Edge-Aware Point Set Resampling

HUI HUANG
Shenzhen Key Lab of Visual Computing and Visual Analytics / SIAT
and
SHIHAO WU
South China University of Technology
and
MINGLUN GONG
Memorial University of Newfoundland
and
DANIEL COHEN-OR
Tel-Aviv University
and
URI ASCHER
University of British Columbia
and
HAO ZHANG
Simon Fraser University

Points acquired by laser scanners are not intrinsically equipped with normals, which are essential to surface reconstruction and point set rendering using surfels. Normal estimation is notoriously sensitive to noise. Near sharp features, the computation of noise-free normals becomes even more challenging due to the inherent under-sampling problem at edge singularities. As a result, common edge-aware consolidation techniques such as bilateral smoothing may still produce erroneous normals near the edges. We propose a resampling approach to process a noisy and possibly outlier-ridden point set in an edge-aware manner. Our key idea is to first resample away from the edges so that reliable normals can be computed at the samples, and then based on reliable data, we progressively resample the point set while approaching the edge singularities. We demonstrate that our edge-aware resampling (EAR) algorithm is capable of producing consolidated point sets with noise-free normals and clean preservation of sharp features. We also show that EAR leads to improved performance of edge-aware reconstruction methods and point set rendering techniques.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

Additional Key Words and Phrases: Point set resampling, normal estimation, upsampling, surface reconstruction, surfel point set rendering

1. INTRODUCTION

The last two decades have seen a considerable amount of work on surface reconstruction from scanned point clouds. The use of points as a modeling and rendering primitive has also been studied extensively [Gross and Pfister 2007]. Both tasks heavily rely on having a quality normal associated with each point sample. In particular, popular surface reconstruction techniques such as Poisson [Kazhdan et al. 2006] and RBF [Carr et al. 2001] are guided by normal information and the well-known surfel-based point set rendering [Pfister et al. 2000] operates on oriented samples. However,

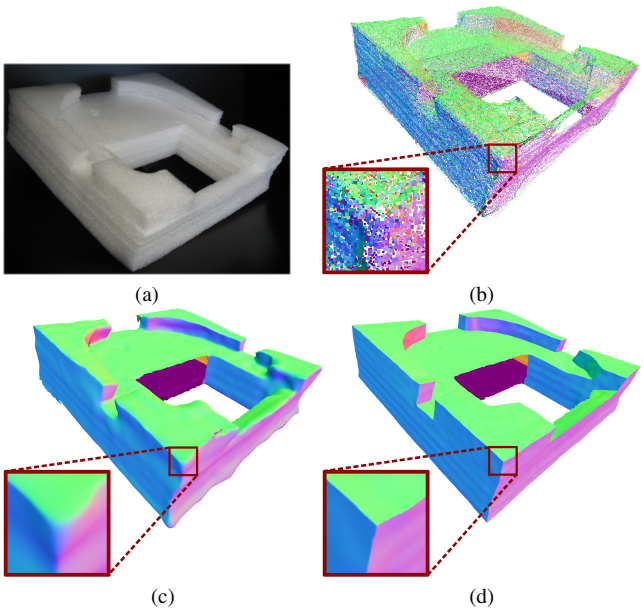
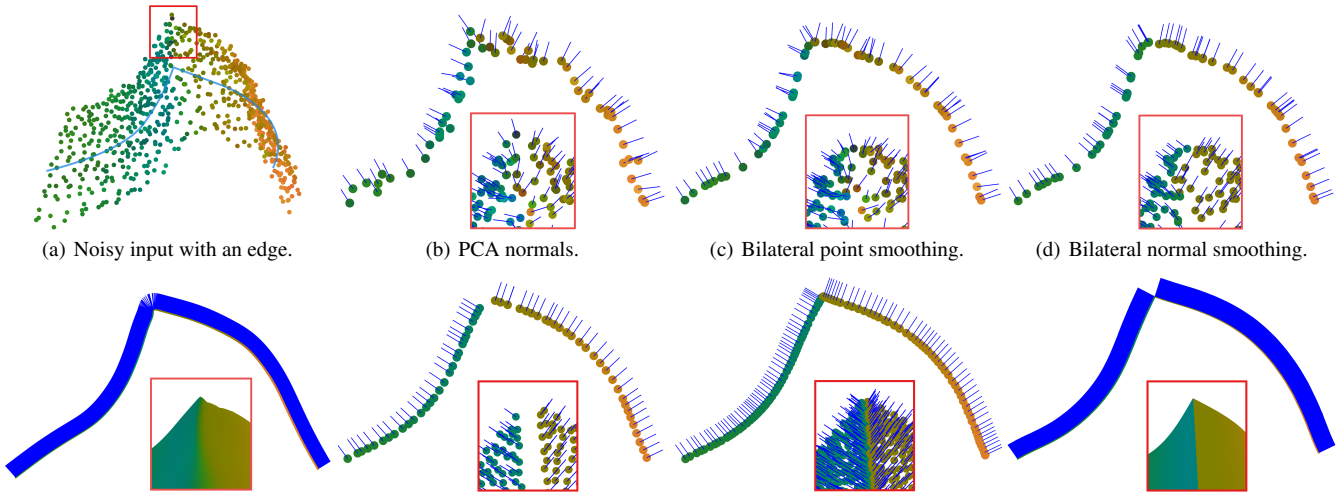


Fig. 1. Points (222K) acquired by a laser scan (b) are corrupted with noise and not intrinsically equipped with normals. Resampling the data without accounting for surface singularities may smear the sharp features after surface reconstruction (c). Our edge-aware resampling (EAR) leads to a piecewise smooth reconstruction (d) while preserving the sharp edges. Point colors are the result of normal maps and the original object is shown in (a).



(a) Noisy input with an edge. (b) PCA normals. (c) Bilateral point smoothing. (d) Bilateral normal smoothing.

(e) RIMLS + Marching Cubes on (d). (f) Resampling (b) away from edge. (g) Resampling (f) close to edge. (h) RIMLS + Marching Cubes on (g).

Fig. 2. Comparison between bilateral smoothing and the proposed EAR approach. Each point is colored according to its normal direction and such normal maps are used throughout the paper for rendering the results. The input (a) represents a fin shape containing a sharp edge; the point cloud (1K points) is corrupted with noise. In each subsequent result (b-h), we show a 2D cross-section view taken at the blue curve indicated in (a), as well as a zoomed-in 3D view into the red window shown in (a). The results demonstrate that our resampling approach can effectively remove noise, provide reliable normals, preserve sharp features, and facilitate edge-aware reconstruction methods such as RIMLS [Öztireli et al. 2009].

points acquired by laser scanners are not intrinsically equipped with normals; they must be estimated from acquired image or geometry data. Acquired data are often tempered with noise and even outliers which hinder the computation of normals. We stress that normal estimation is sensitive to noise since normals are measured as first-order derivatives and numerical differentiation amplifies noise.

Computing noise-free normals is much more challenging in the presence of sharp features, e.g., see Figs. 1 and 2. The desire to preserve the sharp features disallows the use of a naïve point smoothing operator prior to normal estimation, since such smoothing blurs the edges. Alternatively, one can compute normals over the noisy point set, typically via principal component analysis (PCA), and then apply an edge-aware robust smoothing operator, such as bilateral filtering [Jones et al. 2004] or ℓ_1-minimization [Avron et al. 2010], to the normals. While such solutions can generally "separate" the two sides of an edge, in the vicinity of the edge some erroneous normals may still persist since the accuracy of these methods is limited by the noise level and sampling rate. In practice, data near an edge are often unavoidably under-sampled and contain more noise than smooth regions, making it difficult to recover sharpness directly at the edges.

Given a noisy point cloud with, possibly noisy, normals computed by PCA, as shown in Figs. 2(a) and 2(b), respectively, Fig. 2(c) and 2(d) show results of applying bilateral filtering over the points and PCA normals. As can be seen, near the edge some points have incorrectly assigned normal directions to agree with point normals at the wrong side of the edge. These seemingly small errors may be amplified on the reconstructed surface, leading to visible artifacts, as demonstrated in Fig. 2(e) by an edge-aware surface reconstruction scheme using robust implicit moving least squares (RIMLS) [Öztireli et al. 2009].

To circumvent this inherent problem, our strategy is to *resample* the noisy point set judiciously and in an *edge-aware* manner. In particular, since normal estimation close to edges is not reliable, our key idea is to first resample *away* from the edges. The result is a set of oriented points away from the edges that are endowed with reliable normals. Then based on these oriented points, we progressively resample the point set while approaching the edges. Thus, normal estimation proceeds from more reliable regions to less reliable regions (close to edges). Using the resampling strategy, we avoid having to compute noise-sensitive derivative measures in difficult regions. Resampling and normal estimation near edges are guided by reliable data, in particular, reliable normals.

Our resampling algorithm is built on a robust edge-aware projection operator which produces samples away from edges and a novel bilateral projection operator which upsamples progressively so as to fill the edge regions. Resampling away from edges is enabled by incorporating normal information into the projection, allowing the projector to be edge-aware. Robustness to noise and outliers is enabled by the use of the ℓ_1-median for data fitting. As we approach the edges, the bilateral projector considers both positional and current normal information when computing the base, direction, and distance of the projection. While the two latter attributes are the result of optimizing a bilateral objective function, the base location is chosen to achieve even point distribution and fast convergence. The result of our resampling algorithm is a consolidated point set with noise-free normals and uniform point distribution throughout and clean preservation of sharp features; see Fig. 2(g).

Previous works which also resort to resampling for point cloud consolidation assume that the underlying surface is smooth [Alexa et al. 2003; Lipman et al. 2007; Huang et al. 2009; Miao et al. 2009; Öztireli et al. 2010]. Our edge-aware resampling (EAR) scheme respects singularities, allowing effective handling of piecewise smooth surfaces. We show that EAR and its associated normal estimation facilitate the reconstruction of such surfaces by edge-oblivious methods such as Poisson [Kazhdan et al. 2006], algebraic point set surfaces (APSS) [Guennebaud and Gross 2007], and Delaunay-based Cocone [Dey and Giesen 2001]. At the same time,

it enhances the performance of edge-aware reconstruction methods such as RIMLS; see Fig. 2(h). We also show that EAR can be applied to upsample a point set, leading to superior rendering results [Pfister et al. 2000; Vergne et al. 2010] near sharp features.

2. RELATED WORK

Most point set resampling schemes aimed at consolidating a raw point scan assume that the underlying surface is smooth. Early work by Alexa et al. [2001] upsamples a point set through Voronoi point insertion in local tangent spaces followed by moving least squares (MLS) projection. Lipman et al. [2007] introduce the parameterization-free, locally optimal projector (LOP), which is driven by the use of the ℓ_1-median. LOP is shown to be effective in enhancing point sets while being robust to outliers and noise. Huang et al. [2009] propose a weighted version of LOP which better deals with non-homogeneous point density in the input. Miao et al. [2009] develop a simplification scheme for non-uniformly distributed point samples, which adaptively reflects the intrinsic geometric features of the underlying shape. Recent work by Öztireli et al. [2010] presents a high quality isotropic point sampling technique for a given surface representation based on spectral analysis, kernel methods and matrix perturbation theory. None of these schemes were designed to handle sharp features, and some required reliable normals as part of the input. Our resampling approach also relies on ℓ_1-median, but unlike LOP, it estimates and utilizes normals, doing so in an edge-aware manner.

Existing edge-aware point set resampling schemes either assume that the sharp features are given or resort to explicit construction of surface patches. For an accurate display of intersection curves between point set surface sheets, the works in [Pauly et al. 2003; Guennebaud et al. 2004; Guennebaud et al. 2008] present accurate rendering of sharp creases and corners. These methods assume that the intersection curves that define the sharp features are given. Fleishman et al. [2005] develop a piecewise surface-refitting algorithm which allows projecting points and sampling the intersection of surface patches from two sides of a sharp feature. Their method requires a dense point set to start with while in practice, regions near sharp features are often under-sampled [Salman et al. 2010]. Generally speaking, reliable fitting of surfaces is computationally expensive and sensitive to input data noise and outliers. Our method does not rely on explicit feature identification or patch fitting; it deals with heavy noise and under-sampling near edges by first resampling away from edges and then propagating reliable information obtained there to the edges.

Sharp features are dealt with extensively in the mesh processing literature. A typical task is to smooth a mesh surface while preserving sharp features, e.g., [Hildebrandt and Polthier 2004; Sun et al. 2007; Huang and Ascher 2008]. Bilateral filtering [Fleishman et al. 2003; Jones et al. 2003] for mesh smoothing is related to our bilateral projector since both perform optimization under a bilateral objective function. However, mesh smoothing assumes an explicit connectivity among the points.

More relevant are meshless methods which smooth point set surfaces that contain sharp features. The methods of [Adamson and Alexa 2006; Guennebaud and Gross 2007] assume explicit representation of sharp features. Salman et al. [2010] first extract sharp features from the point set and then directly generate piecewise smooth surface triangle meshes. Fleishman et al. [2005] and Lipman et al. [2007] analyze the point set around sharp features and fit

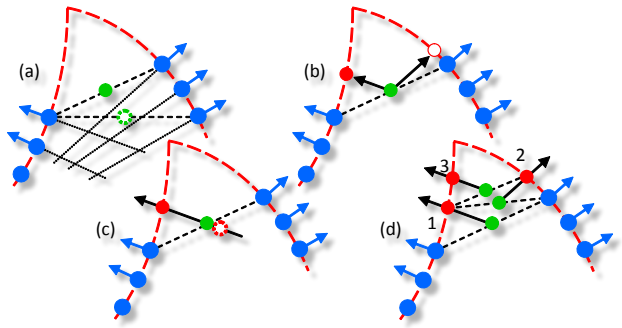


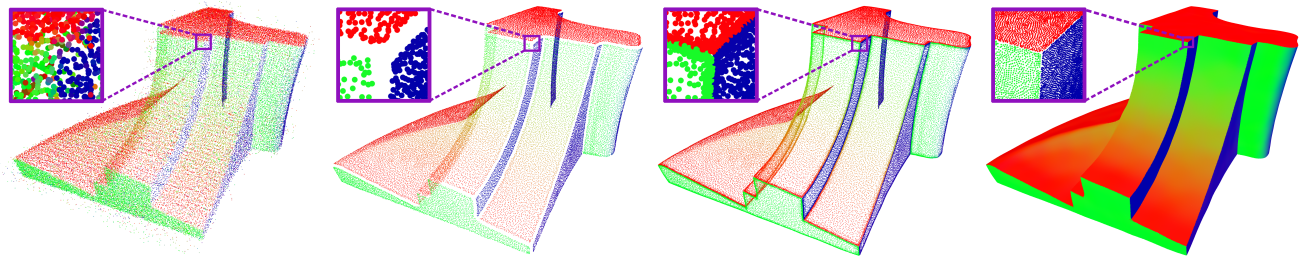
Fig. 4. Upsampling via bilateral projection near a sharp feature: (a) using both positions and normals yields a better point insertion location (solid green dot) than using positions alone (dashed green dot); (b) both projection directions shown fit nearby normals, but the one leading to solid red dot better preserves point uniformity; (c) using bilateral weights properly projects the point onto the latent surface (solid red dot), whereas using unilateral weight pulls the point away (dashed red dot); (d) a series of progressive projection operations upsample the sharp feature.

local surface patches over which points are projected. Algorithms proposed in [Merigot et al. 2009; Weber et al. 2010] mainly focus on detecting sharp features of a piecewise smooth surface from its point cloud sampling. Other recent robust methods [Öztireli et al. 2009; Avron et al. 2010] reconstruct surfaces from point clouds while respecting sharp features. All of these methods can produce less than satisfactory results, e.g., erroneous normal estimates, near sharp features as a result of severe noise and under-sampling. Resampling the point set prior to feature extraction, patch fitting, or surface reconstruction enhances the performance of these methods on a raw point scan.

Given the importance of normals in surface reconstruction and point rendering, it is not surprising that there has been a tremendous amount of work on normal estimation from raw point data. Most methods resort to PCA or its variants [Hoppe et al. 1992; Alexa et al. 2001; Pauly et al. 2002; Mitra et al. 2004; Lange and Polthier 2005; Huang et al. 2009]. Near or on sharp features, PCA normals tend to smear information across discontinuities. Bilateral smoothing of PCA normals [Öztireli et al. 2009] provides some remedy but can still produce erroneous results (see Figs. 2(d) and 2(e)). Moreover, sharp features are often under-sampled in point scans which further hinders the performance of PCA. Another set of techniques use Voronoi poles [Dey and Sun 2006] or Voronoi-PCA [Alliez et al. 2007] to estimate normals. With under-sampling, such interpolation-based methods cannot infer accurate normals near sharp features. In our work, we also rely on bilateral processing of PCA normals, but only to roughly detect edge locations so as to enable a resampling away from edges. Normals at points close to edges are then derived from reliable normals estimated away from edges instead of using PCA.

3. OVERVIEW

Our EAR algorithm takes as input an unorganized and unoriented 3D point scan corrupted with noise, outliers, and under-sampling. It produces a clean, uniform, and feature-preserving set of oriented points that well approximates the underlying surface. An advantage of the resampling approach is that the density of the point set can be adjusted; e.g., for point set rendering, a dense point set is obtained.



(a) Noisy input. (b) Resampling away from edges. (c) Edge-aware upsampling. (d) Upsampling for rendering.

Fig. 3. Overview of EAR scheme. Given a noisy point scan (a) with 163K points, we first resample away from edges, leaving gaps near sharp features (b). Based on reliable normals associated with the point set thus obtained, we upsample while approaching the edges and filling the gaps (c). Point density can be further increased through upsampling to obtain a quality point set rendering (d).

To obtain noise-free normals especially near the possibly undersampled edge singularities, our resampling algorithm is separated into two phases, as shown in Fig. 3. The first phase resamples a set of points away from edges, for which reliable normals can be computed. In the second phase, we upsample to increase point density, progressively filling regions near the edges. Point insertion approaching the edges relies on the reliable normals generated in the first phase, leading to clean reconstruction of the sharp features.

Resampling away from edges. Starting with a noisy scan, such as the one shown in Fig. 3(a), we first perform PCA to estimate normal directions and their consistent orientations as in [Huang et al. 2009]. Next we apply bilateral smoothing of these normals where the weighting scheme accounts for both positional and normal information. Iterative bilateral smoothing alone can generally distinguish normals near edges and roughly reveal edge locations. However, the normals may still smear across edges as shown in Fig. 2(d). Our strategy is then to not compute normals too close to edges but first resample away, where normals can be estimated reliably. The resampling is accomplished by a locally anisotropic projection which accounts for the current normals. The result of this phase is a set of oriented points with reliable normals, but leaving gaps close to sharp features, as shown in Fig. 3(b).

Edge-preserving upsampling. Based on the reliable normals, oriented points are inserted and projected onto the latent surface, which is the unknown underlying surface defined by the input point set. For each inserted point, we first select its base location at a midpoint between two existing points, see Fig. 4(a). Then, the critical question is along which direction to project the point, the green dot in Fig. 4(b), to better preserve the sharp feature. To simplify the problem, we constrain the normal of the inserted point to be the same as the projection direction. Hence, a careful determination of the direction serves two important objectives: (i) it fits the normal distribution of nearby points, and (ii) it helps maintain local point uniformity. Once the projection direction is determined, we compute an optimal projection distance so that the inserted point can be moved onto the latent surface; see Fig. 4(c).

The steps for determining the base, direction, and distance of the projection offer an integrated solution for inserting an oriented point at a sparse spot anywhere on the surface. To properly handle sharp features, both positional and normal information are accounted for in all steps, making the projection operator bilateral and edge-aware. As shown in Figs 4(d) and 3(c), repeating the above upsampling process incrementally fills the gaps along edge singularities and reconstructs the sharp features cleanly. The upsampling process can continue to increase point densities to facilitate rendering of the point set surface, as shown in Fig. 3(d).

4. RESAMPLING AWAY FROM EDGES

The input to our algorithm is an unorganized set of points $Q = \{q_j\}_{j\in J} \subset \mathbb{R}^3$, typically unevenly distributed and containing noise and outliers. The output of the resampling step described in this section is an oriented point set $S = \{s_i\}_{i\in I} = \{(p_i, \mathbf{n}_i)\}_{i\in I} \subset \mathbb{R}^6$, consisting of cleaned point locations p_i that better represent the underlying smooth surface away from edges, as well as their associated reliable and edge-aware normals $\mathbf{n}_i$.

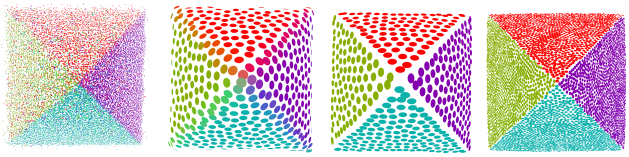
The process of estimating normals near sharp features is particularly delicate and challenging. Traditional normal estimation methods, e.g., [Hoppe et al. 1992], usually work accurately when the underlying surface is smooth, but tend to smear information across singularities such as corners or intersecting planes. Taking it as an initial input, our approach amounts to an iteration between (i) separating and smoothing normals over the obtained point set, and (ii) resampling the points away from edges while holding the normals fixed. Step (i) reveals the location of edges via normal separation around edge regions, which enables an effective anisotropic projection operator in step (ii). The latter resampling away from edges, in turn, can emphasize the edge locations (by sparsity) so that the process in step (i) in the next iteration operates more accurately.

Separating and smoothing normals. The popular PCA-based method for computing surface normal approximations from point cloud data [Hoppe et al. 1992] uses local point neighborhoods. Certain errors may thus be retained when the input is highly noisy, as is typically the case near sharp features; see Fig. 2(b). In order to preserve sharp features, we therefore estimate the normal based on an *anisotropic* neighborhood, as in the framework of bilateral normal smoothing [Jones et al. 2004; Öztireli et al. 2009]. In particular, for a given point $s_i = (p_i, \mathbf{n}_i)$, we measure the difference between its assigned normal $\mathbf{n}_i$ and other normals in its neighborhood as

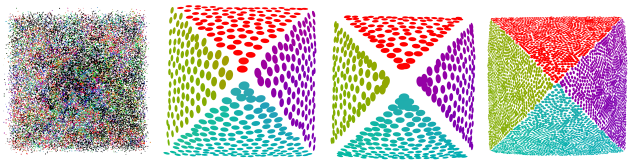
$$f(p_i, \mathbf{n}_i) = \sum_{s_{i'}\in\mathcal{N}_{s_i}} \|\mathbf{n}_i - \mathbf{n}_{i'}\|^2 \theta(\|p_i - p_{i'}\|)\psi(\mathbf{n}_i, \mathbf{n}_{i'}), \quad (1)$$

where $\|\cdot\|$ is the ℓ_2-norm, $\mathcal{N}_{s_i} = \{s_{i'} | s_{i'} \in S \wedge \|p_i - p_{i'}\| < \sigma_p\}$ under a given neighborhood size σ_p. The spatial and normal weight functions are defined by

$$\theta(r) = e^{-r^2/\sigma_p^2}, \quad \psi(\mathbf{n}_i, \mathbf{n}_{i'}) = e^{-\left(\frac{1-\mathbf{n}_i^\top \mathbf{n}_{i'}}{1-\cos(\sigma_n)}\right)^2},$$



(a) 1% noise. (b) LOP. (c) $\sigma_p = h$. (d) Upsampling.



(e) 2% noise. (f) $\sigma_p = h$. (g) $\sigma_p = 3h$. (h) Upsampling.

Fig. 5. EAR under different noise levels and the effect of neighborhood size σ_p. Given a noisy input (a) with 18K points, LOP effectively removes noise (b) but the points are smeared across the edges. In contrast, our resampling (c) yields reliable data including both point locations and normals away from edges, which facilitates the subsequent upsampling (d) towards the edges. When a higher level of noise is present in the input (e), the resampling should employ a larger local neighborhood size σ_p to ensure a clearer patch separation about the edges and better smoothness away from edges, e.g., contrasting (g) with (f), leading to higher quality upsampling (h).

where the angle parameter σ_n scales the similarity of neighboring normals and we set $\sigma_n = 15^\circ$ by default. Our goal is to minimize $\sum_{i\in I} f(p_i, \mathbf{n}_i)$, i.e., the normal differences between all points on the surface and their neighbors. This can be achieved through iteratively updating $\mathbf{n}_i$ for all i with

$$\mathbf{n}_i \leftarrow \frac{\sum_{s_{i'}\in\mathcal{N}_{s_i}} \theta(\|p_i - p_{i'}\|)\psi(\mathbf{n}_i, \mathbf{n}_{i'})\mathbf{n}_{i'}}{\sum_{s_{i'}\in\mathcal{N}_{s_i}} \theta(\|p_i - p_{i'}\|)\psi(\mathbf{n}_i, \mathbf{n}_{i'})}. \tag{2}$$

The above formula works well to distinguish normals across discontinuities, classifying their directions into two disjoint clusters near each sharp edge due to the high variance of PCA normals in the vicinity; see Fig. 2(c). However, we can also see that direction clustering does not really work for the point locations themselves, and several points are incorrectly assigned a wrong normal direction. Noise and outliers in the input make it impossible to directly obtain a clear classification with respect to an edge in both spatial and directional senses. Indeed, given such an oriented point set, RIMLS may fail to reconstruct a surface that preserves sharp features; see Figs. 2(d) and 2(e). Thus, with the normals $\{\mathbf{n}_i\}_{i\in I}$ obtained by (2), we now turn to adjusting the locations $\{p_i\}_{i\in I}$ to complete a split-step iteration for the set S.

Resampling. Several efficient resampling operators have been designed to consolidate a noisy point set [Alexa et al. 2003; Lipman et al. 2007] while being oblivious to normals. These work well when the underlying surface is smooth. For a resampling operator to be edge-aware, however, it must account for normals, even if they may not be entirely accurate. We seek such an operator that would be easily implemented, robust to heavy noise, and utilize the estimated normals around edges. To this end, we alter the LOP operator [Lipman et al. 2007; Huang et al. 2009] and make it normal- or edge-aware, allowing for resampling away from edges.

LOP takes as input a noisy point cloud, possibly with outliers, and outputs a new point set which more faithfully adheres to the underlying shape. The strength of LOP is that it operates well on raw data, without relying on a local parameterization of the points or on their local orientation. In intuitive terms LOP distributes a set of points, within an optimization framework, to approximate their ℓ_1-median so as to achieve robustness to outliers and noise. At the same time, a repulsion force is integrated into the optimization formulation to obtain an even point distribution.

Although LOP works robustly for geometry reconstruction from raw data, it is still an isotropic operator since the spatial weight function $\theta(r)$ used there does not consider sharp geometric features, see Fig. 5(b). However, in our context, following the bilateral smoothing step (2), the processed normal directions indicate where the edges approximately are. Thus we define

$$\phi(\mathbf{n}_i, p_i - q_j) = e^{-(\mathbf{n}_i^\top(p_i - q_j))^2/\sigma_p^2},$$

and adjust the locations of the point p_i by replacing $\theta(\|p_i - q_j\|)$ with $\phi(\mathbf{n}_i, p_i - q_j)$ in the first term of the LOP expressions of [Huang et al. 2009].

Specifically, given a set of points Q as in the beginning of this section, our anisotropic LOP algorithm defines a set of projected points $P = \{p_i\}_{i\in I} \subset \mathbb{R}^3$ by a fixed point iteration $P^{k+1} = G(P^k)$, $k = 0, 1, \ldots$, where

$$G(P^k) = \underset{P=\{p_i\}}{\operatorname{argmin}} \left\{ \sum_{i\in I}\sum_{j\in J} \|p_i - q_j\|\phi(\mathbf{n}_i, p_i^k - q_j) + \sum_{i\in I} \lambda_i \sum_{i'\in I\setminus\{i\}} \eta(\|p_i - p_{i'}^k\|)\theta(\|p_i^k - p_{i'}^k\|) \right\}. \tag{3}$$

The repulsion function is $\eta(r) = -r$. We choose the initial set P^0 as a random subset of Q and its initial normal estimates are computed using PCA and smoothed using (2). The balancing terms $\{\lambda_i\}_{i\in I}$ vary at each point but depend on just one parameter, $0 \le \mu < .5$, controlling the repulsion force, as explained in [Lipman et al. 2007; Huang et al. 2009]. In general, larger values of μ impose higher penalties on points that get close to other points. We set $\mu = 0.45$ by default. Besides the repulsion parameter μ, another important parameter σ_p used in both spatial weighting functions ϕ and θ is tunable based on a rough density measure $h = d_{\text{bb}}/\sqrt{m}$, where d_{bb} is the diagonal length of the bounding box and m the size of the input set. The default setting is $\sigma_p = h$.

The resulting operator is therefore edge-aware and has the visible effect of gently pushing points away from edges, see Fig. 5(c) for example. This is because the normal-dependent weight function ϕ down-weights large variation in geometric similarity, defined as the height difference of point q_j over the tangent plane of the point p_i. If the noise level is high, we should increase the supporting neighborhood size σ_p to ensure the necessary pushing-away "strength" for a clearer point patch separation about the edges and piecewise smoothness away from the edges, e.g., compare the resampling results in Figs. 5(f) and 5(g). The latter leads to higher quality upsampling and hence surface reconstruction results.

The bilateral normal adjusting formula (2) and the anisotropic LOP (3) can be applied alternately several times (our default value is 3), benefiting each other. Referring to the examples in Fig. 5, starting from our resampling with reliable normals, we are able to progressively upsample the point set towards the edges, see Figs. 5(d) and 5(h). Fig. 2 further shows that such resampling enables the RIMLS procedure to retain sharp features, see Fig. 2(h), in marked improvement over Fig. 2(e).

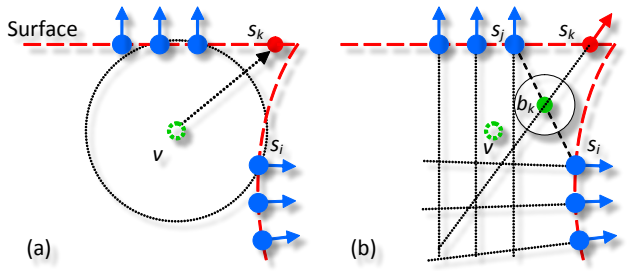


Fig. 6. Base location selection: (a) Inserting a point s_k at the center of the V-cell v then projecting it onto the surface may push it out of the V-cell; (b) selecting midpoint under orthogonal distance picks b_k instead of v. After inserting s_k and projecting it along its normal, the clearance at b_k becomes zero, preventing other points being inserted at b_k.

5. EDGE-PRESERVING UPSAMPLING

Given an oriented point set S with visually bald patches along edges as, e.g., in Fig. 5(c), we now describe how to carefully resample points, approaching edges with reliable data. This is done through a sequence of insertion operations. For each insertion we add a new oriented point $(p_k, \mathbf{n}_k)$ that fulfills three objectives: (i) p_k lies on the underlying surface; (ii) $\mathbf{n}_k$ is perpendicular to the underlying surface at p_k; and (iii) points are evenly distributed in the local neighborhood after insertion.

Finding p_k and $\mathbf{n}_k$ directly under the above objectives requires searching within the 5D solution space, which can be difficult, especially in the vicinity of sharp features. To make the problem tractable, we design a novel projector, where the projection direction is constrained to be along the normal of the inserted point, i.e., $p_k = b_k + d_k\mathbf{n}_k$. The computation of the oriented point $(p_k, \mathbf{n}_k)$ is then realized in three steps: finding a near-sparsest insertion base b_k (objective (iii)), optimizing the projection distance d_k to move the point onto the latent surface (objective (i)), and computing the normal direction $\mathbf{n}_k$ so that it fits the neighborhood normal distribution and best preserves sharp features (objective (ii)).

Base selection. The goal of choosing a good base b_k is to ensure fast convergence to an even point distribution within the local neighborhood, where the ensuing projection must be taken into account. Here we first discuss how to choose a base location in the neighborhood of an existing point s_i. Discussion on how to pick s_i is deferred to the end of this subsection.

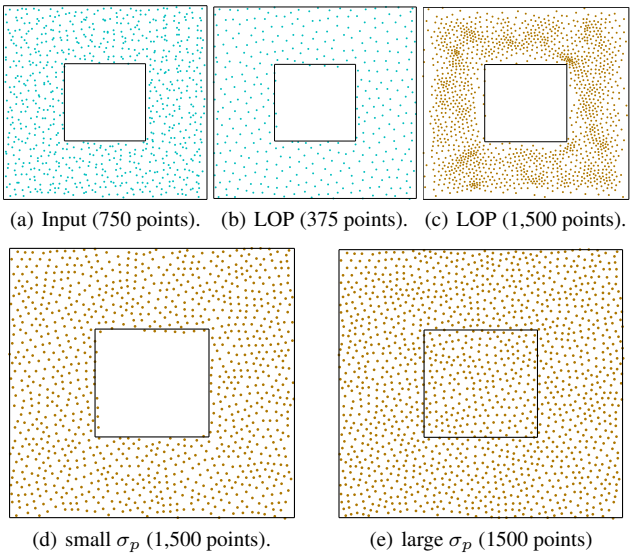
When finding the base location for point insertion the classical approach, e.g., [Alexa et al. 2003], uses a local Voronoi construction and picks the center of the largest empty Euclidean ball. However, the Euclidean distance does not take into account the neighboring normals which dictate the ensuing projection. As shown in Fig. 6(a), in the vicinity of sharp features the projection may push the inserted point outside its clearing space, keeping the ball empty. This would attract subsequent points being inserted into the same base, namely the center of the ball, thus stalling the upsampling.

Therefore, we incorporate projection direction information into the distance consideration when searching for the largest clearance ball. Specifically, considering an existing point s_i and its neighbors in the set $\mathcal{N}_{s_i}$, we wish to insert at a location b in s_i's neighborhood that maximizes $C(b) = \min_{s_{i'} \in \mathcal{N}_{s_i}} D(b, s_{i'})$. Rather than using the Euclidean distance for the distance function $D(b, s_{i'})$, we define it as the *orthogonal distance* between b and the normal extension at $s_{i'}$,

$$D(b, s_{i'}) = \|b - p_{i'} - \mathbf{n}_{i'}^\top (b - p_{i'})\mathbf{n}_{i'}\|.$$

Since the distance calculation considers the normal directions of inserted points, which are also their projection directions, the stalling problem is solved. Once a new oriented point s_k is inserted at b, we have $C(b) = D(b, s_k) = 0$. The chance of further insertion near b is significantly reduced since even after projection, the normal extension at point s_k passes through b; see Fig. 6(b).

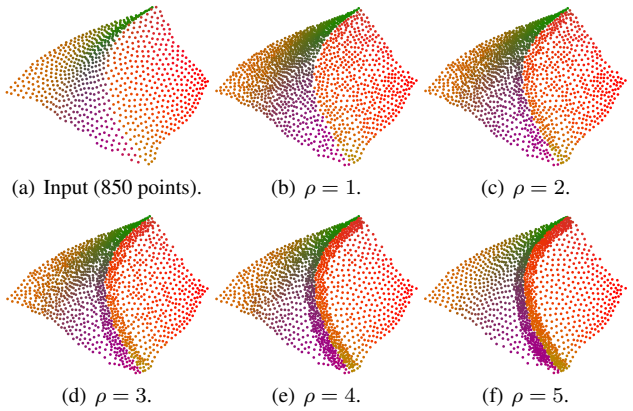
Computing the optimal position b in the neighborhood of s_i based on $D(\cdot, \cdot)$ requires solving a constrained quadratic equation. To



(a) Input (750 points). (b) LOP (375 points). (c) LOP (1,500 points).

(d) small σ_p (1,500 points). (e) large σ_p (1500 points)

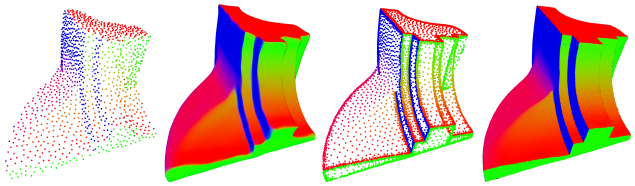
Fig. 7. Resampling on a planar point set, which extends the example in Fig. 6 of [Lipman et al. 2007]. While LOP excels at downsampling (b), using it for upsampling (c) leads to uneven point distributions. Our approach inserts at midpoints of (b) with the largest clearances, yielding an evenly distributed point set (d). The inner boundary of (d) is well preserved under a small neighborhood size. Using a large neighborhood respects the outer boundary only and fills the interior with evenly distributed points (e).



(a) Input (850 points). (b) $\rho = 1$. (c) $\rho = 2$.

(d) $\rho = 3$. (e) $\rho = 4$. (f) $\rho = 5$.

Fig. 8. Upsampling results by EAR for different edge-sensitivity parameter values ρ. The input (a) contains 850 points and was upsampled to 1,500 points in all the cases shown in (b - f). Larger values of ρ give higher priority to inserting points along the sharp edge.



(a) Input. (b) RIMLS. (c) EAR. (d) RIMLS.

Fig. 9. EAR facilitates surface reconstruction. Given a sparse set (a) with only 2,500 points, RIMLS reconstruction (b) does not produce a quality piecewise smooth surface. With upsampling by EAR to 7,500 points (c), the same RIMLS scheme results in a reconstruction of the Fandisk with better quality and sharp feature preservation.

quickly find an approximate solution, here we limit our search to the midpoints between s_i and its neighbors in $\mathcal{N}_{s_i}$ and select the midpoint with the largest clearance as the base b_k. Fig. 7 shows that while selecting among midpoints is only an approximation, it is able to resample points with even distribution.

During the base selection we need to decide globally into which neighborhood the next point should be inserted. Depending on the application we may either want to insert the point where the density is lowest (for uniform resampling) or place it along sharp boundaries (for sharp feature enhancement). To achieve both objectives in a unified approach, we define the priority for the neighborhood of a given point s_i as

$$P(s_i) = \max_{s_{i'} \in \mathcal{N}_{s_i}} (2 - \mathbf{n}_i^\top \mathbf{n}_{i'})^\rho C(\frac{p_i + p_{i'}}{2}),$$

where ρ is an edge-sensitivity parameter. When $\rho = 0$, the neighborhood priority for s_i is determined solely by the largest midpoint clearance. Hence, inserting a new point in the neighborhood with the highest priority actually places its base at the midpoint that has the largest clearance over the whole surface. When $\rho > 0$, higher priority is given to insert points along sharp edges where the normals vary, see Fig. 8. In our context, as we first resample away from potential edges to obtain reliable normals, at this resampling step for edge regeneration, we use the default value $\rho = 5$ to give higher priority near edges, achieving the effect of sharp feature enhancement, as shown, e.g., in Figs. 3(c) and 9(c).

It is worth noting that the ability to adaptively sample the latent surface is an important feature of our method. Careful inspection of Fig. 8 reveals that, with a high ρ value, our approach places *all* newly inserted points along sharp edges and corners. Hence, if the goal is to facilitate sharp feature reconstruction rather than direct point set rendering as in Fig. 3(d), only a small number of point samples need to be inserted; see Figs. 9 and 13 for examples. This allows our approach to handle very large models.

Projection distance under a given normal. With the base location b_k selected, we now discuss how to project it onto the latent surface along a given direction $\mathbf{n}$, through determining the step size d_k. This is achieved by minimizing a weighted total projection distance between $p = b_k + d_k\mathbf{n}$ and existing points in the neighborhood. To handle points near sharp features, the weights of neighboring points are determined based on distances in both Euclidean and directional spaces, giving the objective function

$$\sum_{s_i \in \mathcal{N}_{b_k}} (\mathbf{n}^\top (p - p_i))^2 \theta(\|p - p_i\|)\psi(\mathbf{n}, \mathbf{n}_i).$$

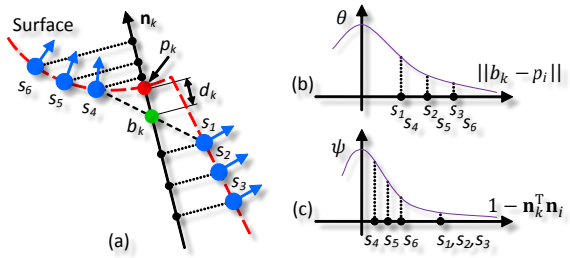


Fig. 10. Projection distance calculation. For each point s_i in (a), the Euclidean distance weight and range space weight are obtained using the corresponding weight functions shown in (b) and (c), respectively. The final projected location p_k is a weighted average of the projections of these points on the straight line defined by b_k and $\mathbf{n}_k$.

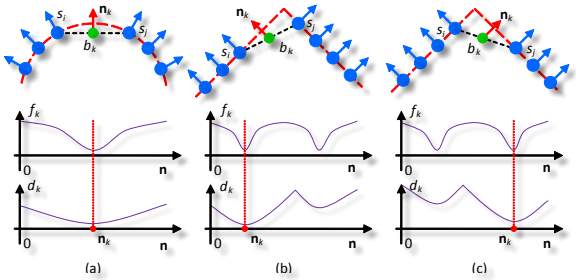


Fig. 11. A 2D example for normal determination. The first row shows locally optimal normal directions (red arrow) under different situations. The next two rows plot the directional difference and the projection distance, respectively. Note that (b) and (c), while appearing similar, have different desired normal directions for preserving even distribution after projection.

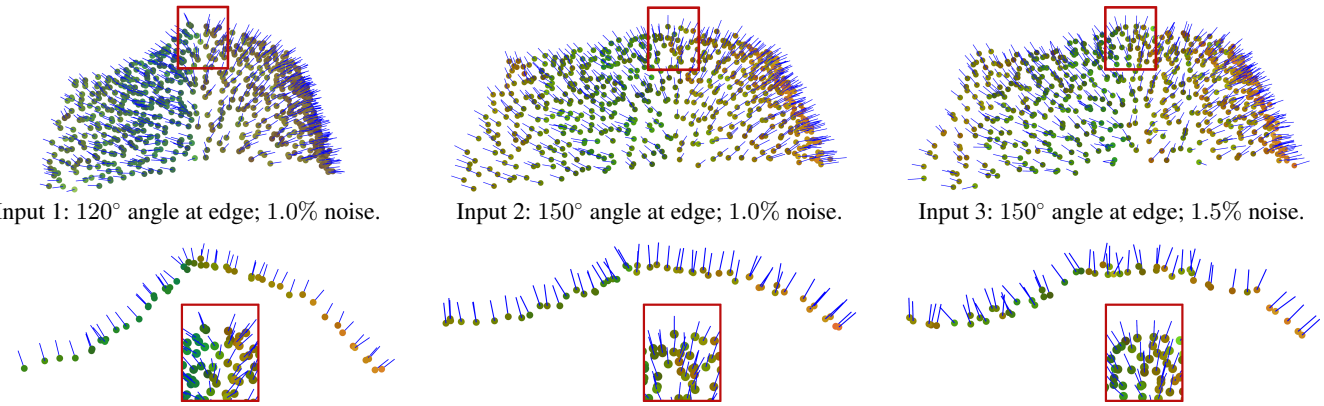
By fixing the spatial weight θ at b_k, we thus obtain the step size

$$d_k(b_k, \mathbf{n}) = \frac{\sum_{s_i \in \mathcal{N}_{b_k}} (\mathbf{n}^\top (b_k - p_i))\theta(\|b_k - p_i\|)\psi(\mathbf{n}, \mathbf{n}_i)}{\sum_{s_i \in \mathcal{N}_{b_k}} \theta(\|b_k - p_i\|)\psi(\mathbf{n}, \mathbf{n}_i)}. \quad (4)$$

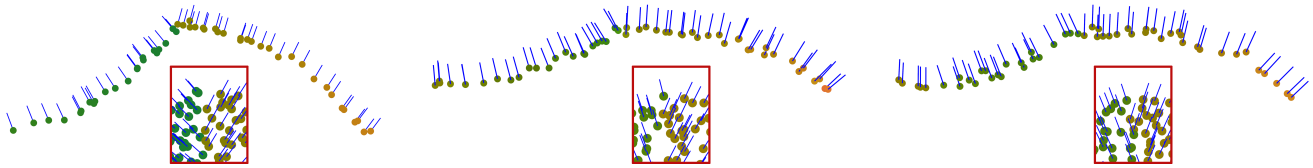
As shown in Fig. 10, the above calculation is equivalent to projecting all existing points onto the straight line defined by b_k and $\mathbf{n} = \mathbf{n}_k$ and then computing the weighted average location based on weights determined using both Euclidean and directional terms. In particular, the directional term ensures that points with normals more similar to the input $\mathbf{n}$ have higher weights. When the input normal $\mathbf{n}$ is accurate, this property ensures that the inserted point can be projected onto the latent surface even in the vicinity of sharp features. Finding a good normal direction is therefore critical.

Normal determination. To finally determine $p_k = b_k + d_k\mathbf{n}_k$, we now compute the normal vector $\mathbf{n}_k$. As shown in Fig. 11, here we have two selection criteria: (i) $\mathbf{n}_k$ fits the normal distribution in the local neighborhood of b_k, i.e., $f(b_k, \mathbf{n}_k)$ of (1) is small; and (ii) the moving distance $d_k(b_k, \mathbf{n}_k)$ defined by (4) is also kept small so that the even distribution we had sought can be better preserved.

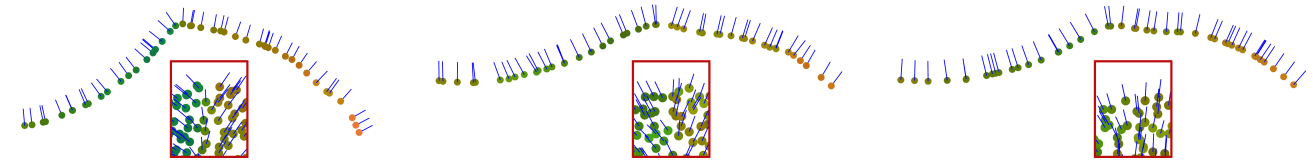
To compute $\mathbf{n}_k$ efficiently, we limit our search to the two neighborhoods surrounding $\mathbf{n}_i$ and $\mathbf{n}_j$, respectively, where $\mathbf{n}_i$ and $\mathbf{n}_j$ are the normals of the two endpoints used for generating b_k; see Fig. 11. We first decide which neighborhood to use based on moving dis-



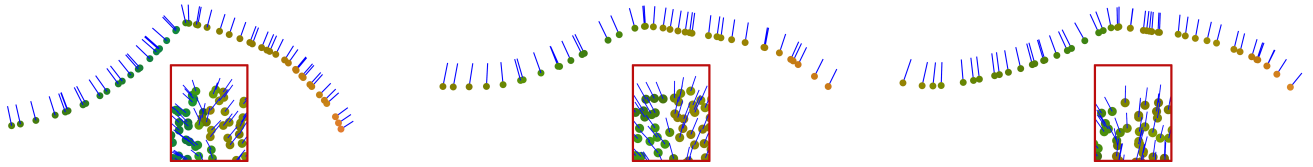
(a) A profile showing normals computed by classical PCA are still noisy (red boxes show zoom-ins near the edge).



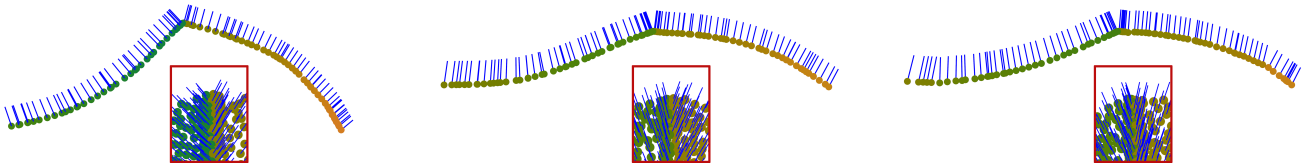
(b) PCA normals after bilateral normal smoothing do not reveal the edges.



(c) Normals after applying RIMLS to (b) showing quality degradation as noise and edge angle increase.



(d) Normals after applying ℓ_1-minimization to (a) showing quality degradation as noise and edge angle increase.



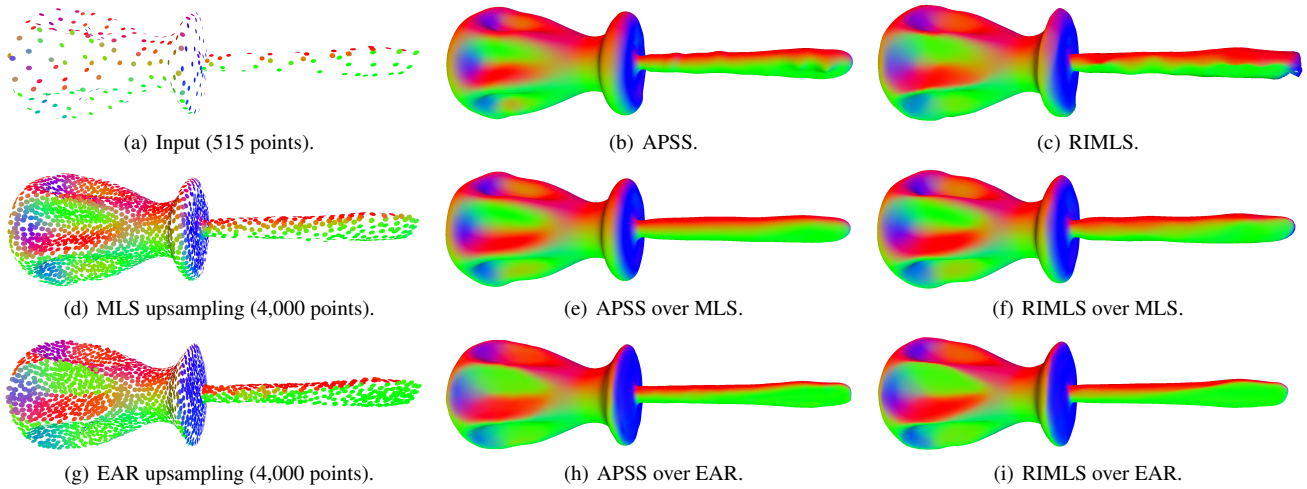
(e) Normals after applying EAR to (a) showing better handling of noise and revelation of soft edges.

Fig. 12. A comparison between EAR (e) and several well-known normal estimation and processing schemes including PCA (a), bilateral normal smoothing (b), RIMLS (c), and ℓ_1-minimization (d) [Avron et al. 2010] (results courtesy of Haim Avron). Three synthetic input point sets (shown in the first row) were tested at two noise levels. For each input, the underlying shape is characterized by a soft edge (120° or 150° angle) with 1K points. The results shown via 2D cross-sections demonstrate that our resampling approach is not only robust to noise but also capable of handling soft edges.

tance d_k, that is,

$$l = \underset{l \in \{i,j\}}{\operatorname{argmin}}\, d_k(b_k, \mathbf{n}_l),$$

and then compute the normal $\mathbf{n}_k$ by minimizing $f(b_k, \mathbf{n})$ while holding the directional weight $\psi(\mathbf{n}, \cdot)$ at the fixed normal direction $\mathbf{n} = \mathbf{n}_l$. This allows us to compute $\mathbf{n}_k$ using (2) with a single iteration. Note that when both endpoints s_i and s_j are on the same smooth surface, for example in Fig. 11(a), the two neighborhoods surrounding $\mathbf{n}_i$ and $\mathbf{n}_j$ overlap. Therefore, the final solution $\mathbf{n}_k$ may be close to the average of $\mathbf{n}_i$ and $\mathbf{n}_j$, providing a smooth interpolation of existing normals.

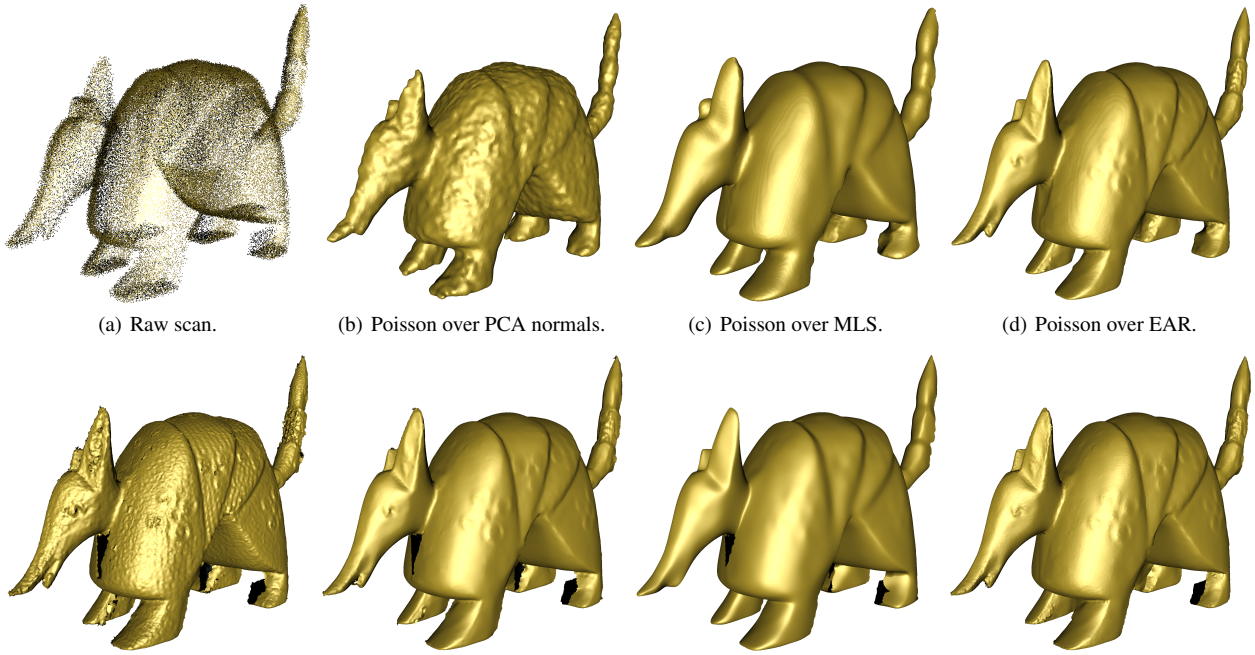


(a) Input (515 points). (b) APSS. (c) RIMLS.

(d) MLS upsampling (4,000 points). (e) APSS over MLS. (f) RIMLS over MLS.

(g) EAR upsampling (4,000 points). (h) APSS over EAR. (i) RIMLS over EAR.

Fig. 13. Power of edge-preserving upsampling in our EAR scheme. For a clean and oriented point set with a rather low density (a), APSS (b) and RIMLS (c) cannot provide a good surface definition without upsampling. MLS upsampling (d) improves the performances of APSS (e) and RIMLS (f), but smear sharp features. With EAR (g), APSS (h) and RIMLS (i) successfully preserve the sharp features.



(a) Raw scan. (b) Poisson over PCA normals. (c) Poisson over MLS. (d) Poisson over EAR.

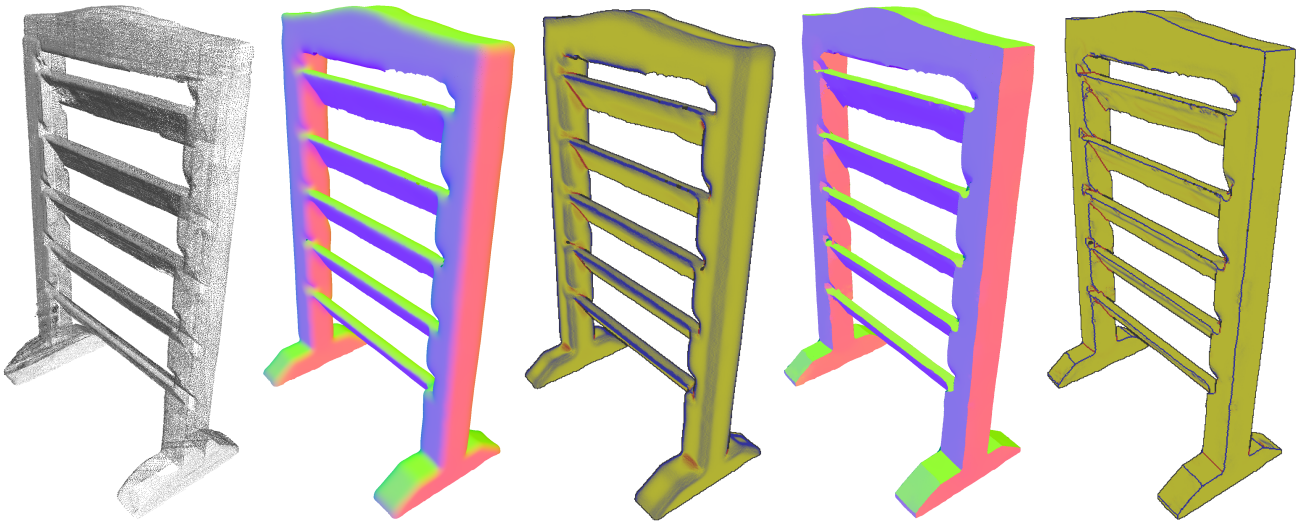
(e) RIMLS over PCA normals. (f) RIMLS over S with $\sigma_p = h$ (g) RIMLS over S with $\sigma_p = 10h$. (h) RIMLS over EAR with $\sigma_p = h$.

Fig. 14. Result comparison on a raw scan (a) using the edge-oblivious Poisson [Kazhdan et al. 2006] and edge-aware RIMLS surface reconstructions. (b) Poisson over oriented PCA normals. (c) Poisson over a filtered and upsampled point set using MLS. (d) Poisson over output of EAR. (e) RIMLS over the same oriented PCA normals used in (b). (f) RIMLS over an oriented point set S after resampling away from edges with $\sigma_p = h$. (g) RIMLS over the same point set S with a much larger σ_p. (h) RIMLS over the output of the complete EAR with the same σ_p as in (f). The ability of EAR to lead to piecewise smooth and feature preserving reconstructions in both scenarios is evident.

6. RESULTS AND DISCUSSION

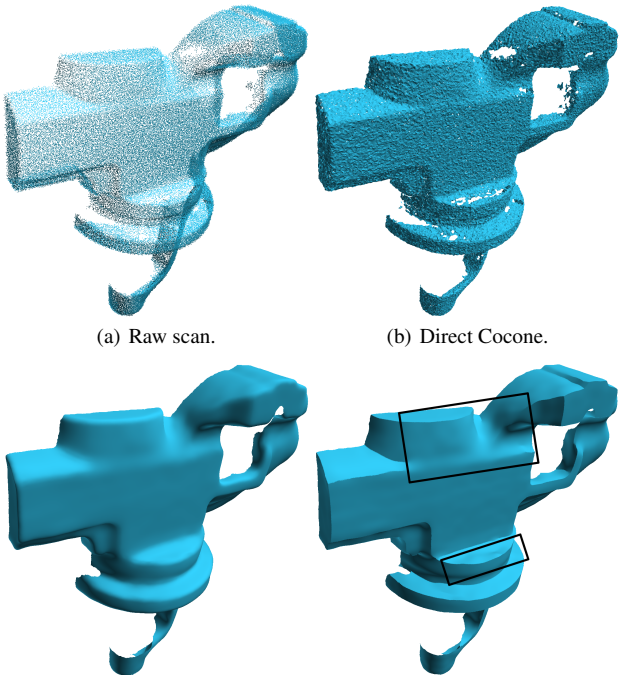
The presented EAR algorithm was tested on a variety of raw and synthetic point scans. Processing times on raw scans are provided in Table I. Unless specifically indicated in the captions, the default parameter values as given in Sections 4 and 5 were applied for obtaining the presented results. Below we first elaborate on some typical obtained results and then discuss limitations.

Results. We first show EAR at work on a few synthetic models containing both sharp and soft features separated by smooth sur-



(a) Raw scan. (b) MLS. (c) Radiance scaling over (b). (d) EAR. (e) Radiance scaling over (d).

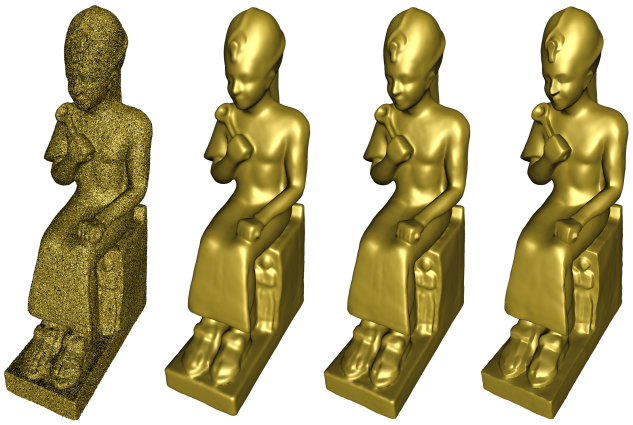
Fig. 17. EAR for surfel point set rendering, where each output surfel is displayed using a single pixel and colored by its normal direction. The input scan (a) of a shutter blind is noisy and unevenly distributed. MLS resampling (b) smears the edges whereas EAR (d) preserves them well. Comparing (c) and (e), dominating edges are enhanced on EAR point set surface using radiance scaling [Vergne et al. 2010].



(a) Raw scan. (b) Direct Cocone. (c) Cocone over MLS. (d) Cocone over EAR.

Fig. 15. Result comparison using the edge-oblivious Cocone surface reconstruction [Dey and Giesen 2001].

face patches. In Figure 3, we use the Fandisk model to demonstrate the capability of EAR to handle noisy input data; the input point cloud was corrupted with both noise (2% of the bounding box) and outliers (10% of the bounding box). The results show that EAR not only smooths out noise in point positions, but also effectively handles smooth variations and sharp changes in normals. Fig. 12 compares our resampling scheme with several well-known



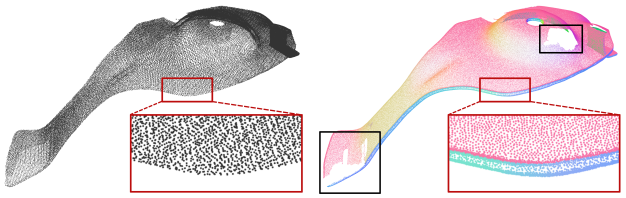
(a) Input. (b) $\sigma_n = 60°$. (c) $\sigma_n = 30°$. (d) $\sigma_n = 15°$.

Fig. 16. Results of applying EAR with different parameters before Poisson surface reconstruction on a dense noisy Ramesses model (800K points). As expected, small values of σ_n tend to better enhance sharp features, but may over-sharpen highly curved areas.

Table I. Timing for EAR on four raw scans.

	IP-N	FP-N	RA-T	OP-N	RC-T
Figure 1	222,543	22K	49 sec	800K	42 sec
Figure 15	161,994	16K	31 sec	700K	36 sec
Figure 14	99,416	10K	17 sec	600K	29 sec
Figure 17	291,365	30K	58 sec	1.8M	73 sec

IP-N: number of input points; FP-N: number of noise-free points; RA-T: time for resampling away from edges; OP-N: number of output points; RC-T: time for upsampling close to edges. All examples were run on an Intel(R) Core(TM) i7 CPU 860@2.80GHz with 2GB RAM.



(a) Raw scan (82K points). (b) EAR results.

Fig. 18. With accurate normals estimated near regions of close-by surface sheets (a), our resampling algorithm performs well; see the middle zoom-in in (b). However, the upsampling step is limited by the choice of the σ_p parameter and may not fill sufficiently large holes present in the point sampling; see zoom-in's with black borders in (b).

normal estimation and smoothing schemes including PCA, bilateral normal smoothing, RIMLS and ℓ_1-minimization filtering on a set of shallow fin-shapes with different edge angles and noise levels. These results demonstrate superior performance of our method in terms of robustness to noise and effective recovery of soft edge features. Fig. 13 demonstrates the power of EAR, compared to MLS, in resampling and more specifically, edge-preserving upsampling, a highly sparse point set surface containing sharp features. With a small amount of new insertions, the performance of reconstruction methods such as APSS and RIMLS can be noticeably improved.

Figs. 1 and 13-16 all demonstrate how our EAR scheme can enhance the performance of existing surface reconstruction methods. Four representative techniques are chosen: RIMLS, APSS, Poisson and Cocone. The implementations of the first three are from MeshLab and the last was provided by its authors. We also implemented the well-known MLS projector which uses Voronoi-based upsampling [Alexa et al. 2003], as the classical resampling operator to compare our EAR scheme with. For all the existing algorithms, we show the best results we were able to generate, following the guidelines provided by the distributed codes and published parameters.

The inputs in Figs. 1, 14 and 15 are raw scans of real objects. The typical imperfections associated with digital scans, such as noise, outliers, non-uniform point distribution, and missing data, are ubiquitous in these data sets. EAR removes noise and outliers, fills in the missing parts, preserves sharp features and, at the same time, facilitates various surface reconstruction methods. In particular, we can observe the benefits of applying resampling prior to surface reconstruction and EAR is evidently out-performing MLS with respect to preservation of sharp features. The proposed two steps, resampling away from edges and edge-preserving upsampling, are both indispensable for a satisfactory reconstruction, as shown in Fig. 14 for a wide variety of parameter values.

Fig. 16 presents results of applying EAR with different σ_n before Poisson surface reconstruction on a large noisy data set, which is more challenging and contains geometric details of different sizes. By decreasing the parameter value σ_n, we are capable of better enhancing sharp features. However, if σ_n is set too small, some highly curved areas may be over-sharpened.

We also show the results of converting noisy and unoriented point sets into clean point set surfaces with reliable, edge-aware normals for direct surfel-based point set rendering. Both synthetic (Fig. 3) and raw (Fig. 17) point scans were employed, demonstrating the capability of EAR for handling smooth surfaces as well as sharp features under a unified framework.

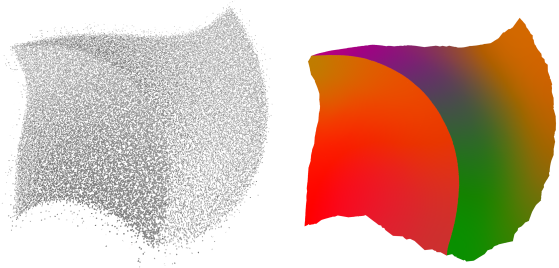


Fig. 19. One limitation of our resampling scheme is that it is not designed to handle open point sets — it may produce rough boundaries as a result.

Limitations. When the noise level in the input point cloud is high or the captured object contains close-by surface sheets, the initial normal orientation using oriented PCA may be erroneous, subsequently causing errors in the resampling. This problem can be alleviated by a more capable normal estimation technique, e.g., [Huang et al. 2009], which can handle the close-by surface sheet problem; this is illustrated in Fig. 18. In the same figure, we also see that relatively large holes in the input point sampling are not filled by our upsampling step in EAR since this step is limited by the choice of σ_p, which measures the size of the neighborhood. A specifically chosen large σ_p would lead to better gap filling. Also, we currently use a fixed neighborhood radius σ_p and angle parameter σ_n. Although the default parameter setting has worked well for most of our test cases, certain special situations may require a careful tuning of these two parameters to deal with elusive sharp features.

Another limitation of our EAR scheme is that it is not designed to handle open boundaries and may produce less than satisfactory results in that situation, as shown in Fig. 19. Finally, under severe noise or under-sampling, our scheme may over-smooth geometric details and sharp features or over-sharpen an edge depending on the edge sharpness, as can be observed in Fig. 14 by comparing (e) with (h) and in Fig. 15(d); e.g., see edges in the black boxes. Currently, our scheme does not have an adaptive parameter to locally control the sharpness of the resampling results.

7. CONCLUSIONS

We have described a novel resampling algorithm for converting noisy scan data into a clean point set surface endowed with reliable normals. The core of our algorithm consists of two phases: a robust edge-aware method which computes reliable normals away from surface singularities is followed by a novel bilateral projector which progressively upsamples toward these singularities. Both phases are edge-aware and hence the resampling results preserve sharp features. Specifically, the presented resampling algorithm enjoys the following properties: (i) sharp edges are preserved and smooth surfaces are maintained under a unified approach; (ii) the resampling can reach any density requirement specified by users; (iii) the reliable normals generated greatly facilitate existing surface reconstruction methods.

In the future, besides research on open boundary handling, we aim to develop a point set surface modeling system, where users can perform local editing such as smoothing, edge enhancing and hole-filling directly using resampling tools presented in this paper. We would also like to design a real-time GPU-based resampling algorithm, providing users with unlimited zoom capabilities when viewing point set surfaces, without the need for mesh generation.

Finally, it is desirable to include an adaptive parameter to locally control the sharpness of our resampling results, in particular when sharp regions are severely corrupted and under-sampled.

Acknowledgments

The authors would like to thank all the reviewers for their valuable comments. This work is supported in part by grants from NSFC (61103166), National 863 Program (2011AA010503), Guangdong Science and Technology Program (2011B050200007), Shenzhen Science and Innovation Program (CXB201104220029A), NSERC (No. 84306 and No. 611370) and the Israel Science Foundation.

REFERENCES

ADAMSON, A. AND ALEXA, M. 2006. Point-sampled cell complexes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 25,* 3, 671–689.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. *IEEE Trans. Vis. & Comp. Graphics*, 21–28.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and rendering point set surfaces. *IEEE Trans. Vis. & Comp. Graphics 9,* 1, 3–15.

ALLIEZ, P., COHEN-STEINER, D., TONG, Y., AND DESBRUN, M. 2007. Voronoi-based variational reconstruction of unoriented point sets. *Proc. Eurographics Symp. on Geometry Processing*, 39–48.

AVRON, H., SHARF, A., GREIF, C., AND COHEN-OR, D. 2010. ℓ_1-sparse reconstruction of sharp point set surfaces. *ACM Trans. on Graphics 20,* 5, 135.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3D objects with radial basis functions. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 67–76.

DEY, T. AND SUN, J. 2006. Normal and feature approximations from noisy point clouds. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*. Lecture Notes in Computer Science, vol. 4337. 21–32.

DEY, T. K. AND GIESEN, J. 2001. Detecting undersampling in surface reconstruction. In *Symp. on Comp. Geom.* 257–263.

FLEISHMAN, S., COHEN-OR, D., AND SILVA, C. T. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 24,* 3, 544–552.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 22,* 3, 950–953.

GROSS, M. AND PFISTER, H. 2007. *Point-Based Graphics*. Morgan Kaufman.

GUENNEBAUD, G., BARTHE, L., AND PAULIN, M. 2004. Real-time point cloud refinement. *Proc. of Symposium on Point-Based Graphics*, 41–49.

GUENNEBAUD, G., GERMANN, M., AND GROSS, M. 2008. Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum (special issue of Eurographics) 27,* 3, 653–662.

GUENNEBAUD, G. AND GROSS, M. 2007. Algebraic point set surfaces. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 26,* 3, 23.

HILDEBRANDT, K. AND POLTHIER, K. 2004. Anisotropic filtering of non-linear surface features. *Computer Graphics Forum (special issue of Eurographics) 23,* 3, 391–400.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 71–78.

HUANG, H. AND ASCHER, U. 2008. Surface mesh smoothing, regularization and feature detection. *SIAM J. Scient. Comput. 31,* 1, 74–93.

HUANG, H., LI, D., ZHANG, H., ASCHER, U., AND COHEN-OR, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH ASIA) 28,* 5, 176.

JONES, T., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature preserving mesh smoothing. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 22,* 3, 943–949.

JONES, T. R., DURAND, F., AND ZWICKER, M. 2004. Normal improvement for point rendering. *IEEE Computer Graphics and Applications 24*, 53–56.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. *Proc. Eurographics Symp. on Geometry Processing*, 61–70.

LANGE, C. AND POLTHIER, K. 2005. Anisotropic smoothing of point sets. *Comput. Aided Geom. Des. 22,* 7, 680–692.

LIPMAN, Y., COHEN-OR, D., AND LEVIN, D. 2007. Data-dependent MLS for faithful surface approximation. *Proc. Eurographics Symp. on Geometry Processing*, 59–67.

LIPMAN, Y., COHEN-OR, D., LEVIN, D., AND TAL-EZER, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 26,* 3, 22.

MERIGOT, Q., OVSJANIKOV, M., AND GUIBAS, L. 2009. Robust voronoi-based curvature and feature estimation. *SIAM/ACM Joint Conf. on Geometric and Physical Modeling*, 1–12.

MIAO, Y., DIAZ-GUTIERREZ, P., PAJAROLA, R., GOPI, M., AND FENG, J. 2009. Shape isophotic error metric controllable re-sampling for point-sampled surfaces. *Proc. IEEE Conf. on Shape Modeling and Applications*, 28–35.

MITRA, N. J., NGUYEN, A., AND GUIBAS, L. 2004. Estimating surface normals in noisy point cloud data. *Int. J. Comput. Geom. and Appl. 14,* 4–5, 261–276.

ÖZTIRELI, C., ALEXA, M., AND GROSS, M. 2010. Spectral sampling of manifolds. *ACM Trans. on Graphics (Proc. of SIGGRAPH ASIA) 29,* 5, 168.

ÖZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Comp. Graphics Forum 28,* 2, 493–501.

PAULY, M., GROSS, M., AND KOBBELT, L. P. 2002. Efficient simplification of point-sampled surfaces. *Proc. of IEEE Visualization*, 163–170.

PAULY, M., KEISER, R., KOBBELT, L. P., AND GROSS, M. 2003. Shape modeling with point-sampled geometry. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 22,* 3, 641–650.

PFISTER, H., ZWICKER, M., VAN BAAR, J., AND GROSS, M. 2000. Surfels: surface elements as rendering primitives. In *ACM Trans. on Graphics (Proc. of SIGGRAPH)*. 335–342.

SALMAN, N., YVINEC, M., AND MERIGOT, Q. 2010. Feature preserving mesh generation from 3D point clouds. *Proc. Eurographics Symp. on Geometry Processing 29,* 5, 1623–1632.

SUN, X., ROSIN, P. L., MARTIN, R. R., AND LANGBEIN, F. C. 2007. Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. & Comp. Graphics 13,* 5, 925–938.

VERGNE, R., PACANOWSKI, R., BARLA, P., GRANIER, X., AND SCHLICK, C. 2010. Radiance scaling for versatile surface enhancement. *Proc. symposium on Interactive 3D graphics and games*.

WEBER, C., HAHMANN, S., AND HAGEN, H. 2010. Sharp feature detection in point clouds. *Proc. IEEE Conf. on Shape Modeling and Applications*, 175–186.