Construction of curvature-aligned meshes from point clouds

E. Kalogerakis, D. Nowrouzezahrai, P. Simari, K. Singh

Technical Report CSRG-569, November 2007, Computer Science Department, University of Toronto

Abstract

We present a robust framework for generating curvature-aligned meshes directly from oriented point clouds. We first present a novel approach to denoising the input point cloud using robust statistical estimates of surface normal and curvature to automatically reject outliers and correct points by energy minimization. We then generate lines of curvature on the corrected point cloud with controllable density. Finally, an anisotropic quad-dominant mesh is directly constructed from the corrected point cloud by detecting the intersections of these lines of curvature, without user interaction. Our approach is applicable to surfaces of arbitrary genus and is statistically robust to noise and outliers, while preserving sharp surface features. We show our approach to be effective over a range of synthetic and real-world input datasets with varying amounts of noise and outliers.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations I.3.5 [Computer Graphics]: Geometric algorithms, languages, and systems

1. Introduction

Incorporating physical objects, scanned into a digital form, is an integral part of many engineering and entertainment applications. The raw output of most shape acquisition methods is a point cloud sampling of the scanned surface. Given the popularity of polygon meshes for representing shape, the construction of meshes from point clouds is an active area of research. Different shape acquisition processes produce a wide range of characteristic point-clouds that commonly exhibit artifacts of irregular sampling, noise and outliers, and make automatic and general purpose mesh construction a challenging problem [JWB*06]. To further complicate the matter, several important applications, including physical simulation, geometry processing and character animation, make demands on mesh structure that are not met by existing surface reconstruction algorithms [ACSD*03]. Curvaturealigned meshes provide an optimal piecewise linear approximation of a smooth surface [Sim94, D'A00] and further mimic the flow-lines along which artists and animators place geometric elements to create 3D models [ACSD*03] or hatch strokes for model illustration [HZ00].

In this paper, we present a new approach to automatically construct curvature-aligned quad-dominant anisotropic meshes with user-controllable density, directly from oriented



Figure 1: Results of applying our method to a point cloud of Einstein's head. We produce a quad-dominant, anisotropic, curvature aligned mesh directly from the point cloud.

point clouds. While a number of quad-meshing techniques exist [ACSD*03, MK04, DKG05, DBG*06a, LPW*06], they operate on an input mesh and are thus likely to inherit any artifacts introduced during the construction of the intermediate mesh from the point cloud. Our approach, in contrast, is statistically robust to irregular sampling and noise, preserves sharp features and operates directly on the input point cloud.



Figure 2: Overview of the stages of our method. (a) Initial point cloud and normal estimates. (b) M-estimation of principal curvature directions. (c) De-noising of normal estimates and point locations using M-estimation weights. (d) Quad-dominant, anisotropic, curvature aligned mesh obtained by following curvature lines over the surface implied by the point cloud and tracking intersections. (e) Mesh close-up.

A schematic overview of our method is shown in Figure 2. The input to our technique is a point cloud with oriented normals. Oriented normal vectors can often be acquired as part of the scanning process or can be estimated with existing algorithms [HDD*92, ACSTD07]. The first three steps of the algorithm generates a corrected point cloud. Here we extend the robust statistical curvature estimation approach of Kalogerakis et al. [KSNS07] to reject outlier points and correct point positions and normals based on the statistical contributions of points while estimating surface curvature (see Figure 2c). We then trace lines of principal curvature with a specified density directly over the surface implied by the corrected point cloud. Finally, we use a Voronoi space partition to efficiently detect intersecting lines of curvature and construct a quad-dominant mesh as the output of our method (see Figure 2d).

We show the results of our approach on synthetic analytic examples with varying noise and sampling quality, models with sharp features, large umbilic regions, as well as commercially scanned real-world examples (Figures 1, 3, 5 and 7) and even highly noisy scans of reflective objects (Figure 8) acquired using scatter-trace photography [MK07].

2. Related Work

The method proposed in this paper is related to surface reconstruction from point clouds, quad remeshing and robust estimation of surface curvature. We discuss the main works in each area and our differences in the following subsections.

2.1. Surface reconstruction from point clouds

A considerable number of techniques for surface reconstruction from point clouds have been proposed in the computer graphics and vision literature. Our approach, however, follows a different methodology from the existing techniques. Our goal is to construct a curvature-aligned quad-dominant mesh directly from oriented point clouds by tracing lines of curvature on the input surface. We use a robust statistical framework in order to increase tolerance to noise, reject surface outliers and preserve sharp features. This framework is based on maximum likelihood estimates of curvature and robust normal correction which also guide the entire meshing procedure.

In the following, we present a brief categorization of the most recent methods. A survey of surface reconstruction methods can be also found in [JWB^{*}06].

Implicit functions: Methods for surface reconstructions based on implicit functions attempt to construct a scalar field of which a level set represents the desired surface. Once such a field is obtained, a marching cubes algorithm [LC87] reconstructs the surface by extracting said level set and producing a tessellation. The methods differ in the way the scalar field is built. One approach is to consider the signed distance to the oriented tangent plane of the closest point [HDD*92]. Signed distance can also be accumulated into a volumetric grid [CL96]. In these approaches, an issue to address is the preservation of sharp features [HDD*94]. Another approach to constructing the scalar field is to use radial basis functions (RBFs) which are fit to the data points [TO99]. Here, issues to consider are fitting RBFs efficiently to large datasets [CBC*01], preserving sharp features [DTS01] and adapting well to local shape complexity [OBA*05, OBS05].

Alternatively, by considering a density function centered at each data point, it is possible to extract the surface as the ridges of the implied scalar field [SG07]. Unsigned distance functions can be used instead with the advantage that no normal information is needed. In such a case, the surface is extracted by computing the minimum cut of a weighted spatial graph structure [HK06]. The reconstruction problem can also be formulated as a spatial Poisson equation where a hierarchy of locally supported functions are admitted [KBH06]. Such a Poisson-based reconstruction can also be performed efficiently with limited memory using a streaming framework [BKBH07].

In all of these methods, while a locally smooth surface is guaranteed by the set of functions used, the resulting mesh is a product of the marching cubes or alternative tessellation algorithm. These algorithms often produce meshes with badly shaped highly irregular faces (see Figures 5d and 7d).



Figure 3: Results of applying our approach to some sampled analytic surface. **From left to right:** regularly sampled non-noisy sphere, randomly sampled noisy sphere, regularly sampled non-noisy torus, randomly sampled noisy torus, randomly sampled cube. Note how in the case of the sphere and cube where umbilics predominate, the meshing is aligned with the principal component axes. Noise is 1% of the bounding-box diagonal.

Moreover, there is the often underlying assumption that one is dealing with a closed surface, which may not necessarily be the case. In contrast, our approach makes no such assumption and the faces produced by our tessellation are predominantly nicely shaped, naturally anisotropic quads oriented with the surface curvature.

Moving Least Squares: In MLS, the surface is defined as an invariant set of a projection operator [Lev98]. The MLS method employs locally weighted least-squares polynomial approximations using fast decaying weight functions [ABCO^{*}03]. Here too it is possible to work without normal information or a local parameterization [LCOLTE07].

Issues to address in this approach include choice of the support size of the weighting function [LCOL06], excluding outliers [FCOS05] and preserving sharp features [RJT*05, LCOL07]. Our approach deals holistically with these issues by using M-estimation to obtain a maximum likelihood estimate of local curvature. This process (as we will show) can integratively remove outliers, determine local sample weighting, and correct point normals and positions while preserving sharp surface features. Moreover, all information yielded by this analysis will be leveraged during the meshing process, rather than being discarded and treating this next step as completely independent.

Computational geometry methods: Several methods approach the reconstruction problem from a computational geometry point of view by using combinatorial structures such as Delaunay triangulations [Boi84, KSO04], alpha shapes [EM94, BBX95, BMR*99] or Voronoi diagrams as in the case of the power crust method [ABK98, ACK01]. Modifications to the power crust method producing more accurate output in the presence of noise have also been proposed [MAVdF05]. Here too it is possible to produce reconstructions from unoriented point sets, with the added guarantee of yielding watertight surfaces [ACSTD07]. A geometric convection technique also makes it possible to reconstruct closed surfaces from very large streaming sets of non-uniformly distributed point [ACA07].

As with RBF approaches, there are underlying assumptions as to the nature of the sampled surface. Namely, these approaches often assume a watertight surface. In contrast, as mentioned above, our approach makes no assumptions on the genus of the sampled surface, making it applicable to a wider range of inputs. In addition, the vertex locations on meshes resulting from computational geometry approaches are largely determined by input point locations in one-to-one correspondence, while our approach naturally resamples the surface as necessary.

Statistical techniques: Machine learning techniques have also been employed for surface reconstruction. These include neural networks [IJS03], support vector machines [SSB05], as well as energy minimizing techniques for surface fitting and registration [YHW06]. A novel surface reconstruction using Bayesian statistics is also presented in [JWB*06,DTB06] using a prior probability distribution over the set of all possible original scenes. A Bayesian method is also employed in [HAW07] for joint surface reconstruction and registration. Finally, another interesting approach is to reconstruct an object via partial matching with shapes in a database [GSH*07].

Our method is most in line with these techniques in that it is based on robust statistics. However, most of these methods correct surface points and sometimes resample them, but still result in a point cloud representation. Our method also removes outliers and corrects surface point positions and normals, in our case using the obtained maximum likelihood estimates of differential properties. However, it then also uses these same estimates on the processed data to obtain a mesh representation of the entailed surface.

2.2. Quad remeshing

Alliez et al. [ACSD*03] propose remeshing an existing polygonal object representation so that lines of minimum and maximum curvatures are used to determine the edges for the remeshed version in anisotropic regions. In order to track the lines of curvature, the initial mesh is globally parametrized, while the curvature tensor field also needs to be pre-smoothed. Marinov and Kobbelt [MK04] instead provide a more efficient framework that does not rely on a global parametrization for anisotropic remeshing. Other quad remeshing techniques have also been proposed using smooth harmonic scalar fields [DKG05] or Laplacian eigenfunctions [DBG*06b].

In our case, we aim at directly constructing a curvature-

E. Kalogerakis & D. Nowrouzezahrai & P. Simari & K. Singh / Construction of curvature-aligned meshes from point clouds



4

Figure 4: Point cloud data used as direct input to our method.

aligned mesh from an oriented point cloud with controllable density, even in the presence of noise or outliers. The information yielded from the robust statistical estimation of surface curvature is used to subsequently remove outliers, correct normals and point positions, and extract the lines of curvature from the initial point cloud to directly yield an anisotropic, quad-dominant, curvature aligned mesh. There is no need to pre-mesh, globally parameterize, or make any topological assumptions regarding the underlying surface.

2.3. Robust curvature estimation in point clouds

There are few curvature estimation techniques applicable to noisy point clouds with outliers [TT05]. Kalogerakis et al. presented a method to estimate principal curvature values and directions over polygon meshes and point clouds using a robust statistical framework [KSNS07]. We will rely on this method in order to extract the maximum likelihood estimates of curvature from point clouds and correct the surface normals. We also extend the approach to remove outlier points and denoise point positions in a principled fashion. Based on these estimates, we will extract the lines of curvature directly from the point cloud in order to proceed with the quad-dominant mesh reconstruction.

3. Surface correction

3.1. Statistical estimation of curvature

In [KSNS07], it was shown that an Iterative Reweighted Least Squares (IRLS) process can be used to achieve a robust estimation of curvature, minimizing the effects of noise. We briefly overview this method here. In subsections 3.2 and 3.3 we introduce the extensions that allow us to (respectively) remove surface outliers from the input point set and denoise point positions using the results from the M-estimation process.

The first step of the algorithm is to determine a minimum



Figure 5: Comparison of results. (a) Results of our method on the cow data set using dense meshing. (b) Results of applying Poisson surface reconstruction [KBH06] to the same dataset using a depth setting of 8 (chosen so as to use approximately the same number of triangles that our results would have if we were to triangulate them.) (c) Close-up of our approach and (d) close-up of Poisson surface reconstruction. (e) and (f) Two progressively coarser meshes produced by our method by decreasing the density parameter.

neighborhood for each point p_i in the initial dataset (see figure 6a). As in [JWB*06], this minimum neighborhood is determined by finding the closest points after projecting them into the local tangent plane of p_i , considering one closest point for each of six 60° slices around p_i on this plane. If there are no nearest points in two or more contiguous slices around p_i within a given threshold, the point is marked as



Figure 6: (a) Boundary point definition and conditions. (b) Normal variation sample for curvature estimation.

boundary point. If p_i is not a boundary point, we consider all pairs of points and their associated normals inside their minimum neighborhood. Each such pair yields a positional variation $\vec{\Delta p}$ and normal variation $\vec{\Delta n}$ which constrain the curvature tensor as follows:

$$\underbrace{\begin{bmatrix} \nabla_{\vec{u}}\vec{N}\cdot\vec{u} & \nabla_{\vec{v}}\vec{N}\cdot\vec{u} \\ \nabla_{\vec{u}}\vec{N}\cdot\vec{v} & \nabla_{\vec{v}}\vec{N}\cdot\vec{v} \\ \nabla_{\vec{u}}\vec{N}\cdot\vec{w} & \nabla_{\vec{v}}\vec{N}\cdot\vec{w} \end{bmatrix}}_{unknowns} \cdot \begin{bmatrix} \vec{\Delta p}\cdot\vec{u} \\ \vec{\Delta p}\cdot\vec{v} \\ \vec{\Delta p}\cdot\vec{v} \end{bmatrix} = \begin{bmatrix} \Delta n\cdot\vec{u} \\ \vec{\Delta n}\cdot\vec{v} \\ \vec{\Delta n}\cdot\vec{w} \end{bmatrix}$$

where \vec{N} is the normal vector field, and \vec{u} , \vec{v} and \vec{w} form a local orthonormal coordinate frame obtained from the tangent plane (see figure 6b).

Given enough variation pairs, we obtain an overconstrained system, which lets us solve for the curvature tensor values in a least squares fashion. This estimation serves as an initial guess to the IRLS process. Then, all normal variations inside an initial operating region are sampled and assigned with geometric weighting scheme according to the inverse of their average squared Euclidean distance to the center point p_i . The initial operating region is heuristically defined as the Euclidean ball centered at p_i with radius 3.0 multiplied by the average distance of p_i from its closest neighbors in its minimum neighborhood. According to the M-estimation literature, the IRLS approach assigns statistical weights to the normal variation samples for each iteration given their observed residual $r_{i,x}$ from the currently fitted linear model x

$$w(r_{i,x}/\sigma) = \frac{2}{(1 + (r_{i,x}/\sigma)^2)^2}$$

where $\sigma = 1.4826 \cdot median(r_{i,x})$. At each iteration, the operating region is refined by considering the normal variation samples whose residuals are less than 2σ . The samples which have larger residuals are considered outliers for the curvature estimation of p_i and are ignored, as in [SAG95]. These statistical weights are chosen so that a cost function of the residuals of the samples is minimized and this corresponds to the maximum likelihood estimates of the curvature tensor [FP02].

As shown in [KSNS07], the initial normals can be corrected using the computed curvature tensors and the final Mestimation weights per each normal variation sample. Firstly,

Copyright November 2007.

the normal differences between p_i and every point in its final operating region are computed using the values for the unknowns as estimated from the IRLS process. Then, the new normal at p_i is computed as the normalized weighted sum of the normals of its neighbor points in the operating region plus the derived normal differences. These weights are the final weights of the IRLS process assigned to each sample. Let us denote with $w_{j,k}^i$ the final weight associated to the variation pair (p_j, p_k) for the estimation of curvature at point p_i (see figure 6b).

3.2. Surface outlier rejection

For each point p_i we can define a sparse weight vector \mathbf{w}_i as follows:

$$\mathbf{w}_i[p_j] = \sum_k w_{j,k}^i$$

which intuitively represents how much p_j contributes to determining curvature at p_i . If the weight is close to 1, then its associated normal variation is strongly related to the curvature of the point (p_i considers it an inlier). If it is 0, its associated normal variation is unrelated (p_i considers it an outlier).

Consider two points in the dataset p_1 with weight vector \mathbf{w}_1 and p_2 with weight vector \mathbf{w}_2 . In the case where $\mathbf{w}_1[p_2] > 0$ and $\mathbf{w}_2[p_1] = 0$, the point p_2 considers p_1 as an outlier in its curvature estimation, while the same does not hold for p_1 (this is a vote from p_2 for p_1 for being an outlier). On the other hand, if $\mathbf{w}_1[p_2] = 0$ and $\mathbf{w}_2[p_1] = 0$, the points are mutually irrelevant to each other's curvature estimation (no vote). If $\mathbf{w}_1[p_2] > 0$ and $\mathbf{w}_2[p_1] > 0$, both points contribute to each others curvature (this is a vote from p_2 for p_1 for being an outlier).

If over half of a point's votes are in favor of considering it an outlier, we mark it as such and ignore it during the next steps of our method (if a more conservative rejection is desired, this threshold may be reduced). Isolated boundary points are also ignored. We show this in the case of the helicoid (figure 2) and fish (figure 8) datasets. After the outliers are rejected, the minimum neighborhood for each point is reselected.

3.3. Point cloud denoising

The recomputed normals from the M-estimation process can be used to correct the position of the rest of the surface points. This is based on a global cost-minimization process where goal is to move the position of the points in such a way so that the local first-order approximation of the normal in the minimum neighborhood of each point p_i matches its robustly corrected normal $\vec{n_i}$. The cost function is defined as follows

$$E = \sum_{i=1}^{N} \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} \sqrt{||\vec{n}_i - s \cdot \hat{n}(p_i, q_j^i, q_k^i)||}$$



6

Figure 7: (a),(b),(c) Results of applying our method to the hand dataset. Note the preservation of features. (d) Closeup of results produced by the Poisson surface reconstruction method [KBH06] to the same dataset using a depth setting of 12. While this setting produces approximately twice the number of triangles that our results would have if we were to triangulate them, the level of detail is comparable. Note the difference in mesh quality.

where q_j^i and q_k^i denote the *j*-th and *k*-th nearest neighbors of p_i respectively, $\hat{n}(p,q,r) = unit((q-p) \times (r-p))$, and s = 1 if $\vec{n}_i \cdot \hat{n}(p,q,r) \ge 0$ and s = -1 otherwise. Intuitively, $s \cdot \hat{n}(p,q,r)$ represents the oriented normal of the plane defined by *p* and its neighbors *q* and *r*. *N* is the number of points in the dataset, and K = 6 is the number of nearest neighbors we consider.

The goal of the minimization process is to correct the point positions so that the local normals of the planes match the corrected normals as given by the M-estimation process. We take the square root of the norm difference to the local corrected normals as we noticed this better preserves features, similarly to the square-root potentials used in [DTB06].

Such an optimization requires an analytic gradient in order to be performed efficiently. We employ the Polak-Ribiere conjugate gradient method [Noc91]. In figure 2 we show an example for a noisy helicoid. Figures 3, 7 and 8 also illustrate this technique. The reduction of noise can reach approximately 65% in the noisy cases of the torus and sphere. The optimization takes a few minutes for a 100K point dataset on a P4 3.0 GHz.

Notice that we do not explicitly place a penalty in our energy term for point movement and they are only locally moved based on their corrected normals. The minimization process does not result in global translation of points, as in such a case, this would not result in lower global energy. Given our choice of the optimization algorithm, the fact that we use the original point positions as the initial guess, and the use of an analytic gradient, it is ensured that we find a minimum for locally optimal point placement which is close to the original point positions.

4. Tracing lines of curvature and their intersections

After rejecting surface outliers and denoising the point dataset, our method starts to create the flow lines of principal curvature. As there is no prior mesh representation in our case, there is a need for special treatment of this process in order to make sure that the flow lines are extracted properly and no intersections are lost.

4.1. Tracing flow lines

A flow line is a piecewise linear curve created by sequentially sampling the surface in adaptive step sizes following the robust curvature estimates. Let us call such a curve $C = \{c_0, c_1, ..., c_n\}$, where c_i is the *i*-th sample point.

The sampling algorithm starts by building a priority queue of seed points selected from the corrected dataset, as well as creating a Voronoi structure over said set which will be used for efficiently implementing intersections. The points with highest priority are those that exhibit the highest confidence during the M-estimation process, i.e. those p_j with highest $\sum_i \mathbf{w}_i[p_j]$, meaning that they contribute the most to the estimation of curvature of the other points in the dataset.

The first point in a given curve, c_0 , is initialized by popping a point from the queue. From this point, we will start to trace flow lines in each of the principle curvature directions $\vec{d} \in \{\vec{k}_1, \vec{k}_2, -\vec{k}_1, -\vec{k}_2\}$.

Each new sampling point is generated firstly as $c_i \leftarrow c_{i-1} + s * \vec{d}$, thus moving towards the current signed principal direction. As we will see, the step size s will be

adaptively chosen and is initialized to half the distance of c_0 to its nearest neighboring point. Of course, c_i currently lies on the tangent plane of c_{i-1} and possibly not on the underlying surface. Therefore, we proceed with a process that corrects this by performing the following steps:

- 1. Retrieve all points neighboring c_i that have non-zero statistical weights for the curvature estimation of c_i (as stored upon completion of the IRLS process). Let us call these points q_1, q_2, \ldots, q_k .
- 2. Define a new point $\hat{q} = \sum w_j q_j$ as their weighted average. As in the curvature estimation process, the weights are a combination of geometric and statistical weights. The statistical weights are the curvature M-estimation weights and the geometric weights are the inverse of the squared Euclidean distance of c_i to each q_j .
- 3. Define the normal of \hat{q} to be the normalized weighted average of the normals of q_j using the same weights. We also define its principal curvatures in the same way. Update \vec{d} and c_i according to this interpolated principal curvature direction.
- 4. Check if the current flow line is crossing into a new dataset Voronoi cell by intersecting one of the separating hyperplanes of the current cell. If so, limit the step size *s* and update c_i (see below as to why.)
- 5. Update the sampling point c_i by projecting it onto the tangent plane defined by \hat{q} and its normal.
- The statistical weights for c_i are set to the weights of the closest point in the dataset. Therefore, find this closest point and update it if necessary.

We continue this process, by repeating steps 1 through 6 for the updated flow point c_i . As noted in step 4, we also keep track of the flow points that belong to each of the Voronoi cells of the dataset and register them accordingly. This will be very important for the tracking of flow line intersections. For each Voronoi site in the dataset, we only need to check for intersections of the corresponding registered flow segments. This is efficient and guarantees no intersections will be lost.

The interpolation we use allows for the efficient tracing of flow lines with satisfactory stability. Alternatively, a global parametrization of the point cloud and a higher-order Runge-Kutta method could be used but this would be prohibitively slow.

Preserving features: Notice that the statistical weights serve to preserve features during the tracing of the flow lines. For example, in the case of a cube (see Figure 3), for points near the cube edges, the weights of the points past the corresponding feature boundaries are zero. Thus, the flow line interpolates correctly along the cube faces and preserves hard edges.

Stopping conditions: Each current flow line stops if one of the following conditions is met: a) if the current flow point has a distance less than $d(\kappa)$ to a point of a different flow line (of the same principal curvature), where

 $d(\kappa) = 2\sqrt{\epsilon(2/|\kappa| - \epsilon)}$. This density threshold is adapted to the corresponding curvature κ of the flow point as in [ACSD*03]. b) If the current flow point is closer than a small multiple of the current step size to the starting point of the line, then there is a self intersection. c) If a flow line reaches an umbilical point (the current flow point has distance less than the step size to an umbilic).

The proximity queries are performed by running a breadth-first search (BFS) based on the six nearest points of each point in the dataset. The BFS stops when there are no more points in the dataset within a distance equal to the density threshold and the current step size. For each retrieved point, we access its Voronoi cell structure and retrieve its registered flow points. Then, we check the above conditions based on the distance of the current flow point to the retrieved flow points of the nearby cells, as given by the BFS. Of course, this is done for efficiency reasons, as a KD-tree query for each flow point would be prohibitively slow, as also noticed in [MK04].

In the umbilical regions, the principal directions are not well defined. These regions are found by gathering all the umbilical points in the dataset. A point is set to be umbilic if the ratio of its principal curvatures is larger than 0.95. If an umbilical region contains more than 3 points (like the points that lie on a face of the cube), we perform principal component analysis and set the principal directions to be the projections of the eigenvectors which correspond to the highest eigenvalues. This amounts to setting the principal directions to the local planar symmetry axes of each patch (see Figure 3 of the cube). If a flow line starting from an umbilical region reaches its boundary, then it stops. If a flow line reaches a point where its normal is perpendicular to the given PCA directions, it also stops (see Figure 3 of the sphere). As PCA can be sensitive to noise, an IRLS process can be followed with an appropriate weighting of the points in order to optimize for the local symmetry axes as done in the framework given in [SKS06].

The result of executing the above process for each point in the priority queue is a network of principal curvature lines (see Figure 3 of the torus) where each Voronoi cell data structure has the registered flow points. We can now track the intersections of the flow lines very efficiently.

4.2. Checking for intersections

For each Voronoi site in the dataset, we search for intersections of the flow segments incident on said site. As the flow segments may not intersect exactly in 3D, we project them onto the tangent plane of the associated point. A sweep-line algorithm [dBvKOS00] is employed to quickly find the intersections. If there is an intersection between two flow segments on the tangent plane, we find their intersecting points and we reproject them. The new intersection (a new vertex) is set to be the midpoint of these reprojected points. We also



8

Figure 8: (*a*),(*c*) Results of applying our method to highly noisy and outlier rich data acquired using scatter-trace photography [*MK07*] (see Figure 4). (*b*),(*d*) The same dataset meshed using the method of Ohtake et al. [*OBS05*].

check if the flow segments of other lines of curvature meet at an existing intersection. In this case, we update the vertex structure with all the meeting flow lines. Moreover, we set the normal of the vertex to be the average of the normals of the flow points of the intersecting segments in order to remain consistent with the original surface orientation.

5. Meshing process

After tracking the new vertices of the intersecting segments, it is easy to proceed with the construction of the half-edge structure. In the vertex structure, we keep indices to the flow points of the intersecting flow lines. Each flow point has pointers to its previous and next flow points in the line. We traverse the flow lines to find the neighboring intersections of each vertex. In this way, we create all the edges between the vertices.

Half-edge structure creation: The meshing process is similar to the one described in [MK04]. For every vertex, we project all its edges onto its tangent plane, as given by its precomputed normal. Having one of the projected edges as a reference, we find the angles of all the other edges to it and we sort them according to this angle in a counter-clockwise direction. This results in the correct cyclic half-edge order. Based on this process, we build all the half-edges for each vertex.

Face generation: We select a half-edge from the list of all retrieved half-edges and traverse the next half-edge until the starting vertex is met. We mark these half-edges as visited and we create a face. Then we continue this process, until all half-edges are visited. Any concave faces can be further partitioned, as done in [MK04].

6. Results

We show the results of our approach on analytic examples with varying noise and sampling quality (see Figures 2 and 3), models with sharp features, large umbilic regions, as well as synthetic and commercially scanned real-world examples (Figures 1, 2, 3, 5 and 7) and even highly noisy, reflective objects (Figure 8), acquired using scatter-trace photography [MK07].

We also show our algorithm to compare favorably against state of the art approaches. Figures 5 and 7 compare our results to Poisson surface reconstruction [KBH06]. In the case of the cow, we used a depth level setting of 8 so that the number of triangular faces produced was the same as our method would produce if it were triangulated. Notice the difference in the meshing quality. In the case of the hand model, we use a setting of 12 which produces approximately twice the number of triangles that our results would have if we were to triangulate them. Still, the level of detail is comparable to our result. Note the difference in mesh quality.

Figure 8 compares our results to those of Ohtake et al.

[OBS05]. Note that in this case, the fish point cloud (see Figure 4), obtained using scatter-trace photography [MK07], is highly noisy and outlier rich. In order to provide a fair comparison, we firstly eliminate outliers and re-estimate normals using our approach before applying the approach of Ohtake et al. We tuned the method so that it produces a triangle mesh with minimum holes and comparable resolution to our result (using parameters $T_{er} = 1e - 6$ and $T_q = 1$; increasing the smoothing parameter T_q increased the number of holes in the resulting mesh.)

Our implementation uses CGAL for its data structures and the QHULL algorithm for the Voronoi cell computation.

Indicative total running times for a 300K point cloud such as the hand model are approximately 1.5 hrs, divided into 45 min. for robust curvature estimation, normal correction and outlier rejection, 15 min for positional denoising and 15-30 (depending on the target resolution) for flow line extraction and meshing on a P4 3.0GHz. Memory requirements are approximately 1GB for 100K points (such as the fish model) or up to 2GB for higher resolution models such as the hand.

7. Conclusion and future work

We presented a method that allows the creation of quaddominant meshes for manifold surfaces of arbitrary genus directly from oriented point clouds. The entire technique is well grounded on a robust statistical estimate of curvature and normals used in the denoising of the point cloud, excluding outliers and smoothly extracting the lines of curvature in a feature-preserving manner.

We acknowledge the increased computational cost and memory requirements of the current implementation of our method. There are many extensions to our work that we are currently exploring, which could further enhance this novel type of surface reconstruction. A statistical technique to automatically improve the sampling density over an arbitrary genus surface, in the lines of the method presented in [JWB*06], could improve the reconstruction quality. A robust statistical detection of boundaries and crest lines from the point clouds could also be helpful. An interesting extension of our work could be to generate isotropic flow lines on the point cloud given a locally smooth harmonic vector field. Finally, our technique could potentially be used for automatic hole filling and repairing of incomplete meshes.

References

- [ABCO*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization* and Computer Graphics 9, 1 (2003), 3–15.
- [ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new voronoi-based surface reconstruction algorithm. In ACM SIGGRAPH (1998), pp. 415–421.
- [ACA07] ALLÈGRE R., CHAINE R., AKKOUCHE S.: A streaming algorithm for surface reconstruction. In *Eurographics Symposium on Geometry Processing* (2007), pp. 79– 88.

Copyright November 2007.

- [ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In ACM Symposium on Solid Modeling and Applications (2001), pp. 249–266.
- [ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. ACM Transactions on Graphics 22, 3 (2003), 485–493.
- [ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Eurographics Symposium on Geometry Processing* (2007), pp. 39–48.
- [BBX95] BAJAJ C. L., BERNARDINI F., XU G.: Automatic reconstruction of surfaces and scalar fields from 3d scans. In ACM SIGGRAPH (1995), pp. 109–118.
- [BKBH07] BOLITHO M., KAZHDAN M., BURNS R., HOPPE H.: Multilevel streaming for out-of-core surface reconstruction. In *Eurographics Symposium on Geometry Processing* (2007), pp. 69–78.
- [BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics* 5, 4 (1999), 349–359.
- [Boi84] BOISSONNAT J.-D.: Geometric structures for three-dimensional shape representation. ACM Transactions on Graphics 3, 4 (1984), 266–286.
- [CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In ACM SIGGRAPH (2001), pp. 67–76.
- [CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In ACM SIGGRAPH (1996), pp. 303–312.
- [D'A00] D'AZEVEDO E. F.: Are bilinear quadrilaterals better than linear triangles? SIAM Journal on Scientific Computing 22, 1 (2000), 198–217.
- [DBG*06a] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. In ACM SIGGRAPH (2006), pp. 1057–1066.
- [DBG*06b] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. ACM Transactions on Graphics 25, 3 (2006), 1057– 1066.
- [dBvKOS00] DE BERG M., VAN KREVELDA M., OVERMARS M., SCHWARZKOPF O.: Computational Geometry: Algorithms and Applications, 2nd ed. Springer-Verlag, 2000.
- [DKG05] DONG S., KIRCHER S., GARLAND M.: Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design* 22, 5 (2005), 392–423.
- [DTB06] DIEBEL J. R., THRUN S., BRÜNIG M.: A bayesian method for probable surface reconstruction and decimation. ACM Transactions on Graphics 25, 1 (2006), 39–59.
- [DTS01] DINH H. Q., TURK G., SLABAUGH G.: Reconstructing surfaces using anisotropic basis functions. In *International Conference on Computer Vision* (2001), pp. 606–613.
- [EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. ACM Transactions on Graphics 13, 1 (1994), 43–72.
- [FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. In ACM SIGGRAPH (2005), pp. 544–552.
- [FP02] FORSYTH D. A., PONCE J.: Computer Vision: A Modern Approach. Prentice Hall, 2002.
- [GSH*07] GAL R., SHAMIR A., HASSNER T., PAULY M., COHEN-OR D.: Surface reconstruction using local shape priors. In *Eurographics Symposium on Geometry Processing* (2007), pp. 253–262.
- [HAW07] HUANG Q.-X., ADAMS B., WAND M.: Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Eurographics Symposium on Geometry Processing* (2007), pp. 213–223.
- [HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In ACM SIGGRAPH (1992), pp. 71– 78.
- [HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDON-ALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In ACM SIGGRAPH (1994), pp. 295–302.
- [HK06] HORNUNG A., KOBBELT L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *Eurographics Symposium on Geometry Processing* (2006), pp. 41–50.
- [HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In ACM SIGGRAPH (2000), pp. 517–526.

- [IJS03] IVRISSIMTZIS I. P., JEONG W.-K., SEIDEL H.-P.: Using growing cell structures for surface reconstruction. In Shape Modeling International (2003), p. 78.
- [JWB*06] JENKE P., WAND M., BOKELOH M., SCHILLING A., STRASSER W.: Bayesian point cloud reconstruction. In *Eurographics* (2006).
- [KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In Eurographics Symposium on Geometry Processing (2006), pp. 61–70.
- [KSNS07] KALOGERAKIS E., SIMARI P., NOWROUZEZAHRAI D., SINGH K.: Robust statistical estimation of curvature on discretized surfaces. In *Eurographics Symposium* on Geometry Processing (2007), pp. 13–22.
- [KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Eurographics Symposium on Geometry Processing* (2004), pp. 11–21.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In ACM SIGGRAPH (1987), pp. 163–169.
- [LCOL06] LIPMAN Y., COHEN-OR D., LEVIN D.: Error bounds and optimal neighborhoods for mls approximation. In *Eurographics Symposium on Geometry Processing* (2006), pp. 71–80.
- [LCOL07] LIPMAN Y., COHEN-OR D., LEVIN D.: Data-dependent mls for faithful surface approximation. In *Eurographics Symposium on Geometry Processing* (2007), pp. 59–67.
- [LCOLTE07] LIPMAN Y., COHEN-OR D., LEVIN D., TAL-EZER H.: Parameterization-free projection for geometry reconstruction. ACM Transactions on Graphics 26, 3 (2007), 22.
- [Lev98] LEVIN D.: The approximation power of moving least-square. Mathematics of Computation 97, 224 (1998), 1517–1531.
- [LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. In ACM SIGGRAPH (2006), pp. 681–689.
- [MAVdF05] MEDEROS B., AMENTA N., VELHO L., DE FIGUEIREDO L. H.: Surface reconstruction from noisy point clouds. In *Eurographics Symposium on Geometry Processing* (2005), p. 53.
- [MK04] MARINOV M., KOBBELT L.: Direct anisotropic quad-dominant remeshing. In Pacific Graphics (2004), pp. 207–216.
- [MK07] MORRIS N. J. W., KUTULAKOS K. N.: Reconstructing the surface of inhomogeneous transparent scenes by scatter trace photography. In *International Conference* on Computer Vision (2007).
- [Noc91] NOCEDAL J.: Theory of algorithms for unconstrained optimization. Acta Numerica 1 (1991), 199–242.
- [OBA*05] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multilevel partition of unity implicits. In ACM SIGGRAPH (2005), p. 173.
- [OBS05] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: An integrating approach to meshing scattered point data. In ACM Symposium on Solid and Physical Modeling (2005), pp. 61–69.
- [RJT*05] REUTER P., JOYOT P., TRUNZLER J., BOUBEKEUR T., SCHLICK C.: Surface reconstruction with enriched reproducing kernel particle approximation. In *IEEE/Eurographics Symposium on Point-Based Graphics* (2005).
- [SAG95] SAWHNEY H. S., AYER S., GORKANI M.: Model-based 2d&3d dominant motion estimation for mosaicing and video representation. In *International Conference* on Computer Vision (1995), pp. 583–590.
- [SG07] SÜSSMUTH J., GREINER G.: Ridge based curve and surface reconstruction. In Eurographics Symposium on Geometry Processing (2007), pp. 243–251.
- [Sim94] SIMPSON R. B.: Anisotropic mesh transformations and optimal error control. Applied Numerical Mathematics: Transactions of IMACS 14, 1–3 (1994), 183–198.
- [SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Eurographics Symposium on Geometry Processing* (2006), pp. 111–119.
- [SSB05] STEINKE F., SCHÖLKOPF B., BLANZ V.: Support vector machines for 3d shape processing. In *Eurographics* (2005), pp. 285–294.
- [TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In ACM SIGGRAPH (1999), pp. 335–342.
- [TT05] TONG M.-W.-S., TANG M.-C.-K.: Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 3 (2005), 434– 449.
- [YHW06] Y. L., H. P., WANG W.: Constrained 3d shape reconstruction using a combination of surface fitting and registration. *Computer-Aided Design* 38, 6 (2006), 572– 583.