MULTIVIEW IMAGE CODING USING GRAPH-BASED APPROACH

Thomas Maugey[1], *Antonio Ortega*[2], *Pascal Frossard*[1]

[1]Ecole Polytechnique Fédérale de Lausanne (EPFL) [2]University of Southern California

ABSTRACT

In this paper, we design a new approach for coding the geometry information in a multiview image scenario. As an alternative to depth-based schemes, we propose a representation that captures the dependencies between pixels in different frames in the form of connections in a graph. In our approach it is possible to directly perform compression by simplifying the graph, which provides a more direct control of the effect of coding on geometry representation. Our method leads to more accurate view synthesis, when compared to conventional lossy coding of depth maps operating at the same bit rate.

***Index Terms* —** Graph representation, multiview image coding, depth maps

1. INTRODUCTION

Compressing image sequences efficiently can be achieved by exploiting redundancies across images, *e.g.*, using motion-based prediction in the case of video sequences. In the case of multiview images the correlation between images depends on the geometry of the scene. Thus, a better representation of scene geometry means that target views can be more easily predicted from reference views. In terms of coding efficiency, however, overall performance will depend on the aggregate cost of sending geometry information and pixel data. In addition to being used for coding, scene geometry also enables synthesis of virtual viewpoints at the decoder side, which might be useful for navigation between the viewpoints. Representing and coding this 3D scene geometry information is thus a critical task for multiview image coding schemes.

In the literature different methods have been proposed to represent scene geometry. A popular approach consists of using disparity vectors [1], which are computed based on color similarities between blocks of two neighboring images in the multiview set. The low precision of geometry information (only one disparity vector per block) leads to suboptimal performance when the views are too distant. Some more advanced image-based representation methods handle this problem by storing a large number of images captured from close viewpoints. This is the ray-space or light field representation [2]. In this case, the geometric information is implicitly captured by the closely spaced images, which can be viewed as samples of the plenoptic function. Whereas coding and interpolation techniques can efficiently exploit this information, such a representation requires very specific acquisition conditions, which are not always practical. Other methods, such as mesh-based representations [3], combine interesting ideas from computer vision and image processing, but require a significant overhead, which makes them inefficient for coding. Improved accuracy of time-of-flight sensors [4] is leading to increased use of depth images to represent scene geometry [5]. Depth images for each view represent the distance between objects in the scene and the corresponding camera's focal plane. They have been widely used for view synthesis and to improve coding efficiency in multiview image/video representations. When depth-based approaches are used for coding, lossy coding of depth images is required in order to minimize overall rate. A major limitation of these approaches is that the resulting distortion in the depth image induces significant errors in the view synthesis or prediction, and these errors are often local and non-linear, which makes it difficult to optimize depth coding [6]. This problem has been addressed in the literature, mostly by introducing better metrics [7, 8], but direct compression of geometric may still lead to undesirable effects. This is why here we start from depth information and propose a novel approach to represent it.

In this paper we extend the *Graph-based representation (GBR)* we recently proposed [9] as an alternative to depth representation for multiview systems. Our method directly represents the connections between the pixels of the different images in a graph structure. The graph links in the proposed representation connect pixels that represent the same information across multiple viewpoints, so that only the pixels that cannot be derived from the previous image in the graph need to be explicitly transmitted (*e.g.*, disoccluded background pixels). We build on the preliminary results of [9], which showed promising performance in a lossless transmission scenario. Here, we provide further evidence of the advantages of our proposed method by introducing lossy compression within the GBR approach. This approach consists of removing images from the graph (to reduce the bit-rate) and interpolating them at the decoder side. This approach allows a better control of the effect of geometry compression on synthesized view quality. While our proposed approach is somewhat simplistic (and more efficient lossy compression methods can be investigated), we show that at similar rates GBR provides a higher quality representation of the scene geometry, as compared to conventional lossy coding of depth maps. In Sec. 2, we summarize the main concepts of our graph-based representation. Then, we present our algorithm for lossy compression of GBR in Sec. 3 and we test the efficiency of the proposed approach in Sec. 4.

2. GRAPH-BASED GEOMETRY REPRESENTATION

We recall here the main ideas of GBR construction process and view reconstruction at the decoder. Readers are referred to [9] for details. Let us consider a scene captured by N cameras with the same resolution and focal length f. The n-th image is denoted by I_n, with $1 \leq n \leq N$, where $I_n(r,c)$ is the pixel at row r and column c. We only consider translation between cameras, and we assume that the views are rectified. In other words, the geo-

This work has been partially supported by the Swiss National Science Foundation under grant 200021-126894.

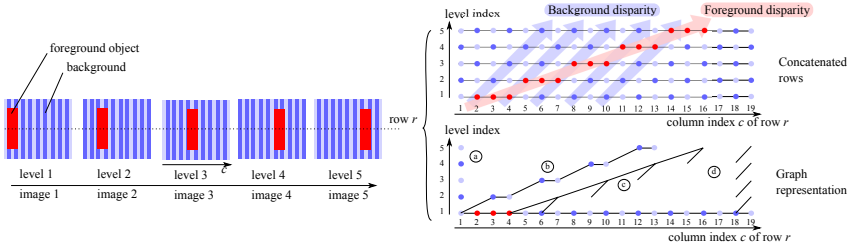


Figure 1. Toy graph construction example: blue texture background has a disparity of 1 at each level and red rectangle foreground a disparity of 3 for each level. Graph contains all different types of pixels: a) appearing, b) disoccluded, c) occluded and d) disappearing.

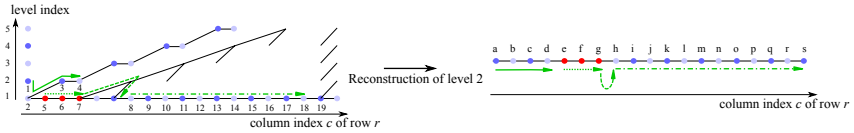


Figure 2. Reconstruction of the level 2 with the toy example of Fig. 1. Green arrows indicates the graph exploration order for view reconstruction.

metrical correlation between the views I_n is only horizontal. We assume that accurate depth images, Z_n, are available at the encoding for every viewpoint, I_n. From the rectification assumption, the link between depth z and disparity d between two camera images is given by $d = \frac{f\delta}{z}$, where δ is the distance between the two cameras. Here, we mainly operate with disparity values, which we compute from the depth maps Z_n and the camera parameters. In what follows we assume there are $N-1$ *predicted* images, which are generated using the *reference* image and additional structure and color information introduced below.

In [9], we categorized the different types of pixels in terms of how they change from one view to another. Because of camera translation, a new part of the scene appears on the right or left of the image (*appearing* pixels) and another part disappears (*disappearing* pixels). During camera translation, foreground objects move faster than the background. As a result, some background pixels may appear behind objects (*disoccluded* pixels). Conversely, some background pixels may become hidden by a foreground object (*occluded* pixels). If we consider a pair of images (reference and target), a row of the target image can be reconstructed by copying pixels from the corresponding row of the reference image, except when the abovementioned types of pixels occur (in which case "new" pixels have to be inserted). Our proposed graph approach directly conveys this information transmitting either i) a link to *the location in reference row* where pixels should be copied from, or ii) *the values of new pixels* to be inserted.

A graph with N levels describes 1 reference image and $N-1$ predicted ones. We show in Fig. 1 a simple graph construction example, with 5 levels (1 reference and 4 predicted images). Its construction requires the depth maps Z_n, $1 \leq n \leq N-1$. Since the object displacement is only horizontal, we consider independent graph construction for each image row[1]. Such a graph is made of two components, which are described by two matrices of size $N \times W$, where N is the number of levels (*i.e.,* the number of images encoded by the graph) and W is the image width. These two matrices are the color values Γ_r and the connections Λ_r and represent color and geometry information for all pixels of all images, where r is the row index (a pair of matrices per row). Γ_r and Λ_r are generated based on the following principles. Pixel intensity values are stored in the level (view) where they appear first. This means that a given level only contains pixels that were not present in a lower level. The connections simply consist of linking these "new" pixels to the position of their neighbor in the previous level.

Refer to the example of Fig. 1. First, the intensities of *appearing* pixels, (a), are stored in Γ_r. No connectivity information is needed, since these pixels appear on the side of the image. *Disoccluded* pixels, (b), do not appear in the lower level, and therefore their intensity is stored in the color matrix Γ_r. A set of consecutive *disoccluded* pixels at level r starts right after a pixel that appeared at level $r-1$. Thus, our graph links the first *disoccluded* pixel at level r to the last copied pixel from level $r-1$ (b). *Occluded* pixels, (c), are pixels at level $r-1$ that are not copied to level r. This situation is represented by links in the graph that go from level $r-1$ to level r and back to level $r-1$ without inserting any pixel values. For example, in Fig. 1 the links between the pixel at position 4 in level 1, through level 2 to the pixel at position 6 in level 1 serves the purpose of "occluding" the pixel at position 5 in the representation at level 2. Finally, *disappearing* pixels, (d), are simply represented by a link (but no pixel intensity value) after the last pixel to be displayed.

To get a more intuitive understanding of the graph representation, refer to Fig. 2, where we show the image of level 2 is reconstructed based on the graph of Fig. 1. By "reconstruction" we mean creating an ouput row containing all pixels value at level 2 based on the sparse graph representation. Reconstruction involves traversing the graph (left to right) and copying pixel values from either level 1 or level 2 to the output, following the links in the graph. In what follows pixel numbering corresponds to their order in the level 2 row output as shown in Fig. 1. The reconstruction starts with the *appearing* pixel 1 at level 2 . Then, it moves to the reference level and copies the corresponding pixels until encountering a link. In the case of Fig. 2, the first connection is after pixel 2 and links it to pixels 3 and 4 in level 2, which are *disoccluded pixels*. After all disoccluded pixels have been copied, the reconstruction goes back to the reference level and copies (5, 6 and 7) until the next non-zero connection (at pixel 7). The connection in 7 indicates an occluded region. Hence, the reconstruction algorithm jumps in the reference frame and restarts the filling (pixel 8 to 19) until the next non-zero connection (disappearing pixel). The reconstruction of the other levels is done recursively. Note

[1]Note that while we construct the graph row by row, compression techniques could be developed that exploit redundancies across rows.

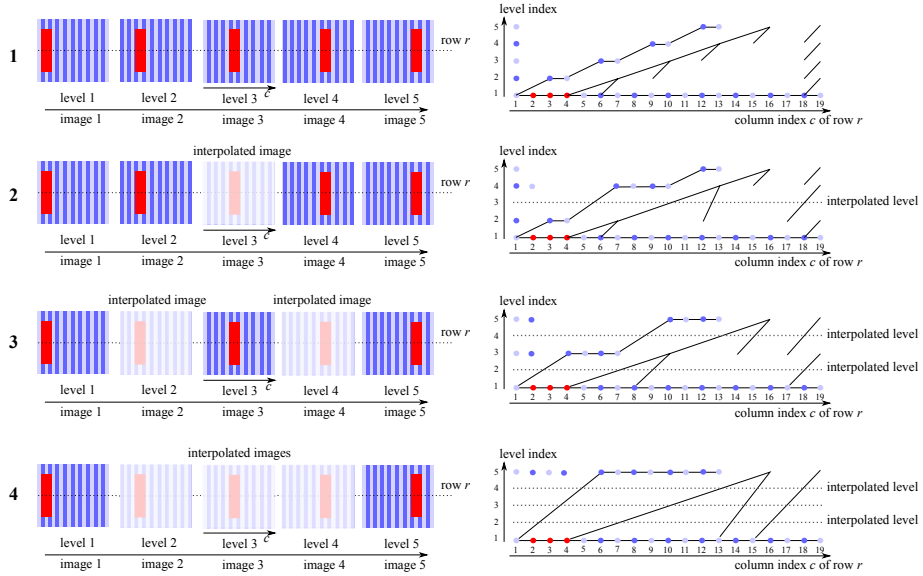


Figure 3. Toy graph compression example, where four different RD points are considered: (1) IPPPP, (2) IPBPP, (3) IBPBP and (4) IBBBP. The graphs connections are coded with an arithmetic coder.

that, in contrast to depth-based representations, our GBR explictly captures the correspondence between levels, making it easier to control the desired level of quality in the representation.

3. COMPRESSION SCHEME

We now introduce a lossy compression scheme for our GBR. The general idea is to remove some images (*i.e.*, levels) from the graph structure and to interpolate them at the receiver. Fewer bits will be required, due to the reduced number of levels, but the interpolation of intermediate images will be less accurate, creating some distortion. In Fig. 3, we provide an illustrative toy example where the graph of Fig. 1 is compressed at four different RD points. The first line is the uncompressed graph version. At a second step, we decrease the size of the graph by removing image 3 from the structure. This image is interpolated at the decoder side. This process can be repeated by removing successively levels 2 and 4 from the graph. The interpolation of a frame at the decoder side is done by disparity-compensating the two closest received images. The two disparity-compensated estimations are then merged resulting to a synthesized frame with no disocclusion. Since the disparity maps are not transmitted in a GBR-based scheme, they are retrieved from the values of the connections in the graph. In other words, the GBR geometry can be used for virtual view synthesis at the decoder, similarly to what can be achieved with depth images. When a level is removed from the graph, the graph links are directly extended (level 2 and 4 connect directly instead of passing through level 3), and the pixel values that were included in the missing level are stored in the upper level and interpolated from those neighboring values in the graph in the levels that are still transmitted. Note that the proposed method is one possible solution to perform lossy coding. Our goal here is to provide further evidence of the benefits of GBR technique (beyond the lossless case of [9]); the study of better coding techniques is left for future work. For example, a scalable GBR representation may be possible, where intermediate levels can be removed without affecting the remaining levels. This would be in contrast with the approach of Fig. 3, where removing pixels level 3, for example, prevents us from using them as a reference for level 4, leading to an error that can propagate to the remaining levels.

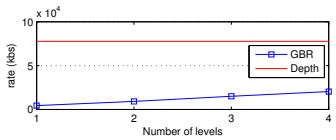
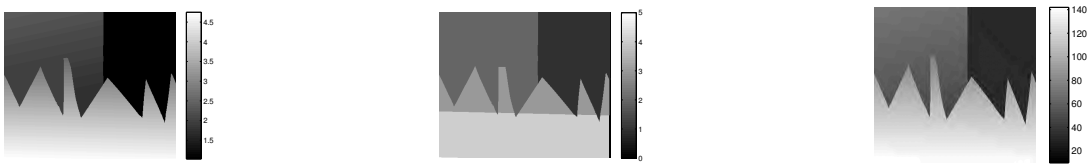


Figure 4. Comparison between depth coding (Lossless JPEG 2000) and GBR for different numbers of levels, for *Sawtooth* sequence. Rate values correspond to geometry (graph or depth) rates for similar decoding quality.

4. EXPERIMENTS

We first evaluate our approach by considering its lossless compression rate for different numbers of levels and comparing it to lossless encoding of depth. For both schemes (depth and GBR), we consider the transmission of one reference image I_1 and N other viewpoints (N is the number of levels). In depth based schemes, these images are predicted by disparity-compensating I_1. In GBR, these N viewpoints are included in the graph representation and predicted at the decoder as detailed in Sec. 2. When we vary the number of levels, we simply vary the number of views (the N first) that are considered at the decoder. Performance is evaluated by measuring the rate required for lossless representation of geometry information (because the representation is lossless there is no distortion). Depth images of viewpoints 1 and N (in order to enable interpolation at the receiver) are compressed using JPEG 2000 [10] since it provides good compression performance for lossless coding. We run the experiments for *Sawtooth* sequence and we show in Fig. 4 the rates obtained. While the depth rate remains constant, the GBR rate increases with N. We see that, for the same level of quality (perfect), GBR requires fewer bits than a depth map representation. This is due to the fact that depth maps provide an unnecessary high level of precision. Moreover, depth approach does not exploit redundancy across depth maps contrary to our GBR technique.



(a) (b) (c)

Figure 5. (a) original depth image of *sawtooth* dataset, (b) the corresponding retrieved disparity from the GBR and (c) the depth image compressed with JPEG 2000 at same bit rate.

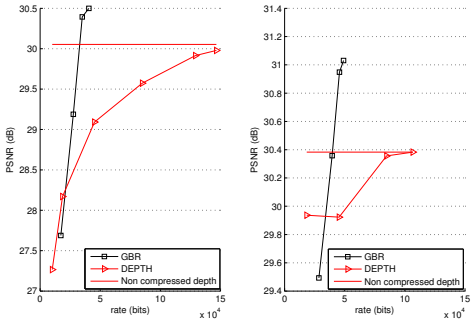


Figure 6. Comparison between depth coding (lossy JPEG 2000) and GBR for different numbers of levels, for *Sawtooth* (left) and *Venus* (right) sequences. Rate values correspond to geometry (graph or depth) rates and PSNR are calculated between reconstructed and original frames.

We consider now lossy compression of both schemes. The dataset is made of 5 views (and their associated depth images) that need to be reconstructed at the decoder. On the one hand, in the depth-based approach, we consider that I_1 and I_5 are transmitted as intra frames, and $((I_i)_{1<i<5})$ are interpolated at the decoder side. We recall that in this work we are interested only in the compression of the geometry structure, so we evaluate the rate of the depth images Z_1 and Z_5, compressed using JPEG 2000 [10] at different QP values. For distortion evaluation, we calculate the quality of the interpolated frames $(\hat{I}_2, \hat{I}_3, \hat{I}_4)$ with respect to the original images. On the other hand, the four rate-distortion points of GBR methods are obtained by varying the number of frames involved in the representation (as explained in Sec. 3). In the following, I corresponds to the reference level, P to a predicted level included in the graph representation and B an interpolated image not considered in the GBR. We build the following scenarios (from high to low bitrate): $IPPPP$, $IPBPP$, $IBPBP$, $IBBBP$ as illustrated in Fig. 3. The B frames are interpolated using disparity maps that are retrieved from the GBR structure. We see an example in Fig. 5 (b) of an estimated disparity map for *Sawtooth* dataset. Comparing to the corresponding depth image (Fig. 5 (a)), we see that the disparity values are well estimated from the simple connections of the GBR. We also notice that depth image compressed with JPEG2000 at equivalent bit rate contains some artifacts. We show in Fig. 6 the RD behavior of both GBR and depth-based schemes for *Sawtooth* and *Venus* sequences. We see that GBR method outperforms the depth-based representation. This result first shows that GBR approach requires less rate than depth maps. In addition, while the loss in depth approach is brought by QP variation, in our method, it is controlled by the number of frames taken into account in the representation. The fact that GBR leads to a better control of the errors is also induced by the compression strategy.

5. CONCLUSION

In this paper, we design a practical lossy compression scheme for GBR by removing some images from the representation and interpolating them at the decoder side. The interest of this work is twofold. First, we confirm the potential of this representation by showing that GBR is a more compact description than depth images and better control the compression artifacts. Secondly, we show that GBR has some of the same advantages as depth, including the capacity of generating virtual views at the decoder side.

6. REFERENCES

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

[2] C. Zhang and T. Chen, "A survey on image-based renderingrepresentation, sampling and compression," *EURASIP J. on Sign. Proc.: Image Commun.*, vol. 19, pp. 1–28, 2004.

[3] N. Stefanoski, P. Klie, X Liu, , and J. Ostermann, "Layered coding of time-consistent dynamic 3-d meshes using a nonlinear predictor," in *Proc. IEEE Int. Conf. on Image Processing*, San Antonio, TX, US, 2007.

[4] G. Alenya and C. Torras, "Lock-in time-of-flight (tof) cameras: A survey," *IEEE Sensors Journal*, vol. 11, pp. 1917–1926, 2011.

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

[6] G. Cheung, A. Ortega, W.S. Kim, V. Velisavljevic, and A. Kubota, *3D-TV System with Depth-Image-Based Rendering: Architectures, Techniques and Challenges, Ce Zhu and Yin Zhao and Lu Yu and Masayuki Tanimoto, Eds.*, chapter Depth Map Compression for Depth-Image-Based Rendering, Springer, 2013.

[7] WS Kim, A. Ortega, P. Lai, Tian D, and C Gomila, "Depth map coding with distortion estimation of rendered views," in *Proc. of SPIE, the Int. Soc. for Optical Engineering*, 2010.

[8] B. Rajei, T. Maugey, and P. Frossard, "Rate-distortion analysis of multiview coding in a DIBR framework," *submitted to Annals of Telecommunications*, 2012.

[9] T. Maugey, A. Ortega, and P. Frossard, "Graph-based representation and coding of multiview geometry," in *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, Vancouver, Canada, 2013.

[10] JPEG-2000, "ISO/IEC FCD 15444-1: JPEG 2000 final comitee draft version 1.0," 2000.