EFFICIENT DEPTH MAP CODING USING LINEAR RESIDUE APPROXIMATION AND A FLEXIBLE PREDICTION FRAMEWORK

Luís F. R. Lucas[1,4]*, Nuno M. M. Rodrigues*[1,2]*, Carla L. Pagliari*[3]
Eduardo A. B. da Silva[4]*, Sérgio M. M. de Faria*[1,2]

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

ABSTRACT

The importance to develop more efficient 3D and multi-view data representation algorithms results from the recent market growth for 3D video equipments and associated services. One of the most investigated formats is video+depth which uses depth image based rendering (DIBR) to combine the information of texture and depth, in order to create an arbitrary number of views in the decoder. Such approach requires that depth information must be accurately encoded. However, methods usually employed to encode texture do not seem to be suitable for depth map coding.

In this paper we propose a novel depth map coding algorithm based on the assumption that depth images are piecewise-linear smooth signals. This algorithm is designed to encode sharp edges using a flexible dyadic block segmentation and hierarchical intra-prediction framework. The residual signal from this operation is aggregated into blocks which are approximated using linear modeling functions. Furthermore, the proposed algorithm uses a dictionary that increases the coding efficiency for previously used approximations.

Experimental results for depth map coding show that synthesized views using the depth maps encoded by the proposed algorithm present higher PSNR than their counterparts, demonstrating the method's efficiency.

Index Terms— Depth map coding, linear modeling function, predictive coding, depth image based rendering

1. INTRODUCTION

The increasing efforts to enrich multimedia viewing experience have motivated the development of new technologies for production, distribution and displaying of 3D video. In the last years, the availability of 3D contents was greatly extended beyond some traditionally restricted markets. In the consumer electronics market, the 3D systems have reached a variety of platforms which include home entertainment equipment, personal computers and mobile devices, among others.

Much of the above has been motivated by advances in 3D displaying technologies. The multi-view video (MV) systems provide much more immersive sensation over traditional stereoscopic systems, by allowing the user to change the scene observation view-point within a limited range. One of the challenges of using MV is to efficiently handle the large amount of data associated with all the views, both for storage and transmission purposes.

A straightforward method for multi-view video representation consists of encoding each view independently. The Joint Video Team of the Video Coding Experts Group (JVT) of the ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC proposed a more efficient approach that exploits the inter-view redundancy, known as Multiview Video Coding (MVC). This algorithm is an extension of the current state-of-the-art single view image and video encoder, H.264/AVC [1].

In spite of its well known efficiency, MVC may be expensive in terms of bitrate, when several views are used. As a consequence, the video+depth format has been considered by the MPEG group and was included in the MPEG-C Part 3 standard [2]. This representation enables the synthesis of virtual views using auxiliary depth data, through a process known as depth image based rendering (DIBR). The use of view rendering reduces the number of views that have to be transmitted, enabling better compression ratios, while maintaining the backward compatibility with 2D systems.

Depth maps represent the distance to the camera of each pixel of the texture image. However, each map may be converted into a grayscale image, where brighter pixels correspond to regions that are closer to the capture camera. These maps are inherently different from ordinary texture images, since they capture the 3D structure of the scene. Usually, smooth regions correspond to pixels of the same object, while sharp variations of pixels' intensity indicate object boundaries at different distances.

Several methods have been proposed in the literature to encode these piecewise-smooth image-like depth maps. The use of state-of-the-art general image and video encoders, like H.264/AVC, allows backward compatibility with the existing

This project was funded by FCT - "Fundação para a Ciência e Tecnologia", Portugal, under the grant SFRH/BD/79553/2011, and Project COMUVI (PTDC/EEA-TEL/099387/2008). The third author acknowledges the financial support of CAPES and CNPq.

technology. Nevertheless, transform-based algorithms tend to produce some coding artifacts around the sharp edges of the depth maps, specialty for low bitrates. These artifacts result in erroneous positioning of the affected pixels in the virtual view, impairing the view rendering process.

Mesh-based image coding has been proposed in [3] for depth map coding, by using an adaptive binary triangular tree (tritree). The main problem of this method is related to the placement of triangular patches on a regular grid, which usually causes the generation of a large number of small patches along the edges. JPEG2000 has also been used for depth map coding by assigning a Region Of Interest (ROI) to each object in the depth map [4]. In spite of avoiding some of the artifacts of the transform-based algorithms, this method is not efficient when many objects exist in the scene. The Multidimensional Multiscale Parser (MMP) algorithm, based on pattern matching, was successfully applied for depth map coding [5]. Nevertheless, this method requires high computational complexity in both the encoder and decoder.

The platelet-based algorithm [6] has been specifically proposed for depth map coding. It approximates the blocks resulting from a quadtree segmentation of the depth map by using different piecewise-linear modeling functions. Smooth blocks are approximated by using a constant or a linear modeling function. Blocks with depth discontinuities are modeled by a wedgelet function, defined by two piecewise-constant functions, or by a platelet function, defined by two piecewise-linear functions, both separated by a straight line. The optimal quadtree block sizes and modeling functions are chosen according to a cost function, that evaluates the rate and distortion.

The growing importance of depth map representation for emerging multimedia technologies has motivated the development of a new coding algorithm, proposed in this paper. As in [6], our algorithm models the image using (a different set of) piecewise-linear functions. However, the piecewise-linear functions are used to model a residual signal that results from the use of intra-prediction. A dictionary-based approach is also applied, providing an efficient method for reusing previously defined linear approximations.

This paper is organized as follows. Section 2 describes the techniques of the proposed framework for depth map coding. In Section 3 the rate-distortion (RD) optimization strategy is presented. Section 4 present and discusses the experimental results, and the conclusions are drawn in Section 5.

2. PROPOSED APPROACH FOR DEPTH CODING

This section describes the main features of the proposed algorithm, namely the segmentation tree, the use of hierarchical prediction and the piecewise-linear residue approximation.

The algorithm first divides the input depth map into 32×32 blocks. A horizontal or vertical segmentation generates 32×16 or 16×32 blocks, which may in turn be divided into 16×16 blocks. These blocks may be useful to approximate large smooth areas, commonly observed in depth maps. From 16×16 blocks, a more adaptive dyadic segmentation (vertical or horizontal) is used. This segmentation generates sub-blocks with dimensions $2^m \times 2^n$, for $m, n = 0, \ldots, 4$, corresponding to 25 different scales. A special symbol is transmitted to indicate whether a given block is segmented vertically or horizontally, or not segmented at all, which generates a binary tree.

The flexible dyadic segmentation has an important role in the representation of depth map edges. Note that very thin blocks may be used (*e.g.*, 16×1) to match image edges. Another important tool, that is also important to encode sharp edges, is the prediction framework. Our algorithm uses 9 intra-prediction modes, similar to those defined in H.264/AVC encoder. Nevertheless, the low-pass filtering applied by the H.264 over the neighborhood of prediction blocks is not used. The use of directional modes in an hierarchical prediction scheme, combined with the flexible segmentation of the predicted residue, results in an efficient strategy for edge estimation.

Another advantage of the used prediction step is the generation of a residue with low energy, that may be more efficiently encoded by piecewise-linear fitting. The smoothness assumption results from the observation that most depth regions are efficiently predicted, since they are constant (*e.g.*, inside an object) or vary smoothly (*e.g.*, on ground plane or walls). The remaining cases (mostly corresponding to the edges) are handled by the adaptive intra-prediction step or by using flexible segmentation.

Thus, the proposed method approximates a given block S, of size $2^m \times 2^n$, by the following linear modeling function:

$$\hat{f}(\tilde{x}, \tilde{y}) = \alpha_0 + \alpha_1 \tilde{x} + \alpha_2 \tilde{y} \quad , \tag{1}$$

where $\tilde{x} = (x - 2^{m-1} + 1)$, $\tilde{y} = (y - 2^{n-1} + 1)$ and (x, y) are the pixel coordinates within the block S. Note that $(\tilde{x}, \tilde{y})$ is a displaced version of (x, y), with an approximately zero mean value. The $(\tilde{x}, \tilde{y})$ values could be determined with exact zero mean by using fractional displacement values. However, fractional representation of $(\tilde{x}, \tilde{y})$ is not favorable because the residue block values are integers. The advantage of using $(\tilde{x}, \tilde{y})$ approximately centered around zero is that, in these conditions, the value of α_0 corresponds to the actual mean value of the target residue block. When the prediction step is successful, the values of the generated α_0 coefficients are highly correlated and centered around zero, which favors their efficient coding.

The linear model coefficients α_0, α_1 and α_2 are estimated in order to minimize the squared error between the original depth values $f(x, y)$ and the approximation $\hat{f}(\tilde{x}, \tilde{y})$. The approximation coefficients for each block are then entropy coded and transmitted to the decoder. In order to reduce the overhead associated with these coefficients, an improved residue coding scheme was developed, that is presented in the following section, together with the adopted rate-distortion optimization procedure.

3. APPROXIMATION MODEL CODING AND RD OPTIMIZATION

The proposed depth map encoder minimizes the coding cost in a rate-distortion sense. This cost depends on some parameters, such as the segmentation decisions, the selected intra-prediction modes and the used piecewise-linear functions.

The optimal segmentation tree for each 32×32 image block is determined by applying a bottom-up pruning technique over the fully expanded tree. The expanded tree represents all possible block partitions and hierarchical prediction combinations for each block scale, as well as all the residue coding tree for each predicted block. The choice of the best prediction mode at each scale uses a sub-optimal approach based on the lowest residue ℓ^1-norm. The pruning procedure choses an optimal tree $\mathcal{T}$ that minimizes the following Lagrangian cost function:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}) \tag{2}$$

where $D(\mathcal{T})$ is the block distortion and $R(\mathcal{T})$ is the bitrate required to encode the optimal tree $\mathcal{T}$.

In order to achieve the best depth map coding results the traditional squared-error measure could be used to compute the distortion value, D. However, since depth maps are not directly observed by the viewers, a different distortion formulation may be considered. We have noticed that squared-error based distortion has some issues for view synthesis. Although the larger error differences are the most important due to the square factor, a squared error-based coding algorithm tends to smooth the object boundaries. To alleviate this we propose to use the absolute error metric as the distortion D for RD cost evaluation. The superiority of the absolute error metric was experimentally validated by the significant RD gains in the view synthesis procedure.

In order to encode the proposed flexible dyadic segmentation five flags are required: two symbols identify the partitioning direction of the prediction block (horizontal or vertical); two other symbols indicate if the block should be segmented or not before the prediction step; and another symbol indicates that neither the prediction block nor the residue block should be segmented. The selected intra-prediction mode is transmitted by using a symbol with 9 possible values.

The linear model coefficients are quantized using a non uniform quantizer, that uses an adaptive quantization step Q. The value of Q for each coefficient is set to 1, 4, 8 or 13 if the values of $|\alpha_0|$, $|\alpha_1.2^m/2|$ and $|\alpha_2.2^n/2|$, belong to the intervals: $[0...10]$, $]10...22]$, $]22...86]$ and $]86...255]$, respectively. Using the adequate value of Q, the quantized coefficients are given by α_0/Q, $(\alpha_1.2^m)/(2Q)$ and $(\alpha_2.2^n)/(2Q)$. In order to further improve the compression efficiency, a method to reuse approximations was introduced. This method is inspired on the pattern matching paradigm used by algorithms such as the Multidimensional Multiscale Parser (MMP) [7]. A dictionary is created for each block scale and is initialized with the all-zero block. Then, each residue block is approximated by using a linear fitting model or by transmitting an element (index) of the dictionary. The choice is made by minimizing the associated Lagrangian cost, given by:

$$J_{\mathrm{fit}}(\mathrm{l}_k) = D_{\mathrm{fit}}(\mathrm{l}_k) + \lambda\Big(R(\mathrm{flag}_{\mathrm{fit}}) + \sum_{j=0}^{2} R(\alpha_j)\Big), \tag{3}$$

if a linear model approximation is transmitted, or by:

$$J_{\mathrm{dic}}(\mathrm{l}_k, i) = D_{\mathrm{dic}}(\mathrm{l}_k, i) + \lambda\Big(R(\mathrm{flag}_{\mathrm{dic}}) + R(i)\Big), \tag{4}$$

if index i of the dictionary is used. In the previous equations, D_{fit} and D_{dic} are the distortions associated with each approximation, and $R(\mathrm{flag}_{\mathrm{fit}})$ and $R(\mathrm{flag}_{\mathrm{dic}})$ correspond to the bitrate of the flags used to indicate whether the residue is approximated by the linear function or by a dictionary index.

For each transmitted linear approximation, a new pattern is used to update the dictionary, becoming available to approximate future residue blocks, with a small coding cost. The proposed dictionary-based method provides an efficient way to reuse approximation patterns, and differentiates the proposed algorithm from previous modeling-based and dictionary-based schemes. It is important to note that the large computational complexity that is traditionally associated with the dictionary search in traditional methods does not have a significant impact in the proposed scheme. This results from the much smaller growth ratio observed for the coefficients' dictionary, when compared with algorithms that use a dictionary to store image or residue patterns (*e.g.*, MMP [5]). The coding complexity is mostly affected by the use of an exhaustive search procedure for RD optimization, although the prediction mode selection is sub-optimal. Future schemes may consider more efficient, sub-optimal, ways to generate the block approximation.

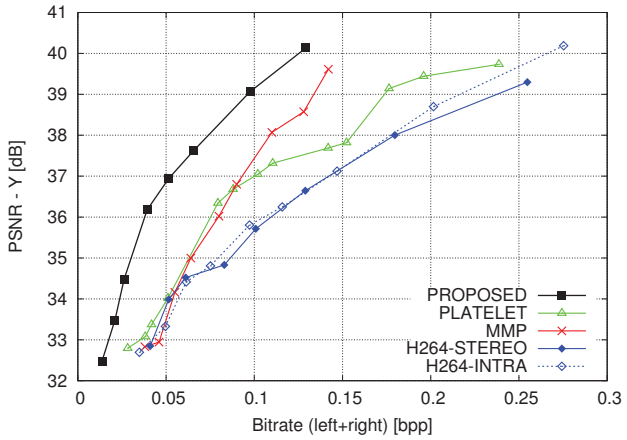
4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, the experimental results were compared against the platelet-based depth map encoder [6], the MMP algorithm [5] and the H.264/AVC (JM-18.0 version) standard using the STEREO and INTRA High-profile at level 4.0 [1]. Several test images were considered from distinct databases, however we only discuss two test images for which the Platelet results are available[1], namely the *Ballet* and *Breakdancers* sequences[2]. The source code of an implementation of the proposed method, as well as extended results can be found in [8].

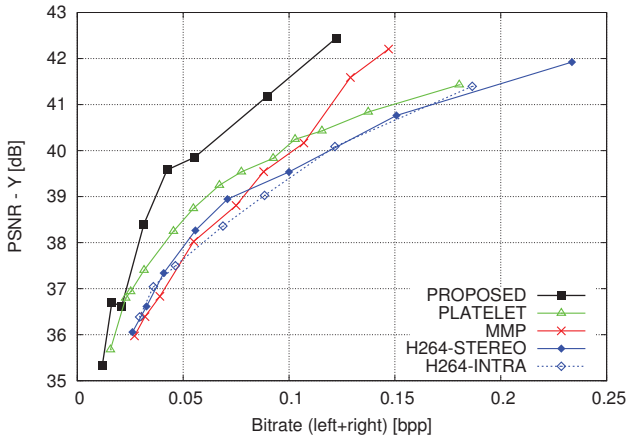
In Figure 1 we present the synthesis results for the frame 0 of the virtual view of camera 4 of *Ballet* and *Breakdancers*, using the coded depth maps and original views associated with cameras 3 and 5. The used DIBR algorithm is the VSRS-3.5 [9]. We present PSNR results for the synthesized virtual camera 4, using original depth maps and views of cameras 3 and 5. The bitrate used in the figures corresponds to the sum

[1] http://vca.ele.tue.nl/demos/mvc/PlateletDepthCoding.tgz

[2] http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/



(a) Ballet



(b) Breakdancers

Fig. 1. PSNR results for the rendered view of camera 4 (frame 0), using the encoded depth maps together with original views of cameras 3 and 5.

of bitrates used to encode left and right depth maps (cameras 3 and 5). For both sequences, the experimental results show that the proposed method achieves the higher reconstruction quality for the same bitrate. The maximum RD gains are close to 3 dB over the Platelet algorithm for the *Ballet* sequence, and almost 2 dB for *Breakdancers*. We have also analyzed the quality of the coded depth maps using PSNR and the SSIM index of the depth maps, and we observed that these results may depend on the tested sequence. In fact, such results are not very significant since the ultimate quality assessment is based on the ability of the depth map to synthesize intermediate views with good quality.

We have also evaluated the view synthesis performance based on the PSNR computed between the rendered views and the original captured view, which are consistent with the presented results, as can be seen in [8].

5. CONCLUSIONS

We present a new efficient depth map coding algorithm, that is able to outperform the current proposals in literature. Our approach takes advantage of an adaptive dyadic segmentation combined with hierarchical intra-prediction, that efficiently estimates both smooth regions and sharp edges, common in depth maps. The residue is encoded using piecewise-linear fitting. Furthermore, a dictionary-based method is used to reduce the coding costs associated with previously defined linear approximations.

Although the proposed algorithm achieves efficient coding results, there is still room for further improvements, through the use of a more efficient approach for the entropy coding of the linear model coefficients. For future work we also plan to develop less computationally complex versions of the algorithm, for example by exploiting alternative sub-optimal RD optimization procedures.

6. REFERENCES

[1] ITU-T and ISO/IEC JTC1, "Advanced video coding for generic audiovisual services," *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2010.

[2] Philips Applied Technologies, "MPEG-C part 3: Enabling the introduction of video plus depth contents," 2008, Suresnes, France.

[3] M. Sarkis, W. Zia, and K. Diepold, "Fast depth map compression and meshing with compressed tritree," in *ACCV*, 2010, vol. 5995, pp. 44–55.

[4] R. Krishnamurthy, B. Chai, H. Tao, and S. Sethuraman, "Compression and transmission of depth maps for image-based rendering," in *ICIP*, 2001, vol. 3, pp. 828–831.

[5] D.B. Graziosi, N.M.M. Rodrigues, C.L. Pagliari, E.A.B. da Silva, S.M.M. de Faria, M.M. Perez, and M.B. de Carvalho, "Multiscale recurrent pattern matching approach for depth map coding," in *PCS*, Dec. 2010, pp. 294–297.

[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. H. N. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Image Communications*, vol. 24, pp. 73–88, Jan. 2009.

[7] N. Rodrigues, E. da Silva, M.de Carvalho, S. de Faria, and V. Silva, "On dictionary adaptation for recurrent pattern image coding," *Image Processing, IEEE Transactions on*, vol. 17, no. 9, pp. 1640–1653, September 2008.

[8] http://www.lps.ufrj.br/profs/eduardo/linear_approx.

[9] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 3.5 (VSRS3.5) Document M16090, ISO/IEC JTC1/SC29/WG11 (MPEG)," May 2009.