Curve-Based Representation of Point Cloud for Efficient Compression

I. DARIBO†, R. FURUKAWA†, R. SAGAWA††, H. KAWASAKI†††, S. HIURA†, and N. ASADA†

† Faculty of Information Sciences, Hiroshima City University Hiroshima, Japan
†† National Institute of Advanced Industrial Science and Technology Tsukuba, Japan
††† Faculty of Engineering, Kagoshima University Kagoshima, Japan
E-mail: †{daribo,ryo-f,asada}@hiroshima-cu.ac.jp, ††ryusuke.sagawa@aist.go.jp, †††jiro@jouhou.co.jp,
††††kawasaki@ibe.kagoshima-u.ac.jp

Abstract Three-dimensional acquisition is emerging scanning systems raising the problem of efficient encoding of 3D point cloud, in addition to a suitable representation. This paper investigates cloud point compression via a curve-based representation of the point cloud. We aim particularly at active scanning system capable of acquire dense 3D shapes, wherein light patterns (*e.g.* bars, lines, grid) are projected. The object surface is then naturally sampled in scan lines, due to the projected structured light. This motivates our choice to first design a curve-based representation, and then to exploit the spatial correlation of the sampled points along the curves through a competition-based predictive encoder that includes different prediction modes. Experimental results demonstrate the effectiveness of the proposed method.

Key words Point cloud, compression, curve-based representation, rate-distortion

1. Introduction

Recently active scanning systems are capable to produce 3D geometric models with millions of pixels. The increase of 3D model complexity rise then the need to store, transmit, process and render efficiently the huge amount of data. In addition to the representation problem, efficient compression of 3D geometry data becomes particularly more and more important.

Traditional 3D geometry representation usually falls in two categories: polygon mesh and point-sampled geometry. Typically, mesh-based representation exploits the connectivity between vertices, and orders them in a manner that contains the topology of the mesh. Such representation is then made of polygons coded as a sequence of numbers (vertex coordinates), and tuple of vertex pointers (the edges joining the vertices), mostly due to its native support in modern graphics cards. Such model requires, however, a time consuming and difficult processing with explicit connectivity constraint. Point-sampled geometry has received attention as an attractive alternative to polygon meshes geometry with several advantages. For example, no connectivity information are needed anymore to be stored, the triangulation overhead is saved, leading to a simpler and intuitive way to process and render object of complex topology through a cloud of points.

Currently active 3D scanners are widely used for acquiring 3D models [1]. Especially, scanning system based on structured light have been intensively studied recently [2], [3]. Structured-light-based scanning is done by sampling the surface of an object with a known pattern (*e.g.* grid, horizontal bars, lines) (see Fig. 1). Studying the deformation of the pattern allows to build a 3D model by means of a point cloud. It is worth to note that structured-light-based scanning systems output the points along the measuring direction, which naturally orders group of points along the same direction: scan lines. This motivates our choice to take advantage of the spatially sequential order of the sampled-points along these scan lines: first, we pre-process the data by partitioning each scan lines in curve of points, and after we exploit the curve-based representation through competition-based predictive encoder specially designed to take benefit from the scanning directions. Our encoder can then reach an efficient compression ratio of the point cloud thanks to the curve-based representation (see Figure 2).

In this work, by proposing a curve-driven point cloud compression, our framework can straightforwardly support for example random access, error recovery, error propagation limitation, where previous work mainly focus on compression efficiency only. These points will be further discussed in Section4..

The rest of the paper is organized as follows. We introduce some related work in Section 2.. Section 3. de-

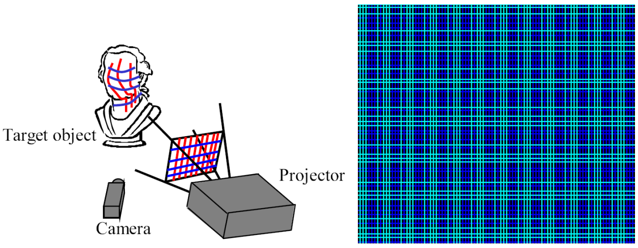


Fig. 1 (left) Grid-pattern-based scanning system: a grid pattern is projected from the projector and captured by the camera. (right) Example of projected grid pattern.

scribes the point cloud pre-processing towards a curve-based representation, and Section 4. addresses the problem of efficiently compressing a point cloud acquired by a structured-light 3D scanning system. Finally, our final conclusions are drawn in Section 6..

2. Related work

The problem of 3D geometry compression has been extensively studied for more than a decade and many compression schemes were designed. Existing 3D geometry coders mainly follow two general lines of research: single-rate and progressive compression. In opposition to single-rate coders, progressive ones allow the transmission and reconstruction of the geometry in multiple level of details (LODs), which is suitable for streaming applications.

Since many important concepts have been introduced in the context of mesh compression, several point cloud compression schemes apply beforehand triangulation and mesh generation, and then use algorithms originally developed for mesh compression [4], but at the cost of also encode mesh connectivity. Instead of generating meshes from the point cloud, other approaches propose partitioning the point cloud in smooth manifold surfaces closed to the original surface, which are approximated by the method of moving least squares (MLS) [5]. On the other hand, an augmentation of the point cloud by a data structure has been proposed to facilitate the prediction and entropy coding. The object space is then partitioned based on the data structure, *e.g.* octree [6]~ [9], spanning tree [10], [11]. Although not strictly a compression algorithm, the QSplat rendering system offers a compact representation of the hierarchy structure [12]. A high quality rendering is obtained despite a strong quantization. A compression algorithm for the QSplat representation has been proposed through an optimal bitrate allocation [13]. Still in a RD sense, an RD-optimized version of the D3DMC encoder has been developed for dynamic mesh compression [14].

To the best of our knowledge, previous point-based coders mainly require at least one of the following issues:

- surface approximation: MLS, *etc.*,
- complexity increasing: point re-ordering, triangulation, mesh generation, *etc.*,
- data structure: spanning tree, octree, *etc.*,

which leads to either smoothing out sharp features, an increase of the complexity, or an extra-transmission of a data structure.

3. Curve-based representation

As discussed before, the points are ordered along scan lines forming naturally lines as illustrated in Figure 2. Under the scan line representation assumption, the curve set generation process partitions the point cloud, wherein each partition is a curve. In some cases, the partitions can be directly obtained from the acquisition process, *e.g.* line detection algorithm [3].

3.1 Curve-based point cloud definition

Let us consider the point cloud $\mathcal{S} = \{p_1, p_2, \cdots p_N\}$ as a collection of N 3D points $p_{k_{1 \leqq k \leqq N}}$. As mentioned earlier, structured-light-based 3D scanning systems fit the sampled points in curves. The point cloud $\mathcal{S}$ can then be represented as a set of M curves $\mathcal{C}^{l_{1 \leqq l \leqq M}}$ as

$$\mathcal{S} = \{\mathcal{C}^1, \mathcal{C}^2, \cdots, \mathcal{C}^M\} \tag{1}$$

where a l-ieme curve $\mathcal{C}^l$ is expressed as

$$\mathcal{C}^l = \{p_r, p_{r+1}, \cdots, p_s\} \text{ with } 1 \leqq r < s < N \tag{2}$$

3.2 Curve-based partitioning

Each curve $\mathcal{C}$ is defined to contain points that share similar proprieties, *e.g.* curvature, direction, Euclidean

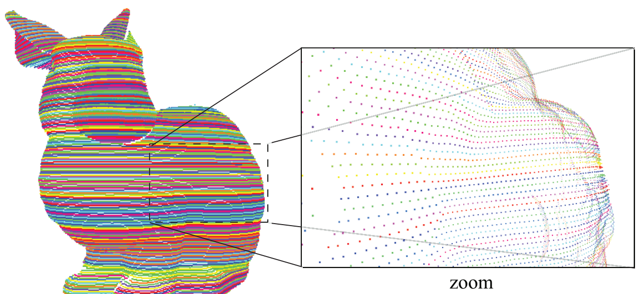


Fig. 2 The Stanford model Bunny partitioned into a set of curves with regard to the scanning directions. Curves are discriminate by different colors.

distance with his neighbor, *etc.*. The point cloud $\mathcal{S}$ is partitioned into a set of curves as defined in Equation (1). The division is controled by defining if the current point p_k to process is an outlier with respect to the current curve $\mathcal{C}$. In this study, we defined an outlier as

$$\|p_k - p_{k-1}\| > \epsilon, \tag{3}$$
$$\text{with} \quad \epsilon = \frac{1}{N-1}\sum_{i=2}^{N}\|p_i - p_{i-1}\|.$$

The current point p_k is considered as an outlier and then add to a new curve, if the Euclidean distance $\|p_k - p_{k-1}\|$ is larger than a defined threshold ϵ: here the average value of the distance between two consecutive points throughout the point cloud.

For instance other outlier definitions can be considered. For example by checking if adding the current point p_k will disturb the normal distribution of the current curve. Otherwise, by computing the inter-quartile-range (IQR) of the current curve, and then use a multiple as a threshold. In this study we only considered the Euclidean distance

4. Point cloud encoding

The proposed framework compresses the points as ordered prior to scanning direction. The object surface is then sampled in curve of points as shown in Figure 2. Unlike previous work in geometry compression, we do not need any connectivity information, or data augmentation by a data structure (*e.g.* octree), to reach a satisfactory compression efficiency.

4.1 Prediction

Let $\mathcal{C}$ be the current curve to encode. Intra-curve prediction attempts to determine, for each point p_k in $\mathcal{C}$, the best predicted point $\widehat{p_k}$ with respect to the previous coded points $\widetilde{p}_{i,i<k}$ in $\mathcal{C}$. Note that previous coded points $\widetilde{p}_{i,i<k}$ have been quantized and inverse quantized. For notation concision, let us define the sub-curve containing the previous coded points by

$$\mathcal{C}|_{i<k} = \mathcal{C} \cap \{p_i | i < k\}, \tag{4}$$

and the intra-curve prediction by

$$\widehat{p_k} = P\left(\mathcal{C}|_{i<k}\right). \tag{5}$$

It is important to note that another curve informations are not utilized, which for instance enables random access and error propagation limitation. The prediction outputs the corrective vector $r_k = p_k - \widehat{p_k}$, also denoted as residual, and transmits it to the entropy coder. The coding efficiency comes with the accuracy of the prediction that is improved by choosing the most suitable prediction method for each point. For each point, instead of using only one prediction method for all the points [10], we propose making compete all defined prediction modes that are known by the encoder and the decoder. The prediction that minimizes the Euclidean distance $\|p_k - \widehat{p_k}\|$ is defined as the best one. A prediction flag is then placed in the bitstream. Since the choice of the prediction modes is related to the scene content, a prediction flag is then placed in the bitstream for each point. In the following, we present the different designed prediction modes.

4.1.1 No-prediction P^{Intra}

No-prediction is applied, which define the current point as key point that can be used, for example, for random access and error propagation limitation.

$$P^{Intra}\left(\mathcal{C}|_{i<k}\right) = (0,0,0). \tag{6}$$

4.1.2 Const P^{Const}

The previous coded point in the curve is used as prediction.

$$P^{Const}\left(\mathcal{C}|_{i<k}\right) = \widetilde{p}_{k-1}. \tag{7}$$

4.1.3 Linear P^{Linear}

The prediction is based on the two previous coded point in the curve.

$$P^{Linear}\left(\mathcal{C}|_{i<k}\right) = 2 \cdot \widetilde{p}_{k-1} - \widetilde{p}_{k-2} \tag{8}$$

4.1.4 Fit-a-line $P^{FitLine}$

The predicted point is an extension of a segment $\mathcal{L}\left(\mathcal{C}|_{i<k}\right)$ defined by all the previous coded points. The segment $\mathcal{L}\left(\mathcal{C}|_{i<k}\right)$ is given by line fitting algorithm based on the M-estimator technique, that iteratively fits the segment using weighted least-squares algorithm.

$$\begin{aligned} P^{FitLine}\left(\mathcal{C}|_{i<k}\right) = \; & 2 \cdot < \mathcal{L}\left(\mathcal{C}|_{i<k}\right) \perp \widetilde{p}_{k-1} > \\ & - < \mathcal{L}\left(\mathcal{C}|_{i<k}\right) \perp \widetilde{p}_{k-2} > \end{aligned} \tag{9}$$

where $< \mathcal{L} \perp p_i >$ is the orthogonal projection of the point p_i onto the line supporting the segment $\mathcal{L}$.

4.1.5 Fit-a-sub-line $P^{FitSubLine}$

As previously, a line fitting algorithm is used to perform the prediction, but a sub-curve $\mathcal{C}|_{i_0 \leqq i<k}$ is utilized instead of all the previous coded points. The starting point p_{i_0} is, however, needed to be signaled to the decoder, and thus an additional flag is put in the bitstream.

$$P^{FitSubLine}\left(\mathcal{C}|_{i<k}\right) = 2\cdot < \mathcal{L}\left(\mathcal{C}|_{i_0 \leqq i<k}\right) \perp \widetilde{p}_{k-1} > \\ - < \mathcal{L}\left(\mathcal{C}|_{i_0 \leqq i<k}\right) \perp \widetilde{p}_{k-2} > \quad (10)$$

4. 2 Quantization

After prediction, the point cloud is represented by a set of corrective vectors, wherein each coordinate is a real floating number. The quantization will enable the mapping of these continuous set of values to a relatively small discrete and finite set. In that sense, we apply a scalar quantization as follow

$$\widetilde{r_k} = \text{sign}(r_k) \cdot \text{round}\left(|r_k| * 2^{bp-1}\right) \quad (11)$$

where bp is the desired bit precision to represent the absolute floating value of the residual.

4. 3 Coding

The last stage of the encoding process removes the statistical redundancy in the quantized absolute component of the residual $|\widetilde{r_k}|$ by entropy Huffman coding. Huffman coding assigns a variable length code to each absolute value of the quantization residual based on the probability of occurrence.

The bitstream consists of:

- an header containing the canonical Huffman codeword lengths,
- the quantization parameter bp,
- the total number of points,
- the residual data for every point.

The coded residual of every point is composed of:

- 3 bits signaling the prediction used,
- 1 bit for the sign,
- a variable-length code for each absolute component value of the corrective vector with regards to the entropy coder.

4. 4 Decoding

The decoding process is straightforward by reading the prediction mode and the residual.

5. Experimental results

The performance of the proposed framework is evaluated using the three models shown in Fig. 3. The objective compression performance of the proposed method is investigated in the rate-distortion (RD) curves plotted in Figure 4 through the average number of bits per points (bpp), in relation to the loss of quality, measured by the peak signal to noise ratio (PSNR). The PSNR is evaluated using the Euclidean distance between points. The peak signal is given by the length of the diagonal of the bounding box of the original model. The RD results correspond respectively to the seven bp quantization parameters: 8, 9, 10, 11, 12, 14 and 16. We compare our competitive-optimized strategy with the simpler decision made only on the prediction error. It can be observed that the proposed method provides better results in RD performance.

In particular Table 1 shows the resulting compression rates in bits per point (bpp) after a 12 bits quantization as defined in Section 4. 2. Around -80% of bit savings is achieved compared to the uncompressed rate that used 96 bits for each point.

Tbl. 1 Compression rates in bits per point (bpp) after 12 bits quantization per coordinate.

model	#points	#curves	rate	bit savings
Dragon	43k	1221	11.98 bpp	-87.51 %
Buddha	79k	2535	11.52 bpp	-87.99 %
Bunny	35k	924	12.17 bpp	-87.31 %

6. Conclusion

We designed and implemented a competition-based predictive single-rate compression for the positions points outputted by a structured-light 3D scanning system. Novel prediction methods has been designed to exploit the inherent spatially organization of the points, that are ordered prior to the scanning direction. First we pre-process the point cloud to efficiently taking advantage of the scan lines in the prediction stage. In addition our method had the advantage to not need any surface approximation, or the transmission of a data structure information (*e.g.* octree, spanning tree).

Several issues remain that warrant further research. In future studies, we integrate other point attributes (*e.g.* color, normal, *etc.*), and extend our encoder to arbitrary point clouds.

Acknowledgment

This work is partially supported by Strategic Information and Communications R&D Promotion Programme (SCOPE) No.101710002, by Grand-in-Aid for Scientific Research No.21200002 in Japan, and by Next Program LR03 of Japan Cabinet Office.

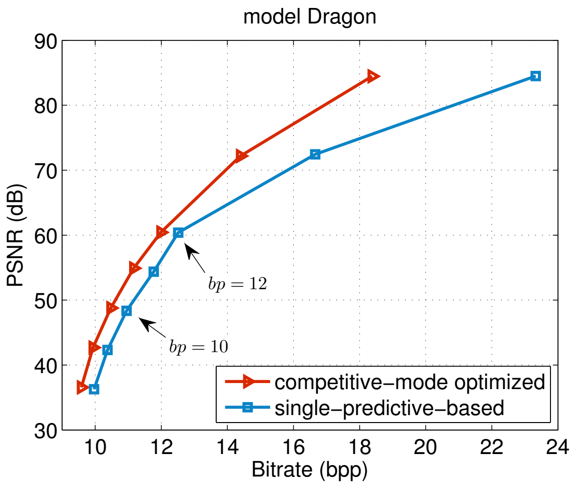
References

[1] J. Batlle, E. M. Mouaddib, and J. Salvi, "Recent progress in coded structured light as a technique to solve the correspondence problem: a survey," *Pattern Recognition*, vol. 31, pp. 963–982, 1998.

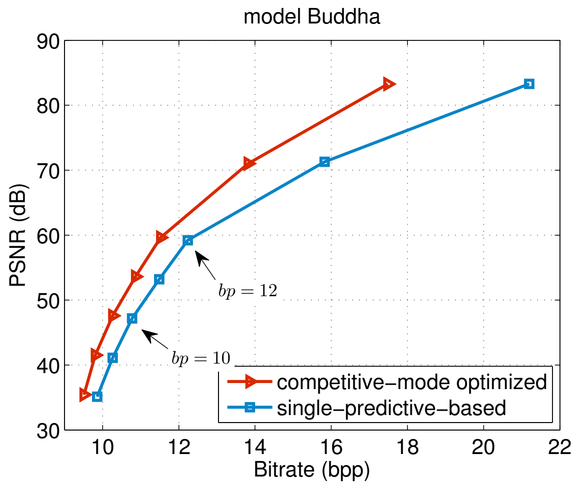
[2] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi,

(a) model Dragon (b) Model Happy Buddha (c) Model Bunny

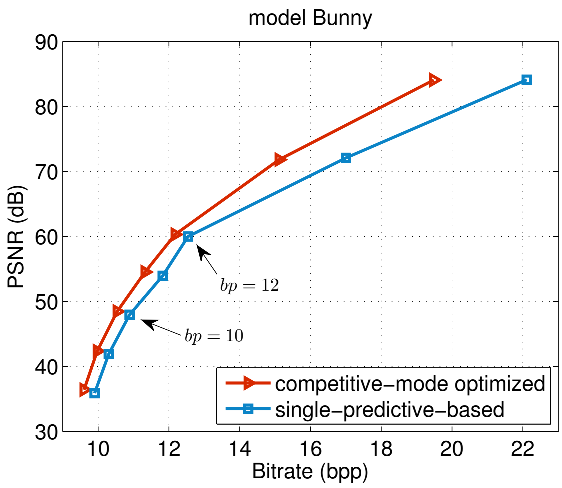
Fig. 3 Test models from Stanford Computer Graphics Laboratory's 3D scanning repository (only one view from each model is used to preserve the scan lines from the acquisition process).



(a) model Dragon



(b) model Buddha



(c) model Bunny

Fig. 4 Rate-distortion performance of the proposed encoder.

"Dynamic scene shape reconstruction using a single structured light pattern," in *Proc. of the IEEE Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, Jun. 2008, pp. 1–8.

[3] R. Sagawa, Y. Ota, Y. Yagi, R. Furukawa, N. Asada, and H. Kawasaki, "Dense 3D reconstruction method using a single pattern for fast moving object," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, Sep. 2009, pp. 1779–1786.

[4] P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes," in *Proc. of the Advances in Multiresolution for Geometric Modelling*, 2003, pp. 3–26.

[5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," vol. 9, no. 1, pp. 3–15, 2003.

[6] M. Botsch, A. Wiratanaya, and L. Kobbelt, "Efficient high quality rendering of point sampled geometry," in *Proc. of the Eurographics Workshop on Rendering (EGRW).* Pisa, Italy: Eurographics Association, 2002, pp. 53–64.

[7] J. Peng and C.-C. J. Kuo, "Geometry-guided progressive lossless 3d mesh coding with octree (OT) decomposition," *ACM Transaction on Graphics*, vol. 24, pp. 609–616, July 2005. [Online]. Available: `http://doi.acm.org/10.1145/1073204.1073237`

[8] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. of the IEEE Eurographics Symposium on Point-Based Graphics*, M. Botsch and B. Chen, Eds. Eurographics, Jul. 2006.

[9] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "A generic scheme for progressive point cloud coding," vol. 14, no. 2, pp. 440–453, 2008.

[10] S. Gumhold, Z. Kami, M. Isenburg, and H.-P. Seidel, "Predictive point-cloud compression," in *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH).* Los Angeles, USA: ACM, Aug. 2005.

[11] B. Merry, P. Marais, and J. Gain, "Compression of dense and regular point clouds," in *Proc. of the Computer graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH).* Cape Town, South Africa: ACM, 2006, pp. 15–20.

[12] S. Rusinkiewicz and M. Levoy, "QSplat: A multiresolution point rendering system for large meshes," in *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Jul. 2000, pp. 343–352.

[13] J.-Y. Sim and S.-U. Lee, "Compression of 3-D point visual data using vector quantization and rate-distortion optimization," *IEEE Transactions on Multimedia*, vol. 10, pp. 305–315, 2008.

[14] K. Muller, A.Smolic, M. Kautzner, and T. Wiegand, "Rate-distortion optimization in dynamic mesh compression," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, 2006.

[15] H. E. III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, May-Jun. 1963.

[16] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, Thessaloniki , Greece, Oct. 2001, pp. 542–545.