novel proposed intra-based PDC algorithm is presented in Figure 2. Similarly to most image/video coding schemes, PDC presents a hybrid coding approach based on intra prediction and residue coding. In the coding process, PDC firstly divides the input image into 64×64 pixel blocks. Each block is intra predicted through the neighbouring reconstructed blocks within the same image. PDC intra prediction modes are inspired by the planar, DC and angular modes proposed in the HEVC standard. Alternatively, the block may be encoded using a constrained depth modelling mode, designed for edges that are difficult to predict. This kind of edges is typically observed in the bottom-right region of the block, which cannot be predicted by directional intra prediction modes using left and top neighbouring block samples. The proposed constrained depth modelling mode allows to explicitly signal an approximation of the edges in the block and surrounding smooth areas.

PDC encodes the residual information, given by the difference between the original and intra predicted signals, using a straightforward and efficient method that applies linear approximations to the residue signal, depending on the chosen intra prediction mode. This dependence on the prediction mode is indicated in Figure 2 by the dotted line. Like 3D-HEVC, PDC uses a depth lookup table to efficiently encode the residue signal values, mainly when depth maps present a very restricted depth range.

On the encoder side, most of the possible combinations of block partitioning and coding modes are examined. The best one is selected according to a Lagrangian RD cost. During the encoding process, each block is reconstructed in the same way as the in decoder. Reconstruction data is further used by the encoder to generate the prediction signal for the next blocks.

The encoded bitstream contains the flags used to signal the block partition, the DLT information and the symbols produced by the encoder blocks: directional intra prediction, constrained depth modelling mode and residue coding,

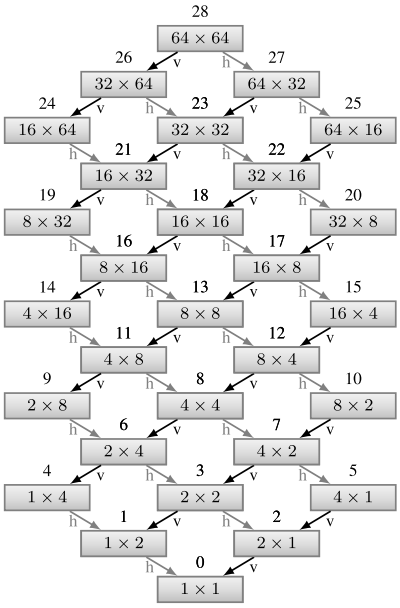


Fig. 3. Possible block sizes in PDC and respective label numbers.

as signalled by dashed lines in Figure 2. For entropy encoding, PDC employs the context adaptive m-ary arithmetic coding (CAAC) algorithm, based on the implementation of [24]. For each frame, the probability models are initialized with uniform distributions. A different context model, which depends on the block size, is used for most of the transmitted symbols.

IV. Coding Techniques Used in PDC

This section details the coding techniques that constitute the PDC algorithm, highlighting the main contributions of this work relative to the current existing techniques used in the 3D-HEVC encoder. The proposed techniques were designed to maximize the RD performance of the encoded depth maps and quality of the synthesised views. However, in order for PDC to have a low computational complexity, some simplifications, leading to sub-optimal decisions have been made. Thus, the proposed algorithm is a compromise between complexity and coding efficiency.

A. Flexible Block Partitioning

Unlike the 3D-HEVC encoder, PDC uses an alternative block partitioning scheme for a more flexible block approximation. Firstly, PDC divides the input depth map into fixed 64×64 sized blocks. During the encoding process, each block can be further segmented through a flexible scheme, which recursively divides the block, either in the vertical or horizontal directions, down to the 1×1 size [25]. In this scheme, vertical partitioning is first applied and left partition is processed before the right one. Then, the block is also partitioned in the horizontal direction and the top partition is processed before the bottom one. Each generated partition is recursively processed using the same partitioning scheme, until the smallest block size is reached. The possible block sizes are labelled from 0 up to 29, as illustrated in Figure 3.

TABLE I
Flexible Segmentation Restrictions per Quadtree Level

Quadtree level	Max. block area	Min. block area
2	(64×64) 4096	(64×64) 256	(16×16)
1	(32×32) 1024	(32×32) 64	(8×8)
0	(16×16) 256	(16×16) 1	(1×1)

Note that, block sizes with a high ratio between horizontal and vertical dimensions (ratios larger than 4, *e.g.* 64×1) are not included in the proposed partitioning scheme because they significantly increase the encoder's computational complexity and have a small impact on the RD performance.

Despite the restriction to 29 possible block sizes, the complexity required to test all the possible block partitioning combinations in the encoder side still remains the main issue of the proposed flexible partitioning scheme. In [20], a rough solution was used to alleviate this problem by dividing the tree (Figure 3) into two sub-trees: one above block size 16×16 and another below. The 64×64 block could then be encoded using the upper sub-tree, where each block could be either partitioned down to 16×16 block size, or it could be encoded as sixteen individual 16×16 sub-blocks, partitioned according to the second sub-tree, from block size 16×16 down to 1×1. By using these two sub-trees, some constraints were created in block partitioning, because the initial 64×64 block could not contain two sub-blocks from both sub-trees simultaneously, *e.g.* sub-block sizes 16×64 and 2×4. This solution provided a significant computational complexity reduction over the one using the full partition tree. This is so, because not all the block partitioning combinations are examined, as it would happen for full tree optimisation from block size 64×64 down to 1×1. Note that the number of block partitioning combinations from a determined block size down to 1×1 increases exponentially with its size.

In this work, a more efficient solution, based on a quadtree, is proposed to mitigate the computational complexity problem of the flexible block partitioning. Three quadtree levels were defined at block sizes 16×16, 32×32 and 64×64. The four partitions, generated by each quadtree partitioning, are processed using a raster scan order. For each presented quadtree level, the flexible partitioning can be used within a restricted range of block sizes, which depends on the block area. Table I presents the proposed maximum and minimum block areas (and block sizes) that are obtained by the flexible block partitioning scheme, for each available quadtree level.

The main advantage the proposed scheme, over the previous solution in [20], is the greater amount of block partition combinations, which allows to use larger block sizes (*e.g.* 32×32) together with smaller ones (*e.g.* 1×1), maintaining a lower computational complexity. Figure 4 presents an example of an optimal segmentation tree (left) with the corresponding block partitioning scheme (right), based on quadtree plus recursive flexible partitioning. Darker nodes represent quadtree partition, while brighter nodes correspond to the flexible partitioning, being complemented with the "h" and "v" labels, for horizontal and

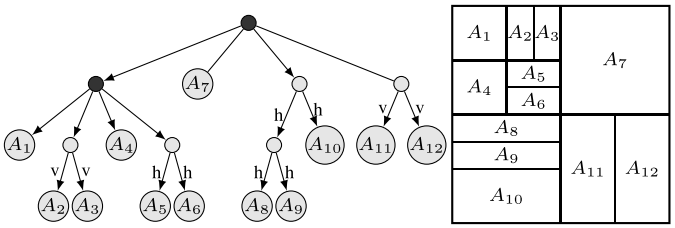


Fig. 4. Example of an optimal block partitioning and corresponding segmentation tree in PDC algorithm using quadtree plus recursive flexible partitioning.

vertical partitioning, respectively. Depth map coding using prediction and residue approximation is used at tree leaves, as labelled in Figure 4, by means of A_n, for $n = 1, .., 12$. In Figure 4, one can also observe the flexible partitioning within a quadtree level. There, sub-blocks as larger as 32×32 (represented by A_7) are used next to smaller non-square blocks, like 16×8 (represented by A_5). Note that some leaves could, eventually, be further segmented down to 1×1 size using flexible partitioning scheme.

B. Directional Intra Prediction Framework

Combined with the flexible block partitioning scheme, the directional prediction is an effective technique in the PDC algorithm. As in the initial version of the PDC algorithm [20], the intra prediction framework is based on the one proposed to the current state-of-the-art HEVC standard [4]. It includes the intra planar, DC and 33 angular prediction modes, as described in Section II-A. However, in this PDC proposal some improvements were made for better prediction of depth map signals.

In order to keep the sharpness of edges and improve prediction results, the proposed framework does not smooth the reference neighbouring samples from which prediction is derived, as typically done in AVC and HEVC standards for prediction of natural image signals. Furthermore, PDC disables another filtering that HEVC uses over some predicted samples of DC, angular 10 (horizontal) and angular 26 (vertical) modes, specifically the samples of the first row and column of the predicted block. This is done in HEVC to smooth the transition between the reference neighbouring samples and the predicted block samples. In the case of depth maps, this filtering is not convenient because it would blur the predicted depth edges.

The used intra prediction model is able to produce a reliable depth map prediction and thus a small residual signal that can be encoded using relatively low bitrate. The smooth and constant regions, which are very frequent in depth maps, are efficiently predicted by DC and planar prediction modes. In addition, angular modes combined with flexible block partitioning are able to represent the sharp edges which also characterise depth maps. The effectiveness of this method results from the piecewise edge approximation using a variety of rectangular sub-block sizes and prediction directions for each partition.

In order to further improve the performance of the proposed directional intra prediction for depth map coding using PDC algorithm, further solutions were investigated. As previously explained, PDC uses a great amount of block sizes which can be predicted based on 33 directional intra modes. Since some block sizes are very small or narrow, some directional intra prediction modes, namely adjacent directions, may be redundant and produce similar prediction patterns. For example, the 1×4 block size presents very few samples in the horizontal direction (1-wide width), which is not sufficient to project the left neighbour reference samples into 17 clearly distinct prediction directions (angular 2 up to angular 18 modes). In this context, in order to avoid unnecessary calculations and to use less bits for directional intra prediction coding, a reduction in the set of prediction directions was investigated for the proposed algorithm according the used block size. Table II presents the proposed restricted direction set of angular prediction modes for each block size in the PDC algorithm. DC mode is enabled at all block sizes, while the planar is active for block sizes larger or equal to 2×2.

TABLE II
RESTRICTED DIRECTION SET OF AVAILABLE ANGULAR PREDICTION MODES PER BLOCK SIZE

Block size $w \times h$	**Active angular modes**
$w \geq 16$ and $h \geq 16$	all modes
$w \geq 16$ and $h = 8$	all modes except 3, 5, 7, 9, 11, 13, 15, 17
$w = 8$ and $h \geq 16$	all modes except 19, 21, 23, 25, 27, 29, 31, 33
$w = 8$ and $h = 8$	even modes
$w = 8, 16$ and $h = 4$	even modes except 20, 24, 28, 32
$w = 4$ and $h = 8, 16$	even modes except 4, 8, 12, 16
$w = 8$ and $h = 2$	even modes except 20, 22, 24, 28, 30, 32
$w = 2$ and $h = 8$	even modes except 4, 6, 8, 12, 14, 16
$w = 4$ and $h = 4$	modes 2, 6, 10, 14, 18, 22, 26, 30, 34
$w < 4$ and $h < 4$	modes 2, 10, 18, 26, 34

Besides this restriction, that is due to the large amount of small and narrow block sizes, the characteristics of depth maps may be exploited to further restrict the used directional intra prediction modes. Because depth maps present large smooth areas, multiple prediction directions may produce the same predicted samples. In order to exploit this prediction redundancy, an analysis of the reference samples in the neighbouring reconstructed blocks is performed and some prediction directions are disabled accordingly. This method provides both speed-up of the encoder and bitrate reduction, since a more limited set of directions is tested, and less bits are required to signal each directional mode.

The proposed method defines three groups of directional prediction modes, which may be disabled as a whole when the associated neighbouring reference samples are exactly constant. These groups of prediction modes and associated neighbouring regions are shown in Table III. The group 1 contains all the directions that generate a prediction signal exclusively based on the top and left neighbourhood including the top-left pixel. When these reference samples are constant, the associated modes of group 1 are disabled. DC mode can be chosen in place of the disabled modes of group 1, since it produces the same predicted samples. When the samples of the neighbour left and down-left regions are constant, the modes of group 2 can be disabled. In this case, the angular

TABLE III
GROUPS OF PREDICTION MODES DEFINED ACCORDING TO THE BLOCK NEIGHBOUR REGIONS

Group	Neighbour regions	Prediction modes
1	top & left & top-left	modes 10 to 26, planar
2	left & down-left	modes 2 to 9
3	top & top-right	modes 27 to 34

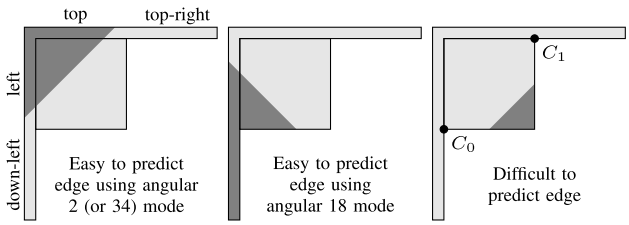


Fig. 5. Example of easy to predict edges (left and middle) and a difficult to predict edge (right).

10 mode (horizontal) is able to substitute these modes, producing the same results. Note that the angular 10 mode, belongs to group 1, and it should be active if left and top neighbour contain varying samples. Otherwise, if left-down, left, top-left and top neighbour regions are constant, the DC mode should be able to replace all the modes of groups 1 and 2. Group 3 contains those modes that depend on top and top-right neighbour regions, and can be replaced by angular mode 26 (vertical).

C. Constrained Depth Modelling Mode

The constrained depth modelling mode (CDMM) is another important novelty of the proposed PDC algorithm. The main idea behind this tool is to boost the intra directional prediction, by providing an alternative method that explicitly encodes depth edges in the bottom-right region of the block, that are hard to predict by directional intra prediction. This method is inspired on depth modelling modes used in 3D-HEVC, but several restrictions were applied to its design, in order to make it more efficient in the context of the PDC algorithm.

Intra prediction angular modes are able to represent most of the straight edges present in depth maps. However, some specific ones are difficult to predict. An example of a straight edge that is difficult to predict is illustrated in Figure 5. PDC intra prediction framework reasonably predicts straight edges coming from the left or top block neighbourhood. As can be observed, the edges illustrated in the left and middle blocks of Figure 5 can be well represented by the proposed intra prediction framework, based on the left and top neighbour reference samples. When an edge does not touch the left or top neighbour samples, like the one shown in the right block of Figure 5, it becomes difficult to predict. In some cases, the top-right or down-left reference samples may provide the necessary information to predict this kind of edges. Unfortunately, these neighbourhoods are often unavailable. Furthermore, as illustrated in the right block of Figure 5, the edge visible in the block may not reach the top-right or left-down neighbour regions, if it does not maintain the straight shape outside of the predicting block.

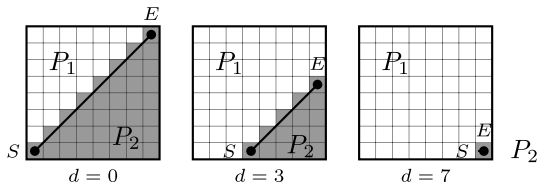


Fig. 6. Block partition examples using the proposed CDMM.

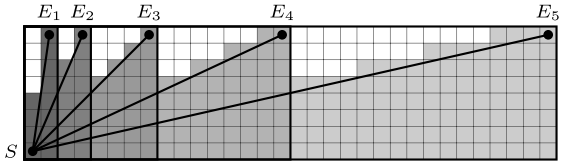


Fig. 7. Different CDMM partition slopes provided by flexible partitioning.

In order to improve the representation of such difficult to predict edges, a depth modelling mode is proposed to represent edges below the block diagonal drawn between the down-left and top-right block corners, indicated in Figure 5 by the C_0 and C_1 points. The principle of the proposed CDMM consists in dividing the block into two partitions, which are approximated by constant values. The block partitioning should occur between two points of the right and bottom margins of the predicting block. As a second restriction to the proposed method, the line drawn between the two chosen points should be parallel to the diagonal defined by the down-left and top-right block corners. This way, it can be specified by just one parameter.

Figure 6 illustrates some partition possibilities of the proposed CDMM for an 8×8 square block. The imposed constraints highly simplify the signalling of the CDMM block partition, requiring a single value only, which is represented by the offset d in Figure 6. In this example, eight different partitions that vary between the minimum offset, $d = 0$, and the maximum offset, $d = 7$, can be employed. It can be observed in Figure 6 that block partitions are performed in the bottom-right half of the block and their slope is the same as the block down-to-top diagonal, satisfying the proposed constraints.

The restriction on the block partitioning slope is advantageous in terms of computational complexity because it avoids testing a lot of block partitions with different slopes. Furthermore, by using a unique partition slope associated with the block size, no bitstream overhead is required for its transmission. The main disadvantage of this partitioning restriction is the reduced flexibility to approximate depth map edges. However, the proposed PDC algorithm is able to alleviate this issue, by combining CDMM with the flexible partitioning scheme.

The large amount of block sizes generated through flexible partitioning provides up to five different CDMM partitioning slopes according to the possible down-to-top diagonals. Figure 7 illustrates these five CDMM line partition slopes generated from different block sizes available in the PDC algorithm. The blocks are overlapped and the available slopes are represented between the points S and E_n, for $n = 1, 2, 3, 4, 5$. The illustrated overlapped block sizes represent all the block width/height ratios available in PDC.

To generate the block partitions in the image sample domain, a simple formulation was derived. Let w and h define the block width and height, respectively, and $r = \max(w,h)/\min(w,h)$ define the ratio between block dimensions. The partition P_2 (see Figure 6) is defined by all the block samples (x, y) (with $x = 0, \ldots, w-1$ and $y = 0, \ldots, h-1$) that satisfy the following conditions:

$$\begin{cases} y + (x/r) \geq (h + d - 1), & \text{if } w > h \\ x + (y/r) \geq (w + d - 1), & \text{otherwise} \end{cases} \quad (4)$$

where $d = 0, \ldots, \min(w,h)$ is the offset variable used to change the partitioning line position, as illustrated in Figure 6.

The above description clearly highlights the importance of flexible partitioning scheme for the proposed CDMM. It allows CDMM to use different block partition possibilities, using a minimal overhead. Only the position of the CDMM block partition is signalled through offset d, while the partition slope is implicitly derived from the used block size.

CDMM block partitioning generates two partitions, whose depth values are approximated by using a constant value. For P_1 partition, the approximation coefficient is derived from the block neighbourhood, namely through the mean of the left and top neighbouring reconstructed samples. The constant approximation of P_2 partition is explicitly transmitted to the decoder. For that, the mean value of the original samples in P_2 is computed and the difference between the constant values P_1 and P_2 is encoded using the DLT technique (detailed in Section II-C), as done for the residue generated by directional intra prediction. The residual information generated by the proposed approximations is bypassed, not requiring any extra bits.

Unlike directional intra prediction, which can use block sizes down to the 1×1, CDMM is disabled at smaller block sizes, namely blocks with an area equal or less than 5 pixels (see Figure 3). This is so because at these levels the right-bottom block partition is not feasible, since the blocks are just one pixel wide in either dimension directions.

D. Residual Signal Coding

The residual signal is given by the difference between the original depth and predicted samples. In PDC, the flexible block partitioning scheme combined with the directional intra prediction and CDMM provides very efficient prediction, resulting in a highly peaked residue distribution centered at zero. For this reason, PDC does not use the DCT, but an alternative approach which often assumes null residue and uses linear modelling in the other cases. The simplicity of the proposed approach is also advantageous in terms of computational complexity. Figure 8 illustrates the schematic of the proposed residue coding method. Four approximation models are available: constant, horizontal linear and vertical linear, as well as a special case of null residue. In this approach, the residue block size matches the prediction block size. Thus, the residue block is no further partitioned. This proposed residue coding solution has been shown to be more efficient than the one presented in [20].

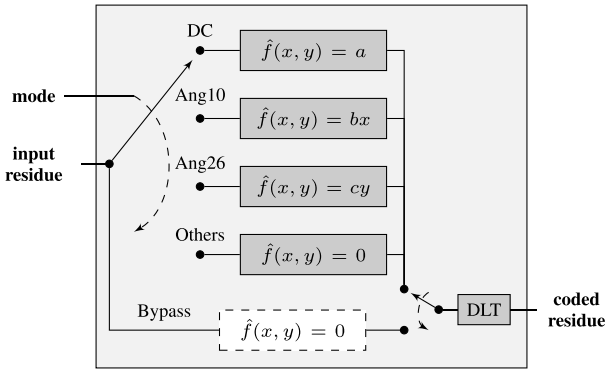


Fig. 8. Detailed schematic of the PDC residue coding method.

Linear residue approximation is mainly intended to encode a residue produced on smooth areas. Constant residue approximation is employed only when DC mode is used. On the other hand, linear fitting is used to approximate the residues generated by the horizontal (angular 10) and vertical (angular 26) directions. Note that, a row- or column-wise formulation is used without any additional offset, as formulated in Figure 8. The main motivation for angular 10 and angular 26 prediction modes is their ability to predict the offset component of depth regions which are similar to horizontal or vertical linear planes, respectively.

Due to the way their residue is encoded, these modes (DC, angular 10 and angular 26) tend to be mostly used at smooth regions, where the residue often can be easily approximated by linear fitting. For the remaining planar and angular modes, PDC always encodes a null residue. Angular prediction modes are mostly intended for depth edge prediction - they usually choose angular modes whose direction matches the predicting edge. When the prediction direction does not match the edge, highly irregular residue patterns, which cannot be efficiently represented by linear modelling, tend to be generated. For this reason, in order to save some bits, we disable residue coding on angular prediction modes. Note that, when depth map edges cannot be efficiently predicted by existing modes, encoder RD control tends to further partition the block into smaller sub-blocks that can be better predicted and encoded.

The linear model coefficient used to approximate the residue (Figure 8) is computed by a straightforward method. For a constant residue approximation by the DC mode, the coefficient is given by the mean value of the residue samples. For the angular 10 and angular 26 modes, the mean of the rightmost column and the mean of the bottom row of the residue block are transmitted, respectively. Note that the transmitted mean values computed from the right column or bottom row do not correspond directly to the value of the linear coefficients b and c represented in Figure 8. However, they can be easily derived at the decoder from the received values.

Depending on the chosen prediction mode, which is known in both the encoder and decoder, one of the residue approximation models should be transmitted. However, for a more efficient RD coding, PDC allows to bypass residue approximation, through the use of a binary flag. The bypass mode is tested as an alternative residue coding mode that can

be chosen, depending on the RD cost evaluation (see Figure 8). In practice, this flag provides a lower bitrate coding for the zero coefficient used in the approximation models.

PDC uses a simple residue quantisation scheme, by rounding the residue model coefficients in the form of mean values to the nearest integer. Note that in opposition to most codecs like 3D-HEVC, bitrate control is not performed by the quantisation step size, that controls only the precision of the transmission of the residue coefficients. As mentioned before, bitrate is controlled through the Lagrangian multiplier. When the intensity range of the input depth map is quantised, the DLT algorithm is used, as presented in Section II-C.

E. Bitstream Syntax and Context Modelling

The bitstream symbols encoded in PDC include the block partitioning flags, intra prediction modes, residue approximation coefficients and DLT. When the optimal partitioning tree and encoded depth data is determined for a 64×64 block, the symbols are organised as a string and represented in the bitstream. Context adaptive arithmetic coding is used to efficiently encode the symbols. For most symbols, an appropriate context model is derived and used. Probability models are initialised with uniform distributions and they are updated whenever a symbol is encoded. In order to guarantee that PDC can independently decode each frame, the arithmetic encoder probability models are reset with uniform distribution for each frame.

For quadtree partitioning, a binary flag is transmitted indicating whether the block is partitioned into 4 sub-blocks (q_1) or not (q_0). For flexible partitioning a ternary flag is used, indicating whether the block is partitioned, in horizontal (f_1), in vertical (f_2), or not further partitioned as it is a leaf of the partitioning tree (f_0). The partitioned block is encoded in the bitstream using the partitioning order as described in Section IV-A, for quadtree and flexible partitioning schemes.

Considering the example of Figure 4, presented in Section IV-A, the block would be encoded by the following strings of symbols:

$$
\begin{array}{l}
q_1 \\
q_1\ f_0\ A_1\ f_2\ \ f_0\ A_2\ \ f_0\ A_3\ f_0\ \ A_4\ f_1\ f_0\ A_5\ f_0\ A_6 \\
q_0\ f_0\ A_7 \\
q_0\ f_1\ f_1\ f_0\ \ A_8\ f_0\ \ A_9\ f_0\ A_{10} \\
q_0\ f_2\ f_0\ A_{11}\ f_0\ A_{12}.
\end{array}
$$

The transmitted string would be given by the concatenation of these smaller strings, which are presented in a convenient way. The first string corresponds to the first quadtree node, while the following ones contain the symbols used to encode the four nodes (quadtree level 1) in a raster scan order, as a result of the first quadtree partitioning. The A_n strings mean that all symbols are used to encode the sub-block (by using directional intra prediction, CDMM and residue coding). They are signalled whenever a non-segmentation flag, f_0, is used. Note that at level 0 of quadtree (block size 16×16), the quadtree flag is omitted, since the block cannot be further partitioned. The same procedure is carried out when the flexible partitioning reaches block size 1×1.

PDC uses an appropriate context modelling, in order to efficiently exploit the fact that larger blocks are mostly used at low bitrate RD points and smaller blocks at high bitrates. For quadtree partitioning flags three independent probability models are used, depending on the quadtree level where the flag is transmitted. For the flexible partitioning scheme, context modelling depends on both the block size (of flexible partitioning) and quadtree partition level from where flexible partitioning was initiated. Thus, based on Table I, 41 probability models (which correspond to $19+11+11$ block sizes used from quadtree levels 0, 1 and 2, respectively) are available to encode the flexible partitioning flags.

Intra prediction mode flags consist of 36 symbols that include planar, DC, directional modes and CDMM. This intra information is transmitted whenever the non-segment flag f_0 is used to signal a new encoded sub-block. The block size in which a prediction mode is transmitted is used as context, to define independent probability models in CAAC (Section III). This context modelling is important, since the number of available prediction modes depends on the block size. Furthermore, as explained in Section IV-B, the probability models can be adaptively adjusted depending on the characteristics of the neighbouring block samples.

In the case of CDMM, two additional symbols are transmitted, specifically, the partitioning offset symbol d and the mean value of the down-right block partition. In the CAAC, an independent probability model associated to the block size is used to encode the partitioning offset d. Note that this context is important because the range of values used in symbol d varies with block size. To encode the mean value of the down-right block partition a fixed context is used for all block sizes.

When DC, angular 10 or angular 26 intra prediction modes are used, the residual information can be linearly approximated. In these cases, the binary flag which indicates whether the linear approximation is applied or not, is encoded using an independent context per block size. If linear approximation is used, the linear coefficient index derived from DLT is encoded using a fixed context for arithmetic coding. For the other directional intra prediction modes, the null residue is assumed and no symbol is transmitted.

In addition to the symbols referred above, PDC also transmits the DLT table used to encode the residual signal. Since the proposed PDC algorithm is designed for intra coding, DLT is computed and transmitted for each encoded depth map frame. Unlike the previous symbols, DLT does not use context-based coding, because it consumes a negligible amount of bitrate. Thus, before the encoded information of each frame, PDC writes 256 bits (one bit per intensity value) directly into the bitstream indicating which intensity values are available in the 8-bit depth intensity range (that varies between 0 and 255). Alternative solutions to encode DLT could be investigated, for example by calculating DLT and transmitting it only once for a group of pictures.

V. PDC Encoder Control

The encoder control plays an important role in the PDC rate-distortion performance, as well as in the

computational complexity. During the encoding process, PDC seeks to minimize the following Lagrangian RD cost function:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}), \tag{5}$$

where $D(\mathcal{T})$ is the block distortion used to represent the partition tree $\mathcal{T}$, and $R(\mathcal{T})$ is its rate. λ is the Lagrangian multiplier that is used to control the target bitrate. The distortion measure used for PDC is the well known mean-squared error (MSE), typically used on standard image/video encoders. The bitrate stands for the number of bits required to arithmetically encode the partition tree (Section IV-A), as well as the symbols used to encode prediction mode and residue signal. Examples of these are the residue coefficient, prediction mode and CDMM related symbols.

PDC optimisation is performed independently for each 64×64 block, by recursively partitioning each sub-block down to the smaller block size and choosing the optimal block representation. This process involves the creation of a fully expanded partition tree, which is then pruned by evaluating the coding costs at the tree nodes. Whenever the parent's node cost is inferior to the sum of the children's node costs, the given node is not partitioned.

At each tree leaf the optimal representation is found by evaluating all intra prediction modes plus residue coding. In the case of CDMM, all the partitions given by each offset, $d = 0, \ldots, min(w, h)$, are examined (see Section IV-C) and the optimal one is selected. When using directional intra prediction, PDC tests all the available prediction modes (including planar and DC) at the current block size and encodes the residue for the case of DC, angular 10 and angular 26 modes. For these modes, the encoder also tests the bypass mode, which assumes null residue, and chooses the solution which minimizes the Lagrangian cost function.

Most of the computational complexity of the PDC encoder is due to the directional intra prediction, CDMM mode and residue coding, because they are repeated thousands of times for the 29 types of block sizes. Although most of the techniques described in the previous sections were designed with the computational complexity in mind, an improved solution for the encoder RD control was proposed to further reduce the encoder complexity. This solution forces an early termination of block partitioning whenever the distortion is smaller than 10% of the Lagrangian cost ($D(\mathcal{B}) < 0.1J(\mathcal{B})$) is true, for some sub-block $\mathcal{B}$ being encoded.

Other improvements were further implemented at PDC encoder control. For example, whenever the cost of the left (or top) child node is higher than the parent's node cost, PDC reduces the encoding time by aborting further partitioning of right (or down) nodes, since the overall cost of the child nodes is known to be higher. These improvements combined with the proposed coding techniques resulted in an efficient depth map coding algorithm with a reasonable computational complexity, that is comparable with that of 3D-HEVC.

VI. Experimental Results

This section presents the experimental results of the proposed intra depth map coding solution in comparison with the current state-of-the-art intra coding techniques present in 3D-HEVC algorithm. The experimental setup is based on the common test conditions (CTC) document for 3D video core experiments [26], with some modifications detailed in the following. Simulations were run under three-view configuration using recommended test sequences and view numbers. The three-view depth maps of each sequence were encoded using PDC algorithm and 3D-HEVC reference software version HTM-13.1 [27] for comparison purposes. Since PDC algorithm is designed for intra coding, the inter-view, temporal and inter-component correlations are not exploited. In order to fairly compare the proposed algorithm with the current state-of-the-art depth intra coding techniques of 3D-HEVC, a reference HTM configuration (*RefHTM*) was created, based on the non-CTC all-intra (AI) encoder configuration provided with the HTM software. The inter-component prediction (which is present on HTM intra frames) was disabled in *RefHTM* configuration by turning off the contour prediction mode. This *RefHTM* configuration makes 3D-HEVC operation similar to PDC, in the sense that only intra coding techniques are used. For direct comparison of RD performance and computational complexity between intra coding techniques of PDC and 3D-HEVC, the View Synthesis Optimisation (VSO) method was also disabled on *RefHTM* configuration. Additional experiments for evaluation of VSO effect on PDC algorithm are also presented and compared with 3D-HEVC.

For all experiments, CTC recommended QP pairs (for texture and depth) were used with 3D-HEVC, namely (40,45), (35,42), (30,39) and (25,34). For PDC, the λ values 1200, 500, 250 and 75 were chosen as the ones that best match the bitrate produced by CTC recommended QPs for 3D-HEVC using *RefHTM* configuration. The PSNR metric is commonly used to evaluate the objective quality of the decoded video. In the case of depth maps, these results are not very meaningful, since depth maps are not directly presented to the viewer, but rather used for view synthesis purposes. An alternative approach has been developed by the experts from ISO/IEC and ITU-T JCT-3V group for depth map evaluation, which is described in the CTC document [26]. The used evaluation methodology is based on CTC recommendation and it consists in assessing the quality of the generated virtual views, using the decoded depth data and original texture views versus exactly the same generated virtual views using both original uncompressed depth and original texture views. The quality of six synthesised views placed between the positions of the encoded depth maps has been measured by luminance PSNR. Note that such a methodology allows to assess depth map coding quality losses, excluding up to some degree the influence of the particular view synthesis algorithm used. Furthermore, by using always original texture data to generate virtual views, a more accurate evaluation of the encoded depth maps performance could be made, without interferences of coding artefacts present on decoded texture views. For the purpose of view synthesis, state-of-the-art view synthesis software for linear camera arrangement implemented in HTM software has been used [11].

The first set of experimental results using the proposed PDC algorithm and 3D-HEVC *RefHTM* configuration,

TABLE IV
RATE-DISTORTION RESULTS FOR DEPTH MAP CODING USING PDC AND 3D-HEVC (*RefHTM* CONFIGURATION) ALGORITHMS

Sequence	RD point	PDC								3D-HEVC (*RefHTM* configuration)								BD-BR
		All depths rate (kbps)	PSNR (dB) of virtual views (vv)							All depths rate (kbps)	PSNR (dB) of virtual views (vv)							
			avg-vv	vv1	vv2	vv3	vv4	vv5	vv6		avg-vv	vv1	vv2	vv3	vv4	vv5	vv6	
Balloons	p4	517,45	44,29	44,39	43,17	45,20	45,04	43,31	44,62	527,22	44,23	44,31	43,18	45,13	45,02	43,25	44,49	**-1,79%**
	p3	794,19	45,58	45,63	44,47	46,49	46,27	44,68	45,96	746,51	45,34	45,38	44,29	46,22	46,10	44,44	45,62	
	p2	1121,34	46,71	46,74	45,64	47,64	47,40	45,83	47,03	1048,90	46,45	46,46	45,41	47,35	47,16	45,54	46,75	
	p1	1989,73	48,92	48,82	47,92	49,92	49,53	48,09	49,24	1808,33	48,40	48,34	47,47	49,34	49,12	47,51	48,63	
Kendo	p4	456,44	40,73	43,87	41,84	41,72	40,08	38,35	38,53	462,36	40,54	43,75	41,60	41,47	39,92	38,04	38,45	**-6,32%**
	p3	683,47	41,88	45,20	43,26	42,98	41,04	39,25	39,53	646,36	41,48	44,78	42,75	42,56	40,66	38,84	39,31	
	p2	940,28	42,95	46,32	44,43	44,35	41,89	40,15	40,54	895,22	42,55	45,85	43,90	43,89	41,51	39,71	40,44	
	p1	1609,54	45,10	48,54	46,63	46,87	43,65	42,07	42,84	1507,68	44,59	47,86	46,02	46,30	43,10	41,56	42,71	
Newspaper	p4	666,30	39,20	38,72	37,82	39,86	40,43	38,35	40,03	678,00	39,15	38,70	37,80	39,85	40,37	38,29	39,89	**-2,70%**
	p3	1067,29	40,43	39,78	39,01	41,08	41,70	39,64	41,34	988,40	40,15	39,52	38,73	40,77	41,41	39,42	41,05	
	p2	1553,10	41,43	40,73	40,06	42,22	42,59	40,61	42,38	1429,92	41,13	40,47	39,77	41,78	42,29	40,40	42,05	
	p1	2805,18	43,25	42,32	41,95	44,13	44,38	42,44	44,27	2532,38	42,86	42,10	41,54	43,72	44,02	42,08	43,73	
GT Fly	p4	543,66	41,40	41,87	40,35	42,09	42,05	40,31	41,73	578,23	41,15	41,62	40,09	41,85	41,80	40,02	41,50	**-13,07%**
	p3	843,53	42,64	43,07	41,57	43,37	43,30	41,56	42,96	806,46	42,08	42,53	41,02	42,78	42,72	40,98	42,43	
	p2	1210,91	43,84	44,27	42,78	44,59	44,51	42,75	44,13	1160,24	43,21	43,64	42,17	43,93	43,86	42,12	43,55	
	p1	2211,44	46,19	46,56	45,15	46,99	46,90	45,11	46,41	2117,57	45,42	45,80	44,38	46,17	46,12	44,33	45,69	
Poznan Hall2	p4	204,67	46,27	46,87	44,76	46,44	47,37	45,29	46,86	203,51	46,12	46,81	44,75	46,13	47,09	45,22	46,69	**-7,31%**
	p3	273,79	47,85	48,49	46,41	47,91	49,04	46,97	48,29	264,36	47,32	47,97	46,00	47,33	48,37	46,48	47,80	
	p2	354,10	49,12	49,86	47,70	49,12	50,37	48,26	49,43	339,91	48,52	49,02	47,19	48,55	49,60	47,76	49,01	
	p1	571,05	51,48	52,20	49,98	51,40	52,88	50,61	51,79	538,12	50,67	51,14	49,25	50,63	51,87	49,96	51,19	
Poznan Street	p4	443,89	43,23	44,32	42,38	44,10	43,63	41,55	43,41	437,17	43,12	44,20	42,28	44,09	43,44	41,42	43,31	**0,07%**
	p3	678,13	44,33	45,43	43,53	45,18	44,70	42,68	44,50	625,95	44,14	45,15	43,28	45,01	44,54	42,51	44,34	
	p2	986,16	45,16	46,20	44,34	46,09	45,56	43,50	45,25	905,54	44,96	46,02	44,13	45,85	45,38	43,31	45,11	
	p1	1979,48	46,64	47,59	45,93	47,63	47,02	44,92	46,73	1772,76	46,34	47,38	45,59	47,24	46,70	44,69	46,46	
Dancer	p4	393,16	37,60	37,75	37,12	37,96	37,73	37,10	37,92	401,49	36,86	37,36	36,07	37,18	37,31	36,09	37,17	**-5,73%**
	p3	541,14	38,80	38,72	38,47	39,39	38,58	38,35	39,29	527,45	38,07	38,43	37,49	38,43	38,33	37,39	38,37	
	p2	697,54	39,90	39,57	39,69	40,53	39,55	39,59	40,46	676,57	39,48	39,55	38,97	40,04	39,51	38,87	39,91	
	p1	1077,78	41,96	41,42	41,92	42,82	41,36	41,73	42,49	1026,09	41,86	41,31	41,58	42,75	41,51	41,47	42,55	
Shark	p4	1135,87	41,71	42,01	40,86	42,21	42,15	40,92	42,10	1189,39	41,44	41,74	40,55	41,94	41,91	40,63	41,89	**-10,97%**
	p3	1893,69	43,30	43,56	42,44	43,83	43,76	42,51	43,71	1832,29	42,78	43,07	41,87	43,28	43,27	41,96	43,22	
	p2	2704,78	44,64	44,87	43,75	45,18	45,11	43,86	45,07	2729,00	44,24	44,52	43,30	44,78	44,75	43,39	44,70	
	p1	4728,10	47,09	47,24	46,16	47,72	47,53	46,28	47,59	4737,46	46,59	46,85	45,61	47,13	47,14	45,75	47,07	

which use SSE as distortion metric, are presented in Table IV. The evaluated virtual views are indicated by *vv1*, *vv2* and *vv3*, which are interpolated between the first and second depth map views, and by *vv4*, *vv5* and *vv6*, which are interpolated between the second and third depth map views. The PSNR of each virtual view as well as the average PSNR for all views (*avg-vv*) are presented for the used RD points (identified by *p1*, . . . ,*p4*). The sum of the bitrate of the three depth map views (in kbits per second) and the average PSNR results of the virtual views were used to compute the Bjontegaard Delta Bitrate (BD-BR) results [28], which are shown in the last column of Table IV. BD-BR represents the average bitrate differences between RD curves for the same PSNR of virtual views, where negative values indicate actual bit-rate savings of PDC algorithm over 3D-HEVC.

The results presented in Table IV clearly show the advantage of the proposed intra coding approach over the current state-of-the-art 3D-HEVC standard. The average bitrate savings of PDC algorithm for all the tested sequences is approximately 6%. The best result is observed for *Ghost Town Fly* sequence, where PDC saves 13% of bitrate in comparison with 3D-HEVC using *RefHTM* configuration. For the worst case, PDC presents approximately the same performance of 3D-HEVC, as observed for *Poznan Street* sequence. Note that PDC performance gains relative to 3D-HEVC are not constant, varying for different sequences. This is expected, since depth maps present distinct features that are differently exploited by PDC and 3D-HEVC algorithms.

Computational complexity results for PDC and 3D-HEVC using *RefHTM* configuration are presented in Table V. The average number of seconds used to encode each depth map frame is shown for each rate point of each recommended test sequence. These time values were obtained using a 2.00 GHz Intel Xeon E5-2620 CPU running GNU/Linux (Ubuntu 14.04) and give us a rough idea of the method's computational complexity. Table V shows that PDC presents a lower computational complexity, with most encoding times being inferior to 50% of the ones of 3D-HEVC. One may observe that PDC presents higher variation of encoding times between different sequences and λ values. This can be justified by the use of a condition for early termination of block partitioning in PDC, which mainly reduces encoding times at lower bitrates or when sequences present large smooth regions.

In Figure 9, we represent some statistical results, namely the distribution of chosen intra prediction modes in PDC algorithm, using all lambdas for each sequence. These results show that DC mode presents a high usage rate, which is mainly due to the large smooth regions present in depth maps. DC mode is also chosen more often than Planar mode. This is because constant residue coding is only available for DC prediction, forcing its usage when encoding of prediction error reduces RD cost. Horizontal and vertical modes also

TABLE V
ENCODING TIME RESULTS (IN SECONDS PER FRAME) FOR EACH RD POINT OF RECOMMENDED SEQUENCES USING PDC AND 3D-HEVC (*RefHTM* CONFIGURATION) ALGORITHMS

Test sequences	PDC (seconds per frame)					3D-HEVC, only depth, (seconds per frame)					Ratio				
	p4	p3	p2	p1	**Average**	p4	p3	p2	p1	**Average**	p4	p3	p2	p1	**Average**
Balloons	3.03	3.59	4.06	4.83	**3.88**	8.69	9.03	8.90	9.31	**8.98**	0.35	0.40	0.46	0.52	**0.43**
Kendo	2.71	3.19	3.59	4.21	**3.43**	8.63	8.77	9.14	8.89	**8.86**	0.31	0.36	0.39	0.47	**0.39**
Newspaper	4.16	5.11	5.93	7.38	**5.65**	9.92	10.07	10.64	11.00	**10.41**	0.42	0.51	0.56	0.67	**0.54**
GT Fly	6.46	7.96	9.33	12.31	**9.02**	26.73	27.57	28.20	28.66	**27.79**	0.24	0.29	0.33	0.43	**0.32**
Poznan Hall2	2.63	3.31	3.90	4.91	**3.69**	19.83	20.67	21.68	22.53	**21.18**	0.13	0.16	0.18	0.22	**0.17**
Poznan Street	6.24	8.58	10.81	15.90	**10.38**	23.84	25.09	27.23	27.62	**25.95**	0.26	0.34	0.40	0.58	**0.39**
Dancer	4.40	5.45	6.29	8.05	**6.05**	24.24	24.97	25.75	26.39	**25.34**	0.18	0.22	0.24	0.31	**0.24**
Shark	6.53	8.20	9.61	12.21	**9.14**	21.80	22.24	24.96	25.40	**23.60**	0.30	0.37	0.39	0.48	**0.38**
Average	**4.52**	**5.67**	**6.69**	**8.73**	**6.40**	**17.96**	**18.55**	**19.56**	**19.98**	**19.01**	**0.27**	**0.33**	**0.37**	**0.46**	**0.36**

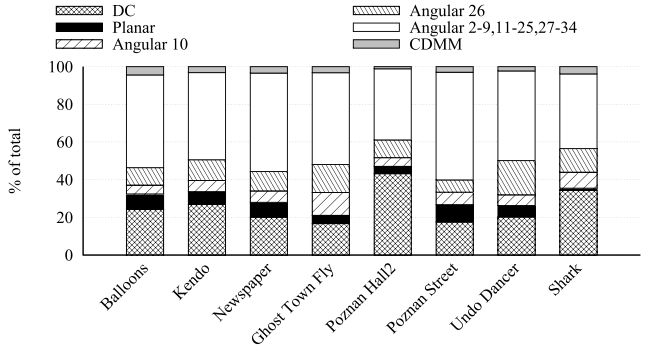


Fig. 9. Average prediction mode usage in PDC algorithm for all tested lambdas and encoded views of each test sequence.

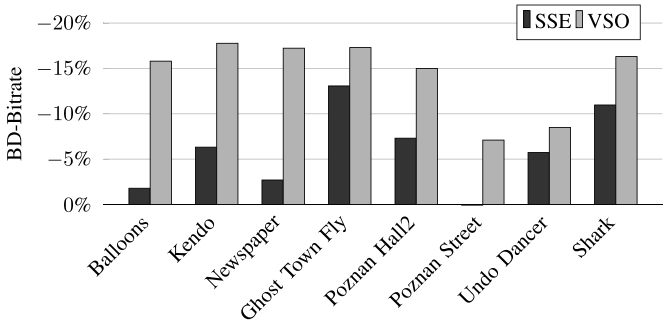


Fig. 10. Average BD-BR values of PDC relative to 3D-HEVC (*RefHTM*) using SSE and VSO distortion metrics.

present a high usage rate in comparison to other angular modes. These two modes are also used in many smooth areas because linear residue approximation is available to encode the prediction error of these modes. Despite presenting a low usage rate, CDMM plays an important role in achieved results. Experiments using PDC without CDMM showed an average BD-BR performance of -2.5% relative to 3D-HEVC using *RefHTM* configuration (in contrast to the previously presented gain of -6%), being inferior to 3D-HEVC for some sequences.

In order to evaluate the effect of VSO on PDC algorithm, we implemented VSO method on PDC based on SVDC approach [21]. As the evaluation methodology uses original texture views and encoded depth maps, the implemented VSO method uses original texture views in its process. In order to compare with 3D-HEVC, the *RefHTM+VSO* configuration was created by enabling VSO method. The VSO string (of HTM configuration file) was also modified to use original texture views instead of encoded ones, for best HTM RD performance in our evaluation setup. PDC λ values were adjusted to approximately match RD distortion curves of 3D-HEVC, resulting in the values 500, 100, 25 and 5.

Average BD-BR results for *PDC+VSO* using 3D-HEVC *RefHTM+VSO* configuration as reference are presented in Figure 10 for each test sequence (bright bars). For comparison purposes, we plotted results of previous tests based on SSE distortion metric (dark bars), specifically the values of the last column of Table IV. These results clearly show that PDC rate-distortion performance highly benefits from VSO method, achieving an average BD-BR gain of -14.3% when compared to 3D-HEVC using *RefHTM+VSO* configuration.

For illustration of RD performance of the discussed configurations at different bitrates, Figure 11 represents several RD curves using PDC and 3D-HEVC algorithms to encode depth maps of *Shark* sequence. These RD curves correspond to the average PSNR values of virtual views generated using encoded maps in function of the total bitrate used to encode the three depth map views. The two lower RD curves compare PDC with 3D-HEVC *RefHTM* configuration, using SSE as distortion metric. The advantage of PDC is clear, matching the BD-BR gain of -10.98% represented in Table IV. The previously discussed results using VSO are represented by *PDC+VSO* and *RefHTM+VSO* curves of Figure 11. We can observe that PDC tends to achieve superior RD performance at higher bitrates when using VSO. This tendency can be related with the highly flexible block partitioning scheme used in PDC, since block partitions are more frequent at higher bitrates. We also compare PDC with *RefHTM+VSO+Contour* configuration which enables inter-component prediction on 3D-HEVC intra frame coding. Although the proposed PDC algorithm does not exploit redundancy from texture frames, we can observe that *PDC+VSO* still presents a superior RD performance when compared to *RefHTM+VSO+Contour* configuration for most RD points of *Shark* sequence. Experiments for all sequences revealed an average BD-BR gain of -11% for *PDC+VSO* in relation to *RefHTM+VSO+Contour* configuration.

The investigation of inter prediction techniques is out of the scope of this paper. However, as a reference, we also present RD results of the full HTM configuration (*HTM-CTC*), which enables temporal and inter-view prediction in *RefHTM+VSO+Contour* configuration.

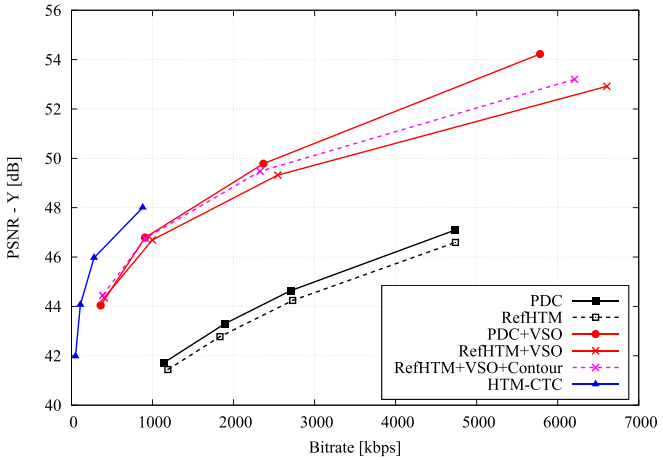


Fig. 11. Average RD performance of shark sequence using PDC and 3D-HEVC algorithms under various configurations.

Figure 11 shows that the use of temporal and inter-view predictions on HTM significantly improves its performance, achieving the highest RD performance on depth map coding. This performance difference can be justified by the large amount of redundancy existing between temporal and inter-view frames. Additional experimental results, as well as, the source code of the PDC algorithm can be found in [29].

The presented results demonstrate that, for intra depth map coding scenarios, PDC is worthy considering as an alternative to the transform-based 3D-HEVC algorithm. These results encourage further research of inter-component, temporal and inter-view prediction methods for PDC, as well as the future integration of PDC in a full 3D video-plus-depth encoder.

VII. Conclusion

A complete intra-based encoder for efficient depth map coding intended for 3D video coding applications was developed. Unlike the state-of-the-art 3D-HEVC algorithm, still under standardisation, the presented approach uses alternative coding techniques, which do not rely on the use of the transform-coding paradigm. By employing a highly optimised approach based on directional intra prediction and a novel constrained depth modelling method combined with flexible block partitioning, PDC efficiently represents depth map signals, characterised by smooth or constant areas delimited by sharp edges.

Experimental results demonstrate the superiority of PDC in terms of rate-distortion and computational complexity performance, when compared to the current state-of-the-art depth intra coding techniques used in 3D-HEVC standard.

In brief, the PDC algorithm is presented as a viable alternative to the current 3D-HEVC intra depth map coding techniques, being worthy of further investigation. As future work, we intend to investigate inter prediction techniques in PDC algorithm for efficient coding of temporal, inter-view and inter-component redundancies.

Acknowledgments

This work was supported by the Fundação para a Ciência e a Tecnologia, Portugal, under grant SFRH/BD/79553/2011, and by the CAPES/Pro-Defesa under grant number 23038.009094/2013-83.

References

[1] P. Benzie *et al.*, "A survey of 3DTV displays: Techniques and technologies," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1647–1658, Nov. 2007.

[2] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[3] *Advanced Video Coding for Generic Audiovisual Services*, document ITU-T Rec. H.264 and ISO/IEC 14496-10 MPEG-4 AVC, 2010.

[4] *High Efficiency Video Coding*, document Rec. ITU-T H.265 and ISO/IEC 23008-2, 2013.

[5] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[6] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Adv. Signal Process.*, vol. 2009, pp. 1–13, Jan. 2008.

[7] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[8] K. Müller *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[9] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[10] P. Kauff *et al.*, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process., Image Commun.*, vol. 22, no. 2, pp. 217–234, Feb. 2007.

[11] K. Wegner, O. Stankiewicz, M. Tanimoto, and M. Domanski, *Enhanced View Synthesis Reference Software (VSRS) for Free-Viewpoint Television*, document ISO/IEC JTC1/SC29/WG11 MPEG2013/M31520, Oct. 2013.

[12] Y. Chen, M. M. Hannuksela, T. Suzuki, and S. Hattori, "Overview of the MVC+D 3D video coding standard," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 679–688, May 2014.

[13] *3DAVC Draft Text 8*, document JCT3V-F1002, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Oct. 2013.

[14] *3D-HEVC Draft Text 6*, document JCT3V-J1001, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Oct. 2014.

[15] H. Schwarz *et al.*, "Extension of high efficiency video coding (HEVC) for multiview video and depth data," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 205–208.

[16] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.

[17] P. Merkle, K. Müller, and T. Wiegand, "Coding of depth signals for 3D video using wedgelet block segmentation with residual adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.

[18] P. Merkle, K. Muller, D. Marpe, and T. Wiegand, "Depth intra coding for 3D video based on geometric primitives," *IEEE Trans. Circuits Syst. Video Technol.*, 2015, doi: 10.1109/TCSVT.2015.2407791.

[19] P. Merkle, C. Bartnik, K. Müller, D. Marpe, and T. Wiegand, "3D video: Depth coding based on inter-component prediction of block partitions," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 149–152.

[20] L. F. R. Lucas, N. M. M. Rodrigues, C. L. Pagliari, E. A. B. da Silva, and S. M. M. de Faria, "Predictive depth map coding for efficient virtual view synthesis," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2058–2062.

[21] *3D-HEVC Test Model 10*, document JCT3V-J1005, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Oct. 2014.

[22] K. Müller, P. Merkle, G. Tech, and T. Wiegand, "3D video coding with depth modeling modes and view synthesis optimization," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2012, pp. 1–4.

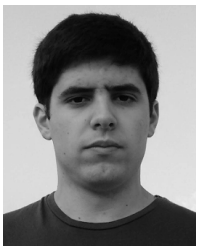
[23] F. Jager, "Simplified depth map intra coding with an optional depth lookup table," in *Proc. Int. Conf. 3D Imag.*, Dec. 2012, pp. 1–4.

[24] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[25] N. C. Francisco *et al.*, "Multiscale recurrent pattern image coding with a flexible partition scheme," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 141–144.

[26] *Common Test Conditions of 3DV Core Experiments*, document JCT3V-G1100, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Jan. 2014.

[27] *3D-HEVC Reference Software: HTM Version 13.1*. [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/3d-hevc/browser/3DVCSoftware/tags/HTM-13.1, accessed Jul. 21, 2015.
[28] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, ITU-T SG 16 Q.6 VCEG, document VCEG-M33, Apr. 2001.
[29] *Predictive Depth Coding*. [Online]. Available: http://www02.smt.ufrj.br/~eduardo/pdc, accessed Jul. 21, 2015.



Luís F. R. Lucas (M'11) was born in Portugal in 1988. He received the Engineering and M.Sc. degrees in electrical engineering from the Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Portugal, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree with Universidade Federal do Rio de Janeiro, Brazil, in collaboration with the Instituto de Telecomunicações, Portugal.

He has been a Researcher with the Instituto de Telecomunicações, Portugal, since 2009. His research interests include digital signal and image processing, namely image and video compression.

Krzysztof Wegner received the M.Sc. degree from the Poznań University of Technology, in 2008. He is currently an Assistant with the Chair of Multimedia Telecommunications and Microelectronics. He is involved in ISO standardization activities, where he contributes to the development of the future 3D video coding standards. He has co-authored several papers on free view television, depth estimation, and view synthesis. His professional interests include video compression in multipoint view systems, depth estimation form stereoscopic images, view synthesis for free view television, face detection, and recognition.

Nuno M. M. Rodrigues (M'99) received the degree in electrical engineering in 1997, the M.Sc. degree from the Universidade de Coimbra, Coimbra, Portugal, in 2000, and the Ph.D. degree from the Universidade de Coimbra, in 2009, in collaboration with the Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

He has been with the Department of Electrical Engineering, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal, since 1997. Since 1998, he has been a Researcher with the Instituto de Telecomunicações, Portugal. He has coordinated, and is currently involved as a Researcher in various national and international funded projects. His current research interests include digital signal and image processing, namely, image and video compression, multiview and plenoptic image and video compression, medical image compression, and many-core programming.

Carla L. Pagliari (M'90–SM'06) received the Ph.D. degree in electronic systems Engineering from the University of Essex, Colchester, U.K., in 2000. She was with TV Globo, Rio de Janeiro, Brazil, from 1983 to 1985. From 1986 to 1992, she was a Researcher with the Brazilian Army Research and Development Institute, Rio de Janeiro. Since 1993, she has been with the Department of Electrical Engineering, Military Institute of Engineering, Rio de Janeiro.

She has experience in television engineering and image processing, acting on the following topics: image/video coding, stereo/multiview video coding, stereo systems, image/video synthesis, video streaming, and digital TV. She participated in the definition of the Brazilian Digital Television System, and is a member of the Educational Board of the Brazilian Society of Television Engineering. She has projects in image processing for defense applications, image/video fusion of the infrared and visible-light signals, video over IP for applications in public safety, and digital TV. She acts as an Associate Editor of the *Multidimensional Systems and Signal Processing* (Springer) journal.

Eduardo A. B. da Silva (M'95–SM'05) was born in Rio de Janeiro, Brazil. He received the Electronics Engineering degree from the Instituto Militar de Engenharia, Brazil, in 1984, the M.Sc. degree in electrical engineering from Universidade Federal do Rio de Janeiro (COPPE/UFRJ), in 1990, and the Ph.D. degree in electronics from the University of Essex, U.K., in 1995.

He was with the Department of Electrical Engineering, Instituto Militar de Engenharia, Rio de Janeiro, in 1987 and 1988. Since 1989, he has been with the Department of Electronics Engineering, UFRJ. He has also been with the Department of Electrical Engineering, COPPE/UFRJ, since 1996. His teaching and research interests lie in the fields of digital signal, image, and video processing. In these fields, he has authored over 200 peer-reviewed papers. He won the British Telecom Postgraduate Publication Prize in 1995, for his paper on aliasing cancellation in sub-band coding. He has also co-authored the book *Digital Signal Processing–System Analysis and Design* (Cambridge University, 2002), that has also been translated to the Portuguese and Chinese languages, whose second edition has been published in 2010.

His research interests lie in the fields of signal and image processing, signal compression, digital TV and pattern recognition, together with its applications to telecommunications and the oil and gas industry. He is a Senior Member of the Brazilian Telecommunication Society, and a member of the Brazilian Society of Television Engineering. He has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I, in 2002, 2003, 2008, and 2009, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II in 2006 and 2007, and *Multidimensional, Systems and Signal Processing* (Springer) since 2006. He has been a Distinguished Lecturer of the IEEE Circuits and Systems Society in 2003 and 2004. He was the Technical Program Co-Chair of ISCAS2011.

Sérgio M. M. de Faria (M'93–SM'08) was born in Portugal in 1965. He received the Engineering degree in electrical engineering from Universidade de Coimbra, Portugal, in 1988, the M.Sc. degree in electrical engineering from the Universidade de Coimbra, Portugal, in 1992, and the Ph.D. degree in electronics and telecommunications from the University of Essex, U.K., in 1996.

He has been a Professor with the Department of Electrical Engineering, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Portugal, since 1990. He has collaborated in master's courses with the Faculty of Science and Technology and with the Faculty of Economy, Universidade de Coimbra, Portugal. He is currently an Auditor with A3ES Organization for Electrical and Electronic Engineering courses in Portugal. He is a Senior Researcher with the Instituto de Telecomunicações, Portugal.

His research interests include 2D/3D image and video processing and coding, motion representation, and medical imaging. In this field, he has published edited two books and has authored five book chapters, 16 journal papers, and over 100 refereed conference papers. He has been participating and he is responsible for several, national, and international (EU), funded projects.

He is an Area Editor of *Signal Processing: Image Communication*. He has been a Scientific and Program Committee Member of many international conferences. He is also a Reviewer of several international scientific journals and conferences, such as the IEEE, IET, and EURASIP.