

## Feature sensitive re-sampling of point set surfaces with Gaussian spheres

MIAO YongWei<sup>1\*</sup>, BÖSCH Jonas<sup>2</sup>, PAJAROLA Renato<sup>2</sup>,  
GOPI M.<sup>3</sup> & FENG JieQing<sup>4</sup>

<sup>1</sup>College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China;

<sup>2</sup>Department of Informatics, University of Zürich, Zürich CH-8050, Switzerland;

<sup>3</sup>Department of Computer Science, University of California, Irvine CA 92697-3425, USA;

<sup>4</sup>State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China

Received November 9, 2010; accepted March 3, 2012; published online July 19, 2012

**Abstract** Feature sensitive simplification and re-sampling of point set surfaces is an important and challenging issue for many computer graphics and geometric modeling applications. Based on the regular sampling of the Gaussian sphere and the surface normals mapping onto the Gaussian sphere, an adaptive re-sampling framework for point set surfaces is presented in this paper, which includes a naive sampling step by index propagation and a novel cluster optimization step by normalized rectification. Our proposed re-sampling scheme can generate non-uniformly distributed discrete sample points for the underlying point sets in a feature sensitive manner. The intrinsic geometric features of the underlying point set surfaces can be preserved efficiently due to our adaptive re-sampling scheme. A novel splat rendering technique is adopted to illustrate the efficiency of our re-sampling scheme. Moreover, a numerical error statistics and surface reconstruction for simplified models are also given to demonstrate the effectiveness of our algorithm in term of the simplified quality of the point set surfaces.

**Keywords** point set surfaces, feature sensitive re-sampling, Gaussian sphere, simplification, surface reconstruction

**Citation** Miao Y W, Bösch J, Pajarola R, et al. Feature sensitive re-sampling of point set surfaces with Gaussian spheres. Sci China Inf Sci, 2012, 55: 2075–2089, doi: 10.1007/s11432-012-4637-0

### 1 Introduction

Large-scale point set surfaces are now becoming popular in computer graphics due to their efficiency for representing complex highly detailed models and maintaining convincing level of realism in computer graphics applications [1,2]. However, the commonly uniform sampling results from 3D automatic scanning devices are not dependent on the surface intrinsic features, and hence the data sets always exhibit a large proportion of redundant information. Many applications, such as remote transmission for 3D point clouds, rapid surface reconstruction by implicit representations, and real-time performance required in entertainment and virtual reality, will benefit a lot from reduced input point sets. Moreover, due to their large memory requirement and high time complexity, efficiently processing such large scale point clouds is still facing great challenges in both shape modeling and real-time rendering. To mitigate these limitations, data reduction and simplification methods provide efficient solutions for subsequent processing tasks [3–5].

\*Corresponding author (Email: ywmiao@zjut.edu.cn)

On the other hand, reducing the data complexity while maintaining their intrinsic geometric features, i.e. feature sensitive re-sampling/re-meshing, is crucial for many applications in digital geometry processing [6–8]. In general, geometric feature regions, such as edges, ridges or valleys are always located in high curvature regions and typically characterized by discontinuities of the surface normal field [9–11]. The unit surface normals always vary sharply along these regions, and their variations can be measured by the derivatives of the Gaussian map from a differential geometry perspective [11,12].

It is widely recognized that an applicable simplification and re-sampling algorithm for point-based surfaces should possess the following properties. 1) Surface quality and user-controllable simplification error: the surface simplification quality is always high in terms of a user-defined quality criterion and it can also provide a convenient way to control the underlying surface simplification error; 2) Efficiency of the procedure: the simplification scheme should be computationally efficient for large-scale point set surfaces; 3) Geometric feature preservation: the simplification algorithm should preserve geometric features of the original data as well as possible.

To develop such a simplification algorithm, using the regular triangulation of the Gaussian sphere and the distribution of the surface normals over the Gaussian sphere, a Gaussian sphere based sampling scheme can be used to simplify the underlying point set surfaces. In a preliminary version of the Gaussian sphere based sampling scheme [13], we have given a theoretical analysis of shape isophotic error metric generated by Gaussian sphere based sampling. Extending from it, we will propose here a novel framework of re-sampling point set surfaces, which can efficiently generate feature sensitive non-uniformly distributed discrete sample points. Our contributions in this paper are summarized as follows.

- A feature sensitive adaptive re-sampling framework for point set surfaces is presented which can inherently preserve the geometric features of the underlying models;
- A novel ellipse-splatting technique is adopted for smooth rendering the simplified point sets;
- Our adaptive re-sampling scheme provides a convenient way to control the surface simplification error by setting different levels of Gaussian sphere subdivision;
- According to the detailed and extensive experimental results, our feature sensitive re-sampling scheme can provide low geometric simplification error and generally produce high quality re-sampling results.

The remainder of this paper is organized as follows: The related work about re-sampling point set surfaces is reviewed in Section 2. Using the adaptive neighbors and Gaussian sphere sampling, a feature sensitive re-sampling scheme is proposed in Section 3. In Section 4, a splat rendering technique is explained to efficiently render the generated simplified models. In Section 5, some experimental results for different point-sampled models are listed and an error analysis of our re-sampling scheme is given. Finally, conclusions are drawn and future research ideas are given in Section 6.

## 2 Related work

Many researchers have developed various techniques for point set surfaces representation [14–19] and simplification [3–5]. A thorough survey can be found in [2]. Here, only the most important techniques related to point-sampled surface simplification are discussed.

One of the most commonly used surface representations for point set surfaces is the moving least squares (MLS) surfaces [14,15]. Their MLS definition is based on the stationary points of an iterative projection operator, in which at each step a polynomial approximation of the local neighborhood is performed from a local planar parametrization. Furthermore, their algorithm gives a down-sampling scheme for representing point set surfaces, and creates a simplified point cloud that is a true subset of the original point set by ordering iterative point removal operations according to a surface approximation error. However, the algorithm involves a non-linear optimization step which is time consuming and memory demanding. It is a pure sub-sampling scheme which unnecessarily restricts potential sampling positions and leads to aliasing artefact.

Using the clustering technique and mesh-based simplification approaches, Pauly et al. [3] explored different approaches from mesh-based to point-based simplification. Clustering-based simplification methods

always split the point cloud into a number of disjoint subsets. The uniform incremental clustering approach is computationally efficient but is reported to cause high approximation errors. Similarly, the hierarchical clustering approach is memory and execution efficient but even in its adaptive version the approximation error is still high in general. Extending directly from the error-controlled mesh simplification scheme [20], Pauly et al. [3] also presented an iterative method to contract point pairs in a point set according to a quadric error metric and generated an optimal result in terms of average geometric accuracy. Moreover, due to the adaptive mean-shift clustering scheme, Miao et al. [21] proposed an adaptive re-sampling method for point-sampled geometry simplification. The generated sampling points are non-uniformly distributed and can account for the local geometric features in a curvature aware manner.

Moenning et al. [4] presented an intrinsic coarse-to-fine simplification algorithm with sampling density guarantee for point clouds. However, their algorithm requires the complicated computation of intrinsic geodesic Voronoi diagrams and does not guarantee any error bounds on surface approximation. Based on implicit radial basis function (implicit RBF) representation for point sets, Kitago et al. [22] presented a simplification technique using a cost function to prioritize points and then sub-sample the underlying point set.

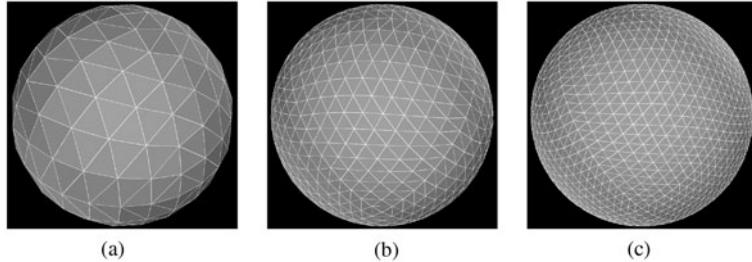
With regard to approximation error metric, Cohen-Steiner et al. [23] proposed a  $k$ -means clustering algorithm for geometric approximation of surfaces under the two definitions of shape approximation error metrics— $L^2$  metric and  $L^{2,1}$  error metric, which measure the generalized distance of a region to its respective proxy. The algorithm iteratively alternates between a geometry partitioning phase and a proxy fitting phase, and the approximation error is decreased by clustering faces into best-fitting regions iteratively. Similar to the  $L^2$  metric and  $L^{2,1}$  approximation error metric definitions of Cohen-Steiner et al. [23], Wu et al. [24] transferred the mesh shape approximation to shape approximation for splat-based geometry. They generate progressive splat representations based on iterative splat merge operations which are arranged in a priority queue according to the error metric. However, these methods are generally quite time-consuming or may lead to high approximation errors.

Recently, by attaching the field of unit normal vectors to the surface vertices as a vector-valued image, Lai et al. [8] presented the so-called image manifold, which associates with each surface point  $x \in \mathbb{R}^3$  a point  $x_f = (x, \omega n(x)) \in \mathbb{R}^6$ . Here, the non-negative weight  $\omega$  can regulate the amount of feature sensitivity of different operations. Then, using the corresponding feature sensitive metric of the image manifold, they proposed a unified framework for various geometry processing tasks. Applying isotropic surface remeshing and sampling scheme to the so-called image manifold, they can achieve feature sensitive remeshing and sampling results which can generate more vertices in highly curved areas than in flat ones. Lai et al. [9] also proposed a B-spline surface fitting approach using a feature sensitive parametrization of the underlying surface, which is generated from an area-preserving parametrization of its image manifold. Moreover, based on the mapping of regular sampling and triangulation of the Gaussian sphere onto a manifold surface, Diaz-Gutierrez et al. [25] applied results from quantization and surface approximation theory to propose a robust and output sensitive algorithm for re-sampling meshes. Their Gaussian sphere based sampling scheme is closely related to our re-sampling algorithm for point-sampled geometry. The sampling algorithm proposed by Diaz-Gutierrez et al. [25] depends heavily on the selection of feature edges and partitions the input triangular mesh into featureless regions. However, many of these simplification and re-sampling schemes are only applicable to polygonal meshes because of their dependence on globally consistent topological connectivity information. In contrast, our method is purely point-based, requiring only vertex positions and associated normals. This allows direct processing of scanned data without the need to construct polygonal meshes beforehand, making it particularly suitable for processing very large raw point models obtained with modern 3D range scanners.

### 3 Feature sensitive re-sampling point set surfaces

#### 3.1 Overview of our algorithm

Due to the superiority of  $L^{2,1}$  shape metric to the traditional  $L^2$  metric in capturing the subtle details and the anisotropy of the underlying surface [23,26], a Gaussian sphere based re-sampling scheme is proposed



**Figure 1** Gaussian sphere subdivision by different levels  $n = 8$  (a), 16 (b) and 24 (c), respectively.

in this section, which generates adaptively distributed discrete sample points in a feature sensitive way for the given point-sampled surfaces. As in [25], our algorithm takes the simple interpretation of the  $L^{2,1}$  shape metric on the point-sampled model to be the Euclidean distance of the normal vectors mapped onto the Gaussian sphere. First, the Gaussian sphere is regularly sampled and triangulated. This can be achieved by approximating the sphere by a recursive subdivision of an inscribed regular polyhedron [13]. In our terminology, the vertices and triangles of Gaussian sphere are called Gaussian vertices and Gaussian triangles. In contrast to [25], however, the key step of our algorithm is to find compact clusters  $\mathcal{C}_i$  of sample points within the same normal space on the Gaussian sphere and replace them by fewer points, usually one representative point  $p_i$  per cluster  $\mathcal{C}_i$ . According to a user specified subdivision of the Gaussian sphere (See Figure 1 for example), the neighboring sample points will belong to different clusters unless the images of their normals on the Gaussian sphere locate in the same Gaussian triangle, which bounds the  $L^{2,1}$  error distance between different clusters.

With the advent of robust normal estimation techniques [3,27] along with the estimated normal information from 3D scanning systems, most of the point based graphics algorithms can naturally assume discrete point data with attributes including normals as their input. So, our algorithm takes as input a set of unstructured surfels rather than a discrete point cloud. Moreover, the point clouds with highly non-uniform sampling or large noise cannot be treated well by our re-sampling algorithm. For these raw scanner data, some pre-processing steps [27] should be performed for subsequent feature sensitive re-sampling task. Now, the input surfels for our algorithm can be represented as clouds of point-normal pairs  $\{(p_i, n_i)\}$ . The high-level outline of our feature sensitive re-sampling algorithm based on the Gaussian sphere can be summarized as follows:

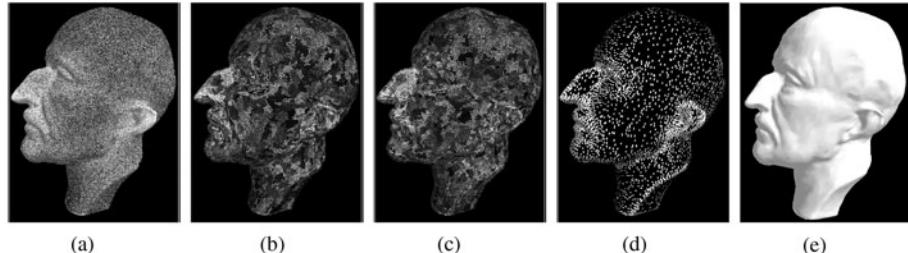
- 1) **Neighbor selection.** Adaptive neighborhoods for each sample point are determined according to the normal deviation;
- 2) **Naive sampling.** All sample points of the underlying model are clustered into regions using our index propagation scheme, and then the singleton points are merged into their nearest clusters;
- 3) **Optimized sampling.** Iterates over the following two steps until convergence,
  - Cluster normalization. The non-normalized clusters are split to generate normalized disk-like clusters;
  - Singleton rejoining. Singleton points which locate in the corner of Gaussian triangles are grouped with their nearest clusters;
- 4) **Simplified surfels generation.** Each cluster region is replaced by a representative surfel;
- 5) **Splat rendering.** The generated simplified models can finally be rendering by elliptic splats.

Our pipeline of feature sensitive re-sampling algorithm is illustrated in Figure 2.

### 3.2 Naive sampling: Clustering points by index propagation

#### 3.2.1 Defining neighbor graph

For each sample point  $p$  in the given point-sampled surface  $\mathcal{S}$ , the adaptive neighbors  $\mathcal{N}(p)$  are obtained by constraining the cone-of-normals as follows: the traditional neighborhood  $\mathcal{N}_k$  is first defined by  $k$ -nearest neighbors and an adaptive neighborhood of sample point  $p$  can then be selected as  $\mathcal{N}(p) = \{q \in \mathcal{N}_k | \langle n_p, n_q \rangle \geq 1 - \frac{1}{2}d_{\max}^2\}$ , where  $\langle ., . \rangle$  means the inner product between the normal vectors. According to the theoretical analysis of shape isophotic error metric [13], if providing  $n$ -levels for the Gaussian sphere subdivision, the maximum length of Gaussian edge on the sphere  $d_{\max}$  can be calculated as  $\sqrt{\frac{7\pi^2}{8(n-1)^2}}$ .



**Figure 2** Pipeline of our feature sensitive re-sampling algorithm. (a) Original uniform sampling of Max Planck model; (b) clusters after the naive index propagation step; (c) clusters after the optimized normalized rectification step; (d) feature sensitive sampling result of Max Planck model; (e) splat rendering result of simplified Max Planck model.

Then, in terms of the position information and neighborhood relation, an undirected non-symmetric abstract neighbor graph  $\mathbb{N} = (P, E)$  can be constructed, which represents the asymmetric neighbor information. In this neighbor graph, the edge  $(i, j)$  belongs to  $E$  if and only if sample point  $p_j \in \mathcal{N}(p_i)$ .

### 3.2.2 Index propagation using a stack data structure

Owing to the defined neighbor graph, the basic idea of our adaptive re-sampling algorithm for point-sampled surfaces is that neighboring sample points whose normal vectors lie inside one Gauss triangle are considered to be located in a featureless region of the underlying surface, and therefore should belong to one common cluster. On the other hand, two neighboring sample points whose normal vectors locate in different Gauss triangles should belong to two different clusters. The goal of our algorithm is to find a solution for labelling each graph vertex with an index and then cluster and replace the sample points with the same index by one representative surfel. So, our index labelling procedure should proceed according to the following criteria: for each graph edge, the two end points whose normals locate in one Gauss triangle should share the same index, while two end points whose normals locate in different Gaussian triangles should be labelled differently.

Now, we describe the main steps for index labelling. The algorithm first selects one seed point and indexes this current sample point. This initial index can then be propagated to the neighboring sample points whose normal directions map to the same Gaussian triangle. In our index propagation procedure, we adopt a stack data structure. The element of the stack records the index of the sample point and its neighbor information. The stack is initialized with an un-indexed seed point. Proceeding with our propagation procedure, we pop the top stack element to deal with. If this seed point has a direct and not yet labelled neighbor sample point and the normal direction of the neighbor is located in the same Gaussian triangle as the seed point, the index of this neighboring sample point can be inherited from the seed. Meanwhile, this neighbor point can be pushed back onto the stack to continue our propagation procedure. If the stack is empty, another un-indexed point can be chosen as a new seed for the above index propagation procedure. Our algorithm will terminate if all of the sample points are given an appropriate index. The index propagation procedure for point-sampled surface models is summarized in Algorithm 1.

### 3.2.3 Extra constraints for index propagation

The index propagation procedure can create many clusters of sample points with the same index. However, in order to avoid introducing badly formed clusters, some extra constraints for index propagation must be introduced. If the normals of two neighboring points are located near two opposite corners of one Gaussian triangle after mapping to the Gaussian sphere, then they possibly belong to two different clusters rather than fall into the same. Hence a distance constraint for normal directions is introduced to avoid this exceptional situation. In detail, the Euclidean distance between two normal directions should not exceed the radius of the inscribed circle in the Gaussian triangle. So, only if the two normal directions of two neighboring sample points belong to one Gaussian triangle and the Euclidean distance of normal directions is less than  $\frac{1}{2\sqrt{3}}d_{\max}$ , the index of the current sample point is propagated to the neighboring point. Here,  $d_{\max}$  means the maximum length of Gaussian edge on the sphere.

**Algorithm 1** Index propagation procedure

---

**Input:** A given point-sampled surface model  $\mathcal{S} = \{(p_k, n_k)\}_{k=1,2,3,\dots,n}$   
**Output:** The adaptive neighbors  $\mathcal{N}$  for every sample point in  $\mathcal{S}$

```

while There is a new un-indexed sample point  $p_l$  do
    Create new index label for  $p_l$ 
    Push  $p_l$  onto a stack as a root point
    while Stack is not empty do
        Pop the top point  $p_i$  from the stack
        for Each sample point  $p_j \in \mathcal{N}(p_i)$  do
            if Sample point  $p_j$  is a direct neighbor of the root point  $p_l$  and
                The normal directions  $n_j$  and  $n_l$  locate in the same Gauss triangle then
                    The label of  $p_j$  is the same as  $p_l$ 
                    Push  $p_j$  onto the stack
            end if
        end for
    end while
end while

```

---

On the other hand, in the case of large and almost planar regions, our sampling scheme may also generate suboptimal clusters. Hence a distance constraint for point positions is adopted, that is, a depth constraint of the recursive graph traversal in our index propagation procedure.

### 3.3 Singleton rejoining: Eliminating the singleton points

After our index propagation procedure, almost all sample points are given an appropriate index. However, in order to eliminate possible singleton points, an index rectification procedure should be performed to group these singleton points with neighboring clusters. For each singleton sample point  $p$  which locates near the corner of a Gaussian triangle, its neighbor clusters  $\mathcal{C}_i$  are first determined. The normal of each neighbor cluster is then taken as the weighted average of the normals of its attached points, and the combining score of the cluster  $\mathcal{C}_i$  can be assigned as  $1 - \langle n, n_i \rangle$ , where  $n$  and  $n_i$  denote the normals of the singleton sample point and the cluster respectively, and  $\langle ., . \rangle$  means the inner product between the normal vectors. Then, each singleton point can be grouped with the neighbor cluster with minimal score.

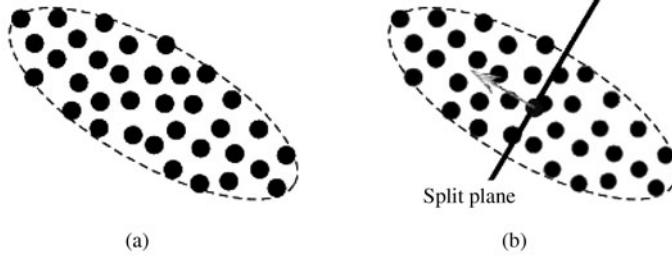
### 3.4 Cluster normalization: Generating normalized disk-like clusters

The clustering result by the naive sampling may generate some non-compact clusters; that is, the clusters deviate from the regular disk-like or common elliptical shapes. The aspect ratios of these clusters can also become very large or very small. So, in order to generate normalized disk-like clusters, we split these bad clusters into two according to their centroid and major extent axis.

For each non-normalized clusters, similar to [3] and [28], we can estimate the local properties of each cluster by the principal component analysis (PCA) [29]. However, unlike the normal estimation in [3], we take the two eigenvectors  $v_1$  and  $v_2$  corresponding to the two larger eigenvalues to represent the splat axes. Furthermore, the two large eigenvalues  $\lambda_1 \geq \lambda_2$  are adopted to estimate the normalization degree of the clusters, which is defined as the ratio of the two large eigenvalues, i.e., normalization degree =  $\lambda_1 / \lambda_2$ . If the normalization degree of one cluster exceeds a given threshold, the cluster will be split into two sub-clusters according to the split plane defined by the centroid point and perpendicular to the longest axis, that is, the eigenvector corresponding to the largest eigenvalue (see Figure 3). Then, these two procedures of singleton rejoining and cluster normalization will be iterated until the generated clusters satisfy the normalization criteria.

### 3.5 Simplified surfel generation: Generating the representative points

Often, point-based rendering methods [30,31] simply use circular oriented disks as object-space surfel representations. For each cluster obtained by the index labelling scheme, the representative surfel can



**Figure 3** Cluster normalization by covariance analysis. (a) Discrete sample points in one cluster; (b) split the cluster into two clusters by the split plane.

be obtained by minimizing the  $L^{2,1}$  metric [23,32]. In detail, the  $L^{2,1}$  metric  $L^{(p_i, n_i)}(p', n')$  for a splat  $(p', n')$  defined by cluster  $\mathcal{C}$  is defined as the normal deviation between the sample point and its proxy, i.e.,

$$L^{(p_i, n_i)}(p', n') = d^2(n', n_i) = (n' - n_i)^T (n' - n_i).$$

So, summing these  $L^{2,1}$  errors for all neighbor sample points belonging to this splat, the total  $L^{2,1}$  error metric for splat  $(p', n')$  can be described as

$$L^{2,1}(p', n') = \sum_{p_i \in \mathcal{C}} \omega_i L^{(p_i, n_i)}(p', n') = \sum_{p_i \in \mathcal{C}} \omega_i (n'^T n' - n'^T n_i - n_i^T n' + n_i^T n_i).$$

The minimum of the error metric  $L^{2,1}(p', n')$  can be found by the following gradient equation:

$$\frac{\partial L^{2,1}(p', n')}{\partial n'} = 0.$$

Now, for the sake of simplicity, if the weight  $\omega_i$  is taken independent from the surface normal, we can obtain the derivative of the error metric  $L^{2,1}(p', n')$  as

$$\frac{\partial L^{2,1}(p', n')}{\partial n'} = 2 \sum_{p_i \in \mathcal{C}} \omega_i (n' - n_i).$$

So, the minimization can be determined easily by the following equation:

$$n' = \frac{\sum_{p_i \in \mathcal{C}} \omega_i n_i}{\sum_{p_i \in \mathcal{C}} \omega_i},$$

which gives the normal of the representative surfel after normalization. Coincidentally, it is similar to the average normal definition in [33]. The weight function is always taken as  $\omega_i = \theta(\|p' - p_i\|)$ , where  $\theta$  is a monotonically decreasing function such as a Gaussian weight  $\theta(r) = e^{-r^2/h^2}$ .

Furthermore, the position information of a circular surfel is irrelevant to the  $L^{2,1}$  minimization. In our implementation, the position of the proxy is simply chosen as the centroid point of the sample points in the cluster, that is,  $p' = \frac{1}{k} \sum_i p_i$ .

However, comparing with the object-space circular surfel representations for simplified models, the elliptical splats can in general provide a better surface coverage [5,28,34]. Perspective accurate splatting of circular or elliptical splats has been treated in [35–38] using shaders and  $\alpha$ -textured polygons respectively. For a better display, in next section we will follow the approach of using elliptical surfels for improved surface coverage which are rendered using a fragment-shader based ellipse ray-caster similar to [35,39].

#### 4 Rendering of simplified point models by ellipse-splatting

Now, the above outlined re-sampling generates point position and normal information for each circular surfel. As in [5], for a given surfel  $(p', n')$  with sample point cluster  $\mathcal{C}$ , we define its elliptical extent based on the covariance analysis of the normals  $\{n_i | p_i \in \mathcal{C}\}$ . That is, we determine the principal curvature

directions, and thus the major and minor ellipse axes for the surfel, from the two (unit-length) eigenvectors  $v_1, v_2$  corresponding to the two largest eigenvalues  $\lambda_1 \geq \lambda_2$  of the covariance matrix  $\sum_{p_i \in C} (n_i - n') \cdot (n_i - n')^T$ , with  $n'$  being the weighted average of the normals. Correspondingly, the ellipse's aspect ratio  $\kappa$  is set to  $\lambda_2/\lambda_1$ . To achieve a good surface coverage we adjust the size of the surfel's area, i.e. the scale factor  $s$  of the ellipse's axes, to match the local sampling density.

Perspective accurate rasterization of object-space elliptical surfels is now achieved using a per-pixel ray-surfel intersection that is implemented using vertex and fragment shaders. Given the parametric ray  $t \cdot v$  through a pixel and the ellipse equation  $p + x \cdot e + y \cdot f$  (with surfel center  $p$  and major/minor axes  $e = sv_1$  and  $f = \kappa \cdot sv_2$ ) in camera coordinates, we must simply solve the system

$$[e \ f \ v] \cdot (x, y, -t)^T = -p \quad (1)$$

for each pixel. The vertex shader gets the surfel position  $p$ , normal  $n$ , major ellipse axis  $e$  and aspect ratio  $\kappa$  to compute  $f = \kappa(n \times e)$  and some per-surfel constants to solve the above ray-ellipse intersection equation. In the fragment shader, for a pixel  $v$  in camera coordinates, we eventually solve for  $x, y$  and  $t$ . A fragment is discarded if  $(\kappa x)^2 + y^2 > |e|^2$ , otherwise we set its  $z$ -depth to  $t \cdot v_z$ .

Then, smooth rendering of the elliptical surfels is performed analogously to standard point-rendering methods using a three-pass approach with  $\varepsilon$ -z-buffer culling and blending between overlapping surfels [30,31,34], although single-pass methods such as [40] could be used as well.

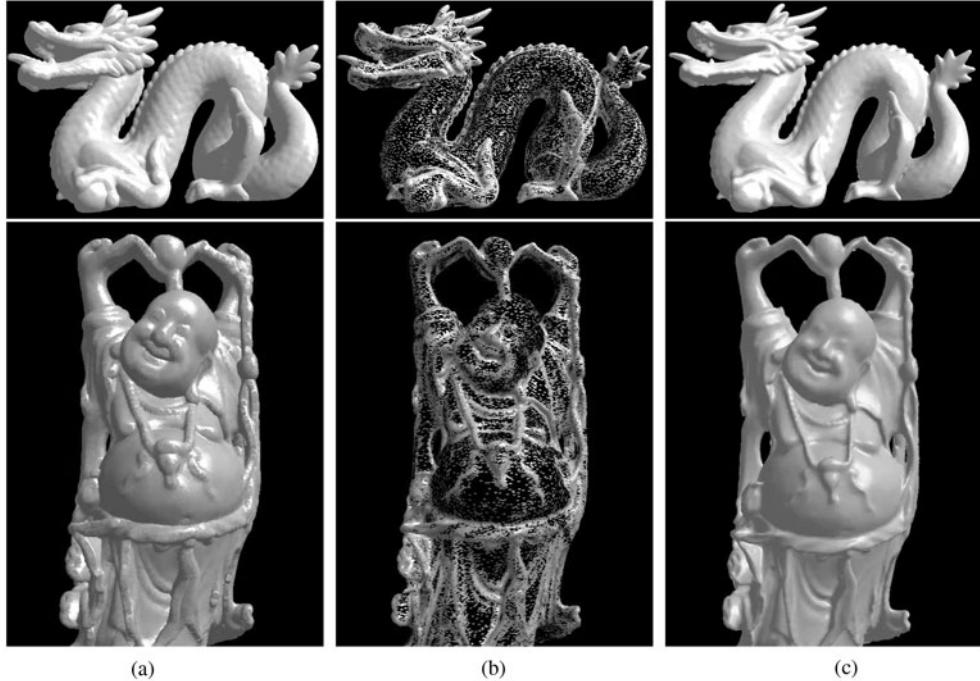
## 5 Results and discussion

All the algorithms presented in this paper were implemented and tested on a PC with a Pentium IV 3.0 GHz CPU, 1024 MB memory. Our re-sampling scheme is adaptive and can account for the local geometric features. It mainly consists of two sampling steps. One is naive sampling by index propagation. The other is optimized sampling, that is an iterative step for cluster normalization and singleton rejoining. In general, we just perform two or three iterations before the stopping criteria are fulfilled; that is, the variation of the number for simplified sample points is very few compared to the number of original sample points (for example, we can take less than 1.0% of the original sample points).

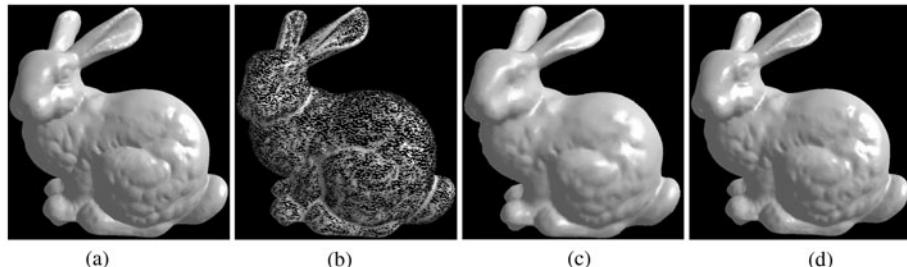
### 5.1 Adaptive sampling of different point sets

In general, surface normals give us first-order information of the underlying surface variation around the sample points, and the variation of normal directions can reflect the intrinsic features. A large variation of normal fields always means geometric features of underlying point set surfaces. In our Gaussian sphere based re-sampling scheme, the essential issue is to cluster sample points due to their normal direction distribution. Hence the proposed algorithm can generate non-uniform adaptively distributed discrete sample points, i.e., in a feature sensitive manner.

For the sake of unification and comparison between different models, the subdivision level of the Gaussian sphere is fixed at 8 and then the size of the adaptive sample point neighborhood determined by the cone-of-normals constraint in each model is 6–16. In the step of naive sampling, the large depth of recursive propagation always means large memory usage, whereas the small depth may generate some bad clusters for planar regions. By trade-off between memory usage and cluster quality, we set the recursive depth at 6 for index propagation. Moreover, in order to generate normalized clusters in the step of optimized sampling, we always split one non-normalized clusters into two sub-clusters. For the sake of splitting the really non-normalized clusters, the threshold of the normalization degree should not be selected as a small value, which will divide some near-normalized clusters and generate inadequate simplification results. Here, the threshold of the normalization degree is chosen as 6.0, which works well in our experiments. Figure 4 shows our adaptive re-sampling results for the Dragon and Buddha models, respectively. The experiments strongly demonstrate the great adaptivity of our sampling scheme to the surface intrinsic features.



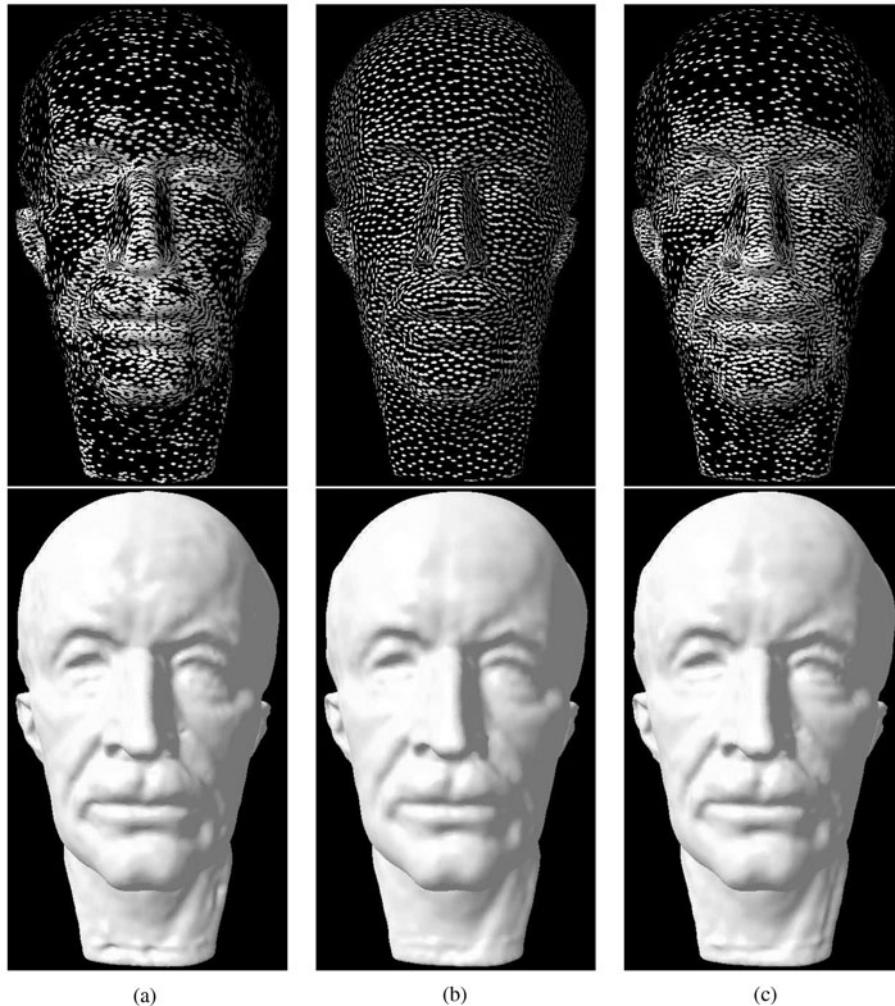
**Figure 4** Feature sensitive re-sampling results for different point set surfaces. (a) Original models; (b) our re-sampling results of the different models; (c) splat rendering results of the different simplified models.



**Figure 5** Feature sensitive re-sampling result and surface reconstruction of the simplified model for the Stanford bunny. (a) Original model; (b) our adaptive re-sampling result; (c) splat rendering result of the simplified model; (d) surface reconstruction of the simplified model by the CSRBF method [16,41].

Moreover, to generate the representative surfel after cluster normalization, we should select the fixed parameter  $h$  introduced in the Gaussian weight function  $\omega_i = \theta(\|\cdot\|)$  (see Subsection 3.5). Unlike the MLS projection in [15] where the result fitting surface will be affected by  $h$  greatly, our weight  $\omega_i$  here only specifies the influence of the neighbor point normals to smoothly blend the normal of each surfel. The final simplified model is unsensitive to the selection of the fixed parameter  $h$ . For example, for Stanford bunny model (the total number of sample points is 280792), if the parameter  $h$  is selected as 0.01 or 0.10, the number of sample points for the simplified model are the same, that is 34255, whilst the normalized RMS geometric  $L^2$  error and the normalized maximum geometric error (see Subsection 5.2) also are the same, that is,  $\Delta_{\text{RMS}}^* = 3.85 \times 10^{-4}$  and  $\Delta_{\text{max}}^* = 0.0020$ . Thus, we set the fixed parameter  $h$  to 0.01 in practice, which works well for our examples.

The simplified models generated by our feature sensitive re-sampling scheme can respect the intrinsic geometric features of the original models. Figure 5 shows the re-sampling result and reconstructed simplified model using relatively sparse down-sampled point set for the Stanford bunny model. For the simplified model, due to the loss of information during the simplification process, the reconstructed simplified surface exhibits the expected smoothing effects. However, the surface reconstruction result directly from the simplified model can still preserve the delicate geometric details of the original model.



**Figure 6** Re-sampling results of different clustering methods and surface reconstruction of the simplified models by the CSRBF method [16,41] for the Max Planck model. (a) Re-sampling result by our feature sensitive re-sampling scheme and surface reconstruction of the simplified model; (b) re-sampling result by uniform incremental clustering [3] and surface reconstruction of the simplified model; (c) re-sampling result by adaptive hierarchical clustering [3] and surface reconstruction of the simplified model. The number of simplified points is 16.5% of the original model in all three simplification schemes.

Figure 6 gives a comparison of our re-sampling scheme to previous clustering methods [3] for point cloud simplification and their surface reconstruction of the simplified models for the Max Planck model. The experimental results indicate that our feature sensitive re-sampling scheme can generate adaptive and non-uniformly distributed discrete sample points in a curvature-aware manner, and the intrinsic geometric features can be preserved well in the reconstructed version of the simplified model. However, the uniform and adaptive clustering schemes exhibit slightly smoothing effects, especially at the regions of sparse-distributed sample points, such as the forehead and the neck of the Max Planck model. Note that the number of simplified points is 16.5% of the original model in all three simplification schemes.

Furthermore, it should be emphasized that we can easily control the re-sampling results by setting different subdivision levels of Gaussian sphere. Figure 7 shows the experimental re-sampling results and geometric  $L^2$  error distributions for different subdivision levels of simplifying the Stanford bunny model (the number of initial samples is 280792). If the subdivision level is chosen as  $n = 8$ , the normalized RMS geometric errors (see Subsection 5.2) are  $\Delta_{\text{RMS}}^* = 3.85 \times 10^{-4}$  and the number of points in the simplified model is 34255 (12.2% of the original points). However, if the subdivision level is chosen as  $n = 16$ ,  $n = 24$ , the normalized RMS geometric errors decrease to  $\Delta_{\text{RMS}}^* = 2.51 \times 10^{-4}$  and  $\Delta_{\text{RMS}}^* = 1.77 \times 10^{-4}$ , respectively. The number of points in the simplified model is 48574 (17.3% of the original points) for  $n = 16$  and 62432 (22.2% of the original points) for  $n = 24$ . The experiments also demonstrate the

great adaptivity of our sampling scheme to the surface features due to different subdivision levels. An increased level of Gaussian sphere subdivision always leads to a larger number of sample points in the final simplified model and a lower geometric  $L^2$  distance error, as expected.

## 5.2 Geometric error analysis of our re-sampling scheme

To evaluate the quality of the simplified geometry generated by our re-sampling algorithm, some standard metric should be adopted to measure the geometric error between the original and simplified versions of the given model. Similar to the 3D mesh Metro analysis tool [42] and [3], we measure the geometric error as both the maximum distance error between the original point-sampled model  $S$  and the simplified version  $S'$ , i.e. the half-sided Hausdorff distance error (or geometric  $L^2$  error),

$$\Delta_{\max}(S, S') = \max_{q \in S} d(q, S'),$$

and root mean square error, i.e.,

$$\Delta_{\text{RMS}}(S, S') = \sqrt{\frac{1}{\|S\|} \sum_{q \in S} d^2(q, S')},$$

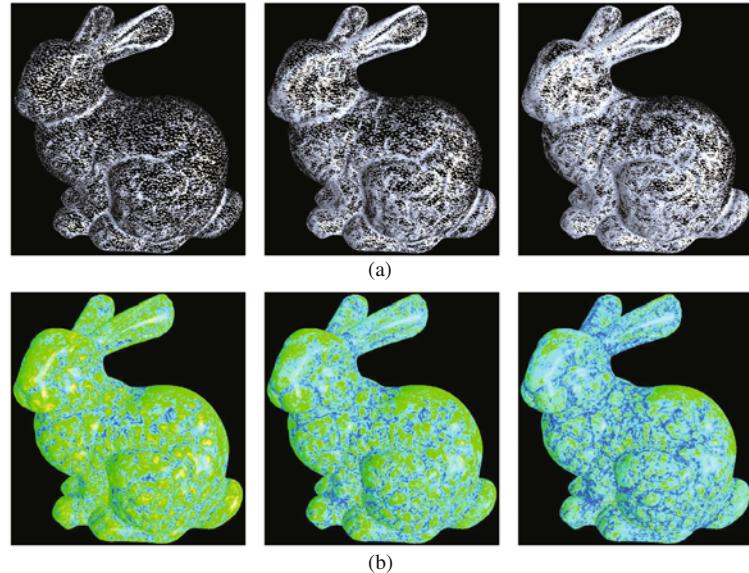
respectively. Then, the corresponding normalized geometric errors  $\Delta_{\max}^*$  and  $\Delta_{\text{RMS}}^*$  can be obtained by scaling the above error measures according to the object's bounding box diagonal. For each sample point  $q \in S$ , the geometric distance error  $d(q, S')$  can be defined as the Euclidean distance between the sample point  $q$  and its projection point  $q^0$  on the simplified surface  $S'$ , that is,

$$d(q, S') = \text{dist}(q, q^0) = \|q - q^0\|.$$

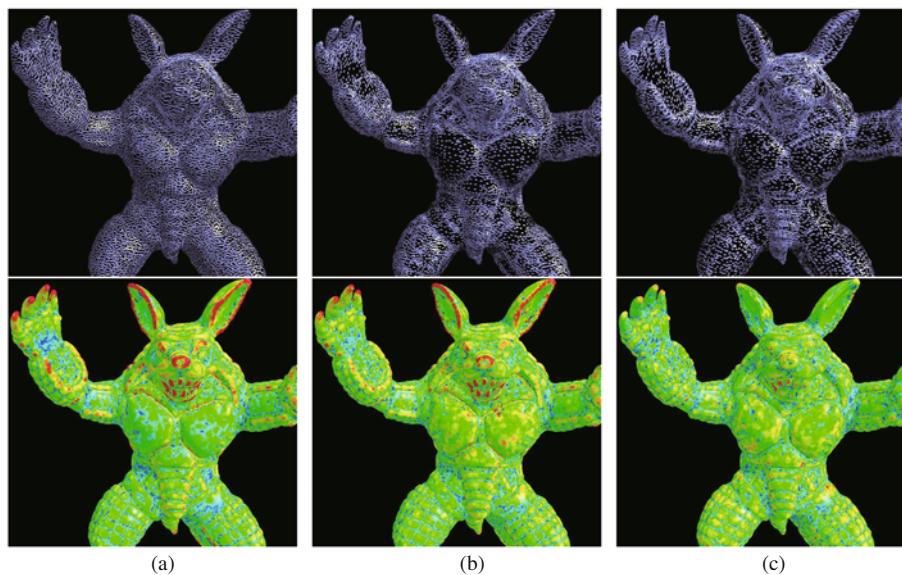
This projection point can be obtained by a simple “almost” orthogonal projection approach, similar as in [33].

Table 1 shows the data and time statistics for the example models and algorithms in this paper. The quantitative error estimates, i.e., the shape  $L^{2,1}$  isophotic error and normalized geometric  $L^2$  error, are listed for each model, which illustrates the efficiency of our adaptive re-sampling scheme. For example, the total number of sample points of Stanford bunny model is 280792, while the number of sample points for the simplified model after optimized sampling is 34255 (12.2% of the original points). The total running-time for our adaptive re-sampling approach is 7.84 s. The root mean square (RMS) isophotic  $L^{2,1}$  error is only  $\Delta_{\text{RMS}} = 0.0369$  and the maximum isophotic  $L^{2,1}$  error is  $\Delta_{\max} = 0.2480$ , whilst the normalized RMS geometric  $L^2$  error is  $\Delta_{\text{RMS}}^* = 3.85 \times 10^{-4}$  and the normalized maximum geometric error is  $\Delta_{\max}^* = 0.0020$ .

Essentially, our feature sensitive re-sampling method is a vertex clustering scheme based on the normal direction distribution mapping onto the Gaussian sphere. Thus, for the sake of uniformity, we limit our comparisons only with other clustering approaches for point-sampled surfaces. In Figure 8, we compare our feature sensitive re-sampling method with Pauly et al. [3] clustering-based simplification approaches for point-sampled surfaces. Table 2 gives the computational time and quantitative error comparison with different clustering-based schemes for the Armadillo model that has been simplified to 22.6% of the original points. Due to its clearness and simplicity for clustering scheme, the uniform and adaptive clustering schemes are slightly faster than our feature sensitive re-sampling approach. However, uniform incremental clustering has the highest normalized RMS geometric  $L^2$  error ( $\Delta_{\text{RMS}}^* = 7.04 \times 10^{-4}$ ,  $\Delta_{\max}^* = 0.0218$ ) and since all clusters consist of roughly the same number of sample points, most of the geometric error is concentrated in the regions of high curvature, such as the brim of Armadillo ear, the finger tip, the nose and teeth. Adaptive hierarchical clustering performs slightly better ( $\Delta_{\text{RMS}}^* = 6.74 \times 10^{-4}$ ,  $\Delta_{\max}^* = 0.0204$ ), in particular in the geometrically complex regions. However, our feature sensitive re-sampling scheme provides lower geometric error ( $\Delta_{\text{RMS}}^* = 6.05 \times 10^{-4}$ ,  $\Delta_{\max}^* = 0.0029$ ), which distributes the error more evenly across the surface, not concentrating in the sharp feature regions of the underlying model.



**Figure 7** Adaptive re-sampling results for the Stanford bunny model according to different levels of Gaussian sphere subdivision. (a) The different adaptive sampling results for the Stanford bunny using  $n = 8, 16$  and  $24$  levels of Gaussian sphere subdivision, respectively; (b) the color map of the geometric  $L^2$  error distribution of the simplified models corresponding to different subdivision levels, respectively. (Note: Yellow means high geometric error zone, blue means low geometric error zone, and green means the zone of middle geometric error.)



**Figure 8** Error analysis and error comparison with other clustering-based simplification schemes for the Armadillo model. (a) Re-sampling result by uniform incremental clustering scheme [3] and the color map of normalized geometric error distribution; (b) re-sampling result by adaptive hierarchical clustering scheme [3] and the color map of normalized geometric error distribution; (c) re-sampling result by our feature sensitive re-sampling scheme and the color map of normalized geometric error distribution. (Note: Red means high geometric error zone, blue means low geometric error zone, yellow and green mean the zones of middle geometric error.)

## 6 Conclusions and future work

Using Gaussian sphere sampling, a novel feature sensitive re-sampling approach for point set surfaces is presented, which includes a naive sampling step and a novel cluster optimization step. Our simplification

**Table 1** Statistics and timings of our re-sampling approach for different models<sup>a)</sup>

Model	#Orig.	Timings (s)	#Opti.	Shape $L^{2,1}$ error		Normal. geom. error	
				RMS	Max.	RMS	Max.
Buddha	543652	19.04	128652	0.0438	0.9636	$3.10 \times 10^{-4}$	0.0047
Dragon	437645	13.97	80295	0.0410	0.9753	$3.20 \times 10^{-4}$	0.0042
Bunny	280792	7.84	34255	0.0369	0.2480	$3.85 \times 10^{-4}$	0.0020
Armadillo	172974	6.02	39161	0.0473	0.2595	$6.05 \times 10^{-4}$	0.0029
Balljoint	137062	4.38	23206	0.0433	0.2865	$5.49 \times 10^{-4}$	0.0033
Santa	75781	2.62	17181	0.0482	0.2444	$6.16 \times 10^{-4}$	0.0044
Planck	52809	1.64	8697	0.0615	0.2903	$10.80 \times 10^{-4}$	0.0094

a) The subdivision level of the Gaussian sphere is chosen to be 8, and the size of the adaptive neighborhood in each model is 6–16. The running-time statistics includes the timing of index propagation and the timing of normalized rectification, which is measured on a PC with a Pentium IV 3.0 GHz CPU, 1024 MB memory. From left to right: Number of sample points of original models (#Orig.), Timings for our re-sampling approach (Timings), Number of sample points of optimized sampling (#Opti.), RMS  $L^{2,1}$  error, Maximum  $L^{2,1}$  error, Normalized RMS geometric error and normalized maximum geometric error.

**Table 2** Computational time and quantitative error comparison with Pauly et al. [3] clustering-based simplification approaches for the Armadillo model, which are measured for a simplification to 22.6% of the input model size

Clustering-based schemes	Timings (s)	Shape $L^{2,1}$ error		Normal. geom. error	
		RMS	Max.	RMS	Max.
Uniform incremental clustering [3]	4.24	0.0878	0.9458	$7.04 \times 10^{-4}$	0.0218
Adaptive hierarchical clustering [3]	4.10	0.0874	0.9458	$6.74 \times 10^{-4}$	0.0204
Our feature sensitive re-sampling	6.02	0.0473	0.2595	$6.05 \times 10^{-4}$	0.0029

scheme can generate non-uniformly distributed discrete sample points for adaptively reflecting the intrinsic geometric properties of the underlying 3D model. The geometric features of the underlying point set surfaces can be preserved due to our feature sensitive re-sampling scheme.

Due to the Gaussian sphere base normal clustering scheme, the proposed re-sampling framework can directly control the surface  $L^{2,1}$  simplification error by setting different subdivision levels of Gaussian sphere. However, one limitation of our re-sampling scheme is its difficulty of directly controlling the number of sample points of the final simplified model, which can be solved by other clustering methods such as [21]. Moreover, due to its inherent clustering and optimization scheme, another limitation of our feature sensitive re-sampling approach is its relative high time complexity if compared with intuitive and simple clustering scheme, such as uniform incremental clustering. Maybe we can consider these limitations as future work.

Moreover, based on the proposed adaptive sampling approach, future research can also focus on geometry processing tasks for point-sampled models, such as model compression and transmission, streaming processing, and real-time point-based rendering.

### Acknowledgements

This work was jointly supported by National Natural Science Foundation of China (Grant Nos. 61170138, 60933007), Science and Technology Planning Project of Zhejiang Province (Grant No. 2009C34005). This work was also partially supported by Swiss National Science Foundation (Grant No. 200021-111746/1). The 3D point-sampled models are courtesy of the Stanford Computer Graphics Laboratory and the Computer Graphics Laboratory of ETH Zürich.

### References

- Zwicker M, Pauly M, Knoll O, et al. Pointshop 3D: An interactive system for point-based surface editing. ACM Trans Graph, 2002, 21: 322–329

- 2 Gross M, Pfister H. Point Based Graphics. Burlington: Morgan Kaufmann Publisher, 2007
- 3 Pauly M, Gross M, Kobbelt L. Efficient simplification of point-sampled surfaces. In: Proceedings of IEEE Visualization, Boston, 2002. 163–170
- 4 Moenning C, Dodgson N A. Intrinsic point cloud simplification. In: Proceedings of the 14th International Conference on Computer Graphics and Vision, Moscow, 2004. 1147–1154
- 5 Wu J H, Kobbelt L. Optimized sub-sampling of point sets for surface splatting. *Comput Graph Forum*, 2004, 23: 643–652
- 6 Vorsatz J, Roessl C, Kobbelt L, et al. Feature sensitive remeshing. *Comput Graph Forum*, 2001, 20: 393–401
- 7 Alliez P, Cohen-Steiner D, Devillers O, et al. Anisotropic polygonal remeshing. In: Proceedings of ACM SIGGRAPH, San Diego, 2003. 485–493
- 8 Lai Y K, Zhou Q Y, Hu S M, et al. Robust feature classification and editing. *IEEE Trans Vis Comput Graph*, 2007, 13: 34–45
- 9 Lai Y K, Hu S M, Pottmann H. Surface fitting based on a feature sensitive parameterization. *Comput Aided Des*, 2006, 38: 800–807
- 10 Ohtake Y, Belyaev A, Seidel H P. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans Graph*, 2004, 23: 609–612
- 11 doCarmo M P. Differential Geometry of Curves and Surfaces. New Jersey: Prentice-Hall. Inc., 1976
- 12 Horn B K P. Extended Gaussian images. *Proc IEEE*, 1984, 72: 1671–1686
- 13 Miao Y W, Diaz-Gutierrez P, Pajarola R, et al. Shape isophotic error metric controllable re-sampling for point-sampled surfaces. In: IEEE International Conference of Shape Modeling and Applications, Beijing, 2009. 28–35
- 14 Alexa M, Behr J, Cohen-or D, et al. Point set surfaces. In: Proceedings of IEEE Visualization, San Diego, 2001. 21–28
- 15 Alexa M, Behr J, Cohen-or D, et al. Computing and rendering point set surfaces. *IEEE Trans Vis Comput Graph*, 2003, 9: 3–15
- 16 Carr J C, Beatson R K, Cherrie J B, et al. Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of ACM SIGGRAPH, Los Angeles, 2001. 67–76
- 17 Ohtake Y, Belyaev A, Alexa M, et al. Multi-level partition of unity implicits. *ACM Trans Graph*, 2003, 22: 463–470
- 18 Amenta N, Kil Y. Defining point-set surfaces. *ACM Trans Graph*, 2004, 23: 264–270
- 19 Guennebaud G, German M, Gross M. Dynamic sampling and rendering of algebraic point set surfaces. *Comput Graph Forum*, 2008, 27: 653–662
- 20 Garland M, Heckbert P. Surface simplification using quadric error metrics. In: Proceedings of ACM SIGGRAPH, Los Angeles, 1997. 209–216
- 21 Miao Y W, Pajarola R, Feng J Q. Curvature-aware adaptive resampling for point-sampled geometry. *Comput Aided Des*, 2009, 41: 395–403
- 22 Kitago M, Gopi M. Efficient and prioritized point subsampling for CSRBF compression. In: Proceedings of Eurographics Symposium on Point-Based Graphics, Boston, 2006. 121–128
- 23 Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation. *ACM Trans Graph*, 2004, 23: 905–914
- 24 Wu J H, Zhang Z, Kobbelt L. Progressive splatting. In: Proceedings of Eurographics Symposium on Point-Based Graphics, Stony Brook, 2005. 25–32
- 25 Diaz-Gutierrez P, Bösch J, Pajarola R, et al. Streaming surface sampling using Gaussian e-nets. *Vis Comput*, 2009, 25: 411–421
- 26 Pottmann H, Steiner T, Hofer M, et al. The isophotic metric and its applications to feature sensitive morphology on surfaces. In: Pajdla T, Matas J, eds. Computer Vision—ECCV 2004, Part IV, 2004. LNCS 3024. 560–572
- 27 Weyrich T, Pauly M, Keiser R, et al. Post-processing of scanned 3d surface data. In: Proceedings of Eurographics Symposium on Point-Based Graphics, Zurich, 2004. 85–94
- 28 Pajarola R. Efficient level-of-details for point based rendering. In: Proceedings of 6th IASTED International Conference on Computer Graphics and Imaging, Honolulu, 2003. 141–146
- 29 Jolliffe I. Principle Component Analysis. New York: Springer-Verlag, 1986
- 30 Kobbelt L, Botsch M. A survey of point-based techniques in computer graphics. *Comput Graph*, 2004, 28: 801–814
- 31 Sainz M, Pajarola R. Point-based rendering techniques. *Comput Graph*, 2004, 28: 869–879
- 32 Hoppe H. New quadric metric for simplifying meshes with appearance attributes. In: Proceedings of IEEE Visualization, San Francisco, 1999. 59–66
- 33 Alexa M, Adamson A. On normals and projection operators for surfaces defined by point sets. In: Eurographics Symposium on Point-Based Graphics, Zurich, 2004. 149–155
- 34 Kim H J, Bickel B, Gross M, et al. Subsurface scattering using splat-based diffusion in point-based rendering. *Sci China Inf Sci*, 2010, 53: 911–919
- 35 Botsch M, Spernati M, Kobbelt L. Phong splatting. In: Eurographics Symposium on Point-Based Graphics, Zurich, 2004. 25–32

- 36 Pajarola R, Sainz M, Guidotti P. Confetti: Object-space point blending and splatting. *IEEE Trans Vis Comput Graph*, 2004, 10: 598–608
- 37 Zwicker M, Räsänen J, Botsch M, et al. Perspective accurate splatting. In: *Proceedings of Graphics Interface*, London, 2004. 247–254
- 38 Guennebaud G, Barthe L, Paulin M. Splat/mesh blending, perspective rasterization and transparency for point-based rendering. In: *Proceedings of Eurographics Symposium on Point-Based Graphics*, Boston, 2006. 49–57
- 39 Botsch M, Hornung A, Zwicker M, et al. High-quality surface splatting on today's GPUs. In: *Proceedings of Eurographics Symposium on Point-Based Graphics*, Stony Brook, 2005. 17–24
- 40 Zhang Y C, Pajarola R. Deferred blending: Image composition for single-pass point rendering. *Comput Graph*, 2007, 31: 175–189
- 41 Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. *Comput Graph Forum*, 1998, 17: 167–174
- 42 Ohtake Y, Belyaev A, Seidel H P. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In: *IEEE International Conference of Shape Modeling and Applications*, Seoul, 2003. 153–161