Saliency Maps for Point Clouds

Victor F. Figueiredo
Electrical Engineering Department
University of Brasília
Brasília, Brazil
fabre@ieee.org

Gustavo L. Sandri
Eletronics Department
Federal Intitute of Brasília
Brasília, Brazil
gustavo.sandri@ieee.org

Ricardo L. de Queiroz
Computer Science Department
University of Brasília
Brasília, Brazil
queiroz@ieee.org

Philip A. Chou
Google Inc.
Seattle, WA USA
pachou@ieee.org

***Abstract*—Algorithms for creating saliency maps are well established for images, even though there is no literature on such methods for point clouds. We use orthographic projections in 2D planes which are subject to well established saliency detection algorithms to create a 3D saliency map. The results of each saliency map are projected to the 3D voxels and the results of the many projections are used to generate a 3D saliency map. Simple compression tests were carried using soft region-of-interest maps. Results have shown an increase in the quality of the voxels inside the selective regions of increased levels of interest.**

***Index Terms*—Saliency map, point cloud, RAHT.**

I. Introduction

The proliferation of computational imaging for 3D detection and the increase of 3D applications such as autonomous navigation and augmented reality made point clouds (PC) increasingly important [1]. A point cloud is a set of points in space represented in a three-dimensional (X, Y, Z) coordinate system. It commonly serves the purpose of representing the outer surface of an object or scene. It is represented by its geometry and attributes [2]. The geometry part of a point cloud is described by a set V with the coordinates of all points:

$$V = \{\mathbf{v}_i\} = \{(x_i, y_i, z_i)\}. \tag{1}$$

Attributes can be represented in a similar way by a set of C attributes where each entry in that set has D attributes:

$$C = \{\mathbf{c}_i\} = \{(a_{i1}, a_{i2}, \ldots, a_{iD})\} \tag{2}$$

Commonly, attributes include color components, but may also include transparency, normal vectors, motion vectors, and more.

Point clouds may have regions of interest (ROI) with special significance or relevance [3]. These regions can be used to selectively increase fidelity during compression, as done for images and videos [4].

Despite the vast available literature for the determination of saliency maps and ROI in images and videos (see, for example, [5] and [6]), there is no literature available on the creation of point cloud saliency maps and the work on point cloud segmentation is still under development [7]–[9]. Here, we propose saliency maps for point clouds.

Saliency maps in 2D have been studied for many years [10]–[14], including technologies such as neural networks and others.They were developed with the purpose of identifying, in images, regions that receive greater attention in human visualization. According to [15], [16] the purpose of a saliency map is to represent the visibility, or salience, at all locations in the visual field by a scalar quantity. It is a topographical organized map that indicates the location of salient objects in the visual field, and not what such objects are.

II. Projection-based point cloud saliency map creation

Many computer vision algorithms have been developed and are extensively studied for the 2D image case, including those to create saliency maps. We recognize the level of difficulty to develop solutions that directly act on a sparse 3D space, and we borrow solutions for the problem in 2D space. Saliency maps are generated on the 2D space, and then mapped from the image pixels to the corresponding 3D voxels.

Information from multiple projections is aggregated to obtain information about the entire point cloud. The idea is illustrated in Fig. 1.

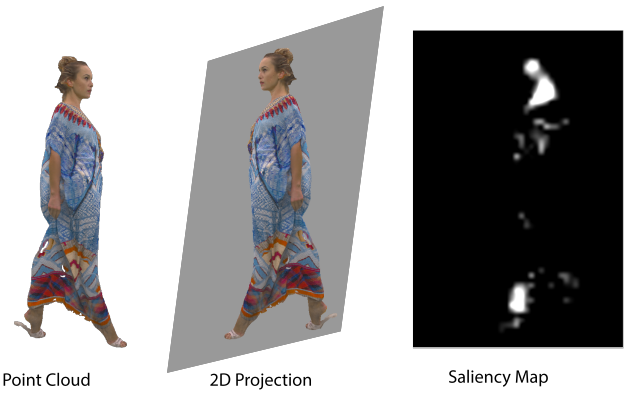

Figure 1. Steps for the creation of a saliency map using two-dimensional projection of a point cloud.

We begin by orthographically projecting the point cloud P onto a 2D plane I. If we imagine a voxel as a 3D cube and a pixel as a square element, the size of the cube side, and if we orthogonally project the cube into any of its six faces,

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal Superior - Brasil (CAPES) - Finance Code 001.

978-1-7281-9320-5/20/$31.00 ©2020 IEEE

we may be able to uniquely map the voxel face to a pixel in the 2D projection plane. Hence, the $P \rightarrow I$ mapping would be reversible. If we project at any other oblique direction, the cube projection would not be square, but a more complex polygon. Such a projection does not fit into a square pixel and partially projects onto many adjacent pixels. To cope with that situation, there are many solutions with varying degrees of accuracy and complexity. In $P \rightarrow I$ and $I \rightarrow P$, one solution is to compute the voxel or pixel color by linear combinations of the various partial projections.

An alternative is to increase resolution by replicating voxels and pixels and simply assigning the voxel color to the pixel with the largest corresponding projection area. In the back-projection $I \rightarrow P$ we can mark the voxel whose center is the closest to the projection line from the center of a marked pixel in the 2D projection plane. After all voxels are marked, the point cloud should be reduced (downsampled or averaged) to the correct resolution. Similar interpolation issues arise if one does not assume cubic voxels nor square pixels. Nevertheless, one should make sure that we are able to map voxels to pixels and to map specific pixels back to voxels.

The projection-based saliency map creation algorithm works as follows:

- Map the 3D voxels into a plane along the direction (θ, ϕ), where $-90^\circ \leq \theta \leq +90^\circ$ is the elevation and $0^\circ \leq \phi \leq 360^\circ$ is the azimuth, see Fig. 2.
- Generate the 2D saliency map, assigning the weight to each pixel that corresponds to a voxel.
- Map the pixels again to the 3D voxels. Since one pixel can be mapped to multiple pixels, one may use rounding or other decision process.

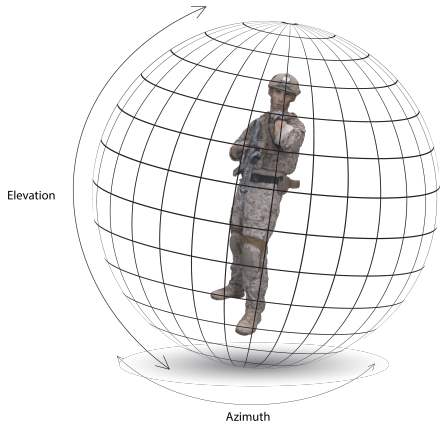

Figure 2. Representation of a point cloud and its elevation angle and azimuth.

With the above algorithm, given a voxelized point cloud and a pair (θ, ϕ) we obtain the salience value for a set of voxels. We, however, test many directions, since we do not know which orientation would be the most relevant for a user visualization. We scan the (θ, ϕ) space by spanning θ from θ_{min} to θ_{max} in steps of $\Delta\theta$ and ϕ from ϕ_{min} to ϕ_{max} in steps of $\Delta\phi$.

The algorithm is generic by nature and to construct the saliency maps we use the algorithm developed by Walther and Koch [5]. In the examples in this work, we used $\theta_{min} = -70^\circ$, $\theta_{max} = 90^\circ$, $\phi_{min} = 0^\circ$, $\phi_{max} = 359^\circ$, $\Delta\theta = \Delta\phi = 10^\circ$. Hence, $N_\theta = 17$ and $N_\phi = 36$ so that we perform $N_a = 612$ projections for each point cloud. After each projection, the salience value of each pixel is re-projected onto the corresponding voxel and added to the already existing voxel salience value. After the 612 projections, the salience value of a voxel is the sum of the salience values of the corresponding pixels in each projection. This method for merging the attributes is similar to the one presented in [16]. At the end of all projections, the voxels' salience values are normalized to a continuous range from 0.0 to 1.0. In order to smooth the transition between the salient region and the non-salient region, we propose a spatial low-pass filter with a cubic kernel of size $9 \times 9 \times 9$.

In summary, the algorithm is:

- For $\theta = \theta_{min} : \Delta\theta : \theta_{max}$
  For $\phi = \phi_{min} : \Delta\phi : \phi_{max}$
  -project the point cloud onto direction (θ, ϕ);
  -run the saliency map creation algorithm;
  -re-project the salience value of the pixels onto the voxels;
- Normalize the salience values to a range of 0.0 to 1.0;
- Filter the saliency map with a low-pass filter.

III. Encoding point clouds with soft regions of interest

When compressing a point cloud, there is a trade off between the number of bits spend to encode the point cloud and the quality of the reconstructed point cloud at the decoder. The higher the quality, the more bits are necessary. Salient regions are supposed to have a higher semantic or perceptual significance than the rest of the point cloud. Therefore, an encoder that prioritizes the quality of salient regions, (or ROI), in detriment to other regions, tend to produce reconstructed point clouds with a better subjective quality, when compared to an encoder that treats all regions equally, for the same number of bits.

In a recent work, it has been shown the compression of point clouds incorporating ROI using the Region Adaptive Hierarchical Transform [3]. We extend this work to allow for the compression using saliency maps. In [3], was assumed that voxels belong to only two regions: ROI and non-ROI. For the saliency maps in this work, there is a smooth transition between voxels that are completely salient to those that are completely non-salient.

The saliency map needs to be conveyed to the decoder. We quantize the saliency map in $L_{salience}$ levels as

$$S_q[n] = \lfloor S[n] \times L_{salience} \rfloor, \tag{3}$$

where $0 \leq S[n] < 1$ is the n-th saliency value and $S_q[n]$ is the n-th quantized saliency value. Thus, the saliency map can be

represented by integers where $S_q = L_{salience} - 1$ represents the most salience.

We sort the quantized saliency map according to the morton codes of the geometry of the corresponding voxel [17]. Morton code sorting preserves neighborhood. We encode the saliency values with adaptive run-length / Golomb-Rice encoding (RLGR) [18]. RLGR performs better when there are long sequences of zeros. As neighboring voxels tend to have similar salience, we take differences of the quantized saliency map prior to encoding with RLGR as $S_d[1] = S_q[1]$ and

$$S_d[n] = S_q[n] - S_q[n-1], \ \forall \ n > 1, \tag{4}$$

where $S_d[n]$ is the n-th differential quantized value.

The encoder in [3] attributes a weight to each voxels as a non-negative integer value. The higher the weight, the better the quality. With the saliency map, the encoder and decoder can compute the weight for each voxel. Unoccupied voxels have a weight of 0, occupied voxels that are completely non-salient have a weight of 1, and completely salient occupied voxels have a weight of $W_{ROI} \geq 1$. For voxels in the transition, the weight is linearly interpolated. Given the weights for each voxel, the encoding of the point clouds follows as in [3].

IV. Results

To test the proposed projection-based method for point cloud saliency map creation, we used 5 point clouds: Boxer, David, Longdress, Loot and Soldier, all voxelized with depth 10 (i.e. $1024 \times 1024 \times 1024$ voxels) [19], [20], [21].

The results are presented in Fig. 3 trough 8 as a saliency map (i.e. gray scale) and in a hot-cold map, where colors closer to red represent a higher saliency value and colors closer to blue are associated with a lower saliency value.

It is noticeable that, in all the examples, the most salient region contains the face, or part of it, and in some cases (as in Fig 6 and Fig. 8) a region close to the face is also considered salient.

In our tests we varied the quantization step from 2 to 128 and the ROI weights (W_{ROI}) from 1 to 64. The saliency map was quantized in 5 levels ($L_{salience} = 5$). The rate is computed as bits per occupied voxels (bpov) and the quality of the reconstructed point cloud by the peak signal to noise ratio of the luminance channel ($PSNR_Y$). Table I summarizes the encoder performance using the saliency maps. The results present the average $PSNR_Y$ difference (BD-PSNR) [22] obtained for the point clouds tested in this work, comparing those curves that prioritize the ROI ($W_{ROI} > 1$) against the curves that equally treats all voxels ($W_{ROI} = 1$). We can observe that as the W_{ROI} increases, the quality of reconstructed voxels that are completely non-salient ($S_q = 0$) decreases, while the quality of those that are salient increases at a larger rate (as in Figures 9 and 10). The gain in quality is higher for higher values of S_q, as expected (see figure 11). The higher the value of W_{ROI}, the more bits are spent to encode the ROI in detriment to non salient voxels. As there are fewer voxels in the ROI compared to those outside the ROI, a small

Figure 3. A view of point cloud "David" and its saliency and hot-cold maps.

Figure 4. A view of point cloud "Boxer" and its saliency and hot-cold maps.

Figure 5. A view of point cloud "Loot" and its saliency and hot-cold maps.

Figure 6. A view of point cloud "Longdress" and its saliency and hot-cold maps.

Figure 7. A view of point cloud "Soldier" (frame 537) and its saliency and hot-cold maps.

Figure 8. A view of point cloud "Soldier" (frame 695) and its saliency and hot-cold maps.

decrease in the quality of the voxels outside the ROI results in a big increase in the quality of those inside. The number of bits spent to encode the saliency map is accounted in the overall bit rate, except when $W_{ROI} = 1$, since there is no need to convey the saliency map to the decoder.

Table I
AVERAGE BD-PSNR COMPARING THE CURVES WITH $W_{ROI} > 1$ AGAINST THOSE WHEN $W_{ROI} = 1$ FOR ALL POINT CLOUDS TESTED IN THIS WORK.

W_{ROI}	$S_q = 0$	$S_q = 1$	$S_q = 2$	$S_q = 3$	$S_q = 4$
2	-0.43	-0.48	1.35	1.16	1.20
4	-0.54	1.30	2.43	2.65	3.06
8	-0.65	2.40	3.88	4.39	5.06
16	-0.82	3.83	5.64	7.07	7.42
32	-1.04	5.60	7.82	9.61	10.28
64	-1.31	7.79	10.61	12.23	16.43

Figure 9 shows results for rate-distortion curves for the point cloud "Longdress" using the weighted $PSNR$. The weighted $PSNR$ uses the weights of each voxel to compute the average squared error as it was shown in [3].

In Fig. 12, the point cloud Longdress is encoded with different W_{ROI}. With $W_{ROI} = 1$ the point cloud is encoded with a quantization step of 128. For $W_{ROI} = 16$ the quantization step is adjusted to 212 so that both encoded files have the same bit-rate of 0.175 bpov. Subjectively, Fig. 12(b) seems to have a better quality. Figure 13 shows a close up of the salient region for the reconstructed point clouds shown in Fig. 12.

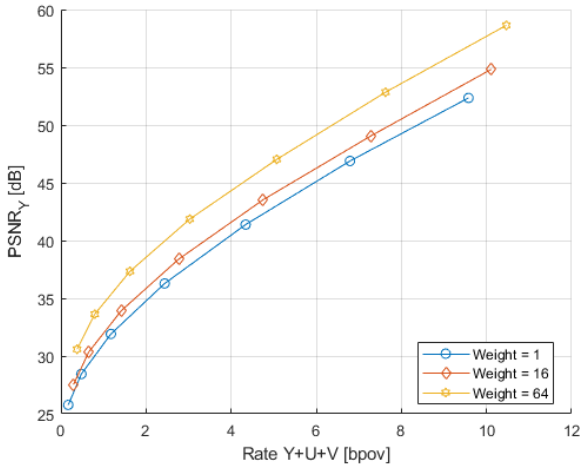

Figure 9. Rate-distortion curves for the point cloud "Longdress" using the weighted PSNR for different values of the ROI weights.

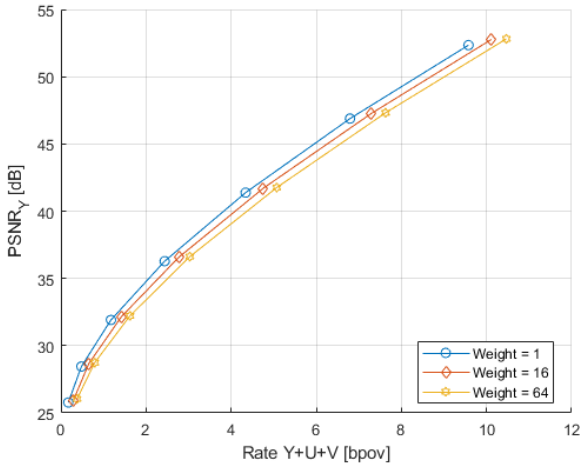

Figure 10. Rate-distortion curves for the point cloud "Longdress" for different values of the ROI weights.

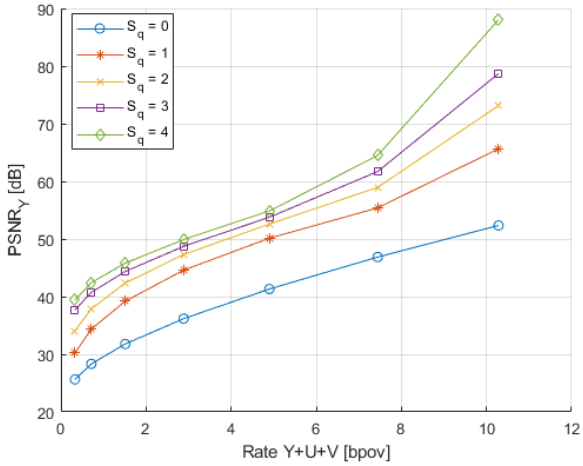

Figure 11. PSNR for the voxels inside the ROI with $W_{ROI} = 32$ for the "Longdress" point cloud.

(a) $W_{ROI} = 1$ (b) $W_{ROI} = 16$

Figure 12. Point cloud Longdress coded with different weights for voxels in the ROI. The Q_step was adjusted in order that the files would have similar sizes.

(a) $W_{ROI} = 1$ (b) $W_{ROI} = 16$

Figure 13. Close up in the salient region of the reconstructed point clouds shown in Fig. 12.

V. Conclusions

We introduced saliency maps for point clouds by using established algorithms and concepts for the creation of saliency maps in 2D images. Two-dimensional projections of different views of the point cloud are used to find saliency maps which are re-projected onto the point cloud. The results for the many views are fused into one saliency map for the whole point cloud. It was also presented a method for point cloud compression based on the saliency map as soft regions of interest. Results have shown a increase in the quality of the voxels inside the selective regions of higher levels of interest.

References

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2019.

[2] G. Sandri, R. L. de Queiroz, and P. A. Chou, "Comments on "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform"," *ArXiv e-prints*, May 2018.

[3] G. Sandri, V. F. Figueiredo, P. A. Chou, and R. de Queiroz, "Point cloud compression incorporating region of interest coding," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 4370–4374.

[4] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2014.

[5] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, pp. 1395–1407, 2006.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec 2001, pp. I–I.

[7] Z. Wen, Y. Yan, and H. Cui, "Study on segmentation of 3d human body based on point cloud data," in *2012 Second International Conference on Intelligent System Design and Engineering Application*, 2012, pp. 657–660.

[8] M. Qiao, J. Cheng, W. Bian, and D. Tao, "Biview learning for human posture segmentation from 3d points cloud," *PloS one*, vol. 9, p. e85811, 01 2014.

[9] P. Mandikal, N. K. L., and R. V. Babu, "3d-psrnet: Part segmented 3d point cloud reconstruction from a single image," *CoRR*, vol. abs/1810.00461, 2018. [Online]. Available: http://arxiv.org/abs/1810.00461

[10] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Matters of Intelligence: Conceptual Structures in Cognitive Neuroscience*, pp. 115–141, 1987.

[11] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *preprint*, 12 2013.

[12] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1265–1274.

[13] A. Maity, "Improvised salient object detection and manipulation," *International Journal of Image, Graphics and Signal Processing*, vol. 8, pp. 53–60, 11 2015.

[14] E. Mendi and M. Milanova, "Image segmentation with active contours based on selective visual attention," 05 2009.

[15] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[16] E. Niebur and C. Koch, "Control of selective visual attention: Modeling the 'where' pathway," *Neural Information Processing Systems*, vol. 8, pp. 802–808, 1996.

[17] G. M. Morton, "A computer oriented geodetic data base; and a new technique in file sequencing," IBM, Ottawa, Canada, Technical Report, 1966.

[18] R. Malvar, "Adaptive run-length / golomb-rice encoding of quantized generalized gaussian sources with unknown statistics," in *Data Compression Conference*. Institute of Electrical and Electronics Engineers, Inc., March 2006. [Online]. Available: https://www.microsoft.com/en-us/research/publication/adaptive-run-length-golomb-rice-encoding-of-quantized-generalized-gaussian-sources-with-unknown-statistics/

[19] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i voxelized full bodies — a voxelized point cloud dataset," ISO/IEC JTC1/SC29/WG1 & WG11 JPEG & MPEG, input documents M74006 & m40059, Jan. 2017.

[20] M. Krivokuća, P. A. Chou, and P. Savill, "8i voxelized surface light field (8iVSLF) dataset," ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914, Jul. 2018.

[21] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, "Microsoft voxelized upper bodies - a voxelized point cloud dataset," 2017, provided by Microsoft https://jpeg.org/plenodb/pc/microsoft/.

[22] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *Visual Coding Experts Group, ITU-T Q6/16 document VCEG-M33*, Apr. 2001.