Dynamic Compression of Curve-based Point Cloud

Ismael Daribo[1], Ryo Furukawa[1], Ryusuke Sagawa[2], Hiroshi Kawasaki[3], Shinsaku Hiura[1], and Naoki Asada[1]

[1] Faculty of Information Sciences, Hiroshima City University, Hiroshima, Japan
[2] National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan
[3] Faculty of Engineering, Kagoshima University, Kagoshima, Japan
[1]{daribo,ryo-f,hiura,asada}@hiroshima-cu.ac.jp
[2]ryusuke.sagawa@aist.go.jp
[3]kawasaki@ibe.kagoshima-u.ac.jp

Abstract. With the increasing demands for highly detailed 3D data, dynamic scanning systems are capable of producing 3D+t (*a.k.a.* 4D) spatio-temporal models with millions of points recently. As a consequence, effective 4D geometry compression schemes are required to face the need to store/transmit the huge amount of data, in addition to classical static 3D data. In this paper, we propose a 4D spatio-temporal point cloud encoder via a curve-based representation of the point cloud, particularly well-suited for dynamic structured-light-based scanning systems, wherein a grid pattern is projected onto the surface object. The object surface is then naturally sampled in a series of curves, due to the grid pattern. This motivates our choice to leverage a curve-based representation to remove the spatial and temporal correlation of the sampled point along the scanning directions through a competitive-based predictive encoder that includes different spatio-temporal prediction modes. Experimental results show the significant gain obtained with the proposed method.

Keywords: Point cloud, compression, curve-based, dynamic, 4D, 3D+t, grid pattern

1 Introduction

Recent evolutions in acquisition technologies allow to produce 3D geometric models with millions of points that evolve over time (see Fig. 1). Problems of efficiently storing, transmitting, processing and rendering spatio-temporal data are then been raised, in addition to classical static 3D data. Towards an efficient compression performance, a suitable 4D data representation becomes particularly more and more important.

Currently active 3D scanners are widely used for acquiring 3D models [2]. Especially, scanning systems based on structured light have been intensively studied in the acquisition of dynamic scene [1, 3, 4], recently. Structured-light-based scanning is done by sampling the surface of an object with a known pattern (*e.g.* grid, horizontal bars, lines). Studying the deformation of the pattern

Fig. 1. Example of a 4D data acquired with a grid-pattern one-shot scanners [1].

allows building a 3D model by means of a point cloud. The huge amount of raw point data has to be stored/transmitted by efficient compact means. Noise and incompleteness, however, make the process more difficult to achieve. While mesh compression is a mature field [5], there is still room for improvement in point cloud compression. To the best of our knowledge, present point-based compression strategies are mainly based on surface approximation [6, 7], and/or hierarchical space subdivision by augmenting the dataset by a data structure (*e.g.* spanning tree [8, 9], octree [10, 11]), which lead to either smooth out sharp features, an extra-transmission of a data structure, or an unavoidable lossy encoding. In addition, the augmentation of the dataset by data structure makes difficult the exploitation of temporal consistency.

With the aim of tackling the aforementioned issues, we first made the observation that structured-light-based scanning systems output points along the measuring direction, which naturally orders groups of points along the same direction: scan lines. We particularly aim at structured-light-based scanning systems that use a grid pattern formed by straight lines distinguishable only as vertical and horizontal lines [1] as illustrated in Fig. 2. When the projected grid pattern is extracted from the captured image, 3D points are naturally fitted into a series of space curves. This motivates our choice to leverage the spatially sequential order of the sampled-points along these scan lines: first, we pre-process the data to retrieve each scan line into a curve of points, and after, we exploit the curve-based representation through a spatio-temporal competition-based predictive encoder specially designed with linear and curved-driven prediction modes. We then formulate the problem of encoding a point cloud as "How to retrieve the scan lines?", and after "How to encode a curve in space?", and then "How to encode a space curve over time?". This formulation has the benefit to simplify the former problem into sub-problems that allow application-oriented functionalities, such as:

- temporal prediction that consists of searching for a similar curve in previous temporal frames,
- random access that allows the decoding of a local part of the point cloud without the necessity of decoding the full dataset,
- error propagation limitation since all curves within a frame are independently encoded,

– possible lossless coding due to the predictive nature of the encoder,
– parallel computation where each curve is simultaneously encoded.

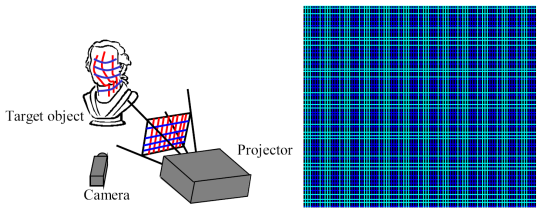


Fig. 2. (left) Grid-pattern-based scanning system: a grid pattern is projected from the projector and captured by the camera. (right) Example of projected grid pattern.

The rest of the paper is organized as follows. Section 2 describes the point cloud pre-processing towards a curve-based representation. Section 3 addresses the problem of efficiently compressing a raw point dataset, followed by the experimental results in Section 4. Finally, our final conclusions are drawn in Section 5.

2 Curve-based representation

In an arbitrary point cloud, the identification of neighbors is a nontrivial task. One approach consist in locally defining as neighbor the point that minimizes an error cost functional based on a prediction rule, which results in the augmentation of the dataset by a predictive data structure such as a spanning tree that also needs to be encoded and transmitted. However, for point cloud outputted by scanners using structured light there is a straightforward way through the scan lines as discussed before. Moreover, the projected grid-pattern makes each scan lines differentiable from others as shown in Fig. 3. It is then possible to address this problem in a more global way, at least at a scan line scale. In some cases, space curves can be directly obtained from the acquisition process, *e.g.* line detection algorithm [4].

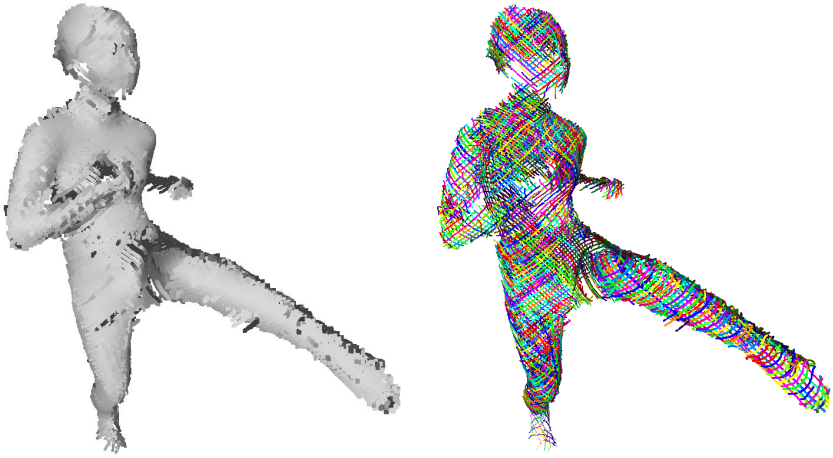
2.1 Curve-based point cloud definition

Let us consider the point cloud $\mathcal{S} = \{p_1, p_2, \cdots p_N\}$ as a collection of N 3D points $p_{k_{1\leq k\leq N}}$. 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 $\mathcal{C}^{l}{}_{1\leq l\leq M}$ 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

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



(a) Sampled surface (b) Curve-based representation

Fig. 3. Sampled point cloud partitioned into a series of curves wrt to the scanning directions. Curves are discriminate by different colors.

2.2 Curve-based partitioning

Each curve $\mathcal{C}$ is defined to contain points that share similar proprieties, *e.g.* curvature, direction, Euclidean distance with his neighbor, *etc.*. Algorithm 1 shows how the point cloud $\mathcal{S}$ is partitioned into a set of curves as defined in Equation (1). The division is controlled 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

$$d\left(p_k, p_{k-1}\right) > \epsilon, \tag{3}$$

$$\text{with} \quad \epsilon = \frac{1}{N-1} \sum_{i=2}^{N} d\left(p_i, p_{i-1}\right).$$

The current point p_k is considered as an outlier and then added to a new curve, if the Euclidean distance $d(.,.)$ is larger than a defined threshold ϵ: here the average value of the distance between two consecutive points throughout the point cloud. In general, 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. Another example is the use of a multiple of the inter-quartile-range (IQR) value of the current curve as a threshold. In this study we consider the Euclidean distance that gives a satisfactory curve set representation.

3 Point cloud encoding

After pre-processing the point cloud data to leverage the spatial order of samples along the scanning directions, we propose applying well-known hybrid-video-encoding techniques: spatio-temporal predictions followed by residual coding. In

Algorithm 1: The partitioning algorithm.

Input: set of points $\mathcal{S}^*$
Output: set of curves $\mathcal{S}$
Data: current curve $\mathcal{C}$

foreach *point* p_k *in* $\mathcal{S}^*$ **do**
  if `size` $(\mathcal{C})$ *is lower than 3* **then**
    `add` p_k in $\mathcal{C}$;
  else if *true* == `isAnOutlier` $(\mathcal{C}, p_k)$ **then**
    `add` $\mathcal{C}$ into $\mathcal{S}$;
    `clear` $\mathcal{C}$;
    `add` p_k in $\mathcal{C}$;
  else
    `add` p_k into $\mathcal{C}$;
  end
end
if *true* == `isEmpty` $(\mathcal{C})$ **then**
  `add` $\mathcal{C}$ into $\mathcal{S}$;
end

opposition to present predictive strategies [12, 8, 9] wherein a spanning tree is built, prior to only one predictive rule, we propose using multiple spatio-temporal predictors that allow sidestepping the utilization of a spanning tree, and moreover, providing an easier and intuitive way to perform temporal prediction. The proposed framework can be denoted as a competition-based predictive encoder in relation with the competition between all spatio-temporal prediction modes for each point.

Let us first introduce some notations. Let $\mathcal{C}^t$ be the current curve to encode at time t. Each point p_k^t in $\mathcal{C}^t$ is predicted by the prediction $\widehat{p}_k^t$ with respect to the previous coded point $\widetilde{p}_i^j$ with $j \leq t$ and $i < k$. Note that previous coded points have been quantized and inverse quantized. The prediction unit outputs then the quantized corrective vector $r_k^t = p_k^t - \widehat{p}_k^t$, 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. The prediction that minimizes the quantized Euclidean distance $\mathcal{Q}\left(\|p_k - \widehat{p}_k^t\|\right)$ is defined as the best one. $\mathcal{Q}$ being the quantization operator defined in Section 3.3. Then for each point the chosen prediction mode is signaled in the bitstream. The following predictions obey the two assumptions: closed points within a curve either evolve in the same direction, or turn constantly.

3.1 Intra-prediction

Intra-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 the same curve $\mathcal{C}$. 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 prediction by

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

Note that in this section, we have not made explicit that $\mathcal{C}$ and $\widehat{p_k}$ are function of t for notation concision.

No-prediction P^{Intra} When no-prediction is applied, this defines the current point as a key point used for random access and error propagation limitation.

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

Const P^{Const} The previous coded point p_{k-1} in the curve is used as predictor.

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

Linear P^{Linear} The prediction is based on the two previous coded point p_{k-1} and p_{k-2} in the curve, assuming then that p_{k-2}, p_{k-1} and p_k belong to the same line.

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

Fit-a-sub-line $P^{FitSubLine}$ The prediction point is an extension of a segment $\mathcal{L}\left(\mathcal{C}|_{i_0 \leq i<k}\right)$. The segment is given by line fitting algorithm based on the M-estimator technique, that iteratively fits the segment using the weighted least-squares algorithm. The starting point p_{i_0} has to be signaled to the decoder, and thus, an additional flag is put in the bitstream.

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

Turning-angle $P^{Turning}$ The current point p_k is predicted under the assumption that the curve is turning constantly around the point p_{k-1}. Given the displacement vector $\mathbf{v_k} = p_k - p_{k-1}$ between two consecutive points, the assumption is equivalent to consider equal the turning angles between two consecutive displacement vectors. The turning angle in 3D space between two vectors being defined by the triplet angles $\alpha\left(\alpha_x, \alpha_y, \alpha_z\right)$ as the difference of their direction angles. Given a vector $\mathbf{v_k}(v_{kx}, v_{ky}, v_{kz})$, his direction angles $\theta\left(\theta_{v_k x}, \theta_{v_k y}, \theta_{v_k z}\right)$ are expressed by

$$\cos\left(\theta_{v_k x}\right) = \frac{v_{kx}}{\|\mathbf{v_k}\|}, \quad \cos\left(\theta_{v_k y}\right) = \frac{v_{ky}}{\|\mathbf{v_k}\|}, \quad \cos\left(\theta_{v_k z}\right) = \frac{v_{kz}}{\|\mathbf{v_k}\|}. \tag{10}$$

The prediction of the current point p_k can then be expressed as

$$P^{Turning}\left(\mathcal{C}|_{i<k}\right) = \widetilde{p}_{k-1} + \mathcal{R}\left(\alpha_{k-1}\right) \cdot \mathbf{v_{k-1}}, \tag{11}$$

where $\mathcal{R}(\alpha_{k-1}) \cdot \mathbf{v_{k-1}}$ being the 3D rotation of $\mathbf{v_{k-1}}$ wrt the turning angle

$$\alpha_{k-1} = \left(\theta_{v_{k-1}} - \theta_{v_{k-2}}\right). \tag{12}$$

3.2 Temporal-prediction

The temporal prediction can be decomposed in two steps: first, find a close curve in previous frames in terms of distance or similarity, and after, use either the direction or curvature of the optimal curve $\mathcal{C}^{t-1}$ to predict the current point p_k^t in $\mathcal{C}^t$. We then highlight the two main strategies:

- find the closest curve in previous frames by segment matching,
- find the most similar curve with closest Euclidean invariant signature in previous frames,

which result in temporal prediction modes presented thereafter.

Close P^{Close} Between two consecutive frames, points occupy roughly the same position due to the small amount of motion, or the presence of static objects in the scene. It is then possible to find an optimal curve $\widehat{\mathcal{C}}^{t-1}$ in the previous frame that can be superposed with the current one $\mathcal{C}^t$ such that

$$\widehat{\mathcal{C}}^{t-1} = argmin_{\mathcal{C}^{t-1}} \left\{ d\left(\mathcal{C}^{t-1}|_{i_0 \leq i < k}, \mathcal{C}^t|_{i_0 \leq i < k}\right) \right\} \tag{13}$$

where the distance between the two curves is expressed by

$$d\left(\mathcal{C}^{t-1}|_{i_0 \leq i < k}, \mathcal{C}^t|_{i_0 \leq i < k}\right) = \sum_{i_0}^{k-1} d\left(\widehat{p}_i^{\,t-1}, \widehat{p}_i^{\,t}\right). \tag{14}$$

The points in $\widehat{\mathcal{C}}^{t-1}$ are then utilized to predict the current point as follows:

$$P^{Close}\left(\mathcal{C}^t|_{i<k}\right) = \widehat{p}_k^{\,t-1} \tag{15}$$

For instance, other alternatives may consist in considering instead $\widehat{p}_{k-1}^{\,t-1}$ or the mean point $\frac{1}{2}\left(\widehat{p}_{k-1}^{\,t-1} + \widehat{p}_k^{\,t-1}\right)$.

Similar $P^{Similar}$ By finding a similar curve in terms of shape prior to a close Euclidean invariant signature, we assign the same turning angle, as defined previously, to the predictor. Let us first define the signature of a space curve, up to a Euclidean transform, by its curvature function $\kappa(n)$ and torsion function $\tau(n)$, both functions of the parameter n. It was shown in [13, 14] that $\kappa(n)$ and $\tau(n)$ can be approximated at the point p_k by

$$\kappa(p_k) = \pm 4 \cdot \frac{\sqrt{s \cdot (s-a) \cdot (s-b) \cdot (s-c)}}{a \cdot b \cdot c}, \tag{16}$$

$$\tau(p_k) = \pm 6 \cdot \frac{H}{d \cdot e \cdot f \cdot \kappa(p_k)}. \tag{17}$$

where H being the height of the tetrahedron form by $p_{i-1}, p_i, p_{i+1}, p_{i+2}$ of base p_{i-1}, p_i, p_{i+1}, and

$$\begin{aligned} a &= d(p_{k-1}, p_k), & b &= d(p_k, p_{k+1}), & c &= d(p_{k-1}, p_{k+1}), \\ d &= d(p_{k+i}, p_{k+2}), & e &= d(p_k, p_{k+2}), & f &= d(p_{k-1}, p_{k+2}), \end{aligned}$$

and $s = \frac{1}{2}(a+b+c)$. Since κ and τ only depends on the Euclidean distance $d(.,.)$ between points, they provide a completely Euclidean invariant numerical signature approximation. The turning angles of the curve having the closest signature is then applied such that

$$P^{Similar}\left(\mathcal{C}^t|_{i<k}\right) = \widetilde{p}_{k-1}^{\,t} + \mathcal{R}\left(\alpha_k^{\,t-1}\right) \cdot \mathbf{v_{k-1}^t}, \tag{18}$$

where

$$\alpha_k^{\,t-1} = \left(\theta_{v_k^{\,t-1}} - \theta_{v_{k-1}^{\,t-1}}\right). \tag{19}$$

3.3 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

$$\mathcal{Q}\left(r_k\right) = \widetilde{r_k} = \text{sign}(r_k) \cdot \text{round}\left(|r_k| * 2^{bp-1}\right) \tag{20}$$

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

3.4 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: a header containing the canonical Huffman codeword lengths, the quantization parameter bp, the total number of points and the residual data for every point; wherein 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 Experimental results

The performance of the proposed framework is evaluated using the two models shown in Fig. 4 over 24 frames. We defined a group of frames consisting of

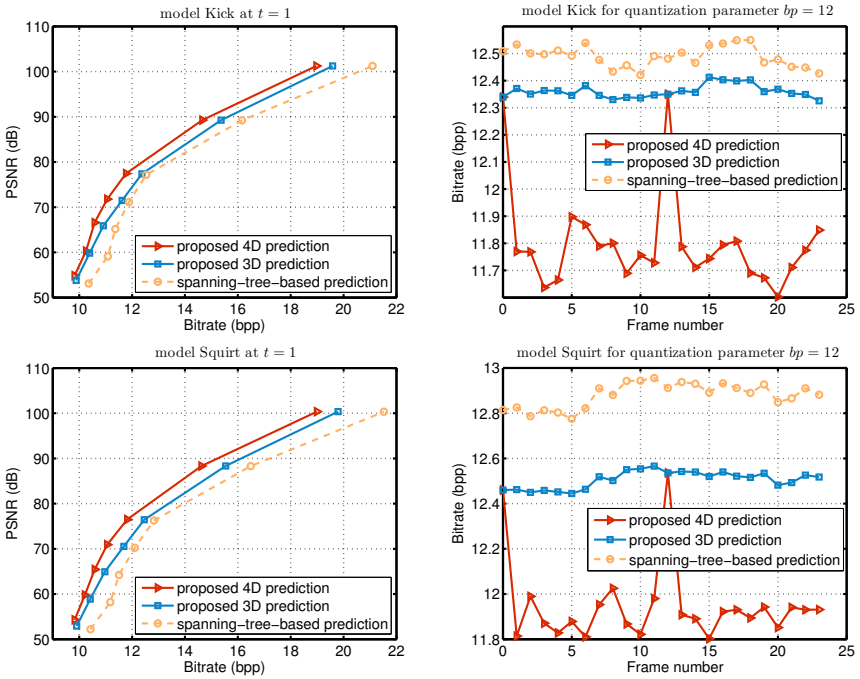
(a) Kick (b) Squirt

Fig. 4. Test models.

twelve frames wherein the first frame is intra-only predicted to limit temporal artifact propagation. The objective compression performance of the proposed method is investigated in the rate-distortion (RD) curves plotted in Figure 5 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 intra-only (*a.k.a.* 3D) competitive-optimized strategy and its temporal extension (*a.k.a.* 4D) with the spanning-tree-based strategy [8]. It can be observed that the proposed method provides better RD results in both cases. Experimental results highlight the advantage of competing multiple predictors instead of the spanning tree strategy optimized for only one predictor. It can be observed in Fig. 6 the average distribution of the different prediction modes at different quantization parameter *bp*. It is important to note that within the prediction unit, the utilized previous coded points have been quantized and inverse quantized, which results in more or less quantization error wrt the quantization parameter *bp*. Fig. 6 illustrates that at strong quantization (*i.e.* low *bp*) temporal predictions are more efficient, while at weak quantization intra-linear prediction are over utilized. Ideally, the choice of the prediction mode should be optimized in a rate-distortion sense, which will be left as future work.

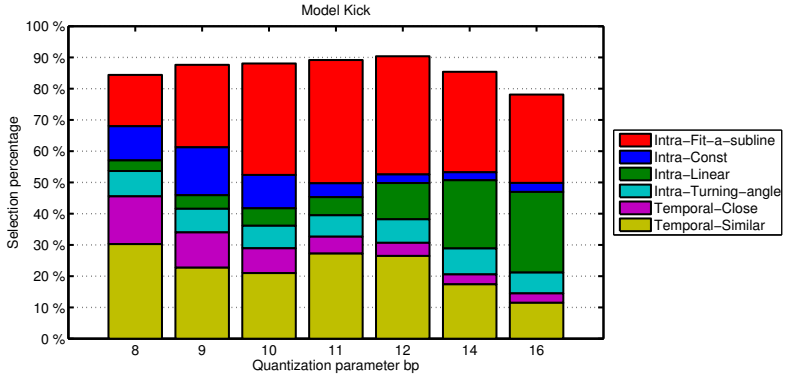
5 Conclusion

We designed and implemented a 3D+t (*a.k.a.* 4D) predictive single-rate encoder for point positions outputted by structured-light 3D scanning systems using a grid pattern. We pre-processed the point cloud to leverage points along a series of curves to exploit the spatio-temporal organization of the points, which are ordered prior to the scanning direction. By using curves as a coding unit we succeed in improving the rate-distortion performance, while enabling application-

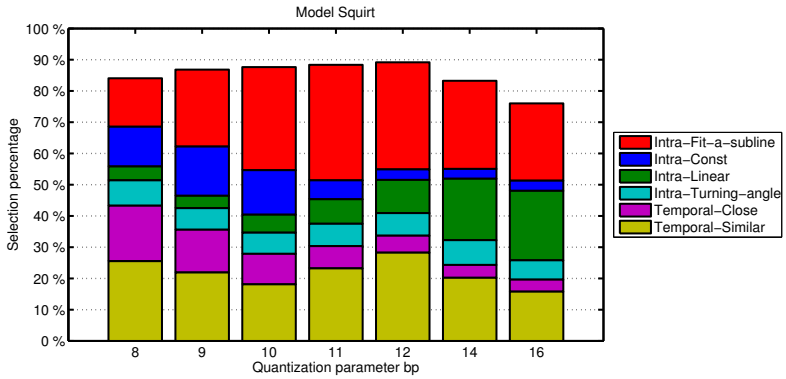


(a) Rate vs. distortion at $t = 1$ (b) Rate evolution over time at $bp = 12$

Fig. 5. Rate-distortion performance of the proposed encoder, using 4D and 3D prediction strategy. For comparison, we also show results for classical spanning-tree-based encoder [8]. Experiments are done for 24 frames, where frame at $t = 0$ and $t = 12$ are intra-only encoded.



(a)



(b)

Fig. 6. Example of average prediction mode distribution at different quantization parameters *bp*.

orientated features such as random access, error propagation limitation. 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.

References

1. R. Furukawa, R. Sagawa, H. Kawasaki, K. Sakashita, Y. Yagi, and N. Asada, "One-shot entire shape acquisition method using multiple projectors and cameras," in *Proc. of the Pacific-Rim Symposium Image and Video Technology (PSIVT)*, Singapore, Dec. 2010, pp. 107–114.
2. 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.
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. E. Hubo, T. Mertens, T. Haber, and P. Bekaert, "Self-similarity-based compression of point clouds, with application to ray tracing," in *Proc. of the IEEE Eurographics Symposium on Point-Based Graphics*. Prague, Czech Republic: Eurographics Association, Sep. 2007, pp. 129–137.
8. 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.
9. 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.
10. 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
11. 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.
12. G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transaction on Graphics*, vol. 17, pp. 84–115, Apr. 1998.
13. E. Calabi, P. Olver, C. S. A. Tannenbaum, and S. Haker, "Differential and numerically invariant signature curves applied to object recognition," *International Journal of Computer Vision*, vol. 26, pp. 107–135, 1998.
14. M. Boutin, "Numerically invariant signature curves," *International Journal of Computer Vision*, vol. 40(3), pp. 235–248, 2000.