Convection-Driven Dynamic Surface Reconstruction

Rémi Allègre Raphaëlle Chaine Samir Akkouche

LIRIS CNRS, Université Claude Bernard Lyon 1, France

{Remi.Allegre, Raphaelle.Chaine, Samir.Akkouche}@liris.cnrs.fr

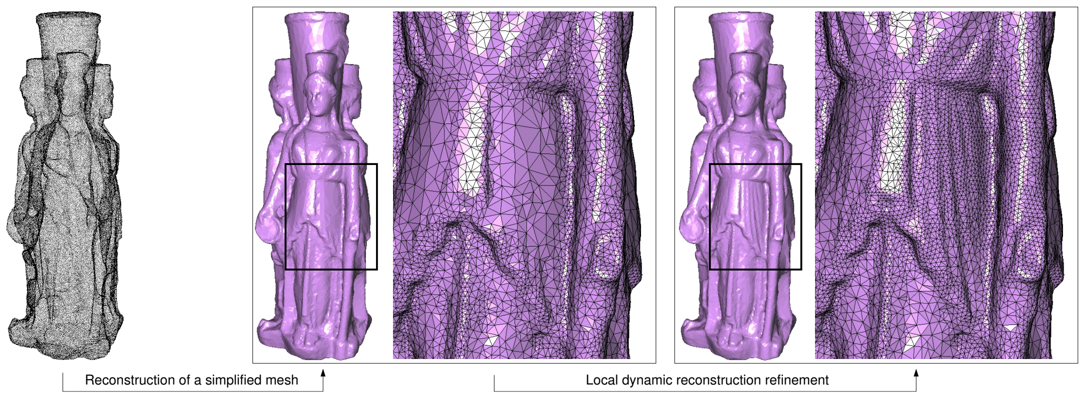

Figure 1. Our reconstruction framework, illustrated on the TRIPLE HECATE model. Starting from a dense input point set, we reconstruct a simplified mesh (center). Benefiting from the connectivity of this initial reconstruction, we can make it to evolve dynamically so as to refine the approximation locally. This refinement can be achieved either in an automatic fashion, for example in order to improve the quality of the elements of the mesh, or interactively, in order to add or remove sample points. Here, the draped dress has been locally enhanced (right).

Abstract

In this paper, we introduce a flexible framework for the reconstruction of a surface from an unorganized point set, extending the geometric convection approach introduced by Chaine [9]. Given a dense input point cloud, we first extract a triangulated surface that interpolates a subset of the initial data. We compute this surface in an output sensitive manner by decimating the input point set on-the-fly during the reconstruction process. Our simplification procedure relies on a simple criterion that locally detects and reduces oversampling. If needed, we then operate in a dynamic fashion for local refinement or further simplification of the reconstructed surface. Our method allows to locally update the reconstructed surface by inserting or removing sample points without restarting the convection process from scratch. This iterative correction process can be controlled interactively by the user or automatized given some specific local sampling constraints.

Keywords: surface reconstruction, point set simplification, geometric convection, dynamic correction.

1. Introduction

Shape modeling from point-sampled geometry has received considerable attention in the past few years, due to the recent advances of 3D digital acquisition and the increasing number of application domains. Today's range scanning devices are able to produce highly detailed digital surface models that can contain millions of sample points. To produce efficient shape representations for interactive visualization or further geometry processing, the complexity of these models has to be reduced.

From a dense input point set, we consider the problem of computing a simplified piecewise linear surface. This goal can be achieved by first reconstructing an initial mesh, and then simplifying this mesh. However, the time and memory costs can be prohibitive if connectivity relations have to be established for all points of the input point set. Another solution consists in first simplifying the input point set, and then reconstructing. The goal of point set simplification algorithms is to extract the relevant data of a dense input point set so as to accelerate a subsequent surface reconstruction process. Most of these algorithms are not designed to perform both point set simplification and surface reconstruction in a single stage, and do not compute their result in an output sensitive manner. The coarse-to-fine surface reconstruction algorithm by Boissonnat and Cazals [5] is a notable exception. One limitation of this work is that the refinement is not achieved in a complete dynamic fashion when a consistent orientation for normals has to be determined. For the purpose of multiresolution shape modeling, when transmitting point samples over networks, or during the 3D acquisition process, updating the reconstructed surface on-the-fly by incorporating or removing points dynamically can be useful.

In this paper, we introduce a dynamic framework for the reconstruction of a simplified triangulated surface from a set of unorganized points sampled from a smooth surface in $\mathbb{R}^3$. We build upon Chaine's geometric convection algorithm [9]. As many surface reconstruction algorithms in the Computational Geometry community [8], this algorithm outputs a triangulated surface embedded in the 3D Delaunay triangulation (or DT for short) of the input point set. Unlike in the original method, we do not require a global 3D DT of the entire input point set. Given a geometric error tolerance, we produce a piecewise linear approximation of the original surface that interpolates only a relevant subset of the input data. We compute this surface in an output sensitive manner by decimating the input point set on-the-fly during the reconstruction process. Our simplification procedure involves a simple criterion that locally detects and reduces oversampling. If desired, the reconstructed surface can be customized in a second stage either to refine the approximation, or to eliminate undesirable features. One key feature of our method is the ability to dynamically insert and remove sample points without restarting the reconstruction process from scratch, taking benefit from the current reconstructed surface. This iterative process can be automatized given some specific local sampling constraints or controlled interactively.

1.1. Background and Related Work

Our work is closely related to surface reconstruction and surface resampling problems. We focus here on point set simplification techniques and review two coarse-to-fine sampling algorithms that output a mesh approximation.

There are mainly two kinds of algorithms for simplifying a dense point set: *subsampling* and *resampling* algorithms. Subsampling algorithms output a decimated point set that is a subset of the original point set. This can be achieved fine-to-coarse by iterative point removal operations. Dey et al. [13] and Funke and Ramos [16] rely on an approximation of the local feature size of the sampled surface. This approximation is computed from the 3D DT of the input point set. Linsen [17] and Alexa et al. [1] estimate local geometric properties using Least Squares techniques. Moenning and Dogson [18] have adopted a coarse-to-fine Farthest Point Sampling strategy based on a distance field representation computed over a regular grid. Wu and Kobbelt [21] compute an optimal set of splats to cover a point sampled surface. For every sample point is computed a circular or elliptical linear surface element called 'splat' that approximates the surface locally. A global optimization process eliminates redundant splats and finds an optimal placement for the remaining ones.

The coarse-to-fine surface reconstruction algorithm by Boissonnat and Cazals [5] starts from a random subset of the input point sample. From the 3D DT of this subset is estimated a signed distance function to the sampled surface based on natural neighbor interpolation. This function is used to enrich the initial set of points till a sufficient number of sample points lie below a prescribed error tolerance. The surface is reconstructed from the 3D DT of the augmented subset. If this surface does not meet the error condition, additional points can be inserted iteratively. The interesting idea is that the 3D DT is computed in an output-sensitive manner. However, priority queue updates are required to maintain the implicit representation and oriented normals, which does not make the final result easily customizable.

Resampling algorithms rely on a global or local estimated representation of the true surface to compute new, well chosen point locations. This representation is generally computed from the input point set. The fine-to-coarse technique by Dey et al. [14] makes use of the local feature size estimated from the 3D DT. Pauly et al. [20] have proposed several algorithms inspired by mesh simplification schemes. Moenning and Dogson [19] have extended their subsampling technique to resampling. Wu and Kobbelt [21] use a relaxation scheme to compute an optimal placement for a set of splats.

Boissonnat and Oudot [6, 7] have recently revisited Chew's Farthest Point sampling technique [11] to generate optimal feature size-dependent triangulations for a fixed implicit or polyhedral surface. Chew's algorithm maintains the restricted DT of a point sample generated incrementally on a smooth surface. The restricted DT of a point sample is the set of facets of its 3D DT such that their dual Voronoï edge intersects the surface. After convergence, the sampling properties of Chew's algorithm make this set of facets a good approximation of the sampled surface, with both geometric and topological guarantees. Following this approach, Boissonnat and Oudot [6, 7] reconstruct a surface from a point set using a Moving Least Squares implicit surface representation. Their method requires to compute the local feature size for some points on the surface, which involves expensive computations. Cheng et al.'s [10] algorithm proceeds in the same way. They do not rely on the local feature size, but on a set of critical points of a Morse function on the surface, which is also a global information whose computation is a difficult task.

1.2. Overview

Let $P = \{\mathbf{p}_i\}$ be a set of sample points that lie on or near a smooth surface S embedded in $\mathbb{R}^3$. We suppose that P is sufficiently dense in the sense this point set forms a ε-sample of S for some constant $\varepsilon > 0$ [2]. This point set can be locally oversampled w.r.t. the local feature size, i.e. the shortest distance to the medial axis. In our framework, we need an (unoriented) normal direction at every sample point $\mathbf{p} \in P$. If normals are not supplied as part of the input data, we estimate the normal direction at a point $\mathbf{p}$ by fitting a least squares plane to $\mathbf{p}$ and its k nearest neighbors in a preprocessing step. For a reliable estimation, a locally uniform sampling distribution is required [3]. We recall the geometric convection algorithm in Section 2. We proceed in two stages, as illustrated in Figure 1.

• In the first stage (Section 3), we compute a linear approximation of S that interpolates a subset P' of P w.r.t. a geometric error tolerance $\rho > 0$ prescribed by the user. Each time a new point is inserted in the reconstructed surface, we decimate the input point set in a small neighborhood around. We achieve this simplification in a feature-sensitive manner thanks to a normal-based error metric. The result is a consistent triangulated surface embedded in the 3D DT of P'. For this reconstruction, we do not require to compute the 3D DT of P explicitly. We only need to compute the 3D DT of P' in prevision of the second stage.

• In the second stage (Section 4), corrections can be applied dynamically to the reconstructed surface. We propose a refinement algorithm to improve the quality of the triangles and give the possibility to the user to customize the result by adding or removing details. For this purpose, we introduce an algorithm to locally update the reconstructed surface by inserting or removing sample points. We store the history of the reconstruction process by maintaining the 3D DT of the set of points that actually belongs to the reconstructed surface.

In Section 5, we present some experimental results. We conclude and discuss future work in Section 6.

2. The geometric convection algorithm

The geometric convection algorithm [9] is based on the convection model introduced by Zhao, Osher and Fedkiw [22]. From a point set P sampled from a surface S, the latter solve the reconstruction problem by computing a closed surface that minimizes a global distance function to the input point set. A convection scheme is used to compute an initial approximation of S. This approximation is obtained by shrinking a surface S' that encloses P. At each step, every point $\mathbf{x}$ of S' evolves along the normal direction $\mathrm{n}(\mathbf{x})$ of S' at point $\mathbf{x}$, with displacement speed proportional to $-\nabla d(\mathbf{x}) \cdot \mathrm{n}(\mathbf{x})$ where $d(\mathbf{x})$ is the distance between $\mathbf{x}$ and its closest point in P.

Zhao et al. [22] compute the convection result on a regular grid with a so-called fast tagging algorithm. Chaine [9] translates the convection scheme into Computational Geometry terms, yielding an efficient, purely data-dependent surface reconstruction algorithm. This work is built on the following theorem.

Theorem (proved in Chaine [9]) *Given a closed surface S' enclosing a point set P, the convection of S' through the velocity field $-\nabla d(\mathbf{x})$ converges to a closed, piecewise linear* pseudo-*surface. All the facets of this pseudo-surface are Delaunay facets oriented consistently towards the interior of the shape that meet an* oriented Gabriel property.

An oriented Delaunay facet f is said to meet the *oriented Gabriel property* if the half of its minimum enclosing sphere located on the positive side of f does not contain any point of P in its interior. The term *pseudo*-surface means that different parts of this surface can be pinched together, i.e. can locally share common geometric information, while remaining topologically independent. More formally, a piecewise-linear pseudo-surface can be defined as the geometric embedding of an orientable manifold polyhedral complex such that the geometric images of two vertices, edges of facets are either identical or disjoint. This structure can represent the evolving surface all along the reconstruction process. It supports topological changes and allows to maintain a consistent mesh at every step of the convection scheme.

Algorithm From the above theorem is derived an algorithm that extracts the reconstructed surface S' from the 3D DT of P. The idea is to shrink an initial piecewise-linear pseudo-surface (or just pseudo-surface for short) through this triangulation, which is equivalent to make it convect through the velocity field $-\nabla d(\mathbf{x})$. Figure 2 illustrates the reconstruction process on a 2D point set.

The pseudo-surface is initialized with the boundary of the convex hull of P, all facets oriented inwards. If an oriented facet does not meet the oriented Gabriel property, it is removed from the pseudo-surface to be replaced by three other 'hidden', consistently oriented facets of the Delaunay tetrahedron it belongs to. An oriented facet can be opened towards a point location that is already attached to a vertex of the current pseudo-surface. Two oriented facets of the

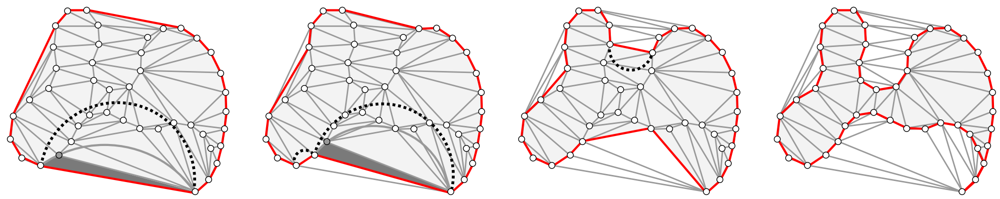

Figure 2. Convection towards a 2D point set. In (a), the pseudo-curve is initialized to the convex hull of the point set. The current edge, enclosed by its (dashed) nonempty half-circle, forms a Delaunay triangle (dark gray) with the dark gray point. This triangle becomes external, and the pseudo-curve is updated consistently in (b), where two other Delaunay triangles have been opened. The pseudo-curve evolves as long as the oriented Gabriel property is not met for any of its oriented edges. In (c), an edge is found to block a cavity. The result in (d) contains coupled oriented edges (the 'tail' at the bottom right of the shape).

pseudo-surface with identical geometry are said to be *coupled* – they necessarily have opposite orientations. When one of two coupled oriented facets of the pseudo-surface does not meet the oriented Gabriel criterion, both are removed and are said to *collapse*, which can involve topological changes of the pseudo-surface. All facets are processed separately, following a breadth-first traversal.

The convection stops as soon as every oriented facet meets the oriented Gabriel property. In presence of concavities larger than the Gabriel half-spheres raised from the pseudo-surface, the basic convection process stops prematurely. If the point sample correctly reflects the local feature size, the pseudo-surface is further shrunk through these *pockets* by detecting inconsistencies between the size of blocking facets and local density. A more global solution for this problem can be derived from a topological persistence criterion [15, 8].

The resulting pseudo-surface is a combinatorial manifold, that can contain coupled oriented facets called *thin parts*. These thin parts are not always significant, so it is important to identify desirable ones. This is achieved by pursuing the convection process in 2D on these thin parts, starting from their boundary.

Complexity The complexity of this algorithm is dominated by the time to compute the 3D DT of the input point set. It can be computed incrementally in time $O(N \log N)$, where N is the size of the input point set [12]. For a locally uniform ε-sample, the number of tetrahedra is almost linear w.r.t. N [4]. Remaining operations correspond to a partial breadth-first traversal of the 3D DT.

The basic convection process does not require any *global* Delaunay-related information such as the poles [2], that can be needed to obtain an estimation of the local feature size for example. Only the pocket detection could benefit from such a global information, but the proposed local solution gives satisfactory results in practice. The 3D DT allows to directly identify the points towards which oriented facets open during the convection process. For redundant input point sets, time and memory are wasted unnecessarily, since not all points are relevant. For our purpose of reconstructing a simplified triangulated surface, we investigate a way to avoid the computation of the 3D DT of the entire point set.

3. Reconstruction and simplification

The first stage of our framework consists in constructing a mesh from the input point cloud while simplifying it. To fit our simplification purpose, we couple the geometric reconstruction algorithm with a subsampling procedure. We take benefit of the locality property of the convection process to avoid the computation of the 3D DT of P. Next, we describe our decimation scheme. Then, we present the complete reconstruction algorithm and take a look at some properties of the reconstructed surface.

3.1. Decimation scheme

Let P_{ev} denote the set of points interpolated by the evolving pseudo-surface at a given time. If a new sample point $\mathbf{p}$ is inserted into the evolving pseudo-surface, we remove some redundant points of $P - P_{ev}$ located in its surface neighborhood following an isotropic region growing strategy. We grow the neighboring region around $\mathbf{p}$ by adding its nearest neighbors in the order of their distance to $\mathbf{p}$. The idea is to make $\mathbf{p}$ a good representative of the geometric information hold by the sample points to be removed in its neighborhood, given an error tolerance $\rho \leq 1$. A *good representative* means that the set of facets that will be incident to this point will have to correctly approximate the surface both geometrically and topologically, at global scale. We thus define two kinds of sub-sampling criteria to control the decimation process: one *geometric approximation* criterion C_{geom}, and one *topological disk* criterion C_{topo}. A point that fulfills both criteria C_{geom} and C_{topo} is eliminated, and the growing stops as soon as one of these conditions is not met. The distance between $\mathbf{p}$ and the farthest point that fulfills both C_{geom} and C_{topo} will be called *decimation radius* of $\mathbf{p}$.

Our geometric criterion (Fig. 3) relies on a normal-based error metric. A point $\mathbf{p}_i$ fulfills the geometric approximation criterion if and only if the following condition is met:

$$(C_{geom}) \qquad |\mathrm{n}(\mathbf{p}_i) \cdot \mathrm{n}(\mathbf{p})| > \rho$$

where $0 < \rho \leq 1$.

The topological criterion must guarantee that the ball $\mathcal{B}(\mathbf{p})$ bounding the decimation region around $\mathbf{p}$ contains only sample points that belong to the local surface neighborhood of $\mathbf{p}$ on S. This criterion is designed to prevent oriented facets from having an half-sphere that 'encroaches' another part of the surface, which would result in a topologically incorrect reconstruction. As illustrated in Figure 4, we derive an intrinsic criterion. A point $\mathbf{p}_i$ fulfills the topological criterion if and only if the following condition is met:

$$(C_{topo}) \qquad |\mathrm{n}(\mathbf{p}_i) \cdot \frac{\mathbf{p} - \mathbf{p}_i}{\|\mathbf{p} - \mathbf{p}_i\|}| < \rho'$$

where $0 < \rho' \leq 1$ is a value that depends on the sampling conditions. If $P = S$, the idea is that if $\mathcal{B}(\mathbf{p})$ touches an other part of S at a sample point $\mathbf{p}_i$, $\mathcal{B}(\mathbf{p})$ becomes tangent to S at $\mathbf{p}_i$ so that $\mathrm{n}(\mathbf{p}_i)$ and $\mathbf{pp}_i$ are collinear. The transposition to the discrete setting is straightforward, provided P reflects the local feature size. Depending on the input type, theoretical bounds for ρ' could be derived. In practice, we found that $\rho' = 0.95$ generally gives satisfactory results. Figure 5 illustrates the effect of this criterion on the screwdriver model.

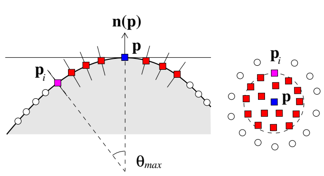

Figure 3. Decimation of the point set in the neighborhood of $\mathbf{p}$. A point $\mathbf{p}_i$ with a normal that makes an angle less than θ_{max} with $\mathrm{n}(\mathbf{p})$ satifies the condition C_{geom}.

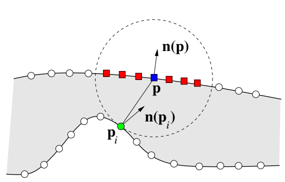

Figure 4. Illustration of the topological disk criterion. Here, $\mathbf{p}_i$ is the first point that does not belong to the topological disk constructed from $\mathbf{p}$. The angle between the normal $\mathrm{n}(\mathbf{p}_i)$ and the edge $\mathbf{pp}_i$ should be small if P is sufficiently dense.

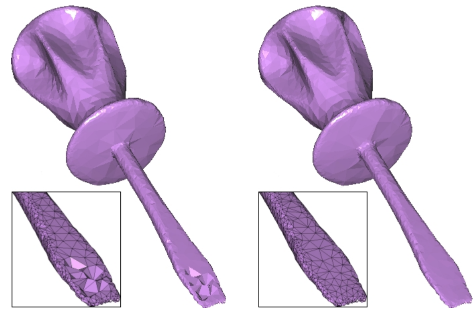

Figure 5. Reconstruction of the SCREWDRIVER model with simplification ($k = 6$, $\rho = 0.98$). The reconstruction on the left does not uses our topological criterion. On the right, the topological criterion is enabled.

3.2. Reconstruction

We extend the original convection algorithm by introducing the previously described decimation scheme to produce a simplified reconstruction. Our algorithm does not compute the 3D DT of the input point set explicitly. We shrink a pseudo-surface mesh around P by computing required Delaunay tetrahedra on-the-fly, while eliminating irrelevant sample points. We will denote as R the set of points eliminated during the reconstruction process, i.e. $R = P - P'$. In the two following paragraphs, we present our data-structures and detail our algorithm.

Data-structures We represent the evolving surface by a pseudo-surface mesh data-structure [9] that is not supported by an explicit 3D DT of the input point set. We delegate spatial searching to a kd-tree data-structure that stores the input point set P. We require this data-structure to perform point locations efficiently.

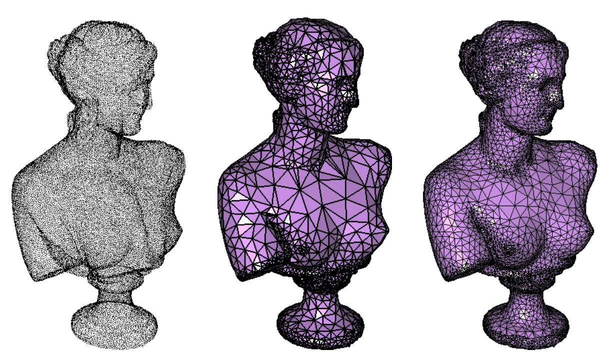

Figure 7. Two reconstructions of the APHRODITE model (left, 46K points) with different values of ρ. With $\rho = 0.9$, the surface is approximated by 3.5K points (center). With $\rho = 0.98$, the surface is approximated by 10.8K points (right).

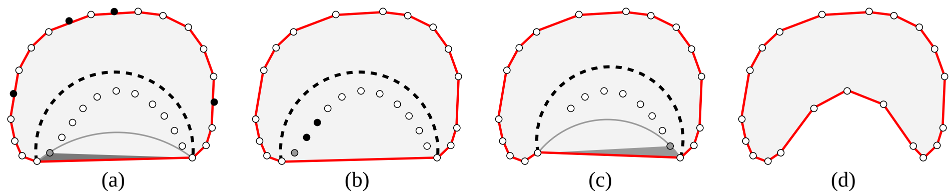

Figure 6. Convection with simplification towards a 2D point set. In (a), the pseudo-curve is first initialized to the convex hull of the point set. The black points have been eliminated while computing the convex hull. The current edge forms a Delaunay triangle with the dark gray point. In (b), the two black points in the neighborhood of the selected point are found to lie below the prescribed error tolerance. These points are removed and the pseudo-curve is updated in (c). The pseudo-curve evolves as long as the oriented Gabriel property is not met for any of its oriented edges. The final result is shown in (d).

Algorithm We first initialize the pseudo-surface S_{ev} by computing an approximate convex hull of P using the Quick Hull algorithm. Each time a new sample point is added to the convex hull, we call our decimation procedure on this sample point. We then start from this mesh for the convection process.

Let **pqr** be an oriented facet of the pseudo-surface. To decide whether this facet should be opened or not towards a point of P, it suffices to check whether its meets the oriented Gabriel criterion. We first report all points of P located in the Gabriel half-sphere $\mathcal{H}$ of **pqr**.

• If the criterion is not verified, we check whether the two coupled oriented facets **pqr** and **qpr** both belong to S_{ev}. If it is the case, the two coupled oriented facets collapse and the connectivity is restored between their neighboring facets [9]. Otherwise, we have to find the point $\mathbf{s} \in \mathcal{H} \cap (P - R)$ such that **pqrs** forms a Delaunay tetrahedron. This point is the one that maximizes the radius of the circumsphere $\mathcal{S}$ of **pqrs**. The sphere $\mathcal{S}$ is a medial sphere that satisfies the Delaunay empty ball criterion. We call the decimation procedure on **s** if **s** does not belong to the pseudo-surface. Then, we just replace **pqr** by the oriented triangles **pqs**, **qrs** and **rps**, which can be achieved by a simple vertex insertion.

• If **pqr** satisfies the oriented Gabriel property, we rely on normals to detect and pursue the convection process through pockets. Let n_f denote the unit normal to the oriented facet f. If the greatest value between $|\mathrm{n}_f \cdot \mathrm{n}(\mathbf{p})|$, $|\mathrm{n}_f \cdot \mathrm{n}(\mathbf{q})|$, and $|\mathrm{n}_f \cdot \mathrm{n}(\mathbf{r})|$ is less than 0.5, we consider that f blocks a pocket. In this case, we search for the point **s** such that the tetrahedron **pqrs** forms a Delaunay tetrahedron. We call the decimation procedure on **s** if **s** does not already belong to the pseudo-surface, and then insert **s** into **pqr** as previously.

The convection stops as soon as every oriented facet of the pseudo-surface meets the oriented Gabriel property and does not pass the pocket detection test. Our reconstruction algorithm is schematized in 2D in Figure 6. Figure 7 illustrates our technique on the APHRODITE point set with different values of the tolerance parameter ρ.

3.3. Sampling density and regularity of the mesh

The sampling density and regularity of the resulting mesh are controlled both by the radius of the decimation region around sample points and by the order in which the convection processes them. We provide here some elements of explanation, but not a complete study, on which we are currently working.

Our simplification procedure is closely related to the geometry of the whole surface, but not directly guided by the local feature size [2]. The radius of decimation is another geometric measure that quantifies the local thickness of the surface. We make a parallel with the local feature size on a simple 2D example, in the continuous setting (Fig. 8). We distinguish two different portions of a parabola. In (a), the light portion identifies the set of points such that the nearest point on the medial axis is the center of the osculating circle at the origin O, where the local curvature is maximized. For every other point on the parabola, the nearest point on the medial axis is the center of a bitangent ball (for example, **p**). In (b), the light portion of the curve is the set of points such that every ball centered on this portion is never tangent to the parabola. The growth of a decimation ball around a sample point on this portion (**p**) would be stopped by the C_{geom} criterion, that only depends on the local surface variations. The radius of a decimation ball built around a sample point on the dark portion (**q**) would be determined by C_{topo} for a sufficiently permissive criterion C_{geom}, i.e. by the distance to the opposite side of the surface.

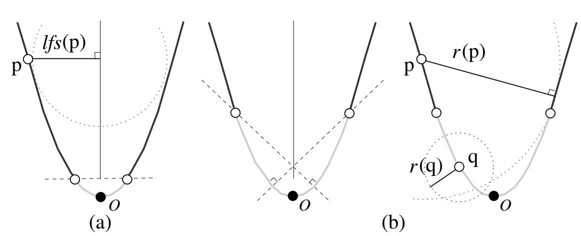

Figure 8. Local feature size (a) vs. radius of decimation (b) on a parabola.

The sampling distribution can be explained by the order in which sample points are incorporated into the evolving pseudo-surface. The neighborhood graph on the pseudo-surface results from a breadth-first propagation around existing vertices that is induced by the geometric convection process. An oriented facet that does not match the oriented Gabriel property tends to open towards a point that forms a neighborhood edge with one of its 3 vertices, except in the case where the entered Delaunay cell corresponds to a branching of the medial axis. In the latter case, a new 'seed' vertex is created. The neighborhood graph is thus grown from several seeds at a time, and advancing fronts merge at the bottom of concavities.

This way of processing yields regular connectivity in regions that exhibit no important change in curvature, i.e. every sample point has exactly 6 neighbors on the surface (Fig. 10(a)). In smooth regions, it can also be observed that retained points are distributed uniformly around a given one. However, strong curvature variations induced, for example, by the presence of sharp features such as edges or corners, produce long and skinny triangles coupled with high valence vertices, which can be inadequate for further mesh processing (Fig. 10(b)). When growing the decimation region around a sample point, we have no cheap means to anticipate on these variations. Sharp features could be detected earlier by precomputing the radius of the decimation region for every input sample point without simplifying, but we prefer to compute this radius only for retained sample points. We improve the quality of the mesh that results from the reconstruction process in a second stage, that we develop in the following Section.

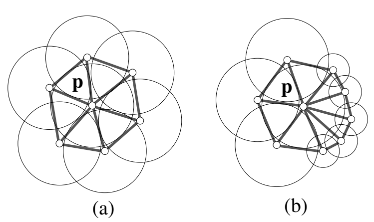

Figure 9. Optimal (a) vs. non-optimal (b) neighborhood configuration. Every sample point is represented with its circular decimation region.

4. Dynamic correction

The purpose of the correction stage is to refine the initial reconstructed surface by inserting or removing sample points. This functionality first serves the purpose of improving the quality of bad shaped triangles. Second, we want to provide the user with means of locally changing the level of detail of the reconstruction. Two kinds of questions naturally arise: Which points are good candidates for further

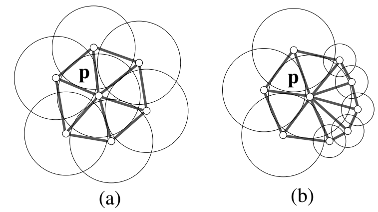

Figure 10. Optimal (a) vs. non-optimal (b) neighborhood configuration. Every sample point is represented with its circular decimation region.

insertions or deletions? How to update the pseudo-surface in a dynamic fashion after these points have been chosen?

For triangle-quality improvement, we devise a simple greedy refinement algorithm inspired from Chew's algorithm [11], recently revisited by Boissonnat and Oudot [6]. Unlike in the latter method, we do not resample a smooth approximation of the surface, but simply reinsert some sample points that have been eliminated in the first stage till a smooth density gradient is achieved on the whole mesh and an aspect ratio criterion is met for every triangle of the reconstructed surface. For user-controlled detail insertion or suppression, we propose to restart the convection process in a prescribed region with a value of the parameter ρ modulated according to a potential field function.

To achieve sample point insertions or deletions in the reconstructed surface, we have designed an efficient algorithm that 'reinflates' the pseudo-surface in an altered region and restarts the convection process only locally. This algorithm requires the computation of the 3D DT of P' during the first stage, and supports either one or several insertion or deletion operations at a time.

In the following paragraphs, we first describe our algorithm for local update of the pseudo-surface. Then, we present our methods for improving the quality of the triangles and for interactive refinement.

4.1. Local update of the pseudo-surface

Let us recall that a pseudo-surface S' that interpolates a subset P' of the original input point set P is embedded in the 3D DT of P'. At the end of the convection process, every oriented facet of S' meets the oriented Gabriel property w.r.t. P'. When a point **p** is inserted (resp. removed) in the 3D DT, the latter is modified only *locally* in a connected region spanned by the set of cells in conflict with **p** (resp. incident to **p**). We exploit this locality property to update the convection result without restarting the convection process from the convex hull of the point set. Figure 11 gives a simple example on a 2D point set, where one new sample point is inserted.

At the beginning of the initial convection process, the

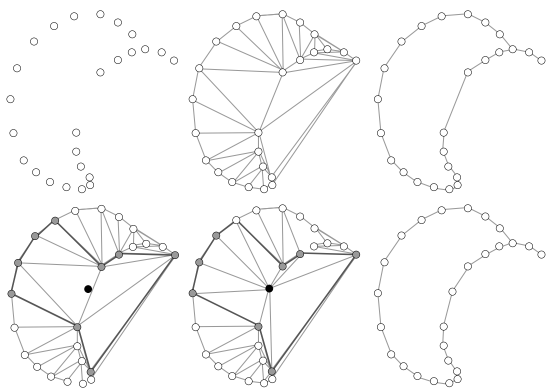

Figure 11. Local update of a pseudo-surface in 2D. The top row illustrates the initial reconstruction process. In the bottom trow, a new (black) sample point is inserted. The conflict region is bounded by the dark contour. The DT is updated and the final pseudo-surface is shown bottom-right.

pseudo-surface lies on the convex hull of the points of P'. All the Delaunay cells of P' are *internal* except the infinite cells [1] that are *external*. During the convection process, the pseudo-surface evolves and the cells it goes through become external. An external cell C_2 is said to have been *discovered* from a cell C_1 if C_2 becomes external when the facet incident to C_1 and C_2, oriented towards C_2, is opened by the convection process. A cell C_2 can be discovered only *once*. In the case a facet oriented from a cell C_3 towards a cell C_2 is pushed towards a cell C_2 that has already been discovered, it means that the pseudo-surface locally has coupled facets between C_2 and C_3 that collapse. The cell C_2 is said to have been *rediscovered* from the cell C_3.

The graph that represents the discovering relation between cells is a forest of rooted trees $\mathcal{D} = \cup \mathcal{D}_i$ where each tree $\mathcal{D}_i$ is rooted on an infinite cell. A given forest $\mathcal{D}$ is not unique in the sense there may exists several equivalent configuration of the discovering relation depending on the order in which Delaunay tetrahedra are opened. Given a cell C_2 discovered from a cell C_1 and rediscovered from a cell C_3, the discovering and the rediscovering relations can be switched if and only if C_2 is not an ascendant of C_3 (Fig. 12).

When points are inserted into or removed from the 3D DT, some cells disappear, that we will call cells *in conflict* in both cases. These are replaced by *new* cells resulting from the local retriangulation. This alters the integrity of the discovering relation. The external cells that do not disappear and that were previously discovered from a conflict cell become roots of the discovering relation, though they are not infinite cells. These cells are called *orphan* cells. To restart the convection process, we reinflate the pseudo-

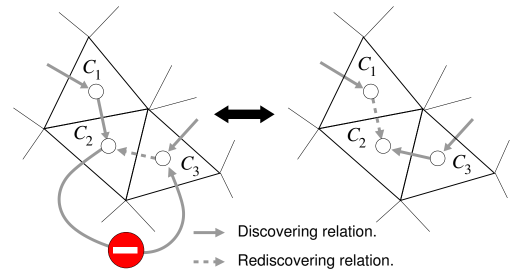

Figure 12. Equivalence between two cell configurations in 2D. The discovering and rediscovering relations can be switched provided there is no cycle creation.

surface so that it encloses the conflict region (Fig. 13,(a)). The newly created cells then become internal. The pseudo-surface is composed of facets oriented towards the interior of the surface, that are of three types:

• The *restart facets*, that are oriented towards new internal cells and that previously encoded discovering or rediscovering relations.

• The *reversable facets*, that are oriented from new internal cells towards orphan cells.

• The other facets.

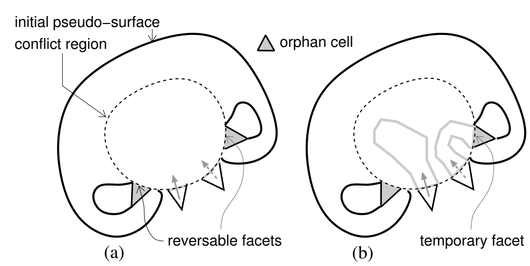

Figure 13. Convection through a conflict region in 2D. The conflict region is bounded by the dashed curve. In (a), the convection starts from two entry points, corresponding to discovering and rediscovering relations. Gray cells are orphan cells. In (b), the convection through the conflict region has stopped. One temporary facet appears.

The update process consists in coherently restoring the discovering relation regarding the new cell configuration induced by the updated new point set. This process can be described through the following three steps:

1. We first launch the convection process from the restart facets and restrict it to the conflict region. The pseudo-surface stops temporarily at the boundary of this conflict region, on standby of an opportunity to re-establish a discovering or rediscovering relation towards the outside. These facets are called *temporary facets* (Fig. 13(b)).

2. This step deals with orphan cells. We have to restore the connectivity between cells so that these orphan cells get discovered, while maintaining a forest. Care must be taken

[1] These cells are artifically created so that the facets of the convex hull are incident to two cells.

to avoid the creation of cycles while re-establishing the discovering relation. If that is not possible for a given orphan cell, this cell cannot be external with the new point set configuration and the pseudo-surface will be reversed locally.

Let C denote an orphan cell. Suppose there exists a cell C_1 adjacent to C, outside the evolving pseudo-surface and such that C could be opened from C_1. If C is not an ascendant of C_1 for the discovering relation, we create a discovering relation between C_1 and C. Now that C has been discovered, the facets that separate C from the interior of the pseudo-surface could be discovered by the convection process. These facets are pushed into the temporary set.

If one of the candidate cell to be C_1 is a new cell (i.e. inside the retriangulated region), then it is chosen rather than the others. If there is no cell candidate to discover C, the cell C becomes internal and the pseudo-surface backtracks consistently. If a cell C_2 was discovered from C, then it becomes an orphan cell (if not infinite) that will be processed in turn. The connectivity of the discovering relation is then restored.

3. The last step consists in launching the convection process from the remaining facets in the temporary set.

4.2. Triangle quality improvement

Let f be an oriented facet of S'. We call r_{min} (resp. r_{max}) the minimum (resp. maximum) decimation radius of the three vertices of f. We define the following two refinement criteria:

- (C_{ar}) f meets the criterion if the ratio between the radius of the circumcircle of f and the length of its shortest edge is less than a positive constant β.
- (C_{den}) f meets the criterion if $\frac{r_{min}}{r_{max}} \geq \gamma$, where γ is a positive constant.

Criterion C_{ar} aims at guaranteeing a minimal aspect ratio for every triangle in S'. Criterion C_{den} prevents big facets from being incident to small facets and so, ensures a smooth gradient of density over the whole mesh.

The idea of the refinement algorithm is to 'break' every triangle of S' that does not meet one of the above conditions. In Chew's algorithm, an interesting idea for producing well-shaped triangles is to create a new sample point equidistant from the three vertices of a facet with bad aspect ratio. In our framework we extend this idea considering the reconstructed pseudo-surface S' and the remaining set of points R. Let f denote a facet that does not pass C_{ar} or C_{den}, and let $\mathcal{B}_f$ be its minimal enclosing sphere. For every such facet, our algorithm proceeds by inserting in S' the point $\mathbf{s}$ of $\mathcal{B}_f \cap R$ that is the nearest from the intersection point between S' and the line that supports the dual Voronoï edge of f. The algorithm stops either when both conditions C_{ar} and C_{den} are met or when $\mathcal{B}_f \cap R$ is empty. Our current implementation does not take boundaries of thin parts into account, but it could be easily extended. Figure 14 illustrates one level of refinement for the APHRODITE model.

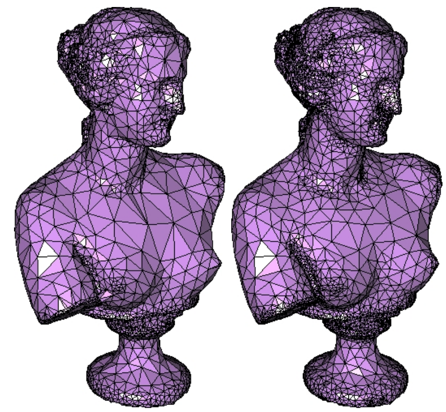

Figure 14. Triangle quality improvement for the APHRODITE model. Starting from an initial reconstruction with $\rho = 0.95$ (left), the refinement was performed with $\beta = 0.8$ and $\gamma = 0.5$ (right).

4.3. Interactive refinement

Our interactive refinement technique consists in rescaling the value of the parameter ρ locally and to update the reconstructed surface according to the new local sampling conditions. By being more or less restrictive locally on the normal variation, we allow to add details or remove some features with an intuitive control. We have implemented a simple brush tool based on a potential field function parameterized by a center $\mathbf{c}$, a radius r and a maximum intensity ρ_c at $\mathbf{c}$. For every point sample $\mathbf{p}$ is computed a local error tolerance $\rho_{loc}(\mathbf{p})$ that depends on the distance between $\mathbf{c}$ and $\mathbf{p}$. We define this local error tolerance as follows:

$$\rho_{loc}(\mathbf{p}) = \rho + \frac{\rho_c - \rho}{r^{2n}} \left(r^n - \|\mathbf{p} - \mathbf{c}\|^n\right)^2$$

where $n \geq 2$ is an integer constant. This function smoothly varies in a monotonically fashion between ρ_c (reached if $\mathbf{p} = \mathbf{c}$), and ρ (reached if $\|\mathbf{p} - \mathbf{c}\| \geq r$). If $\rho_c > \rho$, more points will be inserted. Otherwise, some points will be removed. The C^2 nature of this function ensures a continuous density gradient between the altered region and the remaining of the reconstructed surface.

To refine the reconstruction, we reinflate the pseudo-surface in the region of influence of the tool, and restart the convection process in this region regarding the new local simplification parameters. Figures 1, 15 and 16 illustrate our interactive refinement tool on various point sets.

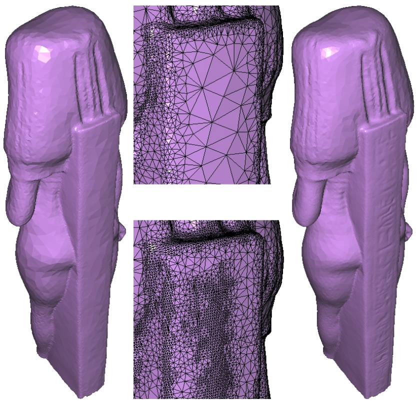

Figure 15. Interactive refinement of the Isis model. The original model exhibits hieroglyphs on the back that are not captured for a too small value of ρ (left, center-top). Our interactive refinement tool allows to make them to appear (center-bottom, right).

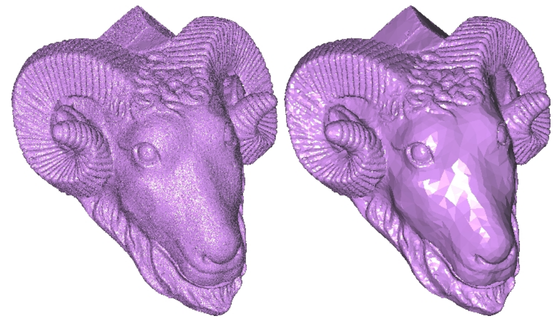

Figure 17. Reconstruction of a noisy point set. In (a) is shown a reconstruction of the original RAM model with 622K points, without simplification. In (b) is shown a simplified reconstruction obtained with our method, with $k = 18$ and $\rho = 0.94$ (43K points). Our normal-based error metric based on a local normal estimation acts as a noise filter.

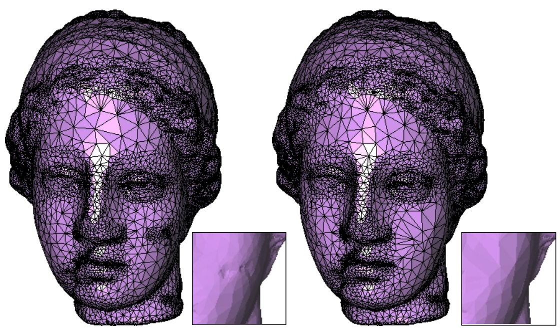

Figure 16. Interactive customization of the IGEA model. The original point set has 134K points. The reconstruction on the left has only 17K points. The ridge on the left cheek was interactively removed to obtain the result on the right.

5. Results and discussion

We have implemented our dynamic reconstruction framework on a Linux platform using the Computational Geometry Algorithm Library, CGAL [2]. We require CGAL's filtered predicates for robust point location and computation of Delaunay tetrahedra.

We demonstrate the effectiveness of our framework on several point set models that were obtained from laser range scanning (Figs. 1, 5, 16, 7), including a particularly noisy point set (Fig. 17). If normal directions are not supplied, the user has to give a value for the size of the k-neighborhood. A value for the geometric error tolerance ρ is required. The reconstruction stage then works automatically. The correction stage can be either skipped or run given user-defined parameters. The triangle-quality improvement procedure requires a maximum tolerated aspect ratio β, and/or a minimum density factor γ. Interactive correction necessitates defining the brush tool properties and areas of interest picked on the reconstructed surface. Table 1 reports the overall timings and final number of points for the initial reconstruction stage and for the correction stage. All the results presented here were obtained on a Pentium IV 3.2GHz, 1GB RAM workstation. Reconstruction timings take into account the incremental generation of the 3D DT of the simplified point set. This generation takes less than 5 seconds for all the point sets we tested.

Our approach for locating points that form a Delaunay tetrahedron in Gabriel half-spheres is not currently optimal, which results in relatively high computation times. Some facets, in particular on the convex hull, may have a Gabriel half-sphere that contains a great part of the input point set. Since the intersection between these half-spheres are not always empty, some point samples can be tested many times before they become part of the surface or they are eliminated. This search could be improved in several ways. We could benefit from the 3D DT of the points inserted in the surface to improve the locality of the point locations or further exploit the normals to guess the position of the next candidate to insertion.

Model		Reconstruction			Correction	
name	#points	ρ	#points	time	#points	time
TRIPLE HECATE	90,180	0.98	28,718	281	34,310	120
SCREWDRIVER	27,152	0.98	7,944	49	–	–
APHRODITE	46,096	0.90	4,507	67	7,644	42
ISIS	187,644	0.95	8,368	96	10,994	38
IGEA	134,344	0.98	17,232	102	17,104	24
RAM	622,716	0.94	43,498	1,472	–	–

Table 1. Performance of our reconstruction framework for various point sets. Computational timings are given in seconds for both initial reconstruction and correction steps.

[2] `http://www.cgal.org`

6. Conclusion and future work

In this paper, we have presented a new framework for reconstructing a surface from an unorganized point set that takes only relevant sample points into account. In a first stage, we construct a triangulated surface that interpolates only a relevant subset of the input data. We decimate the input point set on-the-fly during the reconstruction process. The sampling density is controlled by local geometric and topological constraints. If needed, we then make corrections to the reconstructed surface, which requires the 3D DT of the simplified point set. We improve the quality of the triangles by a refinement algorithm and enable interactive insertion of details or further simplification. These corrections are achieved in a dynamic fashion, without restarting the reconstruction process from scratch, which makes our reconstruction framework very flexible.

Future work will first include the search for a more efficient, dedicated data-structure for point locations. We could then extend our algorithm to automatically produce a multiresolution decomposition of an input point set. This decomposition could be used for progressive reconstruction with our dynamic update procedure. For non-uniformly distributed point sets, it could be interesting to incorporate a relaxation procedure into our algorithm. The question of giving guarantees on the sampling density of the output point set also certainly deserves further investigation.

Acknowledgments The authors would like to thank David Bourguignon for expressing the need for a dynamic approach of geometric convection. The TRIPLE HECATE model is provided courtesy of Christian Lahanier from C2RMF (Musée du Louvre, département des Antiquités Étrusques et Romaines, marbre, MA 2594, C2RMF: FZ31452). The other point set models used in this paper are courtesy of: Cyberware, Polygon Technology, and Fausto Bernardini from IBM.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proc. IEEE Visualization Conference*, pages 21–28, 2001.

[2] N. Amenta and M. Bern. Surface Reconstruction by Voronoi Filtering. In *Proc. Discrete Computational Geometry*, pages 481–504, 1999.

[3] M. Andersson, J. Giesen, M. Pauly, and B. Speckman. Bounds on the k-Neighborhood for Locally Uniform Sampled Surfaces. In *Proc. Symposium on Point-Based Graphics*, pages 167–171, 2004.

[4] D. Attali, J.-D. Boissonnat, and A. Lieutier. Complexity of the Delaunay triangulation of Points on Surfaces: the Smooth Case. In *Proc. ACM Symposium on Computational Geometry*, pages 201–210, 2003.

[5] J.-D. Boissonnat and F. Cazals. Coarse-to-fine surface simplification with geometric guarantees. In *Proc. Eurographics*, pages 490–499, 2001.

[6] J.-D. Boissonnat and S. Oudot. Provably Good Surface Sampling and Approximation. In *Proc. of Symposium on Geometry Processing*, 2003.

[7] J.-D. Boissonnat and S. Oudot. An effective Condition for Sampling Surfaces with Guarantees. In *Proc. Symposium on Solid Modeling and Applications*, pages 101–112, 2004.

[8] F. Cazals and J. Giesen. Delaunay Triangulation based Surface Reconstruction: Ideas and Algorithms. Technical Report 5393, INRIA, November 2004.

[9] R. Chaine. A geometric convection approach of 3-D reconstruction. In *Proc. Symposium on Geometry Processing*, pages 218–229, 2003.

[10] S.-W. Cheng, T. K. Dey, E. Ramos, and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. In *Proc. Symposium on Computational Geometry*, pages 280–289, 2004.

[11] L. P. Chew. Guaranteed-Quality Mesh Generation for Curved Surfaces. In *Proc. of Symposium on Computational Geometry*, pages 274–280, 1993.

[12] O. Devillers. The Delaunay hierarchy. *Internat. J. Found. Comput. Sci.*, (13):163–180, 2002.

[13] T. K. Dey, J. Giesen, and J. Hudson. Decimating samples for mesh simplification. In *Proc. Canadian Conference on Computational Geometry*, pages 85–88, 2001.

[14] T. K. Dey, J. Giesen, and J. Hudson. Sample shuffling for quality hierarchic surface meshing. In *Proc. 10th International Meshing Roundatble Conference*, pages 143–154, 2001.

[15] H. Edelsbrunner. Surface reconstruction by wrapping finite point sets in space. In B. Aronov, S. Basu, J. Pach, and S.-V. M. Sharir, editors, *Ricky Pollack and Eli Goodman Festscrift*, pages 379–404. Springer-Verlag, 2002.

[16] S. Funke and E. A. Ramos. Smooth-Surface Reconstruction in Near-Linear Time. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 781–790, 2002.

[17] L. Linsen. Point cloud representation. Technical Report 2001-3, Universität Karlsruhe, Germany, 2001.

[18] C. Moenning and N. A. Dogson. A new point cloud simplification algorithm. In *Proc. Int. Conf. on Visualization, Imaging and Image Processing*, pages 1027–1033, 2003.

[19] C. Moenning and N. A. Dogson. Intrinsic point cloud simplification. In *Proc. GraphiCon*, 2004.

[20] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient Simplification of Point-Sampled Surfaces. In *Proc. IEEE Visualization Conference*, pages 163–170, 2002.

[21] J. Wu and L. P. Kobbelt. Optimized Sub-Sampling of Point Sets for Surface Splatting. In *Proc. Eurographics*, pages 643–652, 2001.

[22] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast Surface Reconstruction using the Level Set Method. In *Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM)*, pages 194–202, 2001.