Point Cloud Compression for Grid-Pattern-based 3D Scanning System

I. Daribo*, R. Furukawa*, R. Sagawa†, H. Kawasaki‡, S. Hiura* and N. Asada*
*Faculty of Information Sciences, Hiroshima City University, Hiroshima, Japan
Email: {daribo, ryo-f, hiura, asada}@hiroshima-cu.ac.jp
†National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan
Email: ryusuke.sagawa@aist.go.jp
‡Faculty of Engineering, Kagoshima University, Kagoshima, Japan
Email: kawasaki@ibe.kagoshima-u.ac.jp

Abstract— **Recently it is relatively easy to produce digital point sampled 3D geometric models. In sight of the increasing capability of 3D scanning systems to produce models with millions of points, compression efficiency is of paramount importance. In this paper, we propose a novel competition-based predictive method for single-rate compression of 3D models represented as point cloud. In particular we aim at 3D scanning methods based on grid pattern. The proposed method takes advantage of the pattern characteristic made of vertical and horizontal lines, by assuming that the object surface is sampled in curve of points. We then designed and implemented a predictive coder driven by this curve-based point representation. Novel prediction techniques are specifically designed for a curve-based cloud of points, and been competing between them to achieve high quality 3D reconstruction. Experimental results demonstrate the effectiveness of the proposed method.**

I. Introduction

During the last years, the wide-spreading of scanning technologies and applications has finally opened the way to the long-waited 3D acquisition revolution. As a consequence, effective 3D geometry compression schemes are required to face the need to store and/or transmit the huge amount of data.

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.

Currently active 3D scanners are widely used for acquiring 3D models [1]. Especially, scanning systems based on structured light have been intensively studied recently. Structured-light-based scanning is done by sampling the surface of an object with a known pattern (*e.g.* grid, horizontal bars) (see Fig. 1). Studying the deformation of the pattern allows to build a 3D model by means of a set of points, also denoted as point cloud. It is important to note that the spatial point organization is strongly correlated to the pattern shape.

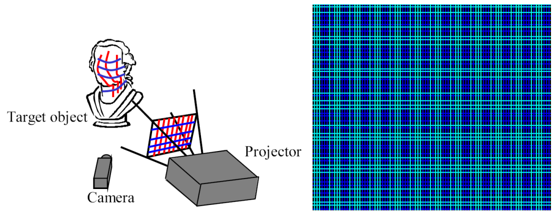


Fig. 1. (left) Grid-pattern-based scanning system: a grid pattern is projected from the projector and captured by the camera. (right) Example of projected grid pattern.

In this paper, we present a general framework compressing efficiently cloud of points acquired by 3D scanning systems using structured light. In particular, we study 3D measurement systems using a grid pattern formed by straight lines distinguishable only as vertical and horizontal lines [2] as illustrated in Fig. 1. When the projected grid pattern is extracted from the captured image, 3D points are naturally fitted into series of curves [3], [4]. Our method compresses point positions by taking advantage of the spatially sequential order of the sampled-points organized along these predefined curves.

One main objective of geometry compression is indeed reducing the amount of data to store and/or transmit, but also supporting application-oriented functionalities such as: splicing, random access, error resiliency and error recovery to name but a few. 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 Section III.

The rest of the paper is organized as follows. We introduce some related work in Section II. Section III addresses the problem of efficiently compressing a point cloud acquired by a grid-pattern-based 3D scanning system. Finally, experiemntal results are presented in Section IV, and our final conclusions are drawn in Section V.

II. 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 compression schemes apply beforehand triangulation and mesh generation, and use algorithms developed for mesh compression [5], wherein mesh connectivity is also encoded. Instead of directly 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) [6]. 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 chosen data structure: octree [7], [8], [9], [10], spanning tree [11], [12] to name a few. Although not strictly a compression algorithm, the QSplat rendering system offers a compact representation of the hierarchy structure [13]. A high quality rendering is obtained despite a strong quantization.

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

- approximation: MLS, *etc.*,
- overhead pre-processing: point re-ordering, triangulation, mesh generation, *etc.*,
- data structure: spanning tree, octree, *etc.*,

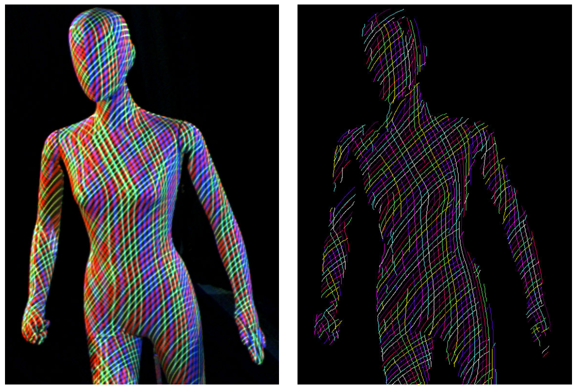
which leads to either smoothing out sharp features, an increase of the complexity, or an extra-transmission of a data structure. In the next section, we discuss our proposed general framework that does not need any of the aforementioned processes.

III. Curve-driven Point Cloud Coding

The proposed framework compresses the points as ordered prior to the way the projected grid pattern, consist of vertical and horizontal lines, has been extracted. The object points are then fitted in curves as shown in Fig. 2. This naturally motivates our choice to take advantage of this curve-like point organization. Unlike previous work in geometry compression, we do not need any overhead pre-processing, connectivity information, or data augmentation by a data structure (*e.g.* octree), to reach a satisfactory compression efficiency.

A. 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 $p_{k_{1\leq k \leq N}}$. As mentioned earlier, grid-pattern-based 3D scanning systems output the sampled



(a) captured image (b) sampled surface

Fig. 2. (a) A grid pattern is projected onto a dummy, (b) the surface object is then sampled in series of curves.

points embedded in curves. The point cloud $\mathcal{S}$ can then be represented as a set of M curves $\mathcal{C}^{l_{1\leq l \leq M}}$ 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

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

The partitioning of the point cloud in set of curves is directly obtained from the line detection algorithm in the acquisition process [4].

B. 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

$$\mathcal{C}|_{i<k} = \mathcal{C} \cap \{p_i | i < k\}, \tag{3}$$

and the intra-curve prediction by

$$\widehat{p_k} = P\left(\mathcal{C}|_{i<k}\right). \tag{4}$$

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 [11], 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. In the following, we present the different designed prediction modes.

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.

$$P^{Intra}\left(\mathcal{C}|_{i<k}\right) = (0,0,0). \quad (5)$$

2) Const P^{Const}: The previous coded point in the curve is used as prediction.

$$P^{Const}\left(\mathcal{C}|_{i<k}\right) = \widetilde{p}_{k-1}. \quad (6)$$

3) Linear P^{Linear}: The prediction is based on the two previous coded point in the curve.

$$P^{Linear}\left(\mathcal{C}|_{i<k}\right) = 2\cdot\widetilde{p}_{k-1} - \widetilde{p}_{k-2} \quad (7)$$

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.

$$\begin{aligned} P^{FitLine}\left(\mathcal{C}|_{i<k}\right) = \; & 2\cdot < \mathcal{L}\left(\mathcal{C}|_{i<k}\right)\perp\widetilde{p}_{k-1} > \\ & - < \mathcal{L}\left(\mathcal{C}|_{i<k}\right)\perp\widetilde{p}_{k-2} > \end{aligned} \quad (8)$$

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

5) Fit a sub-line $P^{FitSubLine}$: As previously, a line fitting algorithm is used to perform the prediction, but a sub-curve $\mathcal{C}|_{i_0\leq i<k}$ 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.

$$\begin{aligned} P^{FitSubLine}\left(\mathcal{C}|_{i<k}\right) = \; & 2\cdot < \mathcal{L}\left(\mathcal{C}|_{i_0\leq i<k}\right)\perp\widetilde{p}_{k-1} > \\ & - < \mathcal{L}\left(\mathcal{C}|_{i_0\leq i<k}\right)\perp\widetilde{p}_{k-2} > \end{aligned} \quad (9)$$

C. 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

$$\widetilde{r_k} = \mathrm{sign}(r_k)\cdot\mathrm{round}\left(|r_k| * 2^{bp-1}\right) \quad (10)$$

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

D. 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.

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.

(a) model Man (b) Model Kick (c) Model Squirt

Fig. 3. Test models.

TABLE I

COMPRESSION RATES IN BITS PER POINT (BPP) AFTER 12 BITS QUANTIZATION PER COORDINATE.

model	nb points	nb curves	rate	bit savings	PSNR
Man	19k	1325	15.75 bpp	-83.58 %	74.77 dB
Kick	18k	1634	15.71 bpp	-83.63 %	77.07 dB
Squirt	26k	2497	14.91 bpp	-89.83 %	75.75 dB

E. Decoding

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

IV. Experimental Results

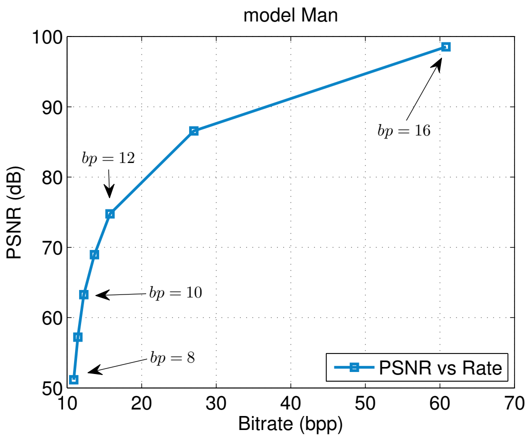
The performance of the proposed framework is evaluated using the three models shown in Fig. 3. The objective compression performance of the proposed method is investigated in the rate-distortion (RD) curves plotted in Fig.4 through the average number of bits per points (bpp), in relation to the loss 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.

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

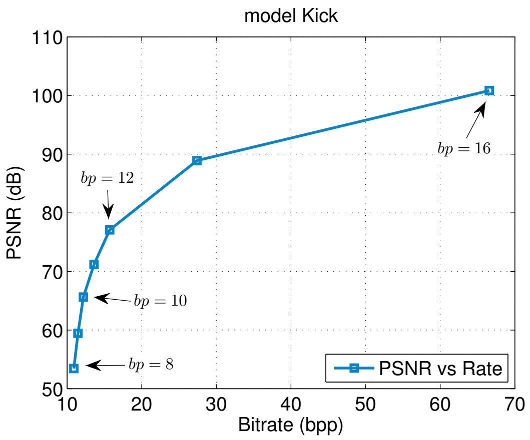
We also provide the distribution of prediction modes over each 3D models. It is observed that the *FitSubLine* prediction mode is mostly selected. Nonetheless, the adopted competitive strategy between all prediction modes ensure a higher quality 3D reconstruction. In the proposed method, 3 bits are consumed to design the prediction *FitSubLine* parameter p_0, which can be: p_{k-5}, p_{k-6}, p_{k-7}, p_{k-8}, p_{k-10}, p_{k-12}, p_{k-16} and p_{k-20} wrt the current point p_k to encode.

V. Conclusion

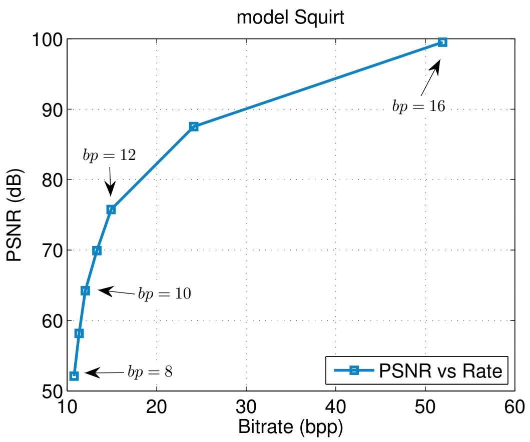
We designed and implemented a competition-based predictive single-rate compression for the positions points outputted



(a)



(b)



(c)

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

TABLE II

SELECTION PERCENTAGE OF EACH PREDICTION MODE.

model	*Intra*	*Const*	*Linear*	*FitLine*	*FitSubLine*
Man	1.80 %	5.12 %	30.94 %	5.11 %	**57.03 %**
Kick	1.52 %	4.12 %	29.23 %	4.03 %	**61.10 %**
Squirt	1.79 %	4.57 %	30.89 %	5.11 %	**57.63 %**

by a grid-pattern-based 3D scanning system. The compression is achieved by exploiting the inherent spatially organization of the points fitted in curves. While our method has the advantage to not require any overhead pre-processing, surface approximation, or the transmission of a data structure information (*e.g.* octree, spanning tree), the curve-driven compression allows to support random access and error propagation limitation.

Several issues remain that warrant further research. In future studies, we intend to design a rate-distortion framework, integrate other point attributes (*e.g.* color, normal, *etc.*). And finally extend this work to arbitrary 3D scanning systems.

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] R. Furukawa, H. Q. H. Viet, H. Kawasaki, R. Sagawa, and Y. Yagi, “One-shot range scanner using coplanarity constraints,” in *Proc. of the IEEE International Conference on Image Processing (ICIP)*, San Diego, CA, USA, Oct. 2008, pp. 1524–1527.

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

[4] 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.

[5] 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.

[6] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, “Computing and rendering point set surfaces,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 1, pp. 3–15, 2003.

[7] 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.

[8] 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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.