Graph-Based vs Depth-Based Data Representation for Multiview Images

Thomas Maugey*, Antonio Ortega†, Pascal Frossard*
*Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL)
Email: {thomas.maugey,pascal.frossard}@epfl.ch
†Department of Electrical Engineering, University of Southern California
Email: antonio.ortega@sipi.usc.edu

***Abstract*—In this paper, we propose a representation and coding method for multiview images. As an alternative to depth-based schemes, we propose a representation that captures the geometry and the dependencies between pixels in different views in the form of connections in a graph. In our approach it is possible to perform compression of the geometry information and to preserve a direct control of the effect of geometry approximation on view reconstruction. This is not possible with classical depth-based representations. As a results, our method leads to more accurate view prediction, when compared to conventional lossy coding of depth maps operating at the same bit rate. We finally show in experiments that our representation adapts the amount of transmitted geometry to the complexity of the predictions that are performed at the decoder.**

I. Introduction

Multiview or 3D data are generally represented as a set of images that correspond to the information captured by cameras at different viewpoints. These images describe the color information acquired by the multiple cameras, along with depth information that becomes easily accessible [1]. Depth images describe the distance between the scene and the focal length of the camera. Obviously this brings interesting challenges for 3D transmission systems. For example, a depth image can be used to project one reference image onto another one [2], [3] with interesting benefits in the compression of multiview images. Despite the huge potential of this tool, one of the important questions linked with depth images relies in the effect of compression on the view prediction performance [4]. More precisely, an imprecise depth value leads to a spatial position uncertainty for the projected pixels in the predicted viewpoint. The modeling of this error has led to several works [5], [6]. Some sophisticated depth image coder has also been proposed [7], [8] to tackle this drawback. However, artifacts due to compressed depth images remain generally difficult to control. This is why, in this paper, we propose a new multiview image representation that permits better control of the geometry information.

We propose a natural form of geometry information that is of moderate size, but leads to effective view reconstruction algorithms. After observing that the knowledge of the scene geometry leads to connections between pixels in images from different viewpoints, we propose to directly represent these links with a graph. The graph contains all the geometrical information needed for multiview image reconstruction at the decoder side. Contrary to depth maps, the geometry information in our graph takes into account the complexity of the prediction when adjusting the proper amount of geometry to compress and transmit. The advantage of such an approach is that it works directly with the inter-pixel connections and offers a better control of the geometry compression artifacts. We have compared this approach with depth map representations in terms of view prediction quality for similar geometry rate budget and shown promising with performance improvements of 4 dB.

The paper is organized as follows. In Section II, we introduce the general concepts of our new representation along with the main differences with depth-based approaches. In Section III, we explain the construction of the graph in detail and finally, in Section IV, we validate the potential of our solution in terms of prediction quality improvement.

II. Dense disparity maps

As mentioned in the introduction, one of the most adopted format to represent and transmit N viewpoints is the MVD one. It consists of N color image and their associated N depth maps. A depth map is a gray-scale 2D image that represents the distance between the scene and the camera plane. Since it takes values between 0 and 255 as a classical image, it is associated to a scaling function that converts the chrominance values into units denoting the distance to objects. This scaling function is generally not linear but follows a evolution in $\frac{1}{z}$ in order to describe more finely the closest points than the furthest objects in the scene. It is justified by the fact that the disparity of a point in the scene between two images evolves in function of the inverse of the depth. Depth images are mainly used to project the corresponding color image onto other arbitrary viewpoints. This is generally done with *depth image based rendering (DIBR)* techniques [2], as illustrated in Fig. 1. Depth is firstly used to find the correspondence between a pixel of the image and points of the 3D world, based on the intrinsic and extrinsic parameters of the cameras. Then, the 3D point is projected back onto another viewpoint.

When lossy compression of the depth data is performed, the DIBR decreases its prediction precision. More precisely, one pixel of the reference image is mapped to a segment of several pixels in the predicted image (represented by Δ in Fig. 1). Due to this imprecise mapping, the error in DIBR is difficult to control, which leads to complex compression algorithms [7], [9], [10]. Depth maps have generally a high precision (256 levels) which is not always necessary, as in the case of simple prediction for example. When a simple lossy coding of depth is performed, one preserves an unnecessary bit

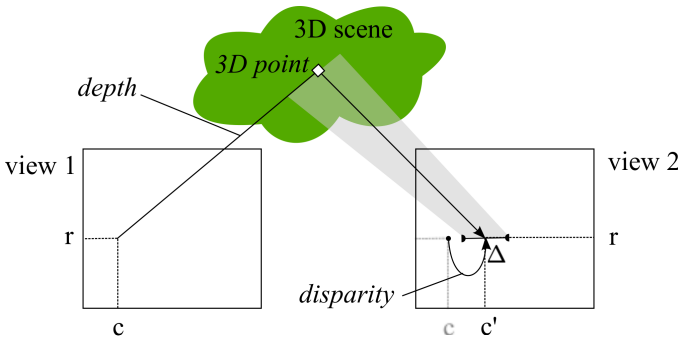


Fig. 1. Illustration of the difference between depth-based prediction and disparity compensation.

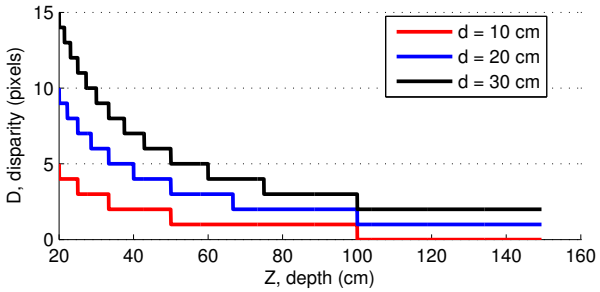


Fig. 2. Integer disparity as a function of depth for different distances between the reference and predicted camera.

depth precision while it introduces losses in crucial regions. In that sense, depth-based representation does not enable a natural control of geometry compression error.

In our GBR representation, we therefore propose another description of geometry information, which constitutes a simpler version of depth and which is eventually losslessly coded. In that way, this geometry signal corresponds to the minimum information that is needed at the decoder for view prediction. During view prediction, one pixel of the reference frame is shifted of D pixels in the predicted viewpoint, with:

$$D = \frac{f.d}{Z}, \tag{1}$$

where f is the focal length, d the distance between the two cameras and Z the depth attached to this pixel. In Fig. 2, we plot the integer disparity $|D|$ as a function of the depth Z which takes 256 values between $Z = 20$ cm and $Z = 150$ cm, for different distances between the reference image and the predicted camera. We clearly see that the disparity requires less precision than the original depth signals. We also remark that the number of levels increases with the inter-camera distance d. In other words, the complexity of the geometry signal that needs to be transmitted varies with the position of the predicted view. Moreover, we remark that the number of disparity levels is larger for small depth values, which also means that the geometry signal complexity depends on the scene content. In our GBR representation, we propose a compact representation of the disparity values, which permit to reconstruct a predefined set of views. Contrary to traditional multiview coders, which transmit one disparity value for blocks of pixels, our solution considers dense disparity maps. The GBR representation is detailed in the next section.

III. Graph-based geometry representation

We recall here the main ideas of GBR construction process and the view reconstruction at the decoder. Readers are referred to [11] 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 geometrical 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. As explained above, we compute $N - 1$ dense disparity maps from these depth images. In what follows we assume that there are $N - 1$ *predicted* images, which are generated using the *reference* image along with structure and color information introduced below.

We categorize 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 graph approach directly conveys this information by 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. 3 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 links these "new" pixels to the position of their neighbor in the previous level.

We look now in more details at Fig. 3. First, the intensities of *appearing* pixels, (a), are stored in Γ_r. No connectivity information is needed, since these pixels appear on the side

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

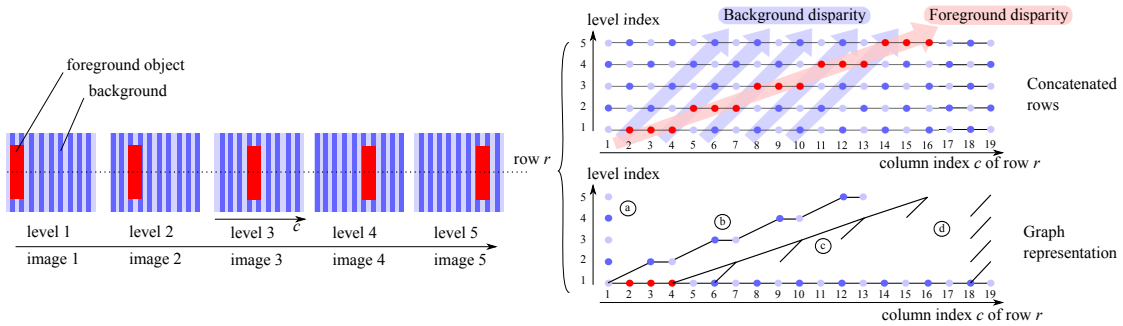


Fig. 3. Toy graph construction example: blue texture background has a disparity of 1 at each level and red rectangle foreground a disparity of 3 for each level. Graph contains all different types of pixels: a) appearing, b) disoccluded, c) occluded and d) disappearing.

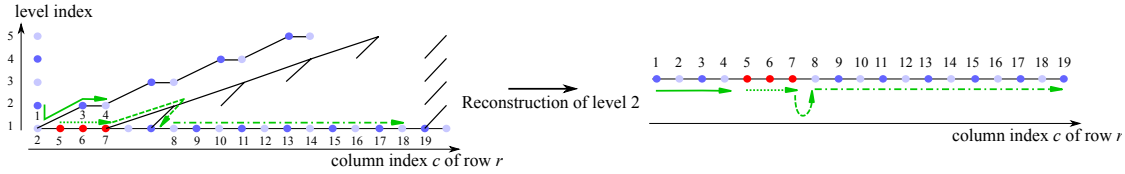


Fig. 4. Reconstruction of the level 2 with the toy example of Fig. 3. Green arrows indicates the graph exploration order for view reconstruction.

of the image. *Disoccluded* pixels, (b), do not appear in the lower level, and 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 appears 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. 3 the links between the pixel at position 4 in level 1, through level 2 to the pixel at position 6 in level 1 represent the occlusion of 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 different view on the graph representation, refer to Fig. 4, where we show the image of level 2 that is reconstructed based on the graph of Fig. 3. By "reconstruction" we mean creating an output row containing all pixel values 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 reconstructed level 2 shown in Fig. 4. 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. 4, 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 process (pixel 8 to 19) until the next non-zero connection (disappearing pixel). The reconstruction of the other levels is done recursively. Note that, in contrast to depth-based representations, our GBR explicitly captures the correspondence between levels, making it easier to control the desired level of quality in the representation.

IV. Experiment

In this section, we show that the geometry sent with our GBR representation corresponds to the proper level of precision that is needed at the decoder side for inter-view prediction, contrary to depth-based scheme. More precisely, we show that GBR offers a higher control of geometry loss impact on the reconstructed quality, compared to depth representation. In the test presented below, we code the geometry while the texture signal is not compressed.

We have implemented a prototype coding scheme for our GBR solution. As we can observe in the example of Fig. 3, the matrix Λ_r has a large number of zero values. We do not code directly Λ_r and rather consider a smaller matrix Φ of size $M \times 4$, where M is the number of non-zero elements in all the Λ_r with $r < H$ (H is the height of the image). The matrix Φ stores all the meaningful connections and it is organized as follows. The first column of Φ contains this row indices r. The second column contains the column indices c, the third column contains the level indices, and finally, the fourth column contains the connection values. Then, we simply consider an arithmetic coding of every column, with, for some of them, a differential operation as preprocessing, in order to decrease the entropy. Alternatively, the depth maps are encoded using the classical image coding tool JPEG2000 [12]. They are used for view prediction at decoder, using DIBR algorithm. As for GBR, no correction is sent since we only consider goemetry compression in these tests.

In Fig. 5, we show the prediction error images using depth or GBR as geometry information. We only consider 2 views in this experiment. First, we build our GBR representation derived from the dense disparity values computed with the depth images. These disparity map is designed for the prediction

(a) Prediction error with lossy depth (27.2 dB)

(b) Prediction error with GBR information (31.8 dB)

Fig. 5. Prediction error images for *sawtooth* dataset using depth (a) and GBR (b). Coding rate of 15.1 kb for geometry rate. PSNR is calculated on the non occluded regions of the image.

of view 2. The geometry size after the coding of the GBR data is 15.1 kb. We then encode the depth map with the same bitrate. For both decoded geometry information we perform the prediction of view 2, we calculate the error image and we evaluate the PSNR on the non occluded regions. We see in Fig. 5 that the GBR-based prediction has error only on the disoccluded regions, while depth coding introduces artifacts also on predicted regions.

Next, we show that GBR adapts the complexity of its geometry signal to the one of the prediction process. We still consider lossless texture and only two views. For different distances between the two views (1 and 2 times the intra-ocular distance), we run the following test. We first build the GBR representation and we obtain a given geometry rate R in bits. Then, we compress the depth map of view 1 with the same rate. We then observe the decoded geometry information. They are presented for *Venus* and *Sawtooth* datasets in Fig. 6. We also show the prediction PSNR values calculated on the non occluded regions. We see that when the distance increases, the GBR provides higher precision. More precisely the foreground objects in (b) and (h) are described with more different values in respectively (e) and (k). At the same time, the depth signal keeps useless precision. More precisely, we see that the right foreground in (c) is represented with several different depth values while one would be enough as it is the case in (b). Moreover, some losses has been introduced in the depth-based coding scheme, for example around the foreground boundaries. In other words, the GBR representation permits to transmit the exact level of geometry required at the decoder side in contrary to depth-based scheme, which leads to more accurate view prediction as shown by the PSNR gain observed for these datasets. The previous experiments show the validity of our solution and prove that GBR constitutes a serious alternative to depth-based data representation schemes in multiview imaging.

V. Conclusion

In this paper, we have presented our new graph-based representation used to describe geometry and texture information of multiview images. In addition to removing spatial redundancies in the data, it provides an intuitive graph structure that permits to efficiently represent the geometry signal. This leads to a better control of inaccuracies due to geometry compression, and their impact on the multiview reconstruction quality, compared to depth-based approach.

References

[1] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, pp. 4–10, 2012.

[2] C. Fehn, "Depth-image-based rendering (dibr), compression and transmission for a new approach on 3d-tv," *Proc. SPIE, Stereoscopic Image Process. Render.*, vol. 5291, pp. 93–104, 2004.

[3] F. Shao, G. Jiang, M. Yu, and Y. Zhang, "Object-based depth image-based rendering for a three-dimensional video system by color-correction optimization," *Opt. Eng.*, vol. 50, pp. 047 006–047 006–10, 2011.

[4] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. M uller, P. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," in *3D TV Conference*, Istanbul, Turkey, May 2008.

[5] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3D video coding," *IEEE Trans. on Circ. and Syst. for Video Technology*, vol. 21, pp. 485–497, 2011.

[6] B. Rajei, T. Maugey, and P. Frossard, "Rate-distortion analysis of multiview coding in a DIBR framework," *Annals of Telecommunications*, 2013.

[7] G. Cheung, W. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsication," in *IEEE Int. Workshop on Multimedia Sig. Proc.*, Hangzhou, China, Oct. 2011.

[8] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video coding," in *Proc. IEEE Int. Conf. on Image Processing*, Orlando, FL, USA, Sep. 2012.

[9] S. Kim and Y. Ho, "Mesh-based depth coding for 3d video using hierarchical decomposition of depth maps," in *Proc. IEEE Int. Conf. on Image Processing*, San Antonio, TX, USA, Sep. 2007.

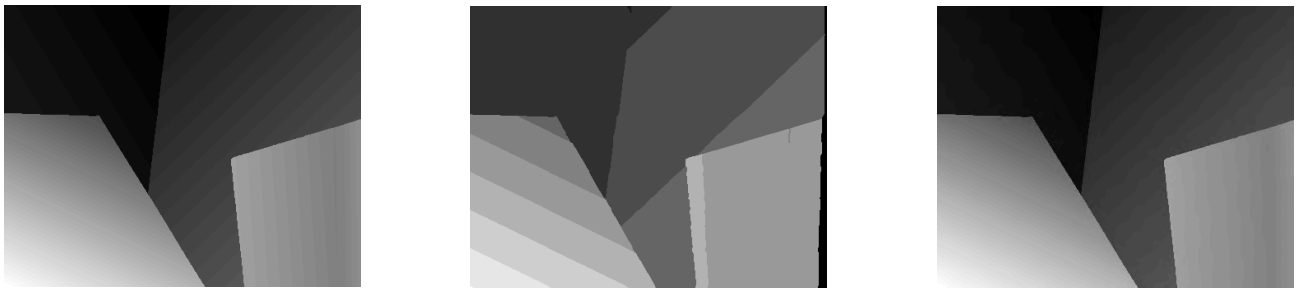
[10] W. Kim, A. Ortega, P. Lai, T. D, and C. Gomila, "Depth map coding with distortion estimation of rendered views," in *Proc. of SPIE, the Int. Soc. for Optical Engineering*, 2010.

[11] 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.

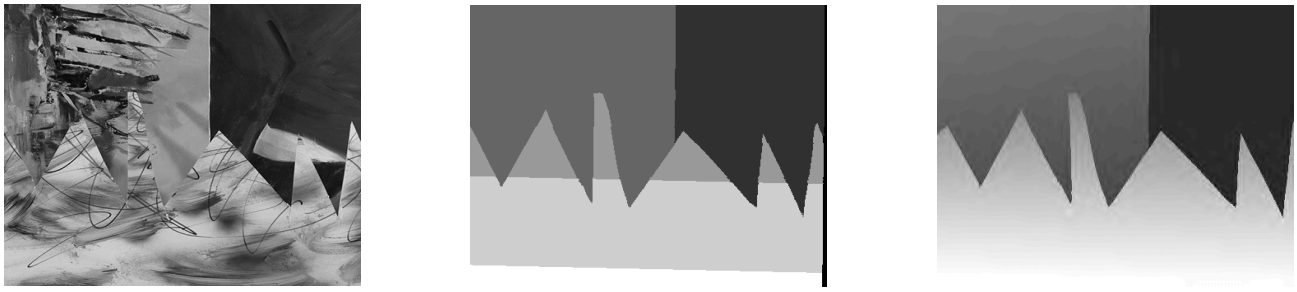
[12] JPEG-2000, "ISO/IEC FCD 15444-1: JPEG 2000 final comitee draft version 1.0," 2000. [Online]. Available: http://www.jpeg.org/FCD15444-1.htm

Original data | **GBR disparity map** | **Corresponding compressed depth**

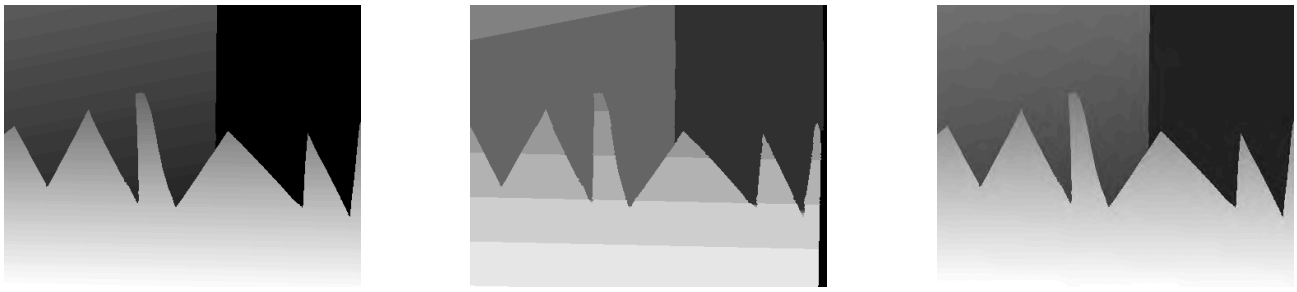
(a) texture image (b) $d = 1$, $R = 15.0$kb, 32.2 dB (c) $d = 1$, $R = 15.0$kb, 28.3 dB



(d) depth map (e) $d = 2$, $R = 22.5$kb, 32.0 dB (f) $d = 2$, $R = 22.5$kb, 26.9 dB



(g) texture image (h) $d = 1$, $R = 15.1$kb, 31.8 dB (i) $d = 1$, $R = 15.1$kb, 27.2 dB



(j) depth map (k) $d = 2$, $R = 23.8$kb, 27.9 dB (l) $d = 2$, $R = 23.8$kb, 23.9 dB

Fig. 6. Illustration on *Venus* (a-f) and *Sawtooth* (g-l) multiview datasets of how the inter-view distance (d) impacts on the geometry signals of both the GBR representation (b,e,h,k) and the depth-based approach (c,f,i,l). This distance is expressed in view index (*i.e.*, multiple of the inter-camera distance) and corresponds to 1 or 2 times the intra-ocular distance (6.5 cm). We expressed for every geometry map its rate R and the PSNR of the prediction using it (on the non occluded regions).