Pacific Graphics (2011)
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin (Editors)

Feature Adaptive Surfel Set Simplification

Yan Huang †1, Feihu Yan[1], Bei Wang[2] and Jingliang Peng ‡1

[1]School of Computer Science and Technology, Shandong University
[2]Intel Corporation

Abstract

In this paper we propose a novel scheme for simplifying a surfel set with the resultant surfels computed and distributed in an effort to preserve prominent geometric and textural features with a reduced number of primitives. It works by iteratively collapsing local neighborhoods around surfels until a desired data reduction ratio is reached. The local neighborhood collapses are prioritized according to a cost metric that takes into account the local complexities of both the geometric and the textural information. Experimental results demonstrate that, besides yielding outstanding model quality at a reduced model size, the proposed scheme provides the flexibility for the users to specify the relative priorities in the geometric and the textural fidelity.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

The use of points as primitives to display 3D surfaces was initially proposed by Levoy and Whitted [LW85]. Since the point-based representation gets rid of the need for maintaining and processing complex topological information, it is particularly suitable for representing highly detailed and densely sampled 3D surfaces. With the advances in 3D laser scanning technology and hardware processing power, the use of point primitives for 3D surface representation and rendering has been gaining increasing popularity in the past decade [PZvBG00, RL00, ABCO*01, KV03, AK04]. In this work, we are concerned with a prevalent form of point primitive, surfel, which is a circular disk with attributes including center location, normal vector, texture color and radius [PZvBG00, ZPKG02].

With the current 3D laser scanning technology, a point-based 3D model with hundreds of millions of point primitives can be easily acquired. It is still not feasible to render and manipulate those huge point-based models at an interactive speed using commodity computers. Therefore, simplification techniques have been investigated by researchers in order to reach a balance point between the point primitive count reduction and the simplified model quality.

† huang.yan74@gmail.com
‡ jingliap@gmail.com

The goal of this research is, for an input surfel set representing a 3D surface and at a desired data reduction ratio, to construct a simplified surfel set whose sufels are adaptively computed and distributed such that geometrical and/or textural features (*i.e.*, local regions with high variation of geometrical and/or textural information) are well preserved.

1.1. Main Contributions

In this work, we propose a feature adaptive surfel set simplification scheme whose main contributions reside in:

- **Geometric and textural feature adaptiveness.**Unlike most related works that focus on pure geometry simplification, we take the textural information into account as well in the simplification process. As a result, the proposed scheme preserves both prominent geometric and prominent textural features well at reduced data sizes.
- **Flexible user control**. Through parameter adjustment, users have the flexibility in controlling the relative degree of fidelity for the geometric and the textural features.

1.2. Related Work

In order to address the issue of data volume explosion, research in point-based model compression has been conducted as well. Typical works include [HPKG08, FCOAS03, WGE*04, OS04, HPKG06, PL09, SK06]. Although some of

DOI: 10.2312/PE/PG/PG2011short/043-048

them produce reduced levels of detail (LODs) during the progressive encoding and decoding processes, their focus is on the compact coding of the whole LOD sequence but not on how to reach an optimal balance point between the primitive count reduction and the approximation quality, which is in general the goal of point-based model simplification.

The previous works on 3D point-based model simplification can be divided into two classes: distortion-driven and cardinality-driven approaches. A distortion-driven approach produces primitives each approximating a local surface region under a distortion threshold while striving to reduce the number of primitives to the minimum. A cardinality-driven approach produces a given number of primitives while striving to keep the approximation quality at the maximum. Works belonging to the former and the latter class include [PG01, WK04, MD04, RT09, HL09] and [PGK02, SF06, SF08, SF09, YWPM10], respectively.

Pauly and Gross [PG01] proposed to segment the surface into patches each represented as a height field and resample the patches based on the spectral analysis. Wu and Kobbelt [WK04] proposed to generate a splat around each original point based on a flatness criterion, greedily select a subset of the splats to cover the whole surface and perform global relaxation to improve the regularity of patch distribution. Moenning and Dodgson [MD04] proposed to start with one sample and iteratively add new samples to the result based on the intrinsic farthest point sampling techniques. Reniers and Telea [RT09] proposed to aggressively segment an input surface into large patches using an algebraic multigrid algorithm, and represent each patch with a textured planar splat. The work by He and Liang [HL09] performs splat generation and selection to construct the simplified model, similar to [WK04], but utilizes the Moving Least Square (MLS) projection and avoids the relaxation procedure.

Pauly *et al.* [PGK02] extended polygonal mesh simplification techniques to point-based model simplification and made a comprehensive comparison. Song and Feng [SF08] proposed to partition the input point cloud into a fixed number of clusters and represent each cluster with a point minimizing the geometric deviation between the simplified and the original data sets. Later, Song and Feng [SF06, SF09] proposed an algorithm suitable for mechanical model simplification which identifies and preserves points on edges while iteratively removing non-edge points that can be reliably represented by their respective neighbors. Yu *et al.* [YWPM10] proposed a two-step approach to point-based model simplification: the K-means algorithm is utilized to first construct a cluster tree; leaf nodes satisfying a certain simplification criterion are iteratively collapsed until a desired data reduction ratio is reached.

All the above-mentioned simplification works seek for geometric fidelity of the simplified model. To the best of our knowledge, very few published works, if any, have tried to systematically reduce the degradation in both geometric and textural approximation quality in the context of 3D point-based model simplification, which is exactly the focus of this work.

2. Overview

Given an input surfel set, we first construct the neighbor relationship between the surfels in it. Each surfel and its neighbors form a local neighborhood for which a cost of collapse is computed. During the runtime, the local neighborhood with the minimum cost of collapse is always picked and collapsed at any iteration. The iteration stops when a desired data reduction ratio is reached.

Details of the key components of this work – the local neighborhood construction, the local neighborhood collapse cost computation and the local neighborhood collapse operation – will be given in the following sections.

3. Local Neighborhood Construction

We denote the input surfel set as $S^0 = \{s_i^0 | i = 1, \ldots, N^0\}$ where N^0 ($N^0 > 0$) is the cardinality of S^0. The i-th ($1 \leq i \leq N^0$) surfel s_i^0 is defined as a tuple of its position, $\mathbf{p}_i^0$, its normal, $\mathbf{n}_i^0$, its radius, r_i^0, and its color, $\mathbf{c}_i^0$, *i.e.*, $s_i^0 = (\mathbf{p}_i^0, \mathbf{n}_i^0, r_i^0, \mathbf{c}_i^0)$. In order to construct a surfel's neighborhood, we first need a criterion by which to determine if two given surfels, s_i^0 and s_j^0 ($1 \leq i, j \leq N^0, i \neq j$) are neighbors.

Since we work with surfels that are compactly tiled to cover a 3D surface, we propose to utilize both the location and the dimension information of surfels in the test of neighbor relationship between two surfels. Given two surfels, s_i^0 and s_j^0 ($1 \leq i, j \leq N^0, i \neq j$), they are neighbors if and only if $|\mathbf{p}_i^0 - \mathbf{p}_j^0| < r_i^0 + r_j^0$, *i.e.*, the distance between the centers is less than the sum of the radii of the two disks at s_i^0 and s_j^0.

In order to accelerate the local neighborhood construction, we subdivide the input model's axis-aligned bounding box into $32 \times 32 \times 32$ grids and, for each surfel, we only search for its neighbors within a local range of $3 \times 3 \times 3$ grids.

After the local neighborhood construction is completed, we compute for each surfel a cost of local neighborhood collapse. A heap data structure, H, is used to maintain the sorted list of all the surfels and their associated local neighborhoods, with the order determined by their local neighborhood collapse costs.

4. Local Neighborhood Collapse Cost

Humans' visual perception is usually sensitive to the variation in the curvature and the texture on a surface. Therefore, we base the local neighborhood collapse cost metric on the measurement of the variations in the normal and the color information of the surfels in a local neighborhood. High variation in normals and colors means sharp geometric and textural features, respectively, and should lead to high local neighborhood collapse cost.

We denote the surfel set at the current iteration as $S = \{s_i | i = 1, \ldots, M\}$ where M ($M > 0$) is the cardinality of S and a surfel s_k ($1 \le k \le M$) is represented as a tuple of its position, $\mathbf{p_k}$, its normal, $\mathbf{n_k}$, its radius, r_k, and its color, $\mathbf{c_k}$, *i.e.*, $s_k = (\mathbf{p_k}, \mathbf{n_k}, r_k, \mathbf{c_k})$. For a surfel, s_i ($1 \le i \le M$) with a local neighborhood $B_i = \{s_j \,|\, 1 \le j \le M, |\mathbf{p_j} - \mathbf{p_i}| < r_j + r_i\} \cup \{s_i\}$, the local neighborhood collapse cost, h_i, of B_i is defined as

$$h_i = a_i \cdot e_i \cdot [\alpha \cdot w_i \cdot f_i + (1-\alpha) \cdot (1-w_i) \cdot g_i] \qquad (1)$$

where a_i, e_i, f_i and g_i measures the neighborhood size, the surfel size regularity, the normal variation and the color variation for B_i, respectively, w_i is a weight adaptively determined based on the variation of normals and colors and α is a parameter input by the user to control the relative degree of fidelity between the geometric and the texture features.

4.1. Regularity of Surfel Sizes

When the surfels in a local neighborhood have similar sizes, the local surface can often be better approximated with a circular disk than when there is much variation of the surfel sizes. Thus, we incorporate into the collapse cost metric a factor of surfel size regularity, e_i, which is formulated as the ratio of the standard deviation to the the mean of the radii of the surfels in B_i, *i.e.*,

$$e_i = \frac{1}{\bar{r}_i} \sqrt{\frac{1}{|B_i|} \sum_{s_j \in B_i} (r_j - \bar{r}_i)^2} \qquad (2)$$

where $\bar{r}_i = \frac{1}{|B_i|} \sum_{s_j \in B_i} r_j$. The size of the neighborhood B_i is approximated with

$$a_i = |B_i| \pi \bar{r}_i^2 \qquad (3)$$

4.2. Normal and Color Variation

We formulate the normal variation of a local neighborhood collapse as the area-weighted standard deviation of the surfels' normals in this neighborhood normalized by 2, *i.e.*,

$$\begin{aligned} f_i &= \frac{1}{2} \sqrt{\frac{1}{\sum_{s_j \in B_i} (\pi r_j^2)} \sum_{s_j \in B_i} [\pi r_j^2 \cdot (\mathbf{n}_j - \bar{\mathbf{n}}_i)^2]} \\ &= \frac{1}{2} \sqrt{\frac{1}{\sum_{s_j \in B_i} r_j^2} \sum_{s_j \in B_i} [r_j^2 \cdot (\mathbf{n}_j - \bar{\mathbf{n}}_i)^2]} \end{aligned} \qquad (4)$$

where $\bar{\mathbf{n}}_i = \frac{\sum_{s_j \in B_i} (\pi r_j^2 \mathbf{n_j})}{\sum_{s_j \in B_i} (\pi r_j^2)} = \frac{\sum_{s_j \in B_i} (r_j^2 \mathbf{n_j})}{\sum_{s_j \in B_i} r_j^2}$.

Similarly, we formulate the color variation of a local neighborhood collapse as the area-weighted standard deviation of the surfels' colors in this neighborhood normalized by the maximum distance between the color vectors of two surfels in the input model, *i.e.*,

$$\begin{aligned} g_i &= \frac{1}{\max_{s_u^0, s_v^0 \in S^0} |\mathbf{c}_u^0 - \mathbf{c}_v^0|} \sqrt{\frac{\sum_{s_j \in B_i} [\pi r_j^2 \cdot (\mathbf{c}_j - \bar{\mathbf{c}}_i)^2]}{\sum_{s_j \in B_i} (\pi r_j^2)}} \\ &= \frac{1}{\max_{s_u^0, s_v^0 \in S^0} |\mathbf{c}_u^0 - \mathbf{c}_v^0|} \sqrt{\frac{\sum_{s_j \in B_i} [r_j^2 \cdot (\mathbf{c}_j - \bar{\mathbf{c}}_i)^2]}{\sum_{s_j \in B_i} r_j^2}} \end{aligned} \qquad (5)$$

where $\bar{\mathbf{c}}_i = \frac{\sum_{s_j \in B_i} (\pi r_j^2 \mathbf{c_j})}{\sum_{s_j \in B_i} (\pi r_j^2)} = \frac{\sum_{s_j \in B_i} (r_j^2 \mathbf{c_j})}{\sum_{s_j \in B_i} r_j^2}$.

4.3. Adaptive Weighting

In order to enhance the contribution from the more prominent type of features, we compute the weight, w_i, for the normal distortion variation term in Equation 1 as

$$w_i = \frac{f_i}{f_i + g_i} \qquad (6)$$

where f_i and g_i represents the degree of the surfel normal variation and the surfel color variation, respectively (see Equations 4 and 5). From Equations 1 and 6, we can see that the variation of the more prominent type of feature is weighted more in the computation of the collapse cost.

5. Local Neighborhood Collapse

Assuming that the local neighborhood B_i around the surfel s_i in the current sufel set S is picked for collapse, we need to compute a representative surfel $s = (\mathbf{p}, \mathbf{n}, r, \mathbf{c})$ for the surfels in B_i, and update the current surfel set and the local neighborhoods as necessary, details of which are given in Section 5.1 and Section 5.2, respectively.

5.1. Computation of Representative Surfel

5.1.1. Geometric Attributes Computation

We compute a best fitting plane, P, to the surfels in B_i through linear least squares regression based on their positions and sizes. The unit normal of P sets $\mathbf{n}$, the normal of s.

Denoting the projection of the center of surfel s_j ($s_j \in B_i$) on plane P as $\mathbf{p}'_j$, we compute the position, $\mathbf{p}$, and the radius, r of surfel s as:

$$\mathbf{p} = \frac{\sum_{s_j \in B_i} |r_j^2 \cdot \mathbf{n}_j \cdot \mathbf{n}_P \cdot \mathbf{p}'_j|}{\sum_{s_j \in B_i} |r_j^2 \cdot \mathbf{n}_j \cdot \mathbf{n}_P|} \qquad (7)$$

$$r = \max_{s_j \in B_i} (|\mathbf{p}'_j - \mathbf{p}| + |r_j \cdot \mathbf{n}_j \cdot \mathbf{n}_P|) \qquad (8)$$

where $\mathbf{n}_j$ and $\mathbf{n}_P$ denotes the normal of s_j and the normal of plane P, respectively.

5.1.2. Visibility-Based Color Computation

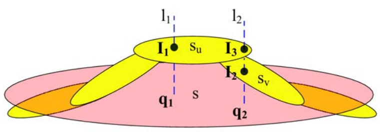

Figure 1: *Visibility test for surfel samples.*

The color, $\mathbf{c}$, of s could be straightforwardly computed as the area-weighted average of the colors of the surfels in B_i. However, a more accurate approach should weight the contribution of $\mathbf{c}_j$ in proportion to the visible area of s_j when it is viewed from the outside (which is often the case). We randomly sample the circular disk of the representative surfel, s, with M (which is proportional to the disk area) samples. Thereafter, through each sample, q_m ($1 \leq m \leq M$), we shoot a line l_m that is perpendicular to the disk and intersects with a subset, T_m, of the surfels in B_i. Assuming that the viewer looks at the model from far outside, we pick the surfel, if any, in T_m whose intersection with l_m is the closest to the viewer and denote its color as $\mathbf{c}_m$. If $T_m = \emptyset$, we set $\mathbf{c}_m = 0$. An example is shown in Fig. 1 where two lines l_1 and l_2 are shot from two sampling points, q_1 and q_2, which intersects with one (s_u) and two surfels (s_u and s_v) in the local neighborhood, respectively. Thus, $T_1 = \{s_u\}$ and $T_2 = \{s_u, s_v\}$. Since the intersection, I_2, of s_v with l_2 is occluded by the intersection, I_3, of s_u with l_2, $\mathbf{c}_2$ is set to be the color of s_u. We use a flag bit, b_m, to indicate whether T_m is empty or not. If it is, we set $b_m = 0$; otherwise we set $b_m = 1$. Finally, the color $\mathbf{c}$ of the representative surfel s, is computed as

$$\mathbf{c} = \frac{\sum_{m=1}^{M} b_m \cdot \mathbf{c}_m}{\sum_{m=1}^{M} b_m} \tag{9}$$

5.2. Updates of Surfel Set and Local Neighborhoods

With the newly computed representative, s, we replace all the surfels in B_i, *i.e.*, we update S to $S' = S - B_i + \{s\}$, remove from the heap, H, the entries corresponding to the surfels in B_i. All the neighbors of the surfels in B_i except those in B_i are set as neighbors of s. Thus, the neighborhood, B, of s is defined as $B = \bigcup_{s_j \in B_i} B_j - B_i$. As shown in Fig. 2, when the local neighborhood around s_i, as indicated by a shaded pink circle in Fig. 2(a), is collapsed, the local neighborhood is replaced with a representative surfel, s, and the neighborhood around s is shaded yellow as shown in Fig. 2(b). The collapse cost for B is then accordingly computed and a new entry corresponding to s is inserted to H.

In addition, for the set of surfels, $R_i = \bigcup_{s_j \in B_i} B_j - B_i$, in S around B_i, we decide for each surfel s_k ($s_k \in R_i$) whether its local neighborhood, B_k, needs to be updated. It should be noted that, although some surfel(s) in B_k has(have) been removed from S with the collapse of B_i, we still maintain their copies in B_k and we mark those surfels as virtual. As shown in Fig. 2(c), after the collapse of the local neighborhood around s_i (as illustrated in Fig. 2(a)), two but not all surfels in the local neighborhood of surfel s_k ($s_k \in R_i$) become virtual (as illustrated with dashed circles) and therefore B_k is unchanged (as illustrated with the green-shaded region). Only when all the surfels in B_k become virtual do we update it to the set, B'_k, of surfels including s_k and its neighbor surfels in the current surfel set. Otherwise, we keep the B_k unchanged. Correspondingly, only when B_k is updated do we re-compute its collapse cost and adjust its position in the heap, H, accordingly. As a result, the local neighborhood update overhead is significantly reduced while satisfying simplification results are still obtained in our experiments.

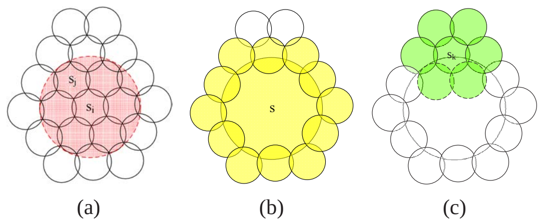

Figure 2: *Collapse of (a) s_i's local neighborhood B_i (enclosed in the shaded pink circle) leads to (b) newly constructed local neighborhood (shaded yellow) around the representative, s, and (c) updated local neighborhood (shaded green) of a surfel, s_j, around B_i.*

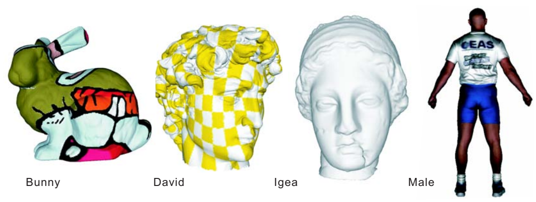

Figure 3: *Models used in our experiments.*

6. Experimental Results

We use four surfel set models in our experiments as shown in Fig. 3. They are Igea and Male by the courtesy of Pointshop 3D, and Bunny and David by the courtesy of Stanford Computer Graphics Laboratory. It should be noted that we obtained the surfel set models for Bunny and David from the original 3D meshes by subdividing each triangle into four sub-triangles and placing a surfel at the center of each. Igea, Male, Bunny and David contains 134345, 148138, 277888 and 323432 surfels, respectively.

We compare with one most recent state-of-the-art

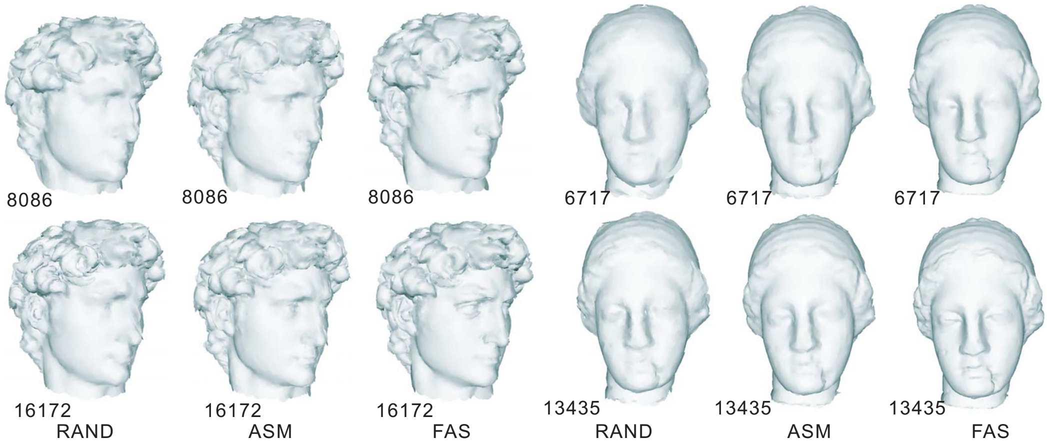

Figure 4: *Simplification results for pure geometric models.*

work [YWPM10], denoted as ASM, which is a cardinality-driven simplification method. It should be noted that the original ASM algorithm works on pure points. In our experiments, we apply the ASM algorithm to the positions of surfels and reconstruct a surfel set from the simplified point sets for the purpose of comparison. In addition, we implemented an algorithm, denoted as RAND, which is based on iterative local neighborhood collapse as well but randomly picks one local neighborhood to collapse at each iteration with no prioritization. Comparison between RAND, ASM, and our algorithm (denoted as FAS) is made in our experiments. In addition, we demonstrate the flexibility of FAS in user control.

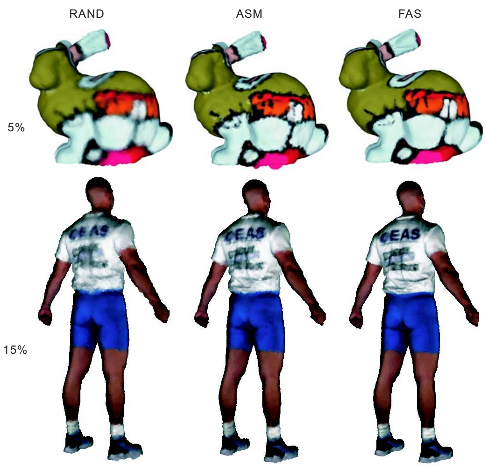

Figure 5: *Simplification results for textured models.*

6.1. Pure Geometric Model Simplification

As shown in Fig. 4, we experimented on two surfel sets (David and Igea) with only geometric information using RAND, ASM and FAS. The results shown in each column are obtained with one algorithm and the cardinality of each simplified model is marked besides it. From Fig. 4, we see that ASM yields better quality of simplified models than RAND, while FAS performs still better than ASM.

6.2. Textured Model Simplification

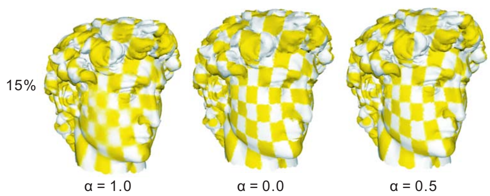

Figure 6: *Effect of α values on the simplified David models at the data reduction ratio of 15%.*

As shown in Fig. 5, we experimented on two surfel sets (Bunny and Male) with both geometric and textural information using RAND, ASM and FAS. The models in each column (row) are obtained with the same algorithm (data reduction ratio) as marked on the top (left). From Fig. 5, we see that ASM performs better in the preservation of textural detail than RAND, while FAS performs still better.

We also experimented on the textured David model with different α values in Equation 1. The results are shown in Fig. 6 which demonstrate that bigger (smaller) α values lead to better (worse) geometric approximation quality but worse(better) textural approximation quality. Therefore, the user may specify their relative priority in the preservation of geometric and textural features by adjusting the value of α.

6.3. Timing Statistics

Timing statistics for the simplification of the selected surfel sets are given in Table 1, which was obtained with our experiments on a PC platform with Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz 2.93GHz and 3.21GB RAM.

Table 1: *Timing statistics for the simplification of selected surfel sets at various data reduction ratios.*

	Igea	Male	David	Bunny
20%	9s	16s	46s	47s
10%	9s	16s	46s	67s

7. Conclusion

The novel surfel set simplification scheme proposed in this work is, to the best of our knowledge, the first that systematically reduces the degradation in both geometric and textural approximation quality. It works by iteratively picking the local neighborhood with the least cost to collapse until a desired data reduction ratio is reached. The proposed cost metric for local neighborhood collapse is based on the local variations in the geometric and the textural information, both of which provide important hints to the humans' visual perception. Furthermore, extra flexibility is provided for the users to control the relative priority in the preservation of the geometric and the textural features.

Acknowledgements

This work was supported by the Independent Innovation Foundation of Shandong University (Grant No. 2009TB017), National Natural Science Foundation of China (Grants No. 61070102, No. 61070103 and No. U1035004) and Program for New Century Excellent Talents in University.

References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *Proceedings of IEEE Visualization* (2001).

[AK04] AMENTA N., KIL Y. J.: Defining point-set surfaces. *ACM Transactions on Graphics 23*, 3 (2004), 264–270.

[FCOAS03] FLEISHMAN S., COHEN-OR D., ALEXA M., SILVA C.: Progressive point set surfaces. *ACM Transactions on Graphics 22*, 4 (2003), 997–1011.

[HL09] HE Z., LIANG X.: A novel simplification algorithm based on mls and splats for point models. In *Proceedings of Computer Graphics International 2009* (2009), pp. 45–52.

[HPKG06] HUANG Y., PENG J., KUO C.-C. J., GOPI M.: Octree-based progressive geometry coding of point clouds. In *Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics 2006* (2006), pp. 103–110.

[HPKG08] HUANG Y., PENG J., KUO C.-C. J., GOPI M.: A generic scheme for progressive point cloud coding. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 440–453.

[KV03] KALAIAH A., VARSHNEY A.: Modeling and rendering of points with local geometry. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (2003), 30–42.

[LW85] LEVOY M., WHITTED T.: The use of points as display primitives. In *Technical Report TR 85-022, University of North Carolina at Chapel Hill* (1985).

[MD04] MOENNING C., DODGSON N. A.: Intrinsic point cloud simplification. In *Proceedings of the 14th International Conference on Computer Graphics and Vision (GraphiCon)* (2004).

[OS04] OCHOTTA T., SAUPE D.: Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields. In *Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics 2004* (2004), pp. 103–112.

[PG01] PAULY M., GROSS M.: Spetral processing of point-sampled geometry. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 379–386.

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *Proceedings of the 13th IEEE Visualization Conference* (2002), pp. 163–170.

[PL09] PARK S.-B., LEE S.-U.: Multiscale representation and compression of 3-D point data. *IEEE Transactions on Multimedia 11*, 1 (2009), 177–183.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *Proceedings of ACM SIGGRAPH 2000* (2000), pp. 335–342.

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: a multiresolution point rendering system for large meshes. In *Proceedings of ACM SIGGRAPH 2000* (2000), pp. 343–352.

[RT09] RENIERS D., TELEA A.: Extreme simplification and rendering of point sets using algebraic multigrid. *Computing and Visualization in Science 12*, 1 (2009), 9–22.

[SF06] SONG H., FENG H.-Y.: A point cloud simplification algorithm for mechanical part inspection. *Information Technology for Balanced Manufacting Systems 220* (2006), 461–468.

[SF08] SONG H., FENG H.-Y.: A global clustering approach to point cloud simplification with a specified data reduction ratio. *Computer-Aided Design 40*, 3 (2008), 281–292.

[SF09] SONG H., FENG H.-Y.: A progressive point cloud simplification algorithm with preserved sharp edge data. *The International Journal of Advanced Manufacturing Technology 45*, 5-6 (2009), 583–592.

[SK06] SCHNABEL R., KLEIN R.: Octree-based point cloud compression. In *Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics 2006* (2006), pp. 111–120.

[WGE*04] WASCHBÜSCH M., GROSS M., EBERHARD F., LAMBORAY E., WÜRMLIN S.: Progressive compression of point-sampled models. In *Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics 2004* (2004).

[WK04] WU J., KOBBELT L.: Optimized sub-sampling of point sets for surface splatting. *Computer Graphics Forum 23*, 3 (2004), 643–652.

[YWPM10] YU Z., WONG H.-S., PENG H., MA Q.: ASM: An adaptive simplification method for 3D point-based models. *Computer-Aided Design 42*, 7 (2010), 598–612.

[ZPKG02] ZWICKER M., PAULY M., KNOLL O., GROSS M.: Pointshop 3d: An interactive system for point-based surface editing. *ACM Transactions on Graphics 21*, 3 (2002), 322–329.