Studying Platelet-Based Depth Map Coding

Dhiraj Kumar Shah[a], João Ascenso[b], Catarina Brites[a], Fernando Pereira[a]

[a]Instituto Superior Técnico – Instituto de Telecomunicações, Lisbon, Portugal
[b]Instituto Superior de Engenharia de Lisboa – Instituto de Telecomunicações, Lisbon, Portugal

***Abstract*—The multi-view plus depth (MVD) format is a promising representation approach for 3D and free viewpoint video systems as it allows synthesizing more views at the decoder than those explicitly coded at the encoder. This format represents each view of a visual scene with both texture (array of color pixels) and a depth map which provides information about the 3D scene geometry. Thus, for the transmission of MVD data, both texture and the associated depth maps need to be efficiently coded. For depth maps, coding artifacts should be minimized (especially at the object boundaries) since they can lead to severe geometric distortions in the virtual intermediate views. A popular approach to code depth maps is the so-called *platelet coding scheme* which uses piece-wise linear functions and quad-tree decomposition to model the depth information. In this context, this paper studies the performance of platelet-based depth map coding, notably through the analysis of several statistics such as the frequency of each modeling function, the selected block sizes and the bits consumed for the function coefficients. The platelet-based depth map coding rate-distortion (RD) performance is evaluated and compared to other standard based depth map coding solutions. This study is important not only to understand the weaknesses and strengths of this coding solution but also to design new techniques to improve the MVD coding efficiency.**

Keywords— depth map; quad-tree; platelet coding; multi-view plus depth coding;

I. INTRODUCTION

The multi-view video plus depth (MVD) 3D video representation format is currently one of the most promising approaches to provide an enhanced 3D visual experience [1]. With this representation paradigm, each view of the visual scene has two data components: texture and depth. Nowadays, there are very efficient video codecs to compress the texture part, such as those based on the H.264/AVC standard and the upcoming High Efficiency Video Coding (HEVC) standard. However, depth maps must also be efficiently encoded and transmitted to the decoder, to be later used to render some virtual intermediate views of the scene. With accurate depth maps, it is possible to render a large amount of views (across a wide view-angle) with good quality, while reducing the amount of data that needs to be transmitted. The MVD format is, therefore, essential to improve the experience in rate constrained emerging applications such as free viewpoint video (FVV) and next-generation 3DTV. For FVV, almost any desired view point may be chosen by the viewer while for auto-stereoscopic displays (glasses-free), intermediate views can be created at the decoder avoiding the need to transmit a large number of views, i.e. without a significant increase in rate when compared to stereoscopic displays [2].

In practice, each pixel in the depth map represents the relative distance from the camera to a part of the object in the 3D space. While the lighter gray regions represent nearer objects, the darker gray regions represent farther objects, as shown in Fig. 1. While depth data may be coded with textures codecs, many depth map coding schemes have been proposed in the past aiming to exploit the specific characteristics that make depth maps different from texture images. For example, depth maps are rather smooth between sharp transitions and thus may be well represented with the so-called *piece-wise smooth functions.* In depth maps, the sharp edges are rather important, since they separate relatively constant depth areas corresponding to objects that lie in different depth planes. Thus, a depth coding scheme should be able to represent the depth maps with high quality, and especially preserve well the depth edges as edge artifacts may have a significant impact on the subjective quality of the synthesized views. For this reason, most efficient depth map coding schemes avoid any filtering (e.g. removing DCT high frequency components) across the depth map edges.

Figure 1. Example of the depth map for the 1st frame of the Mobile sequence.

The popular platelet-based coding solution [3] is a edge-aware coding method, modeling the depth data with piece-wise constant (i.e. wedgelet) and piece-wise linear (i.e. platelet) functions over a quad-tree decomposition (variable block size support) of the depth map. In [3], a set of four different depth modeling functions were defined, notably the constant, linear, wedgelet and platelet functions. In addition, a pruning-based algorithm is used in the quad-tree decomposition to determine the best (image) block partitions and modeling function considering a rate-distortion (RD) optimization criterion.

Other different techniques have been used to model depth maps, such as contours [4], edges [5] and tri-trees [6]. The contour selection technique [4] starts by iteratively computing segment boundaries (i.e. closed contours) describing the depth map; the boundaries are losslessly encoded with a JBIG image codec and two modeling functions are used to represent the depth values inside each segment. In [5], edge adaptive transform and edge aware prediction techniques have been proposed as an alternative to the Discrete Cosine Transform (DCT). These techniques avoid filtering across edges in each depth block to prevent creating large high frequency coefficients. The tri-tree depth map compression technique [6] is based a mesh binary partition which divides the depth map into triangular trees (tri-tree).

In the context above, this paper aims to provide a detailed study on the performance of the platelets-based depth map coding scheme, which is not available in the literature. With this purpose, some relevant statistics related to platelets-based depth coding are obtained and analyzed,

notably the frequency of each modeling function, the block sizes obtained with the quad-tree decomposition and the bitrate spent to encode each function coefficient; all these relevant statistics are obtained after each depth map coding operation. This platelet-based depth map coding scheme analysis, performed with rather diverse test material, is important to steer the design of new depth map coding techniques aiming to improve the overall MVD coding efficiency. Finally, a RD performance evaluation of the platelets-based depth coding scheme is also provided. The focus of this evaluation is on the synthesized views distortion since these are the displayed view where the depth quality has an impact.

This paper is organized as follows. Section 2 briefly describes the platelet-based depth map coding approach. Section 3 presents and analyses some platelet-based depth coding relevant statistics and Section 4 provides a comparative RD performance analysis. Finally, Section 5 presents some final remarks and future work directions.

II. PLATELET-BASED DEPTH MAP CODING

In this section, a short description of the platelet-based depth map coding scheme proposed in [3] is provided, as this coding scheme will be analyzed and evaluated in the next sections. The platelets-based coding scheme defines two primary block types: *i)* blocks with constant depth values; and *ii)* blocks with a gradually changing gradient; these two primary block types are modeled with piece-wise constant and piece-wise linear functions, respectively. The hierarchical segmentation of each image is performed with a quad-tree decomposition method which recursively divides the image into blocks of different sizes (see Section II.B).

A. Modeling Function

Consider that the depth value at pixel position *(x,y)* ∈ S is represented by *f(x,y)*, where S is the support area of an *n*×*n* quad-tree block. In this context, each quad-tree block can be modeled (i.e. approximated) by the following four modeling functions [3]:

- Modeling function, $\hat{f}_1$, for blocks with constant pixel values, as shown in Fig. 2(a):

$$\hat{f}_1(x,y) = \alpha_0 \tag{1}$$

- Modeling function, $\hat{f}_2$, for blocks with a linear gradient, as shown in Fig. 2(b):

$$\hat{f}_2(x,y) = \beta_0 + \beta_1 x + \beta_2 y \tag{2}$$

- Modeling function, $\hat{f}_3$, for blocks with a sharp edge between two constant regions, *A* and *B*, as shown in Fig. 2(c), each modeled by function $\hat{f}_1$.

$$\hat{f}_3(x,y) = \begin{cases} \hat{f}_{1A}(x,y) = \gamma_{0A} \ , \ (x,y) \in A \\ \hat{f}_{1B}(x,y) = \gamma_{0B} \ , \ (x,y) \in B \end{cases} \tag{3}$$

- Modeling function, $\hat{f}_4$, for blocks with a sharp edge between two gradient regions, *A* and *B*, as shown in Fig. 2(d), each modeled by function $\hat{f}_2$.

$$\hat{f}_4(x,y) = \begin{cases} \hat{f}_{2A}(x,y) = \theta_{0A} + \theta_{1A} x + \theta_{2A} y, (x,y) \in A \\ \hat{f}_{2B}(x,y) = \theta_{0B} + \theta_{1B} x + \theta_{2B} y, (x,y) \in B \end{cases} \tag{4}$$

The functions $\hat{f}_1, \hat{f}_2, \hat{f}_3, \hat{f}_4$ are known as *constant*, *linear*, *wedgelet* and *platelet* functions, respectively [3]. For functions $\hat{f}_3$ and $\hat{f}_4$, the boundary line between the regions *A* and *B* is denoted by P_1 P_2, as shown in Fig. 2(c) and Fig. 2(d), respectively.

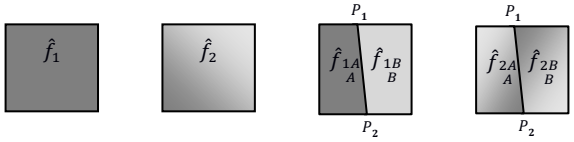

Figure 2.(a)-(d) : Examples of functions $\hat{f}_1, \hat{f}_2, \hat{f}_3$, and $\hat{f}_4$

B. Quad-Tree Decomposition

To accurately model the blocks with a detailed structure with the minimum bitrate, the quad-tree decomposition algorithm recursively subdivides the initial block into smaller blocks while collecting the bitrate and the distortion of each block partition. Therefore, the main goal of the quad-tree decomposition phase is to provide the best block partition while preventing that too many small blocks are created (since this would demand spending a significant amount of bitrate). The image quad-tree decomposition proceeds as follows:

- **Quadtree computation (top-down):** A full quad-tree decomposition is obtained with an iterative process where each block (parent node) is recursively divided into four smaller blocks (child nodes), following a top-bottom approach. Then, each node (block) of the tree is modeled by one of the four modeling functions defined in Section II.A. The modeling function selection is made considering the rate (*R*) and distortion (*D*) resulting from using that modeling function to estimate the block. During this recursive block partitioning process, all decomposition levels (i.e. block sizes) will have a Lagrangian (coding) cost ($J = D + \lambda R$) assign to them.
- **Quadtree pruning (bottom-up):** In this phase, the quad-tree decomposition is pruned from the lowest level to the top level (also known as a bottom-up approach). Consider four children nodes denoted by N_1, N_2, N_3 and N_4 with a common parent node N_0, as illustrated in Fig. 3. The four children nodes are pruned whenever (5) is verified.

$$\sum_{k=1}^{4} (D_{N_k} + \lambda R_{N_k}) > (D_{N_0} + \lambda R_{N_0}) \tag{5}$$

When (5) is not verified, the sum of the coding cost J_{N_k} is assigned to the parent node. Thus, in the next iteration, the parent node becomes the child node with the updated coding cost. This tree pruning technique is recursively performed following a bottom-up approach.

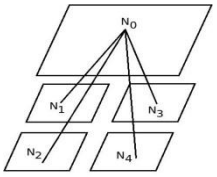

Figure 3. Quad-tree pruning

Note that the platelets-based coding algorithm does not have any rate control mechanism; thus, for a certain RD point, the value of λ (defining the weight of the rate versus the weight of the distortion) is kept fixed while several quantizers are tested to encode the modeling functions

coefficients. For each node, the quantizer, q_i, selected is the one minimizing the coding cost ($D_{q_i} + \lambda R_{q_i}$).

C. Quantization and Entropy Coding

The obtained quad-tree decomposition corresponds to the best trade-off between the rate and the (lowest) distortion. The bitrate resulting associated to the quad-tree coding depends on the modeling functions and block sizes chosen. The modeling functions have zero-order coefficients ($\alpha_0, \beta_0, \gamma_{0A}, \gamma_{0B}, \theta_{0A}, \theta_{0B}$), which are uniformly distributed and are coded with a fixed length-code. The remaining coefficients, the first-order coefficients ($\beta_1, \beta_2, \theta_{1A}$, $\theta_{2A}, \theta_{1B}, \theta_{2B}$) follow a Laplacian distribution (non-uniform) and thus an adaptive arithmetic encoder is used to encode them.

III. PLATELETS-BASED DEPTH CODING STATISTICS

To study the performance of a platelet-based depth map coding scheme, the main objective of this paper, the statistics of some relevant metrics should be analyzed, notably: *i*) the selection frequency of each modeling function; *ii*) the block sizes selected by the quad-tree decomposition; and iii) the amount of bits consumed to encode the coefficients of the selected modeling functions.

A. Test Conditions

The MPEG FTV Ad-Hoc group provides a detailed, clear and complete set of test conditions that are widely used in the literature [7] and are also adopted here for the coding statistics and the RD performance analysis presented in Section IV. Table I shows the sequences used in this experiment and their characteristics with the interpolated view identified in bold.

TABLE I. TEST SEQUENCES CHARACTERISTICS.

Sequence	Spatial resolution @ frame rate	Nr. Frames	Nr. Views
Mobile	720×528@25Hz	200	3-**5**-7
Kendo	1024×768@30Hz	150	1-**3**-5

B. Modeling Functions Frequencies

In the coding solution, each depth map block is approximated by one of four modeling functions, in fact the one leading to the lowest RD cost. Fig. 4(a) and Fig. 5(a) show the frequency of each modeling function (in percentage) for each adopted RD point where a RD point is defined by λ. This frequency was normalized according to the block size, thus, it is proportional to the area of the depth map (and averaged for all frames) using a certain modeling function. The lowest RD point corresponds to the lowest depth map (decoded) quality while the highest RD point corresponds to the highest depth map (decoded) quality. As shown, the most often selected function is the constant function, along with either linear and wedgelet functions (for the mobile and kendo sequences, respectively) depending on the spatial characteristics of the depth map.

C. Block Sizes

The quad-tree decomposition provides the optimal depth map block partition considering various sizes, each associated to an approximated depth map block corresponding to a modeling function. Typically, smaller block sizes have higher RD cost than larger blocks. Fig. 4(b) and Fig. 5(b) show the percentage of use of each block size (ranging from 2×2 to 64×64) according to the RD point. As for the modeling functions, this frequency was normalized to units of 2×2 block size. From Fig. 4(b) and Fig. 5(b), it is possible to conclude that, as expected, the depth map quality increases for smaller block sizes, notably 8×8, until the rate increase impact overcomes the distortion reduction impact.

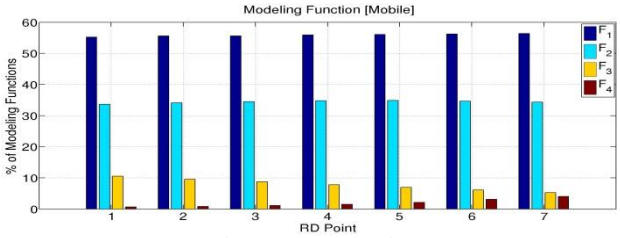

Figure 4(a) : Modeling functions statistics for the Mobile sequence.

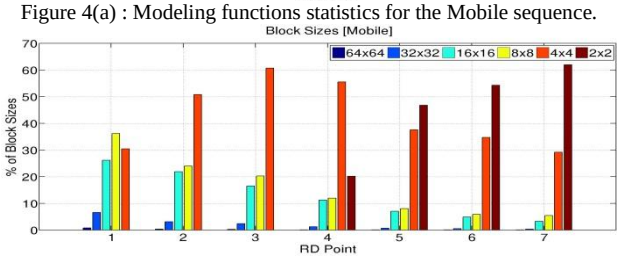

Figure 4(b): Block size statistics for the Mobile sequence.

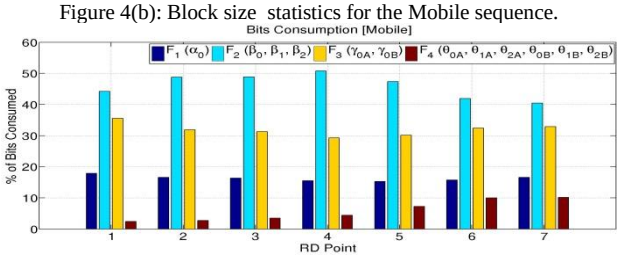

Figure 4(c): Bitrate statistics for the Mobile sequence.

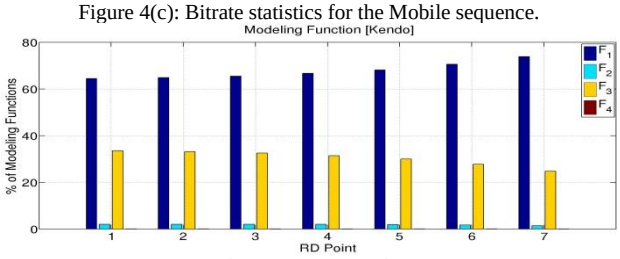

Figure 5(a): Modeling function statistics for the Kendo sequence.

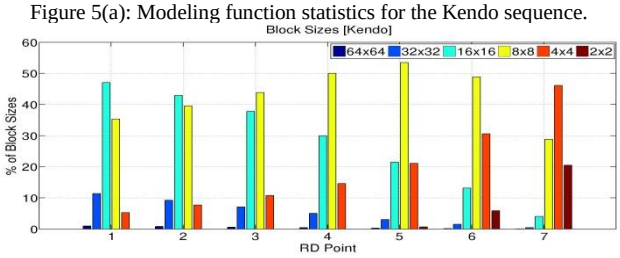

Figure 5(b): Block size statistics for the Kendo sequence.

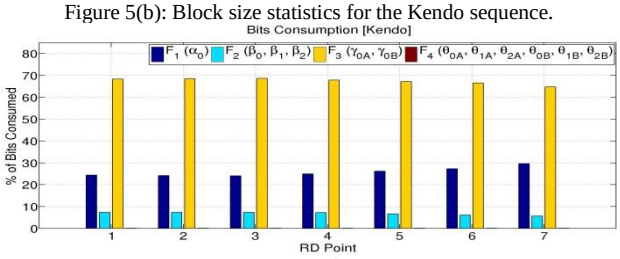

Figure 5(c): Bitrate statistics for the Kendo sequence.

D. Bitrate Consumption

The bitrate consumption depends on the modeling functions and block sizes selected along all the frames of the video sequence. Fig. 4(c) and Fig. 5(c) shows the percentage of bitrate spent to encode each modeling function coefficients for the various RD points. As expected, it is not

the most often selected modeling function coefficient (α_0 coefficient) that consumes the largest amount of bitrate (Fig. 4(a) and Fig. 5(a)). For the Mobile sequence, the largest amount of bitrate is spent on the modeling function $\hat{f}_2$ coefficients (three $\beta_0, \beta_1, \beta_2$ coefficients) where for the kendo sequence, function $\hat{f}_3$ coefficients occupy the largest amount of bitrate (four $\gamma_{0A}, \gamma_{0B}, P_1, P_2$ coefficients).

IV. RD Performance Evaluation

This section presents a comparative depth coding RD performance evaluation, another main goal of this paper. For this purpose, the MVD framework used in the assessment process is defined first.

A. MVD Evaluation Framework

Fig. 6 depicts the MVD architecture used in this paper for the RD performance evaluation [8].

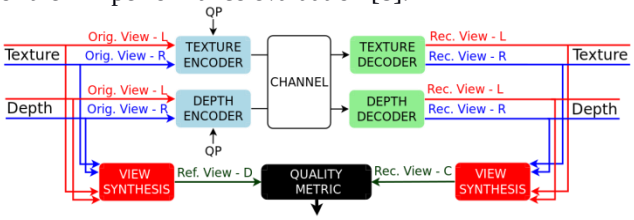

Figure 6: Multi-view plus depth evaluation framework.

In the MVD framework, a view synthesis algorithm is used to generate intermediate views. The texture and depth maps of the reference views have to be provided as input to the view synthesis module. The view synthesis framework of [7] was used for rendering the intermediate view. Naturally, the overall RD performance of the MVD framework depends on the coding solutions used to code the texture and depth maps. In this case, the texture is encoded with H.264/AVC Inter main profile, while the depth maps are encoded with the codecs in Table II. Table II also shows some relevant parameters, such as the quantization parameter (QP), the motion estimation search range (SR), as well as the software version used for each specific coding solution. For the RD performance assessment, only the synthesized view (PSNR) quality is calculated since it corresponds to the view where the depth quality has impact. However, all the bitrate is included, namely the bitrate spent in the transmission of the texture and depth maps of the adjacent views with the respect to the synthesized view.

TABLE II. Video Coding Solutions Used for Depth Map Coding.

Depth codec	QP	SR	Software
JPEG2000	26, 31, 36, 41	-	Jasper
H.264/AVC Intra		-	JM18.2
H.264/AVC Inter		96	JM18.2
Platelets		-	[3]

B. Experimental Results

According to the test conditions, Fig. 7 shows the RD performance of the synthesized view between platelets, H.264/AVC Inter, H.264/AVC Intra and JPEG2000 codecs. As shown, JPEG2000 has a lower RD performance than H.264/AVC Intra which is nowadays one of the most efficient Intra codecs available. By using the platelet-based depth map codec, better RD performance was obtained compared to H264/AVC Intra for low to medium bitrates while lower RD performance was obtained for higher bitrates. This is a very promising result, since the platelet-based coding solution does not perform any spatial prediction as the H.264/AVC Intra codec. The H.264/AVC Inter codec outperforms all the other codecs due to its motion estimation/compensation techniques. The platelet-based coding solution does not exploit any temporal correlation and, thus, suffers from lower RD performance.

V. Conclusions And Future Work

In this paper, a statistical analysis of the platelet based depth map coding solution was made along with a RD performance evaluation under precise and well known test conditions. This study may help guiding the developing of future improvements in this type of coding scheme. To further improve the RD performance of the platelet-based depth map coding scheme, spatial and temporal prediction techniques will be developed, especially to obtain better RD performance when compared to H.264/AVC Inter.

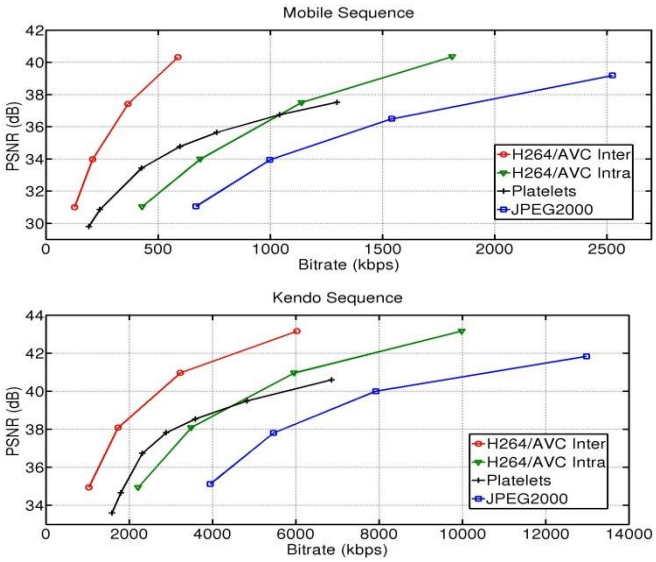

Figure 7. RD performance for a synthesized Kendo sequence view.

REFERENCES

[1] P. Merkle, A. Smolic, K. Muller, T. Wiegand, "Multi-view video plus depth representation and coding," IEEE International Conf. on Image Processing, San Antonio, TX, USA, Sep. 2007.

[2] A. Vetro, S. Yea, A. Smolic, "Towards a 3D video format for auto-stereoscopic displays," SPIE Conf. on Applications of Digital Image Processing XXXI, San Diego, CA, USA, Aug. 2008.

[3] Y. Morvan, P.H.N. de With, D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," Proceeding of SPIE, Stereoscopic Displays and Applications, vol. 6055, pp. 93-100, San Jose, CA, USA, Jan. 2006.

[4] F. Jager., "Contour-based segmentation and coding for depth map compression," IEEE Visual Communications and Image Processing, Tainan, Taiwan, Nov. 2011.

[5] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, Jaejoon Lee, Hocheon Wey, "Edge-adaptive transforms for efficient depth map coding", Picture Coding Symposium, Nagoya, Japan, Dec. 2010.

[6] M. Sarkis, W. Zia, K. Diepold, "Fast depth map compression and meshing with compressed tritree", 9th Asian Conference on Computer Vision, Xian, China, September 2009.

[7] H. Schwarz and D. Rusanovskyy, "Common test conditions for 3DV experimentation", ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, Geneva, Switzerland, May 2012.

[8] D.K. Shah, J. Ascenso, C. Brites, F. Pereira, "Evaluating multi-view plus depth coding solutions for 3D video scenarios", 3DTV Conference, Zurich, Switzerland, Oct. 2012.