COMPRESSION OF PLENOPTIC POINT CLOUDS USING THE REGION-ADAPTIVE HIERARCHICAL TRANSFORM

Gustavo Sandri[*] — University of Brasília, Department of Electrical Engineering, Brasília, Brazil

Ricardo de Queiroz[*] — University of Brasília, Department of Computer Science, Brasília, Brazil

Philip A. Chou — 8i Labs, Los Angeles, United States

ABSTRACT

Point clouds have recently gained interest for the represention of 3D scenes in augmented and virtual reality. In real-time applications point clouds typically assume one color per point. While this approach is suited to represent diffuse objects, it is less realistic with specular surfaces. We consider the compression of plenoptic point clouds, wherein each voxel is associated to colors as seen by different angles. We propose an efficiently compressible representation to incorporate the plenoptic information of each voxel. We have proposed three compression methods, one based on a cylindrical projection and two others based on the intersection of the line of view with the voxel's face, one using flat boundaries and the other using a spherical boundary. Extensive tests have shown that the last two have the best performance, which are much superior than independently encoding the color attributes from each of the cameras point of views.

Index Terms— point cloud, plenoptic, compression, augmented reality, virtual reality.

1. INTRODUCTION

The region-adaptive hierarchical transform (RAHT) [1] is an algorithm for compression of voxelized point clouds (PCs) with a quality comparable to coders based on the Graph Transform [2] and Gaussian Process Model [3], at a fraction of their complexity. An occupied voxel is associated to a color attribute (in RGB or YUV space). When rendering a view of the scene, these voxels act as a source of light, emitting the same color in all directions. The representation of a scene by single color voxels might not be realistic for specular surfaces where the color of a given point varies according to the viewing angle. The extreme case is a mirror, which reflects its surroundings. A more realistic representation should allow a voxel to change its color according to the viewing angle. For that, we need to attribute to a voxel the color as seen in a plurality of directions. We refer to this data as the plenoptic information, as it is based on the plenoptic function representing a scene.

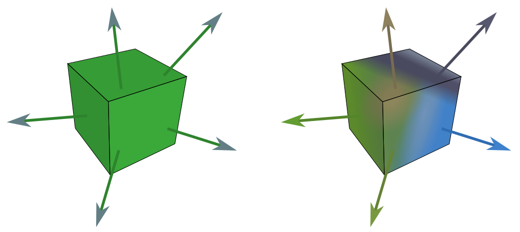
[*]This work was partially funded by CAPES and CNPQ under grant 308150/2014-7.

The 5-dimensional plenoptic function represents the chromaticity of light observed from every position and direction in a 3-dimensional (3D) space [4] as

$$P(x, y, z, \theta, \phi), \tag{1}$$

where (x, y, z) are the coordinates of a point in space, θ the azimuth and ϕ the elevation angle. The plenoptic information of a voxel is obtained by fixing (x, y, z) at the voxel position and letting θ and ϕ vary according to the viewing angle.

Plenoptic PC can be produced by processing the information captured by an array of cameras combined with depth maps [5, 6], or from light-field cameras [7]. In this fashion, the number of sampled viewing directions is determined by the number of cameras employed and the plenoptic information is derived from the colors as seen by each of the cameras. Hence, it is more practical to encode the colors from each camera (sample) instead of encoding the continuous function covering all (θ, ϕ). In this sense, the information is a vector of color components per voxel.



(a) Non-plenoptic voxel (b) Plenoptic voxel

Fig. 1. A non-plenoptic voxel has no directional color information. Such information is present in a plenoptic voxel and can be used to represent a scene in a more realistic way.

In this work, we propose to incorporate the sampled plenoptic information into each voxel by two methods: subdividing the voxel into subvoxels were the subvoxels position represents the cameras displacement and by using a projection map of the cameras displacement. We assume that

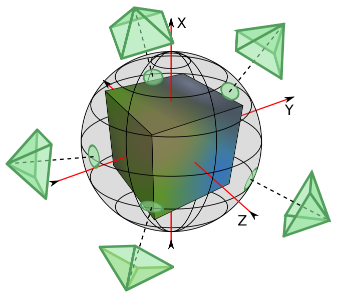


Fig. 2. Capture of the plenoptic information of a voxel.

both the encoder and decoder know the original geometry of the PC and camera displacement (encoded with another algorithm) and we focus only in color compression. It contrasts to other methods such as those using surface light field representation [8, 9].

2. VOXEL SUBDIVISION

Consider the voxel in Fig. 2, whose colors are captured by five cameras placed on the depicted directions.

The sampled plenoptic information comprises not only the color, but also the direction of the cameras. If we divide the voxel into M partitions along each axis ($M = 4$ in Fig. 3), we obtain M^3 cubes with $1/M$ of the original width. Each of these cubes resulting from the division resemble voxels and we refer to it as subvoxel. We will show that the plenoptic information can be associated to subvoxels by means of the subvoxel position.

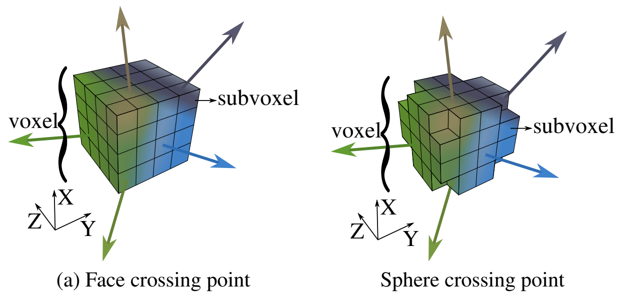


Fig. 3. A voxel is divided into subvoxels and its subvoxels are employed to incorporated the plenoptic information.

After attributing the plenoptic information to the subvoxels, we can now apply RAHT-based coder [1] to the cloud of subvoxels. This process is transparent for RAHT because it treats the subvoxels as voxels.

In order to associate the viewing direction and color to the subvoxel position we devised two different methods. For the first one, named "face crossing point", the line connecting the voxel center and a camera, simply referred as viewing line, can be used to represent the viewing direction. This line crosses one of the subvoxels at the voxel's faces. Hence, in this method, the direction is represented by indicating the subvoxel position on the voxel's faces crossed by the given viewing line. The color as viewed in that direction is associated to this subvoxel (see Fig. 3-(a)). All subvoxels that were not crossed by any viewing line remain unoccupied, as well as all the subvoxels not belonging to any of the voxel's faces.

We can improve the face crossing point method by, instead of using the subvoxels on the voxel's face, using those that are crossed by a sphere surface tangent to the voxel's faces. In this fashion, the occupied subvoxels will be distributed in a spherical-like way, instead of a cube-like way, thus avoiding the distortions near the voxel's corner (see Fig. 3-(b)). This method is named "sphere crossing point".

3. CYLINDRICAL PROJECTION

A third method that we propose to represent the plenoptic information is by means of a projection map. The direction of the cameras relative to each voxel can be described in cylindrical coordinates by the azimuth angle $-\pi \leq \theta < \pi$ and the elevation $-1 \leq h \leq 1$, resulting in a $\theta \times h$ plane.

One may divide the camera directions ($\theta \times h$) plane into sub-regions of equal area as depicted in Fig. 4.

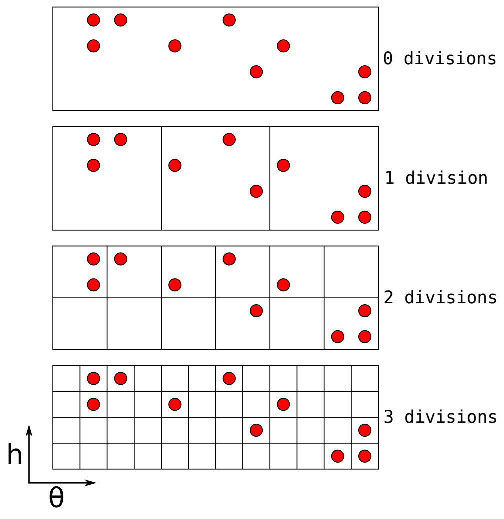


Fig. 4. Subdivision of the $\theta \times h$ plane

We may further divide each sub-region several times until attaining the desired precision. The smaller the sub-region, the more precise the camera position is represented. After dividing the plane, several sub-regions remain unoccupied. This representation is similar to voxelized point clouds in the 3D space. Therefore, we apply the RAHT-based coder to the colors associated with each camera (sub-region), through a 2D quad-tree decomposition rather than the 3D octree.

The RAHT results in several high-frequency components and one DC value. The resulting DC value for each $\theta \times h$ plane represents the average voxel color as seen by all cameras. This DC value is then associated to each voxel and we apply the RAHT-based coder to all voxels in their spacial (x, y, z) positions.

4. EXPERIMENTS

We carried tests on 5 realistic real-time-captured scenes. They were recorded with up to 13 cameras and around 3 million points (see Table 1 and Fig. 5). These images were voxelized using 11 bits of spatial resolution (octree with a depth level $L = 11$), resulting in around 2 million voxels. For the subdivision of the voxels into subvoxels was chosen $M = 2^6$. The $\theta \times h$ plane was divided 6 times.

Table 1. Database

	Number of	
Image	**Occupied voxels**	**Cameras**
boxer	2056256	13
longdress	1860104	12
loot	1858707	13
redandblack	1467981	12
soldier	2365732	13

Fig. 5. Rendered Images. Point clouds are a courtesy of 8i®.

Colors are represented in the RGB space. In our experiment, the quantization step was varied between 15 and 500. We compared our methods to RAHT applied independently to each camera, simply refered as 'independent'. The results in terms of rate-distortion (RD) curves are shown in Fig. 6 and 7 where the PSNR was calculated using all color components.

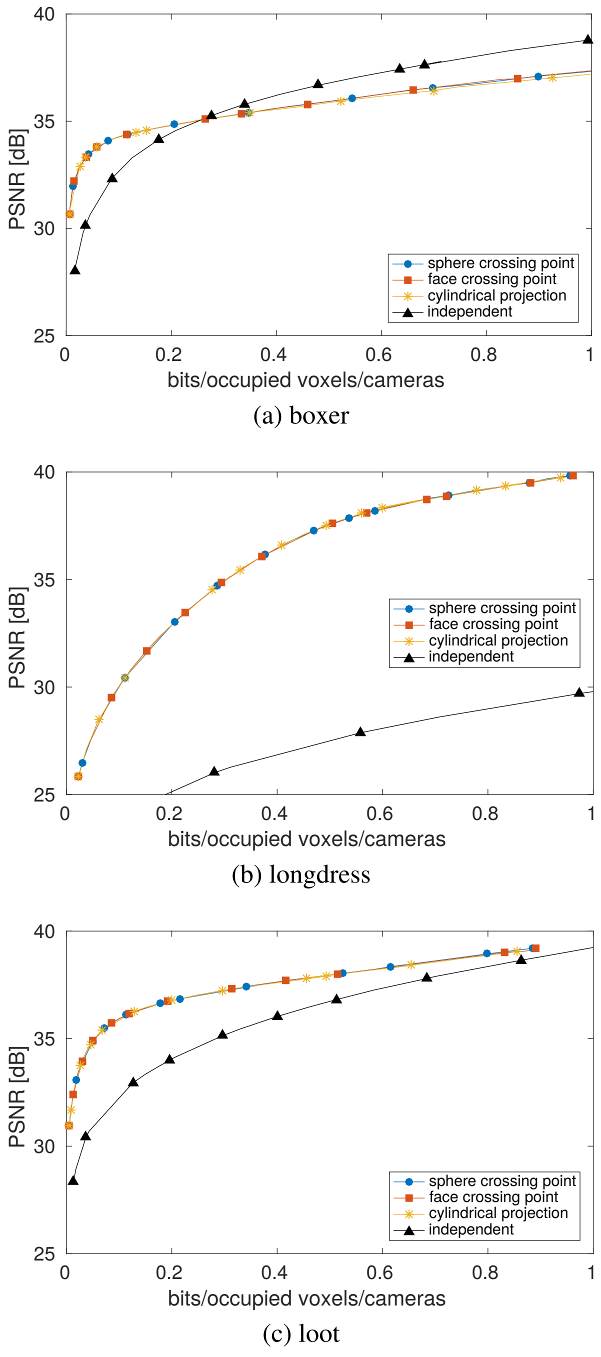
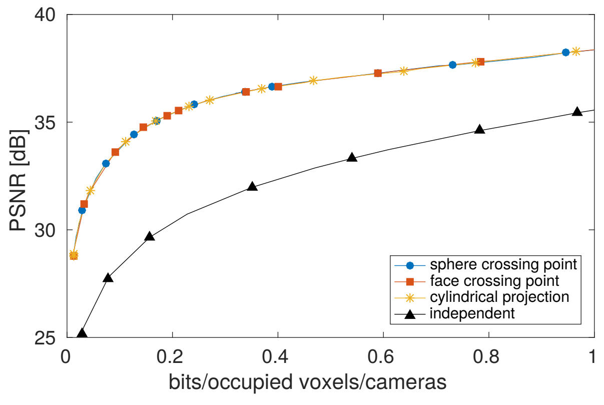
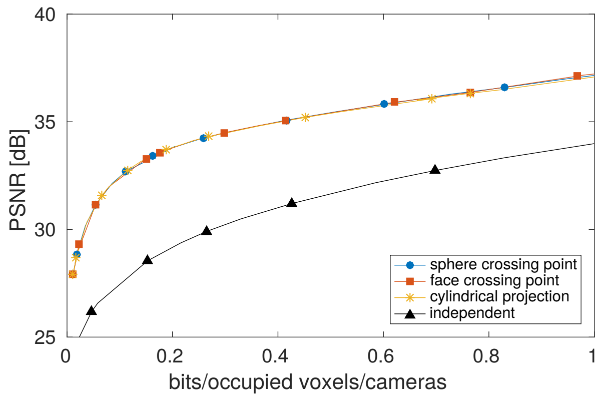


Fig. 6. Rate-distortion curves for the point clouds "boxer", "longdress" and "loot".

From Fig. 6 and 7 we can observe that all three methods perform similarly, the cylindrical projection method being slightly worst. This can be more clearly seen in Fig. 8, which shows the PSNR difference between the methods when fix-

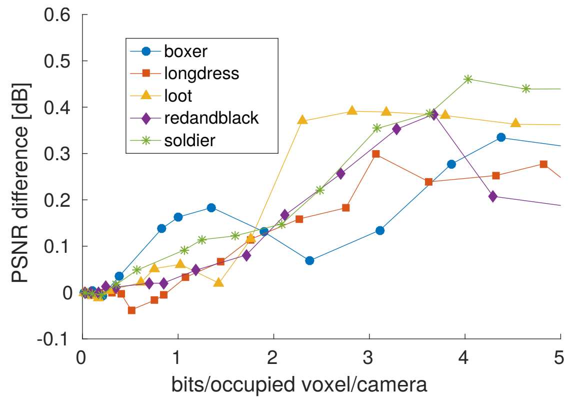


(d) redandblack

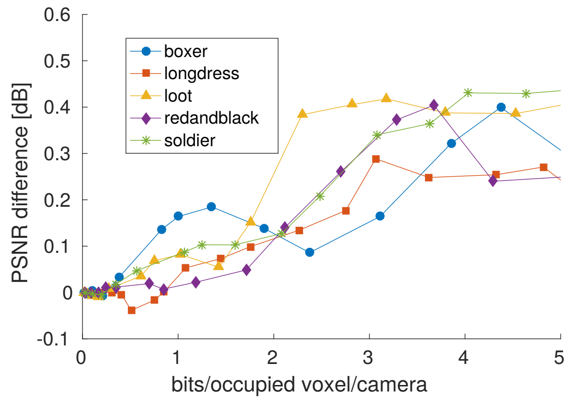


(e) soldier

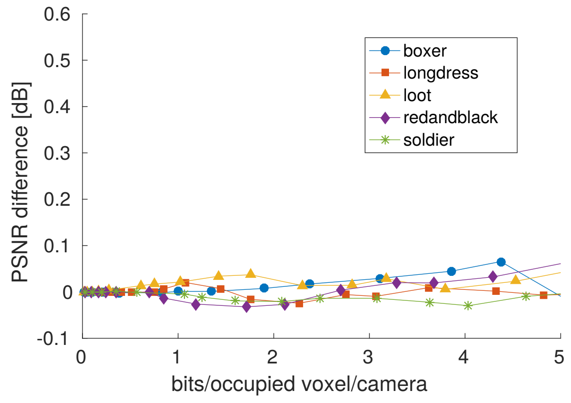
Fig. 7. Rate-distortion curves for the point clouds “redandblack” and “soldier”.



(a) face crossing point - cylindrical projection



(b) sphere crossing point - cylindrical projection



(c) face crossing point - sphere crossing point

Fig. 8. Difference of PSNR between methods for the same rate. We can see that the cylindrical projection performs slightly worse than the face crossing point and the sphere crossing point, while the latter two have virtually identical performance

ing the rate. In Fig. 8-(a) and (b) we observe that both the face and sphere crossing point methods outperform the cylindrical projection and the difference is higher for higher rate values. Face and sphere crossing point methods, on the other hand, have very similar curves (see Fig. 8-(c)), presenting a virtually identical performance.

The methods presented here were able to outperform RAHT when applied independently to each camera information, with the exception of the PC “boxer” at rates higher than 0.2 bits/occupied voxels/cameras.

5. CONCLUSIONS

In this work we proposed three methods to incorporate the plenoptic information of a voxel: cylindrical projection; face crossing point and sphere crossing point. Their performance were very similar, the two latter being slightly better for higher rate values.

These methods are compliant with any single-color point cloud compression algorithms. In this work, we employed the RAHT to encode the color of the PC as it is a low-cost high-performance algorithm. Nevertheless, it is easy to readapt them to other compression algorithms.

The results were compared to RAHT individually applied to each camera . We observed that the proposed modifications largely improved the compression.

6. REFERENCES

[1] R. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947–3956, 2016.

[2] C. Zhang, D. Florêncio, and C. Loop, "Point cloud attribute compression with graph transform," in *IEEE International Conf. Image Process. (ICIP)*, pp. 2066–2070, 2014.

[3] R. de Queiroz and P. A. Chou, "Transform coding for point clouds using a Gaussian Process Model," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3507–3517, 2017.

[4] E. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," *Comput. Models of Visual Process.*, pp. 3–20, 1991.

[5] S. Orts-Escolano *et al.*, "Holoportation: Virtual 3d teleportation in real-time," in *Proc. of Annual Symp. User Interf. Soft. and Tech. (UIST)*, pp. 741–754, 2016.

[6] A. P.-Miro, J. R.-Hidalgo, and J. R. Casas, "Registration of images to unorganized 3D point clouds using contour cues," in *European Signal Process. Conf. (EUSIPCO)*, pp. 81–85, 2017.

[7] C. Perra, F. Murgia, and D. Giusto, "An analysis of 3D point cloud reconstruction from light field images," in *International Conf. Image Process. Theory, Tools and Applications (IPTA)*, pp. 1–6, 2016.

[8] D. N. Wood *et al.*, "Surface light fields for 3D photography," in *Proc. Annual Conf. Computer Graphics and Interactive Techniques*, pp. 287–296, 2000.

[9] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," *ACM Trans. on Graphics*, vol. 21, no. 3, pp. 447–456, 2002.

[10] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Eurographics / IEEE VGTC Conf. Point-Based Graphics*, pp. 111–121, 2006.

[11] Y. Huang, J. Peng, C.-C. Jay Kuo, and M. Gopi, "A generic scheme for progressive point cloud coding," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 2, pp. 440–453, 2008.

[12] T. Ochotta and D. Saupe, "Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields," in *Proc. Eurographics Conf. on Point-Based Graphics*, pp. 103–112, 2004.

[13] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based motion estimation and compensation for dynamic 3D point cloud compression," in *IEEE International Conf. Image Process. (ICIP)*, pp. 3235–3239, 2015.

[14] J. Kammerl *et al.*, "Real-time compression of point cloud streams," in *IEEE International Conf. Robotics and Automation (ICRA)*, pp. 778–785, 2012.