

SURFACE LIGHT FIELD CODING FOR DYNAMIC 3D POINT CLOUDS

Deepa Naik, Sebastian Schwarz

Nokia Technologies

ABSTRACT

Surface Light Field (SLF) representations aim to provide photo realistic, free-viewpoint viewing experiences. They produce high-quality sense of presence by producing motion parallax and extremely realistic textures and lighting. Dynamic and static point clouds are used in several interesting applications such as autonomous navigation, AR/VR content viewing, cultural heritage etc. Surface Light Field point clouds captured with rig of cameras from several different directions can be key enablers for many emerging applications. However, such data is huge in size, thus distribution needs compact representation. MPEG recently started standardization activity on compression of point clouds. Current MPEG point cloud test model supports compression of only single-color attributes. In this regard, we provide the support for reading the additional attributes for SLF representations, color transforming the attributes, packing these extra attributes in separate frames and encoding and decoding these frames. We show that the MPEG Test Model is capable of handling SLF data. Furthermore, we also show the comparative results of encoding point clouds with and without the surface light field data.

Index Terms — Surface Light Field, Point Cloud Coding, Volumetric Compression.

1. BACKGROUND AND OBJECTIVES

Volumetric video data represents a three-dimensional scene or object and can be used as an input for AR (Augmented Reality), VR (Virtual Reality) and MR (Mixed Reality) applications. One form of volumetric data are point clouds, which are a set of points (x,y,z) in 3D space, where each point is associated with geometry and attribute values. One of the most typical point cloud attributes is texture color. There may exist several other attributes such as color, transparency, surface normals, curvature, and specularities which may help in advanced rendering.

Volumetric data is generated typically from 3D models. Real-world scenes are also captured using multiple cameras or combination of both video and dedicated geometry sensors. Point clouds are represented in several ways, selection of the method depends on the application, for example, medical data may need dense voxel arrays, polygon meshes may be more suitable for computer graphics applications. Sparse voxel arrays are another way to represent point clouds, point clouds represented as sparse voxel arrays are also known as voxelized point clouds.

Augmented and virtual reality applications need given scene to be rendered from any arbitrary view point. Light fields aim to provide such photo realistic free-viewpoint viewing experiences. There are two types of LF representations 1) multi-view and 2) lenslet. A multi-view representation is a collection of images captured from different viewpoints [1]. In lenslet representation, set of micro lenses are placed in front of optical direction to capture the light in each direction and from any angular direction. LF is captured in different ways, In plenoptic an array of micro-lenses

are placed in front of a conventional image sensor; to sense intensity, color, and directional information. LF is also captured using cameras mounted on moving gantries, or using robotic arm or rig of cameras. Lenselet representation can be captured using a large aperture [2] where as multi-view representation is captured using dense array of cameras [3]. Surface Light Field point clouds captured with rig of cameras from several different directions can be key enablers for many emerging applications. Unlike conventional 2D images, light field also records directional information that allows for wide range of immersive applications. However, such additional information requires substantially more data that poses challenges for storage and transmission. So, practical application of such light field is limited due to its huge memory size. Hence, distribution of such data needs compact representation. Recently, ISO/IEC JTC1/SC29/WG11 (MPEG) has started standardization activity on point cloud compression. However, the current Video based Point Cloud Compression (V-PCC) test model supports compression of only single color attribute.

The remainder of this paper is structured as follows. Section 2, describes the relevant work in this area. Section 3, describes the current architecture of V-PCC. Proposed solution for surface light field coding is explained in section 4. Evaluation of the proposed method is discussed in Section 5 and section 6, concludes the paper.

2. RELEVANT WORK

Several literature exist on representation and compression of light field data. A high level over-view of various methods for acquiring point cloud representations can be found in [4]. Authors of [5], give comprehensive overview about light field imaging including representation, acquisition, depth estimation and compression. Authors also talk briefly about lossy and lossless compression. In [6], authors discuss about light field image compression based on multiview video plus depth coding, where authors first estimate the depth according to epipolar plane image and then they use some filter to smooth inaccurate region and finally use MVD (Multiview Video plus Depth) to encode the depth. LF (Light Field) codec with disparity guided sparse coding is discussed in [7]. Here, authors select limited number of views called structural key views in such a way that, most of the spatial information is preserved. Authors report that they recover entire LF from the coding coefficients. In [8], a scheme to partition Sub-Aperture Image (SAI) into key SAI and no key SAI, is discussed. In order to exploit inter view correlation, authors perform learning-based angular super-resolution. Finally, they use model based rate distortion to optimize the bits allocated. In [1], authors map each viewing direction efficiently using B-Spline wavelet. The coefficients of the B-Spline wavelet representation are then compressed spatially. In [9], depth image-based view synthesis technique is discussed. Here, authors compress a subset of views using HEVC (High Efficiency Video Coding) encoder. The entire light field is then reconstructed using these subset of images. Authors claim that the results outperforms similar view synthesis methods which uses convolutional neural

networks. In [10], authors provide deep neural network-based approach to compress SLF (Surface Light Field) data.

Recently, ISO/IEC JTC1/SC29/WG11 (MPEG) has started standardization activity on point cloud compression. The work presented in this paper is based on the MPEG video-based point cloud compression (V-PCC) framework for dynamic point clouds. In this work we provide simple solution to handle SLF data which will be described in more detail in the following section.

3. VIDEO-BASED POINT CLOUD COMPRESSION (V-PCC)

Volumetric data is generally huge in size each point here is associated with geometry, positions and attribute information. The main challenge for such data is to find the relationship between temporal and spatial correlation among different points to achieve compression. Recently MPEG has established Point Cloud Compression as new MPEG-I standards group within MPEG, and set to develop novel solutions to compress 3D geometry and attribute information. The main idea is to leverage the existing video codecs to compress the geometry and texture information of a dynamic point. Here, 3D point cloud is converted to 2D sequences one for capturing the geometry information and another for capturing the texture information. Converted 2D sequences are compressed using existing video codecs, e.g. using HEVC Main profile. Additional meta-data that are needed to interpret the video sequences are also multiplexed to form a single bitstream. Fig. 1 illustrates the V-PCC encoding block structure.

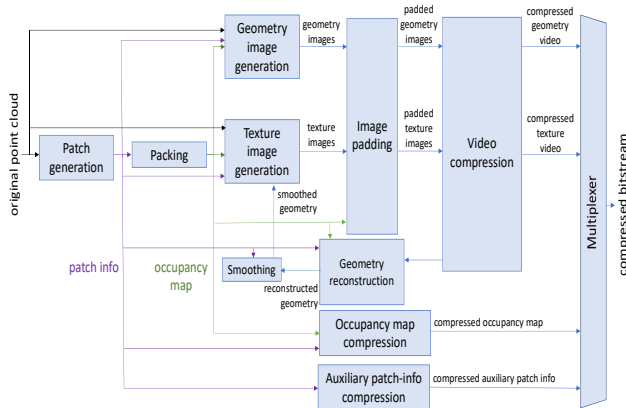


Figure 1: MPEG V-PCC encoder block structure.

3.1. Patch generation, packing, and image generation

First step of V-PCC block is patch generation. First, surface normals at every point are estimated. Clustering of the point cloud is performed by associating each point with one of the six orientation planes. Initial clustering is then refined by updating the cluster index associated with each point based on its normal and the cluster indices of its nearest neighbors. Finally patches are extracted by connected component extraction procedure. packing process maps extracted patch onto a 2D grid while minimizing the unused space, packing process also assigns unique patch to every $T \times T$ blocks

3.2. Geometry and texture image construction

The image generation process exploits the 3D to 2D mapping computed during the packing process to store the geometry and

texture of the point cloud as images. Since multiple points may get projected on to the same pixel, each patch is projected onto two different planes. The first layer or the nearest plane stores the point with lowest depth and the second layer or farthest plane stores the point with highest depth. Thus two image layers are created for both, geometry and texture. Once the geometry image is created, it is dilated and compressed. Texture image exploits the reconstructed 3D points. For each reconstructed 3D points, its point to pixel position is evaluated. Color of each point is then mapped to 2D grid by using point to pixel position. Texture image then goes for dilation and compression in the same way as for geometry.

3.3. Auxiliary and occupancy information coding

Auxiliary patch metadata such as index of the projection plane, 2D bounding box (u_0, v_0, u_1, v_1) , 3D location of the patch (x_0, y_0, z_0) are encoded/decoded for every patch. Furthermore, size of each $M \times M$ block and its associated patch index is encoded/decoded.

Occupancy map is a binary map that indicates for each cell of the grid whether it belongs to the empty space or to the point cloud. For each patch, if the point of the patch has depth less than infinite depth value, the point is considered as valid and its position is marked as occupied in the occupancy map which is 2D binary image. Occupancy map thus created is later used during point cloud reconstruction process. Every occupied cell of the 2D grid would produce a pixel during the image generation.

3.4. Geometry reconstruction

Non-empty pixels in the geometry/texture images/layers are detected based on the occupancy map. The 3D positions of the points associated with non-empty pixels are computed by leveraging the auxiliary patch information and the geometry images. More precisely, let P be the point associated with pixel position (u, v) and let (δ_0, s_0, r_0) be the 3D location of the patch to which it belongs and (u_0, v_0, u_1, v_1) its 2D bounding box. P could be expressed in terms of depth $\delta(u, v)$, tangential shift $s(u, v)$ and bi-tangential shift $r(u, v)$ as shown below in equations 1, 2 and 3.

$$\delta(u, v) = \delta_0 + g(u, v) \quad (1)$$

$$s(u, v) = s_0 - u_0 + u \quad (2)$$

$$r(u, v) = r_0 - v_0 + v \quad (3)$$

where $g(u, v)$ is the luma component of the geometry image.

3.5. V-PCC decoder

V-PCC decoder block is shown in fig. 2. At the decoder side first, the auxiliary information is decoded. Followed by auxiliary information, occupancy and geometry are decoded. 3D points are reconstructed using decoded, auxiliary information, occupancy and geometry. Finally, texture mapping performed for the reconstructed 3D points using decoded texture frame.

4. METHOD

Current V-PCC frame work does not support more than single color attribute, so the main emphasis of the research work is to provide simple solution to support more than single color attribute. For this purpose, we modified the current V-PCC test model [11] to be able to load the above described data set and perform the 3D to 2D conversion for all the input texture channels (up to 13). The outcome is a single geometry video stream, thirteen texture video streams

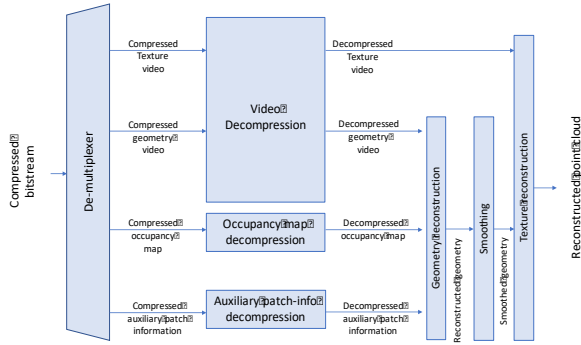


Figure 2: MPEG V-PCC decoder block structure.

and the auxiliary information, packed in a V-PCC bitstream. The modified decoder can read this bitstream and reconstruct a PLY file including all encoded rigs.

4.1. Color transformation, Padding and Texture generation

The first few blocks of V-PCC test model i.e, patch generation, occupancy map generation, geometry encoding follows the same pipe line as current V-PCC without any modification. However, texture generation pipeline is modified to handle all the other texture attributes. Texture generation process exploits the available geometry and occupancy map information to reconstruct the 3D point clouds. Pixel position of the 3D points are also determined during the reconstruction of the 3D points. However, all the points that are reconstructed may not have corresponding source color associated with it. Thus, color refinement process is performed. In color refinement process, for each target point its nearest source point is evaluated using a KD tree. Color of the closest source is assigned to the target. Color transformation is performed in similar way for all the available attributes.

The padding process fills the empty space between patches in order to generate a piece-wise smooth image suited for video compression. Each $T \times T$ block is processed independently. Each empty block is filled by copying either the last row or column of the previous $T \times T$ block in raster order. If the block has both empty and filled pixels, then the empty pixels are iteratively filled with the average value of their non-empty neighbors.

3D points are transferred to 2D grid using previously evaluated point to pixel information. All the available attributes(13) are placed in separate video channel and compressed. Required meta data is also encoded along side the bitstream required for decoding.

4.2. Attribute metric

V-PCC's error metric frame work is also updated to handle the extra attributes. The color distortion is measured in YUV space with 3 separate mean-squared error (MSE) distortions, which are reported as PSNR (Peak Signal to Noise Ratio) for each channel: Y, U, and V. In order to compute the MSE (Mean Square Error), squared distance of the color between the reference point cloud and the reconstructed point cloud is performed. For each point in decoded point cloud, its corresponding nearest point in reference point cloud is evaluated. KD tree search is used to locate the nearest neighbor. The MSE then corresponds to the squared distance between these

Table 1: Single color and geometry result.

Case	Total Bitrate(bits)	Geometry PSNR in dB		Color PSNR in dB		
		D1	D2	Y	U	V
AI	49997288	83.29	86.76	40.78	45.53	45.56

Table 2: SLF (Color attributes) and geometry result.

Case	Total Bitrate (bits)	Geometry PSNR in dB		Attribute PSNR in dB			Attribute PSNR in dB			Attribute PSNR in dB		
		D1	D2	Y1	U1	V1	Y5	U5	V5	Y13	U13	V13
606020560		83.29	86.76	40.01	45.23	45.26	40.14	45.28	45.35	40.45	45.47	45.50

two points. Attribute distortion is evaluated in similar way for all the attributes of the light field.

5. RESULTS

We make use of data set provided by 8i [3], for MPEG standardization activity for the development and test purpose. The data here is captured by 39 synchronized RGB cameras configured in either 12 or 13 rigs, or clusters. For each rig, there exists a separate red, green, and blue color component, thus there are 13 extra attributes than the normal point cloud data.

For coding performance evaluation, the MPEG Common Test Condition (CTC) for Point Cloud Compression [12] were followed. Objective evaluation is performed using two distortion metrics, namely point-to-point error (D1) and a point-to-plane error (D2) for geometry, as well as PSNR for the color attributes [13]. For the objective evaluation, All Intra (AI) and Random Access (RA) coding conditions [12] for 15 frames were assessed.

Table 1, show the result of total encoded bitrate, geometry PSNR and color PSNR. For the sake of simplicity, in table 2 color PSNR of only 3 attributes are shown along with total encoded bitrate and PSNR of the geometry. It can be seen from the Table 1 and 2, there is an increase in bit rate of around 13 Mb due to SLF support instead of single color. Fig. 3, shows uncompressed and decoded files. Fig. 3a, shows uncompressed file, Fig. 3b, shows reconstructed file. Fig. 3b, is mainly averaged color from all the viewing directions. The decoded PLY file has all the 13 textures (from different viewing directions) but for the sake of simplicity the reconstructed Fig. 3c with viewing direction 9 is shown. As can be seen from the figure, viewing direction 9 shows better reflection as it is not fused color from all the direction as in the case of single color.

6. CONCLUSION AND FUTURE WORK.

In this work, we provided a simple solution to handle SLF data with the current V-PCC test model. We have proven that the test model is in general capable of handling such SLF data, although memory requirements are still a hurdle. As of now, the memory requirement for handling the massive amount of input data was too large as we are encoding and decoding all the 13 attributes of the SLF. We plan to deliver further improvements to reduce the compression of these extended color sets in to smaller sub sets yet providing the same specularly experience.

7. REFERENCES

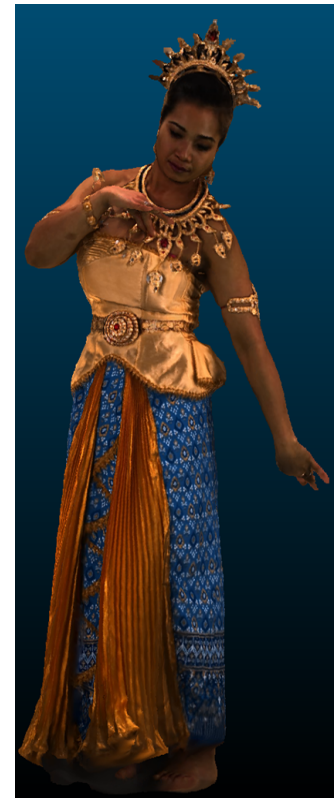
- [1] P A Chou M-T. Sun M. Tang S. Wang S. Ma W. Gao. X. Zhang, P A. Chou, "Surface light field compression using a point cloud codec," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2018.



(a) Uncompressed



(b) Reconstructed



(c) Recon. viewing direction 9

Figure 3: Uncompressed and reconstructed point clouds.

- [2] M. Brédif Mathieu G. Duval-M. Horowitz R. Ng, M. Levoy and et al. P. Hanrahan, “Light field photography with a handheld plenoptic camera,” *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.
- [3] V. Vaish E-V. Talvala-E. Antunez A. Barth A. Adams M. Horowitz B. Wilburn, N. Joshi and M. Levoy., “High performance imaging using large camera arrays,” in *ACM Transactions on Graphics (TOG)*. ACM, 2005, vol. 24, pp. 765–776.
- [4] V. Baroncini M. Budagavi P. Cesar P A. Chou R. Cohen M. Krivokuća Maja S. Schwarz, M. Preda and et al. S. Lasserre, Z. Li, “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, 2018.
- [5] A.Jarabo Y. Zhang L. Wang Q. T. Chai G. Wu, B. Masia and L. Yebin., “Light field image processing: An overview,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, 2017.
- [6] L. Shen X. Huang, P. An and K. Li., “Light field image compression scheme based on mvd coding standard,” in *Pacific Rim Conference on Multimedia*. Springer, 2017, pp. 79–88.
- [7] J. Hou J. Chen and L-P. Chau., “Light field compression with disparity-guided sparse coding based on structural key views,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 314–324, 2017.
- [8] J. Chen J. Hou and L-P. Chau., “Light field image compression based on bi-level view compensation with rate-distortion optimization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 517–530, 2018.
- [9] M.Le Pendu X.Jiang and C. Guillemot, “Light field compression using depth image based view synthesis,” in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2017, pp. 19–24.
- [10] Y. Zhang N. Li J. Lu S. Gao A. Chen, M. Wu and J. Yu., “Deep surface light fields,” *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 14, 2018.
- [11] “V-pcc test model v4, iso/iec jtcl/sc29 wg11 doc. n71996, macau, october 2018,” 2018.
- [12] Krivokuća, S. Schwarz, P A. Chou, and M. Budagavi, “Common test conditions for point cloud compression,” *arXiv preprint arXiv:1810.00484*, 2018.
- [13] D. Tian, H. Ochimizu, C. Feng, Robert Cohen, and A. Vetro, “Geometric distortion metrics for point cloud compression,” in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3460–3464.