

Surface Reconstruction from Point Clouds by Transforming the Medial Scaffold

Ming-Ching Chang^a Frederic Fol Leymarie^b Benjamin B. Kimia^a

^a*LEMS, Engineering, Brown University, USA*

^b*Dept. of Computing, Goldsmiths College, University of London, UK*

Abstract

We propose an algorithm for surface reconstruction from *unorganized points* based on a view of the sampling process as a deformation from the original surface. In the course of this deformation the *Medial Scaffold* (\mathcal{MS}) — a graph representation of the 3D *Medial Axis* (\mathcal{MA}) — of the original surface undergoes abrupt topological changes (transitions) such that the \mathcal{MS} of the unorganized point set is significantly different from that of the original surface. The algorithm seeks a sequence of transformations of the \mathcal{MS} to invert this process. Specifically, some \mathcal{MS} curves (junctions of 3 \mathcal{MA} sheets) correspond to triplets of points on the surface and represent candidates for generating a (Delaunay) triangle to mesh that portion of the surface. We devise a greedy algorithm that iteratively transforms the \mathcal{MS} by “removing” suitable candidate \mathcal{MS} curves (gap transform) from a rank-ordered list sorted by a combination of properties of the \mathcal{MS} curve and its neighborhood context. This approach is general and applicable to surfaces which are: non-closed (with boundaries), non-orientable, non-uniformly sampled, non-manifold (with self-intersections), non-smooth (with sharp features: seams, ridges). In addition, the method is comparable in speed and complexity to current popular Voronoi/Delaunay-based algorithms, and is applicable to very large datasets.

Key words: surface mesh reconstruction; unorganized points; 3D medial axis; medial scaffold; symmetry transforms; non-manifold, non-closed, non-smooth, non-orientable; non-uniform samplings.

1 Introduction

We consider the problem of constructing surface meshes from a sampling in the form of an *unorganized cloud of points*. The recovery of this connectivity amongst points is typically based on assuming (i) some surface continuity, possible smoothness, and (ii) sufficient sampling density to capture all surface features.

Our approach is meant to be general and applicable to various object topologies (Fig.2); in particular it does not assume that the surfaces are smooth, nor closed

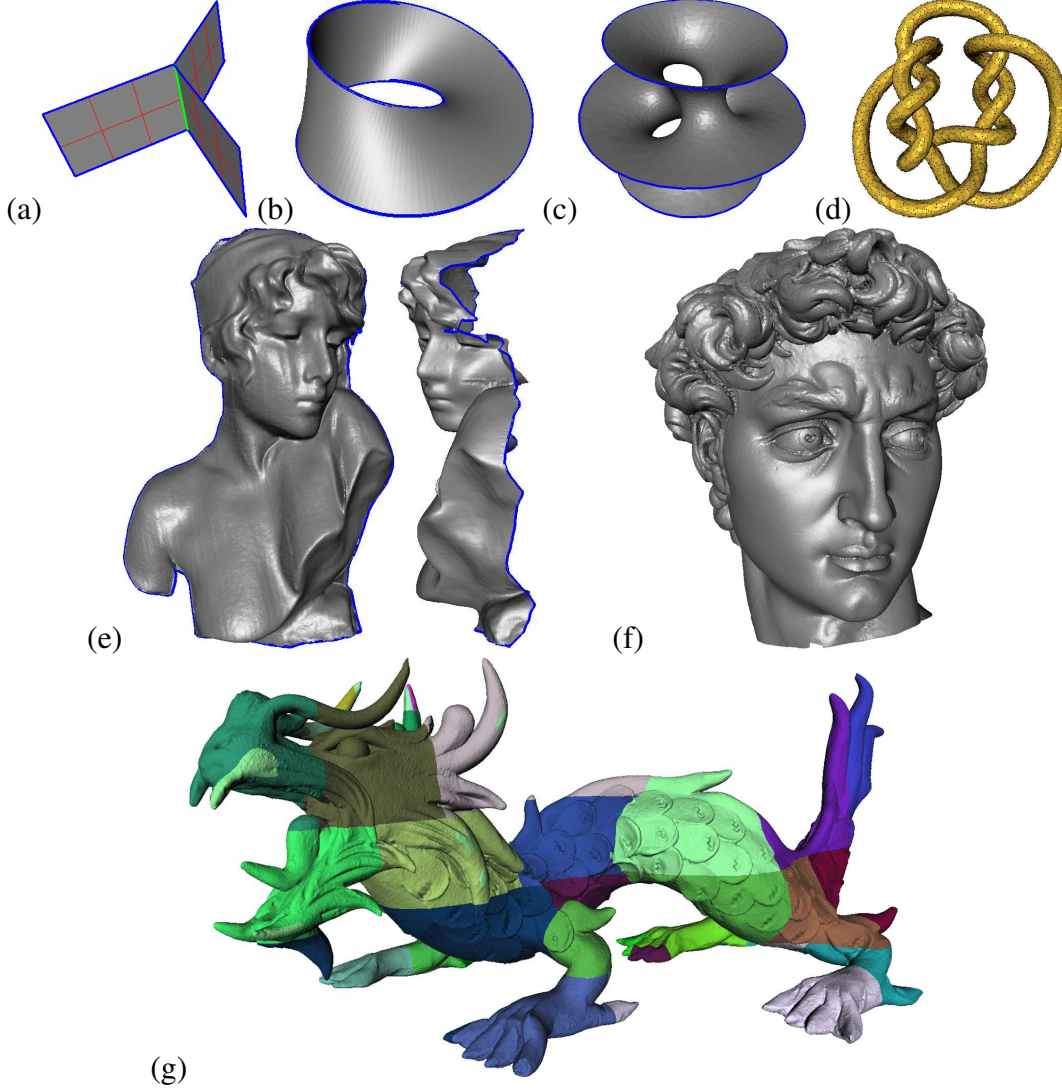


Fig. 1. Surface reconstruction from *unorganized* point clouds using \mathcal{MS} transforms of several topology types: (a) non-manifold surface with self-intersections, (b) Moebius strip (non-orientable), (c) Costa’s minimal surface with the topology of a torus thrice punctured, (d) connected water-tight knot in highly non-uniform sampling of 10K points, (e) a complex surface with recovered boundary (in blue): the Sappho raw scan dataset (121K points, Stony Brook), (f) Michelangelo’s David head (250K points, Stanford), (g) the combination of several reconstruction results (in various colors) from a 3D bucketing and stitching of the Stanford Asian Dragon dataset ($\approx 3.6 \times 10^6$ points).

(enclosing a volume), nor orientable, it can handle *non-manifold* surfaces (a case not considered in most main-stream methods), it does not require uniform sampling, it can handle varying noise levels, and it is scalable to arbitrarily large input datasets (Fig.1), as discussed below.

Our approach is based on the notion of representing shape via deformations and their local successive topological variations. It is directly inspired from previous work by Kimia *et al.* in 2D and 3D [27,40,19,20] who represent shape as a member

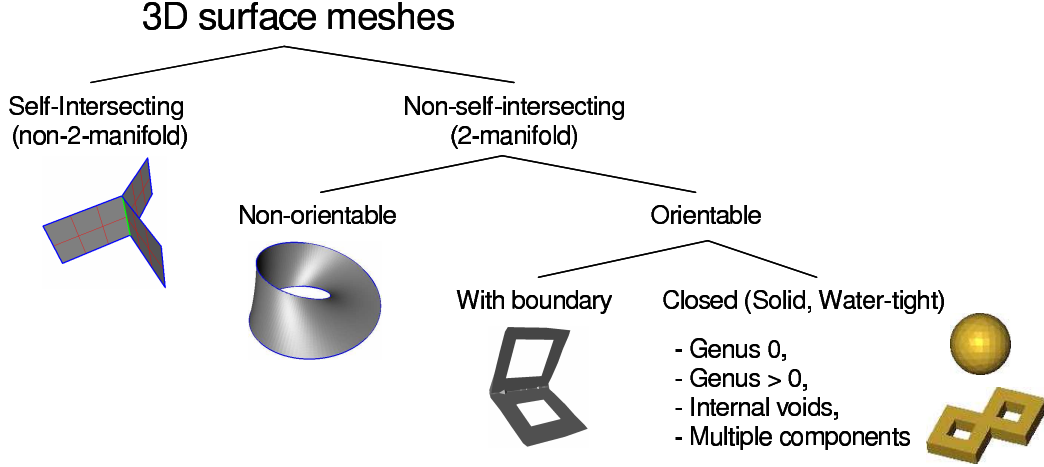


Fig. 2. Classification of surfaces in 3D based on the topology.

of an equivalence class, each defined as the set of shapes sharing a common *shock graph* topology: “shocks” are simply the *Medial Axis* (\mathcal{MA}) points endowed with a sense of dynamics of flow which arises when the \mathcal{MA} is considered as the quench points of a propagating “grass fire” from the contour samples. In 2D the shock graph is a planar directed graph while in 3D the classification of shocks into five types, namely two types of vertices, two types of curves, and one type of sheet, first gives rise to a *hypergraph*, referred to as the *Medial Scaffold* (\mathcal{MS}) which explicitly retains the topology of \mathcal{MA} vertices, curves, and sheets, and second reduced to a directed graph in 3D, which explicitly retains the topology of \mathcal{MA} vertices and curves, while the \mathcal{MA} sheets are implicitly represented as loops in the graph [30]. All 3D shapes sharing the same \mathcal{MS} are considered equivalent.

In addition to defining an equivalence class of shapes, which effectively reduces the dimensionality of the “shape space,” an equivalence class of deformation paths is also needed. Observe that the \mathcal{MS} (equivalently the \mathcal{MA}) undergoes abrupt changes or *transitions* (Fig.3) [20] as the shape is deformed. Two deformation paths sharing the same sequence of transitions are similar and thus are considered to be equivalent. This equivalence class of deformations acting on an equivalence class of shapes effectively *discretizes the shape space*.

Our approach can now be described in the above context: *complete surfaces and sampled surfaces are simply two points in the shape space that are closely related*. In an approach to 2D object recognition [40], the optimal deformation path between two shapes is found by establishing similarity between them, leading to a highly successful object recognition scheme. In an approach to 2D perceptual grouping [43], a greedy approach finds more regular shapes from unorganized data. Similarly, in our case in 3D, a sampled shape is simply a deformation of the original surface, as illustrated in Fig.4. In this process, the \mathcal{MS} undergoes abrupt transitions (topological events). Fig.5 shows the transitions for deforming curves in 2D by removing or adding points.

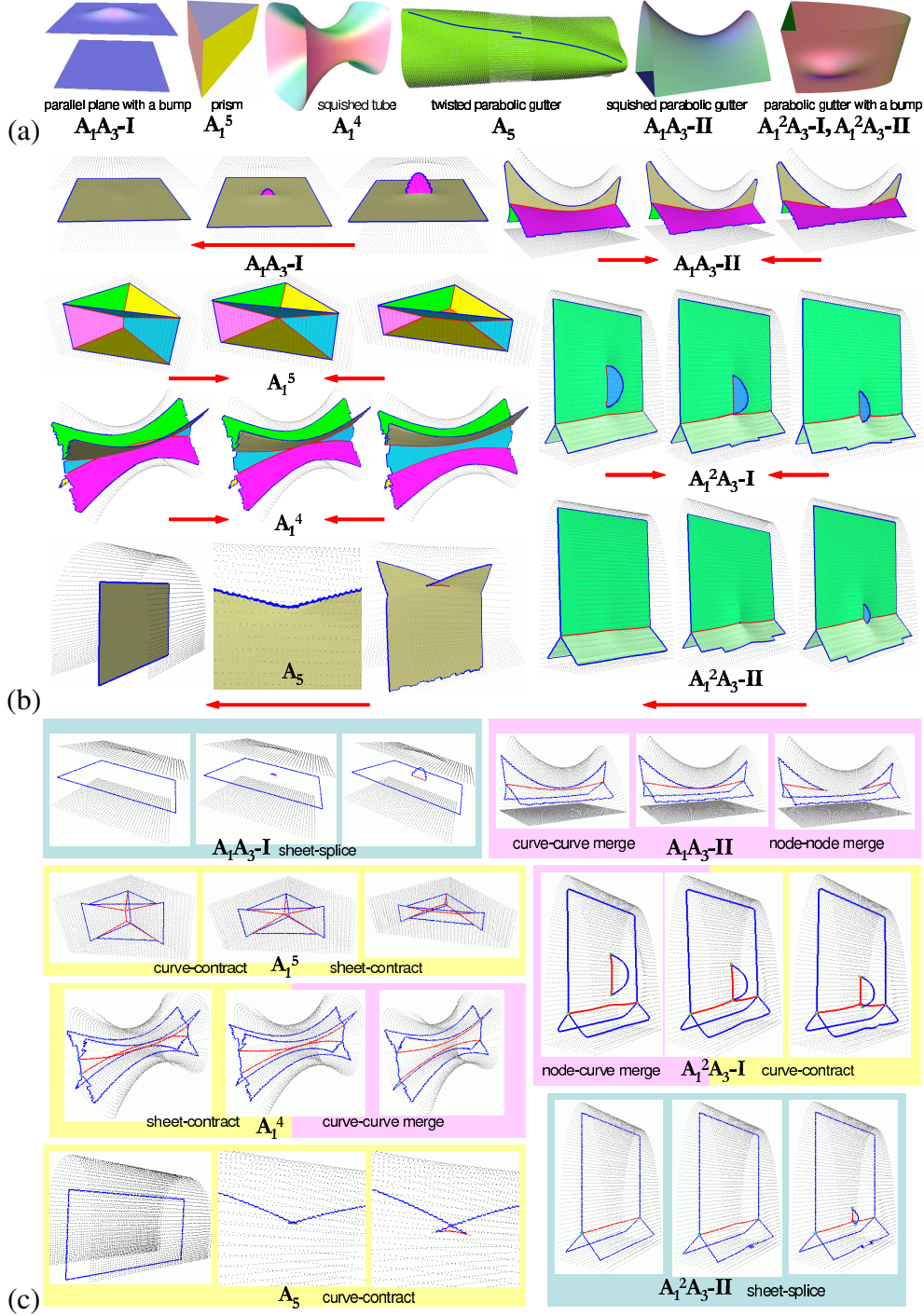


Fig. 3. Illustration of some \mathcal{MS} (or \mathcal{MA}) transitions in 3D [20]. (a) Seven example shapes near the transition point across a one-parameter family of shape deformation. (b) Simulation of the 7 cases of \mathcal{MS} transitions. In (c) all medial sheets are hidden to better visualize their structures. There are 11 \mathcal{MS} transforms (red arrows) in total corresponding to the 7 transitions defined in [10].

The process of achieving a transition on an existing shape is called a *symmetry transform*, i.e., by changing the \mathcal{MS} we are jointly deforming the shape. Ideally,

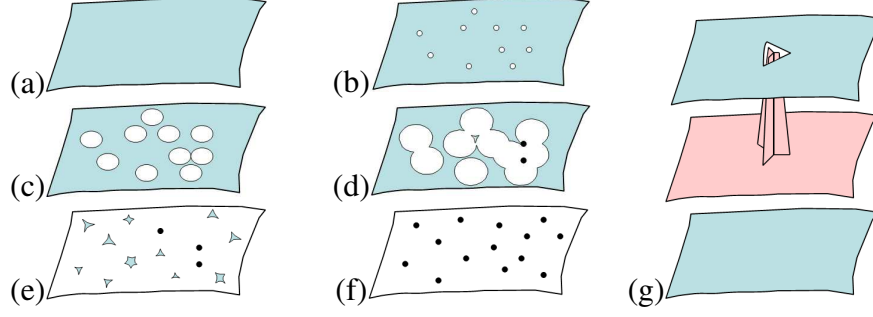


Fig. 4. This figure illustrates that “sampling” a surface to produce an unorganized cloud of points can be viewed as a *deformation process*: (a) the original surface; (b) some points are taken away from the surface, creating holes; (c) these holes grow thus reducing the area of the remaining surface patch, and (d) eventually start joining to form larger gaps and isolated patches; (e) the remaining surface patches each tend to shrink to a point; (f) the resulting point cloud represents samples of the surface. (g) An infinitesimal but generic hole creates three \mathcal{MA} sheets intersecting at at \mathcal{MA} curve.

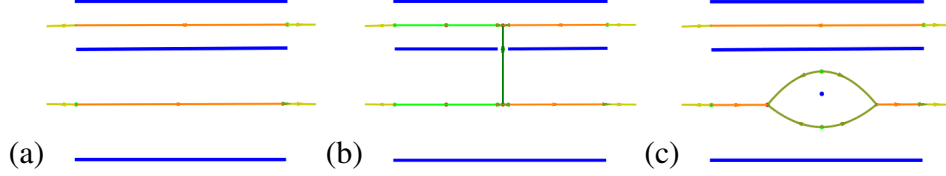


Fig. 5. Example of two shock transforms in 2D. (a) The dark blue curves are 2D object contour segments and other colors represent the shocks. (b) The removal of an object point adds a line to the shock graph, while (c) adding a point creates a loop; we are only concerned with the process in (b) in this paper.

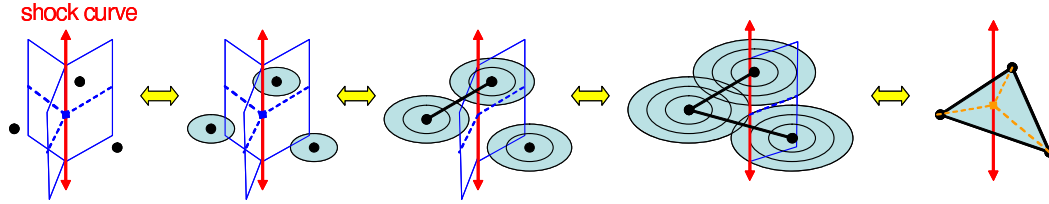


Fig. 6. This figure illustrates how a \mathcal{MS} curve (with three sheets) is transformed into a surface triangle in a deformation process simulated as a gap transform.

all sequences of transitions should be considered, *i.e.*, the ones described in Fig.3 and additional ones having to do with adding and removing isolated points in Fig.5. In this paper, however, we restrict ourselves only to the transition arising from removing a patch from the surface (the first step in our view of the sampling process) and consider only the symmetry transform pertaining to that, called the **gap transform**. Specifically, removing a surface patch generically results in adding to the \mathcal{MS} a group of three \mathcal{MA} sheets and one \mathcal{MA} curve at their intersection. Thus, the inverse process of reconstructing a surface patch corresponds to consider the removal of such curves and their attached sheets from the \mathcal{MS} , Fig.6.

As in perceptual grouping in 2D [43], the gap transform in 3D is considered in a *greedy* sequence, proceeding first with the least cost transform. This cost is based

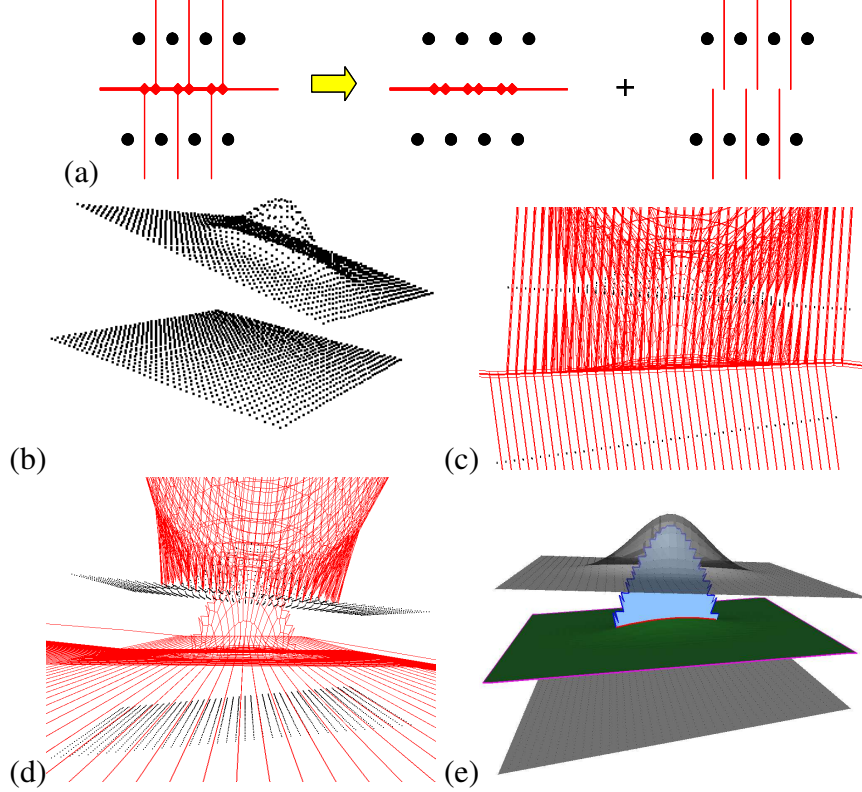


Fig. 7. (a) Illustration of the proposed *shock segregation* process in (a) 2D and (b-d) 3D. (b) A set of 3,200 points are uniformly distributed on a pair of planes, one of which is deformed by an elongated Gaussian kernel; (c) side-view of the corresponding \mathcal{MS} . (d) The remaining scaffold after undergoing a series of *3D gap transforms*. (e) The results of this shock segregation are two-fold: (i) the reconstructed surface and (ii) its corresponding \mathcal{MS} hypergraph.

on the likelihood that a \mathcal{MS} curve represents a ‘missing’ surface patch, as well as on the local context (*i.e.*, other reconstructed surface patches which may already exist in the vicinity). Fig.7 illustrates this process, which can be viewed as a **segregation process** (Fig.7a) of the 3D shocks on a pair of planar sheets, one of which is somewhat protruding (Fig.7b). The \mathcal{MS} (Fig.7c), after a sequence of gap transforms, then resembles that of the original two complete un-sampled surfaces (Fig.7d). This in turn enforces a connectivity of the surface samples thus creating a surface mesh together with an approximated \mathcal{MA} (Fig.7e).

This paper is an extension of a conference version [11]. The remaining of this paper is organized as follows. We first review the background for surface reconstruction below (Section 2). We then describe our approach in greater detail (Section 3), followed by experimental results and other extensions (Section 4) and conclusions.

2 Background

Surface reconstruction is an important problem in computational geometry, computer aided-design, computer vision, graphics, and engineering. Methods related

to our approach can be classified into three main categories, depending if they are based on (1) implicit (distance) functions, (2) propagation (region growing methods), and (3) Voronoi/Delaunay geometric constructs.¹

Implicit methods can be classified into two sub-categories: (i) local approximants to the global distance function constructed by blending primitives (implicit functions) and (ii) global approximants to the distance function constructed by (volume) propagation.

First, *local approximants* mainly differ in three ways: how clusters are generated, the type of implicit function used, and how local functions are blended together. Most recent papers evolve from Blinn’s original work [6] and vary mainly in the types of local implicit functions used, *e.g.*, linear combination of Gaussian blobs [34], bounded polynomial in three variables [42], blended union of spheres [31], and, more recently, mixtures of globally supported radial basis functions (RBF’s) [9,46], as well as blended compactly supported RBF’s [39,36,28]. A line of recent improvements is the multi-level partition of unity implicits (MPU’s) [35,37,45].

Second, *global approximants* explicitly build a distance field on a grid which tessellates a volume containing the input dataset. Following the early work of Hoppe *et al.* [23], Curless & Levoy meshed range scans by volumetric integration using “lines of sight” [13]. More recent work is based on the level-set methods [24,49]. Closely related is the method of having an active surface model deform and be attracted towards sample points or data under the influence of a global distance function [44,41,18]. Recent work by Kazhdan *et al.* [26] reconstruct surfaces from *oriented* points (with normals) by extracting the iso-surface of an “indicator field” (defined as 1 at points inside the model, and 0 outside) casted as a global Poisson system (*i.e.*, resolving the divergence of a vector field, ϕ , bounding a solid as a scalar function, μ : the famous Laplace-Poisson equation $\nabla^2 \phi = \mu(x, y, z)$).

Implicit methods make a strong assumption that the surface normal at each sample point is available in some form — equivalently that a consistent gradient of a vector (distance) field can be retrieved everywhere in the vicinity of samples. A normal field is very informative in that the surface is essentially made available near each sample point in the form of a locally oriented tangent plane (patch). This usually implies that an “inside” and “outside” of an object’s surface is available leading most implicit methods to further require that the surface to recover bounds a solid. These methods are well adapted to deal with noise, at the expense of smoothing away sharp features.

Methods such as [48,24] reconstruct surfaces without directly referring to the (oriented) normal information. However, in these methods an initial computation of a distance field surrounding points is required, providing a consistent normal field,

¹ Other classification schemes have been proposed; *e.g.*, the addition of “hybrid” methods combining the main features of two or more categories.

albeit from the provided sampled data, and without the need for an a priori identification of an inside/outside.²

Propagation-based methods can be traced back to the algorithm of Boissonnat [7], where a seed edge is picked to find a triangle in the locally estimated tangent plane. More recently, the *ball pivoting algorithm* [5] “rolls a ball” of fixed (parametric) size to reconstruct a mesh, leaving holes in the mesh, a function of the radius parameter which may be iteratively increased to handle non-uniform sampling (but then, risks introducing additional unwanted interpolants). In [25], the propagation is restricted by using the k -nearest-neighboring points and locally varying the ‘radius’ parameter automatically. Another related alternative is the Intrinsic Property Driven (IPD) algorithm [32], based on the ratio of the lengths of the longest and shortest edges incident to a point. Typically a space-division such as an octree or a voxel grid is used to efficiently lookup for neighboring candidates.

These methods still suffer from topological errors, in particular when two distinct surface patches come close to each other (Fig.8a). Gopi *et al.* introduce further restrictive assumptions to help with local ambiguities [21]: locally uniform sampling is assumed, minimum distance between samples belonging to different surface patches is set, and smoothness of the original surface is imposed. An important set of propagation-based algorithms uses Delaunay triangles in their scheme; we consider some of these below.³

Voronoi/Delaunay methods can be organized in two subcategories, whether they are surface or volume oriented.

First, *incremental surface-oriented* methods select suitable Delaunay triangles interpolating sample points either in a batch or in a greedy incremental fashion. A popular recent set of such methods was initiated by the works of Amenta *et al.* who proposed a Voronoi filtering method [1] where they first consider furthest apart Voronoi vertices of a Voronoi cell of a sample point p to define *poles* by pairs, to approximate the local surface normal. They also define the *local feature size* as the minimum distance from a sample p to the *theoretical MA* (for a smooth surface with bounded curvatures), which is used to derive theoretical constraints on the sampling and guarantees on the resulting meshes. The method requires a second pass of Delaunay computation using the poles as additional vertices. Triangles through triplets of original sample points are kept to construct a final mesh called the *crust*. Difficulties occur as the *poles* do not always approximate well surface normals, and, thus, a post-processing step is needed to trim results and fix such problems. This was further refined by defining a local neighborhood in the vicinity of a sample p taken as the complement to a cone intersection with its Voronoi cell, called “co-cone,” giving a heuristic to approximate the local tangent space at p to restrict the search for neighboring samples to create candidate triangle interpolants.

² Nevertheless, these methods have been developed for meshing solids only.

³ They could well be considered as a set of “hybrid” methods.

This improved *cocone* method [2] has for main advantage the bypassing of the second Delaunay computation in computing a crust mesh. However, it still requires heuristics to try repair the computed mesh. Note that the theoretical guarantee of obtaining a correct reconstruction in the above *crust* and *cocone* methods is only valid for a strict requirement of densely sampled points, for a smooth surface where the \mathcal{MA} nowhere reaches the surface, a condition which is not necessarily true in all practical situations, such as is the case of objects found in CAD-CAM (with sharp surface features and boundaries).

Aware of these issues, Petitjean and Boyer proposed another approach by defining a notion of r -regularity measured from the samples alone and combined with a discrete (rather than theoretical) \mathcal{MA} [38]. A surface mesh is then constructed by a propagation scheme by selecting Delaunay faces meeting the r -regularity criterion. Kuo and Yau [29] improve the *crust* algorithm to preserve sharp features by a combined adaptive sculpting and region-growing scheme.

Cohen-Steiner and Da have proposed a greedy incremental algorithm that uses for its main heuristic in selecting interpolants the dihedral angle between Delaunay face pairs, which postpones difficult decisions on a queue, and uses additional heuristics to detect surface boundaries [12]. The greedy β -skeleton in [22] iteratively meshes suitable Delaunay faces satisfying certain topology constraints, but an overall error repairing heuristic is lacking.

Another recent line of work relies on the notion of a discrete Morse theory induced from the dual Voronoi–Delaunay complex: the critical points of the (discrete) distance function, h_Σ , from the point samples are exactly given as the intersection of Voronoi k -faces and their dual Delaunay simplices [15]. Giessen *et al.*'s consider properties of the “flow” induced by h_Σ , *i.e.*, the unique direction of steepest (distance) ascent of h_Σ at any non-critical point, x . All critical points, c , are given a 0 vector, while every other point in \mathbb{R}^3 is associated to the unique unit vector of steepest ascent (of h_Σ); this defines a vector field on which discrete Morse theory can be applied. A “stable manifold” of a critical point c of h_Σ is defined as the set of points whose orbit ends in c (*i.e.*, flow into c). This creates 4 types (indices) of stable manifolds in 3D: (i) an index 0 critical point, *i.e.*, a local minimum of h_Σ — identically a sample point — has for stable manifold the sample point itself; (ii) an index 1 critical point is a 1-saddle which sits at the intersection point of a Delaunay edge with its dual Voronoi facet, and has for associated stable manifold a Delaunay (Gabriel) edge; (iii) an index 2 critical point is a 2-saddle which sits at the intersection point of a Delaunay facet (triangle) with its dual Voronoi edge, and has for associated stable manifold a piecewise linear surface patch; (iv) finally, an index 3 critical point is a local maximum (of h_Σ), a Voronoi vertex. Stable manifolds of index 2 are used to reconstruct a surface in this scheme. Under high density sampling conditions and for smooth surfaces, Giessen *et al.* show that the critical points can be separated into two classes: one class made of critical points remaining nearby the original surface, and the other made of critical points near the (theoretical) \mathcal{MA}

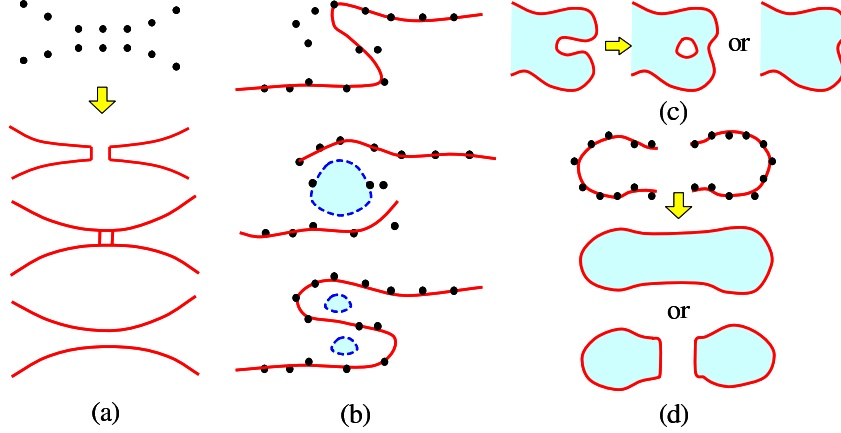


Fig. 8. Illustration of some local ambiguity problems (on topology) of several categories of methods in reconstructing curves from unorganized samples in 2D. (a) Incremental methods based on a local analysis could result in curves of very different topologies. (b) Implicit methods could model regions into very different topologies. (c) Deformable methods could merge near-by contours and split them into artificial holes or remove them completely. (d) Volumetric methods, which enforce the assumption of a solid, could face the ambiguity to either merge the contours or split them.

of that (assumed) smooth surface.

Second, *volume-oriented methods* consider the restrictive problem of reconstructing the closed surface bounding some solid. An early “volume sculpting” method was proposed by Boissonnat where various tetrahedra faces can be removed one by one [7], which was later refined in particular by Attali with a notion of r -regular shape taken from mathematical morphology [4].⁴ Another variant, the *Power Crust*, is based on the power diagram (a weighted Voronoi diagram), with theoretical guarantees under a proper sampling assumption, and with polygonal interpolants rather than triangles [3]. The *tight cocone* is based on the original cocone method (see above), but specialized to produce a water-tight mesh for solids [16]. More recently, robustness to noise for this approach is based on a model where both the sampling density and noise level can vary locally [17]. A recent improvement of the Power Crust in [33] handles noisy dataset and allows arbitrary over-sampling densities.

The problem of reconstructing surfaces with arbitrary topology from unorganized points is *ill-posed* mainly in that the sampling condition (density) is unknown. Methods generally face ambiguities (as Fig.8 illustrates) and impose restrictive assumptions on the topology or the sampling, more or less, one way or the other. In summary, implicit methods face difficulties on the availability/reliability of surface normal and the volume-enclosing assumption. Propagation-based methods are

⁴ A well-known variant of this type of methods is the α -shapes of Edelsbrunner *et al.* [ACM TOG’94] and the recent conformal α -shape [Cazals *et al.* PBG’05] and weighted α -shape [Park *et al.* LNCS’05].

generally efficient, simple to implement, but are less powerful in dealing with ambiguities, *e.g.*, when two surfaces come close together or near sharp features. Voronoi/Delaunay methods explicitly tessellate the space in between samples, and have in the recent years being studied with theoretical guarantees under restrictive sampling assumptions (*i.e.*, with high density of sampling for smooth surfaces such that the (theoretical) \mathcal{MA} never comes too close to the original (unknown) surface) [3,15,17].

Handling large datasets: Recent needs in handling huge collections of samples, *e.g.* from laser scanned points, often demand out-of-core techniques which are scalable to such large datasets. Typically, the computation is first performed locally and then iteratively refined to produce a globally consistent result. For example, the *super-cocone* in [14] uses an octree subdivision and runs the *cocone* algorithm multiple times towards a final surface. Notably, a recent multi-level streaming Poisson-based approach [8] handles up to 400 million oriented points. In our novel method discussed below, we rely on an adaptable and scalable partition of space function of local sampling densities to also produce a multi-pass version of our reconstruction scheme, allowing us to handle arbitrarily large datasets.

3 Surface Recovery via \mathcal{MS} Transforms

In the remaining of this paper we discuss in greater details our approach whose main features are as follows:

- The input data consists of only 3D positions of (sample) points — no assumptions (on sampling density, normals) is needed to process a raw dataset, although additional knowledge, such as on the sampling density and the local connectivity as a partial mesh, can be used to refine our results as discussed below.
- Our method is one of the few to deal with generic topological configurations: non-closed, non-smooth, non-orientable, non-manifold, capable of handling various topologies, and yet able to also recover water-tight surfaces for solids, identify ridges and intersection curves (via recovered \mathcal{MA} features).

The computation relies on a first stage of deriving a medial scaffold (\mathcal{MS}), from the original dataset as presented in [30]. This step is as efficient as the computation of a Voronoi or Delaunay tessellation for the input point set.⁵ Our meshing approach can be summarized with the following steps:

- (1) Compute the \mathcal{MS} of the input point cloud [30].
- (2) Create two queues sorting all \mathcal{MS} A_1^3 curves⁶ such that their associated A_1^3 -2

⁵ The \mathcal{MS} is directly computed from the input set in 3D, contrarily to the main method of computing the Voronoi/Delaunay complex in a higher dimensional space.

⁶ The A_k^n notation indicates the order (k) of contact of a maximal ball with n surface points. A_1^3 shock curves delimit A_1^2 \mathcal{MA} sheets; for a point cloud, these curves are identically Voronoi edges, their A_1^4 endpoints are Voronoi vertices, and their duals are Delaunay

critical points are either along these curves or not.⁷

- (3) Rank order these queues based on the local geometry and topology of the associated triangle set.
- (4) Start creating a reconstructed surface set by using the order of triangles associated to the two queues satisfying topological constraints. As triangles are meshed, both their costs and the neighboring triangle's costs are updated accordingly.
- (5) The meshing is completed when all ranked ordered shock curves are considered. Additional steps are taken to handle non-manifold intersections (if any).

We now present the details of our method, once the \mathcal{MS} is made available by the method of Leymarie and Kimia [30].

We propose to reconstruct a surface mesh from unorganized point clouds with minimalistic assumptions, by considering the *gap* transforms on all A_1^3 shock curves in a rank-ordered greedy approach. The greedy nature of the algorithm, in contrast to the ideal but impractical algorithm considering all sequences of gap transforms on the \mathcal{MS} , implies that local decisions in ambiguous cases may lead to erroneous results. Since the ranking is partially decided on the basis of a local surface neighborhood, these errors can then potentially propagate. To prevent the negative effects of such “local minima”, we adopt a three-fold strategy. (i) The set of A_1^3 curves is divided into two distinct categories, based on whether the gap transform decision is categorically easy or difficult, leading to two passes of greedy iterations. The first pass aims at constructing valid surface triangles (*i.e.*, without ambiguity about their shape or local topology) in a greedy iterative sequence. Given a well-sampled input (*i.e.*, the *intra*-sampling distance is smaller than the *inter*-sampling distance, Fig.9),⁸ most surface triangles can be successfully built in this step. All remaining cases are then handled in the second pass, where the surface built in the first pass can be used to help resolve the initially ambiguous situations as discussed below. (ii) Since applying a gap transform can invalidate another candidate gap transform, in situations when a pair of gap transforms are close in ranking, *i.e.*, the difference in costs $\delta\rho$ is smaller than a threshold Δ_ρ (where “ ρ ” denotes a cost function), their

triangles through the $n=3$ sample/contact points. A_1^3 curves are computed from a sequence of critical points of the radius flow (the radius of maximal balls projected on the \mathcal{MA}): sources for \mathcal{MA} sheets (A_1^2-2) are paired to find sources for shock curves (A_1^3-2), which in turn are paired to find A_1^4 endpoints; details in [30].

⁷ In comparison to the *flow complex* approach of Giesen *et al.* based on a discrete Morse theory analysis [15], our definition of the \mathcal{MS} , which is based on singularity theory, produces a richer superset of critical points. That is, all the “relay” type of *shock* vertices we identify and use (A_1^2-3 , A_1^3-3 , see below) [30] are *not* identified as *critical* points in the Morse analysis within the flow complex framework. Omitting these singular points makes their subsequent analysis and algorithms different from ours.

⁸ We consider the theoretical sampling issues for our method elsewhere; the focus here is on introducing the algorithm for surface reconstruction and main philosophy behind the use of \mathcal{MS} transitions/transforms.

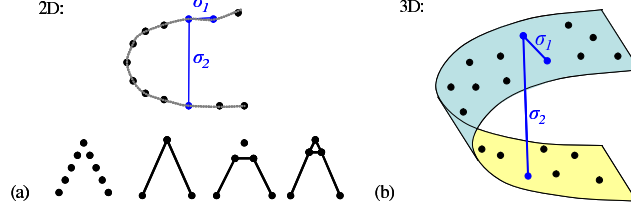


Fig. 9. The sampling condition: intra-sampling distance $\sigma_1 \ll$ inter-sampling distance σ_2 . Observe that this condition is not satisfied near sharp features.

ranking is reduced by increasing their cost by a factor ζ , so as to delay a decision until additional local context is available.⁹ (iii) An *error recovery* option is available: after each gap transform, costs associated to neighbor surface interpolants (including already selected and candidate triangles) are re-evaluated: if the newly computed cost of a previously accomplished gap transform exceeds our top candidate's cost, we “undo” it by putting back the associated A_1^3 curve into a (priority) queue. This essentially increases the robustness of the algorithm in recovering from errors, which will be addressed in more details later.

Initialization: The surface reconstruction is initialized by sorting all A_1^3 curves into the two priority queues, Q_1 and Q_2 (for the 1st and 2nd greedy iterations). Each A_1^3 curve identifies three sample points, its *generators*, which can potentially be meshed with neighbors on the original surface. Fig.10 shows the *three possible types* of candidate surface triangular interpolant. When there are no other sample points nearby the three generators, the minimum-radius \mathcal{MA} point for these three generators is always on the A_1^3 curve: it is the critical point of the radius flow, A_1^3-2 , which sits at the circumcenter for the three generators and which is used to build the \mathcal{MS} [30]. This critical point is then either inside (type I) or outside (type II) the interpolating (Delaunay) triangle, Fig.10a,b. Cases corresponding to types I and II are straightforward and used to create queue Q_1 . The remaining possibility of having the A_1^3-2 critical point not belonging to part of the A_1^3 curve indicates that a nearby forth generator is preventing its “formation” (*i.e.*, as being part of the coarse-scale \mathcal{MA}) and that the local connectivity is ambiguous (type III), Fig.10c. Cases corresponding to type III are more likely to lead to meshing ambiguities and are therefore used to create a separate queue Q_2 .

Rank ordering A_1^3 curves: The rank of each A_1^3 curve is based on (i) the likelihood that the corresponding triangle could have arisen from the original surface given the \mathcal{MS} curve length,¹⁰ and (ii) the consistency of the gap transform in reconstructing a surface given already neighboring meshed sample points. We discuss each case in turn.

⁹ Experimentally, we found that values of $\Delta_\rho = 5\%$ and $\zeta = 4\Delta_\rho$ lead to robust results.

¹⁰ Intuitively, the longer an A_1^3 curve is, the likelier it can approximate well the local normal field.

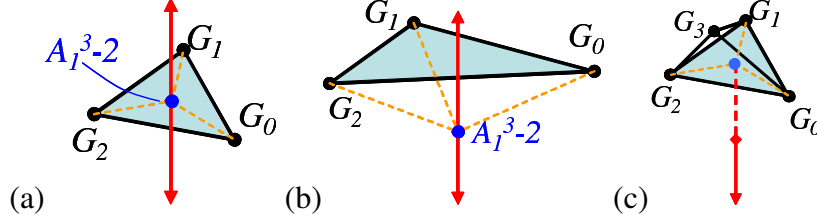


Fig. 10. The three types of A_1^3 shock curves (red) and their corresponding Delaunay triangles (cyan) used in our method: (a) an A_1^3 curve with an *acute* triangle (type I), (b) an A_1^3 curve with an *obtuse* triangle (type II), (c) an A_1^3 curve not containing the A_1^3-2 critical point (type III).

Ranking A_1^3 curves *without* local context: Such a ranking can be decided on the basis of the shape and size of the candidate triangle with respect to the length of the \mathcal{MS} curve. Let the sides of the triangle through three generators be d_1 , d_2 and d_3 , and the shortest length (among the two sides) of the A_1^3 curve with respect to its A_1^3-2 critical point be R , and define:

$$\begin{aligned} d &= \max(d_1, d_2, d_3) \\ P &= d_1 + d_2 + d_3 \\ m &= (d_1 + d_2 - d_3)(d_3 + d_1 - d_2)(d_2 + d_3 - d_1) \\ A &= \sqrt{(P \cdot m)/16} \\ C &= 4\sqrt{3} \cdot A / (d_1^2 + d_2^2 + d_3^2), \end{aligned}$$

where P is the triangle's perimeter, A is its area (Heron's formula), and C measures its compactness (Gueziec's formula). Then, the cost:

$$\rho_1 = \begin{cases} \frac{P}{R} \cdot \frac{1}{C^2}, & \text{if } d < d_{\max} \\ \infty, & \text{if } d \geq d_{\max} \end{cases}, \quad (1)$$

favors compact triangles rather than elongated ones, and triangles with smaller size (*w.r.t.* the shortest length R of their associated A_1^3 curve); d_{\max} represents the maximal length of an expected triangle side and is set in practice as: $d_{\max} = \eta \cdot d_{\text{med}}$, where d_{med} is the median of the histogram of A_1^3-2 radii of all type I and II shock curves (Fig.11).¹¹ Observe that this cost will delay the completion of triangles which are near corners and ridges in favor of those in flat regions, away from such “close encounters” (Fig.12), a direct influence of the factor $1/R$ in Eq.(1).

Ranking A_1^3 curves *with* local context: We now consider the relationship between an A_1^3 curve's putative surface triangle which can either share an *edge* or a *vertex* with its neighboring already reconstructed surface triangles, Fig.13. First, a candidate triangle is more likely to interpolate the surface if it is oriented similarly to its neighboring triangles, sharing an *edge* as determined by the dihedral angle

¹¹ d_{med} is used as the estimate of the expected *unit distance between samples*, while $5 < \eta < 15$ is a parameter which experimentally varies according to the sampling uniformity.

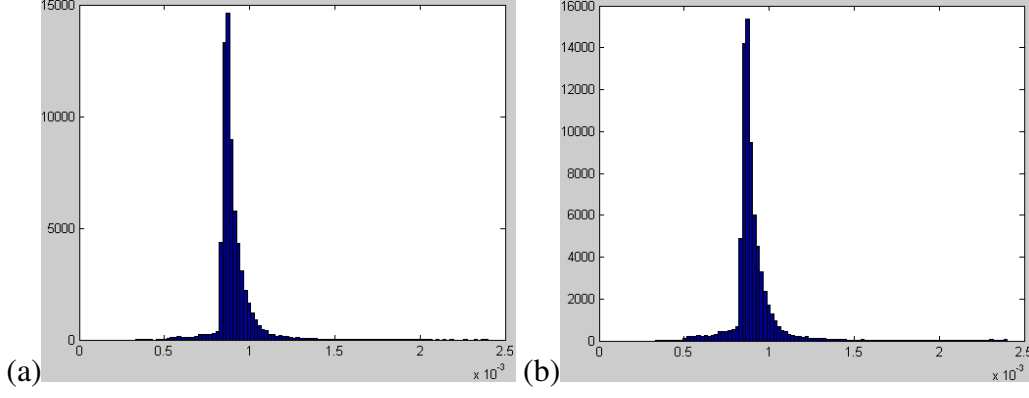


Fig. 11. (a) The distribution of the A_1^3 -2 radii of all shock curves corresponding to triangles in the original mesh of the Stanford Bunny model (Fig.20). (b) The distribution of the A_1^3 -2 radii of all shock curves of type I and II in the \mathcal{MS} of the point cloud. Observe how the shape of the distributions resemble each other. The median of the A_1^3 -2 distribution (d_{med}) is used to set d_{max} in Eq.(1).



Fig. 12. Steps in reconstructing a “corner shape” from 800 sample points which illustrate how our method favors meshing triangles in the region distant to the corner prior to meshing sharp features, where difficult decisions (due to under-sampling) have to be made. From Left to Right: the full \mathcal{MS} (in red, with unbounded elements not shown); reconstructing the first 10%, 50%, and 90% of candidate triangles; final surface reconstruction result.

θ between the two. When θ is small ($< 45^\circ$) we expect the observed continuity to offset the cost ρ_1 . When θ is large ($\geq 45^\circ$) we expect the lack of continuity to rather augment the cost ρ_1 . The function $f(\theta) = [\exp^\theta - 1]^2 - 1$ captures this notion well: at $\theta = 0$, $f(\theta) = -1$; at $\theta = 40^\circ$, $f(\theta) \simeq 0$; at $\theta = 80^\circ$, $f(\theta) \simeq 8.24$, giving us:

$$\rho_2 = \frac{d_i}{R} \cdot \frac{1}{C^2} \cdot f(\theta), \quad (2)$$

where d_i is the length of the shared edge. Thus, for a triangle with its three edges part of existing smooth surface patches, the contributions for ρ_2 add up to completely cancel ρ_1 in Eq.(1).

The second form of local context for a triangle is when it shares a *vertex* with an existing triangle. This is a locally ambiguous situation that can potentially lead to (non-manifold) topological errors (Fig.13 bottom right) and must thus be delayed, avoided, or undone, to maintain a “one-ring” vertex topology, as described below in three possible cases. (i) If the shared vertex topology is already a *one-ring*, the new triangle should be rejected (Fig.13 bottom rightmost), since further meshing is

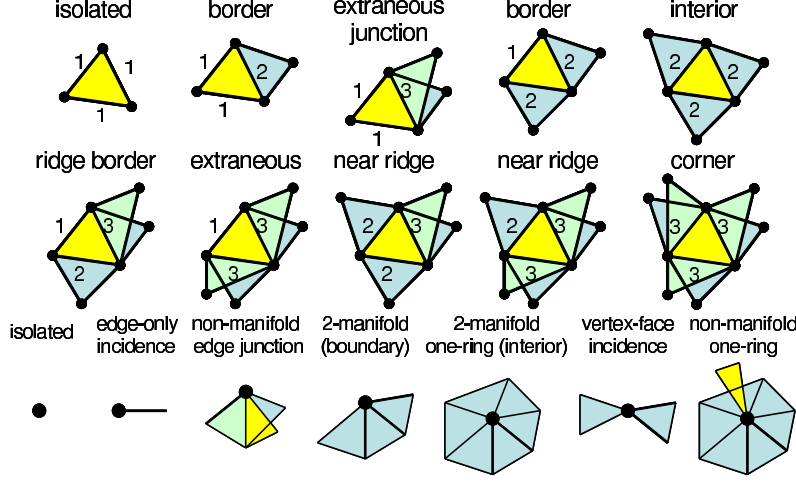


Fig. 13. (Upper) Typology of mesh triangle topology based on simplicial adjacency at edges. (Bottom) Typology of mesh vertex topology based on its incident edges and faces.

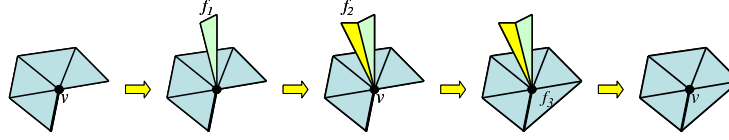


Fig. 14. Schematic steps of meshing and fixing vertex topology dynamically in our meshing process to retain a “one-ring” around a vertex.

unlikely to yield a better 2-manifold mesh with lower cost. (ii) If the shared vertex is a *vertex-face incidence*, (Fig.13 bottom second from right), the triangle should be delayed by increasing its cost and reinserted into its queue to be considered later. (iii) If the shared vertex is a *non-manifold one-ring* (the second last of Fig.14), the *one-ring* topology will be recovered because the gap transforms will be undone as described earlier.

In summary, we mesh surfaces in a *best-first* manner considering the suitability of each candidate triangle: its shape, corresponding shock curve type and length, continuity from neighbors, and the local mesh topology. Our method is implemented as a multiple-pass greedy iterative scheme. The first pass builds all confident surface triangles (with costs estimated via ρ_1 and ρ_2) from \mathbf{Q}_1 ; the second pass uses candidates from \mathbf{Q}_2 and resolves difficult cases using the local supports built from the first pass. Fig.15 depicts an example of \mathbf{Q}_1 and \mathbf{Q}_2 where the input is a set of 5,728 points sampling a toy sheep model. The meshing process is depicted in Fig.16. Our method can be viewed as a *multiple seeded propagation scheme*, where each A_1^3 curve without local context serves as an initial seed, whose selection is optimal (in a greedy sense) and is integrated with the meshing and error-recovery process. We note that the ability to *retract* improper meshing from errors is a major advantage over other greedy/propagation-based methods. Fig.17 shows an example of error recovery often observed in practice.¹²

¹² The proposed algorithm is *not* optimal and thus could produce meshes with artifacts such as the *twisted surface* in Fig.17, depending on the sampling condition. One solution

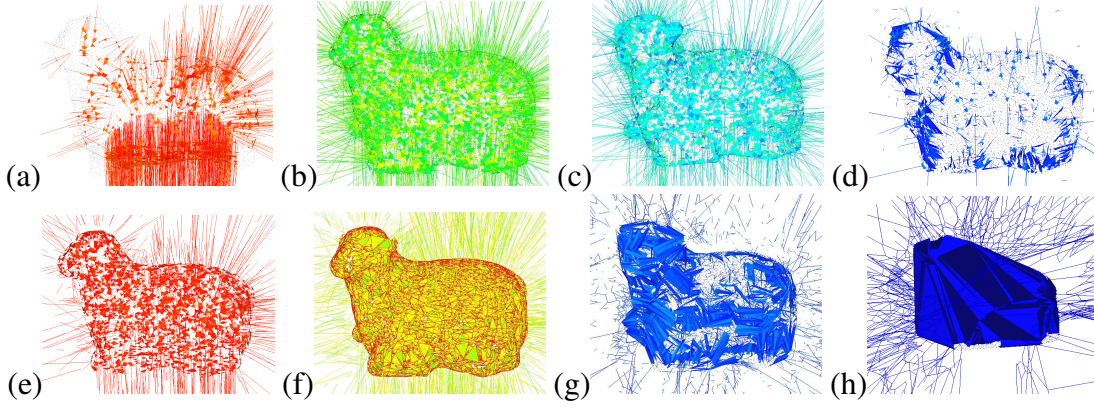


Fig. 15. This figure illustrates how the surface interpolants (of A_1^3 shock curves) are sorted according to their suitability in the classified queues. (a-d) The most relevant 10%, 30%-60%, 60%-90%, and last 10% of type I and II shock curves in Q_1 . (e-g) The most relevant 5%, 20%-40%, and last 10% of type III shock curves in Q_2 . (h) The oversized ($d > d_{max}$) surface interpolants, which are not considered in the algorithm.

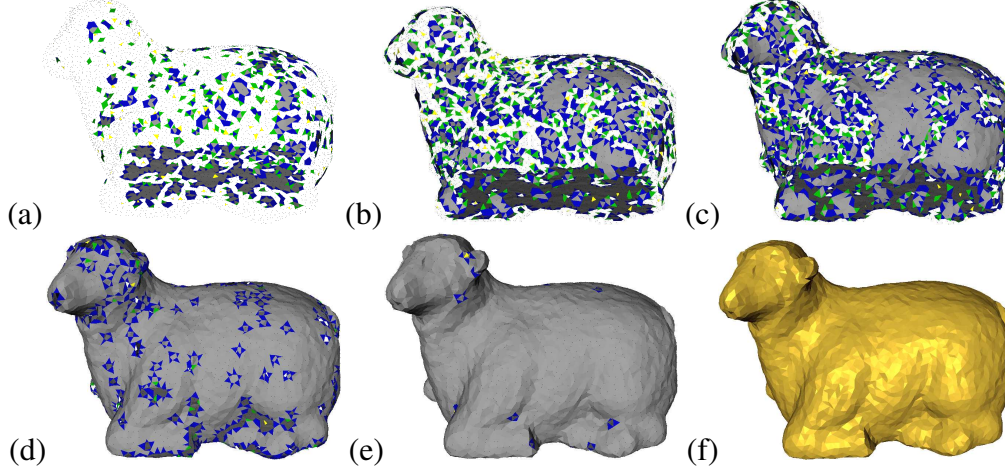


Fig. 16. Intermediate result of meshing the set of 22, 619 points sampling a toy sheep model at (a) 10%, (b) 30%, (c) 50%, (d) 75%, (e) 90% completeness of the queue Q_1 , which simulates an “inverse sampling” process as described in Section 1. Color scheme: in gray are shown “interior” triangles (Fig.13), in blue, triangles with 1 boundary side, and in green those with 2 boundary sides. (f) In this case, a water-tight surface is obtained after completion of meshing the A_1^3 curves in Q_1 .

Extensions: Our method can be further fine-tuned when additional information is available. For example, each A_1^3 curve can be shown to robustly estimate a surface normal, a result related to Amenta *et al.*’s analysis of *poles* [1]; thus, when normals are available in conjunction with a point cloud (as many main-stream methods assume), the candidate A_1^3 curve in the normal direction should be prioritized. When some existing mesh connectivity is already available (*i.e.*, a partial mesh to be re-meshed), the local structure of the \mathcal{MS} sharing such connectivity can be used to

is to use the (coupling with the) remaining \mathcal{MS} to guide the recovery process: the surface artifacts can be detected and fixed as other \mathcal{MS} transforms are applied [10].

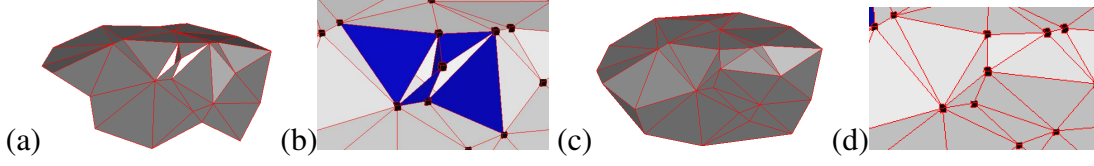


Fig. 17. This figure shows a “local minima” typically observed in greedy meshing algorithms and a solution to recover from it. (a) After our first greedy iteration we are left with a “twisted surface” with two holes on both sides. (b) Detailed view of the twisted surface. The introduction of the half-occluded triangle in the middle prevents the meshing to be 2-manifold and no further meshing is possible unless a retraction is allowed. (c,d) A retraction and re-meshing of all local triangles (blue) give a slightly different (but valid) manifold mesh.

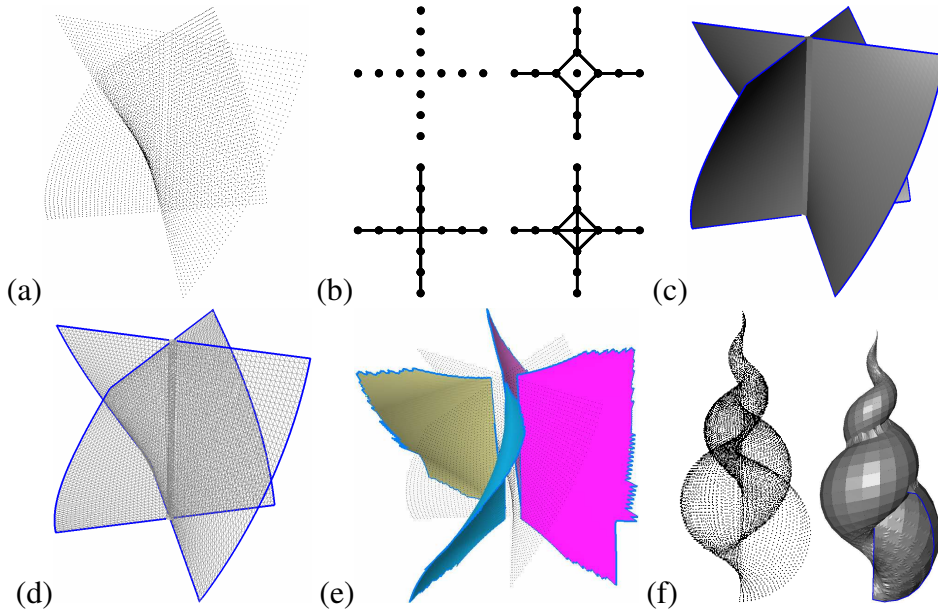


Fig. 18. Reconstructing surface with coarse-scale self-intersecting topology: (a) an artificial non-manifold surface with ‘+’ like cross-sections rotated along an axis. (b) The sample points in the cross section and possible reconstructions in 2D. (c) The resulting surface where the mesh structure is depicted in (d) and the corresponding \mathcal{MA} in (e). (f) Another example of meshing a seashell model with non-manifold seam junctions.

correctly select A_1^3 curves with associated surface triangles (detailed below).

Handling non-manifold surfaces: A second extension of our method is to reconstruct surfaces with coarse-scale non-manifold *seams* (path of intersecting junctions) [50] typically observed in CAD models and other multi-media objects, Fig.18. This problem is further ill-posed and rarely considered in the literature [47], since without the manifold constraint, any sample point in the vicinity could potentially be meshed. We propose to add a third queue Q_3 (after Q_1 and Q_2) to handle the re-meshed surfaces closer to such (coarse-scale) seams and surfaces with (true) boundary and large holes. This extends our approach to handle all surface topolo-

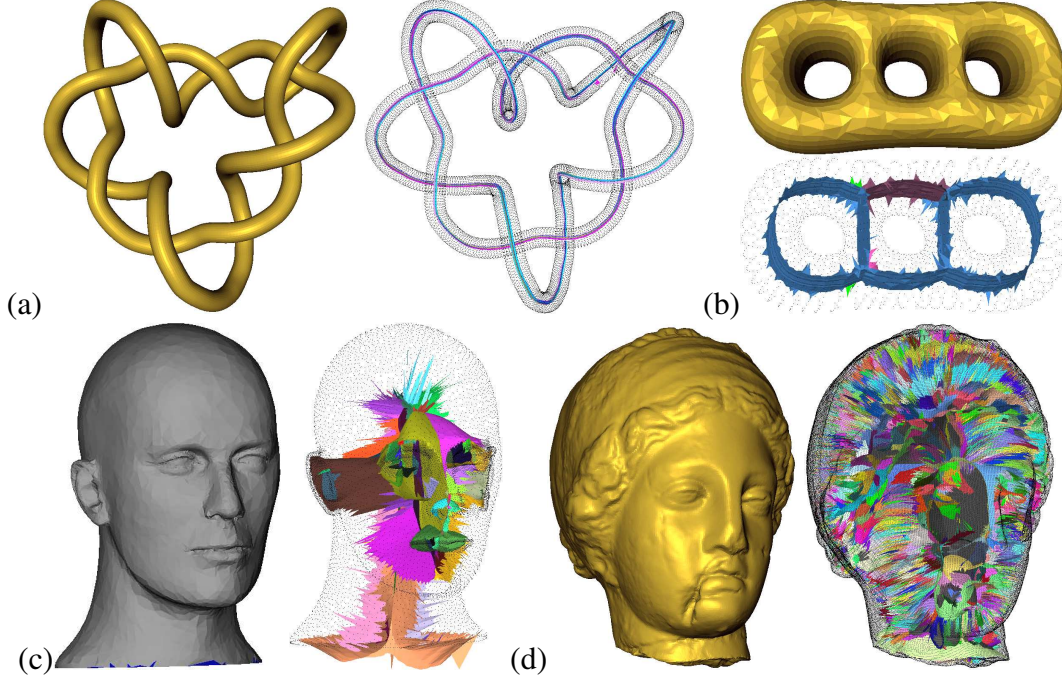


Fig. 19. Results of our reconstructed surfaces and corresponding \mathcal{MS} 's. (a) Uniformly sampled knotted figure (28,653 points, 57,306 faces, data from MPII). (b) Non-uniformly and low-sampled triple donut shape (1,996 points, 3,999 faces). (c) Cyberware Mannequin (6,386 points, 12,727 faces); NB: not a solid, bottom is left open. (d) Cyberware Igea (134,345 points, 268,686 faces). The initial \mathcal{MS} after the segregation typically contains abundant details about the shape. Further regularization of the \mathcal{MS} into a succinct form such as the ones in Fig.22 is detailed in [10].

gies in Fig.2. ¹³ The coupled \mathcal{MA} (Fig.18e) can be used to organize the resulting surface: an A_3 *rib* curve corresponds to a *ridge* or *seam* on the surface.

4 Experimental Results

We have implemented our method and extensively tested it in reconstructing surfaces with *various topologies*: closed (Fig.1d,g, 16, 19a,b,d, 22, 23, 25e), with multiple components (Fig.7, 21), non-orientable (Fig.1b), multiply punctured (Fig.1c), with multiple holes (genus) (Fig.19b, 22e, 23), closely knotted (Fig.1d, 19a, 25e), with boundaries (Fig.1a-c,e-f, 3, 7, 12, 18, 19c, 20, 21, 24, 26), with sharp ridges (discontinuous in curvature) (Fig.1e-f, 3, 12, 23, 24, 26), non-manifold surfaces intersecting at seams (Fig.1a, 18c,f), and with (relatively) low sampling (Fig.1d, 17, 19b-c, 24 (in each bucket), 25e). ¹⁴

In addition to meshing unorganized points, we point out three other useful applications: (1) re-meshing/repairing a partial mesh, (2) computing a tightly-coupled

¹³ Additional heuristics can be used *a posteriori* to fill remaining (large) holes if needed.

¹⁴ Color scheme: all surfaces in “gold” are closed (water-tight) by our method; otherwise they are left in gray, with boundaries in blue.

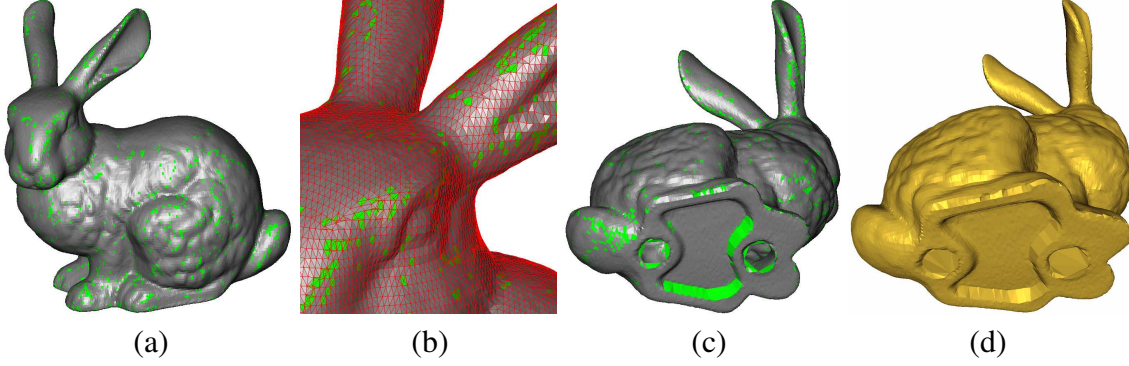


Fig. 20. Validation: Superimposing our resulting mesh on the Stanford Bunny shows that most of the original mesh is recovered. (a,b) The minor differences (green) are the result of different triangulations of geometrically similar surface patches. (c) The narrow strips of holes at the bottom of the model are filled and the two larger circular holes are left unfilled. (d) Increasing η to be 30 (and thus increasing d_{\max}) allows larger triangles to close these holes, producing a water-tight model.

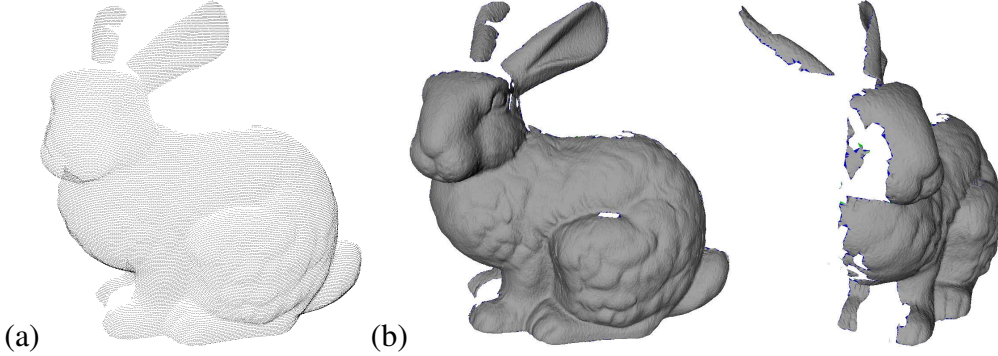


Fig. 21. Surface reconstruction from (a) a single scan of the Bunny (40,256 points) from the Stanford scanning repository. (b) Two views of our meshing result.

\mathcal{MA} with the meshed shape, which is useful in further modelling and matching applications [10], and (3) handling large datasets.

Re-meshing partial meshes: This can be done in two ways, either (i) by keeping all existing triangles known *a priori* to be correct and letting other candidate triangles compete to grow surface patches from these, or (ii) by assigning high priority to existing triangles (in a “*polygon soup*”) over other new candidates and letting the algorithm re-mesh them to a final surface. Fig.20 validates our reconstruction result against the original model. For applications, Fig.1e and 21 show the result of meshing *raw scan* data, which suggests the potential in fusing laser scans by meshing each scan for *e.g.* applying a point-to-mesh iterative closest point (ICP) algorithm to accurately align them. The low-sampling boundary in Fig.21 can be re-meshed after the fusion with more scan views.

Extending to robust \mathcal{MA} computation: The proposed method leads to a successful $\mathcal{MA}/\mathcal{MS}$ computation scheme in producing a coupled shape-skeleton representation [10], which is useful in a range of shape modelling and matching appli-

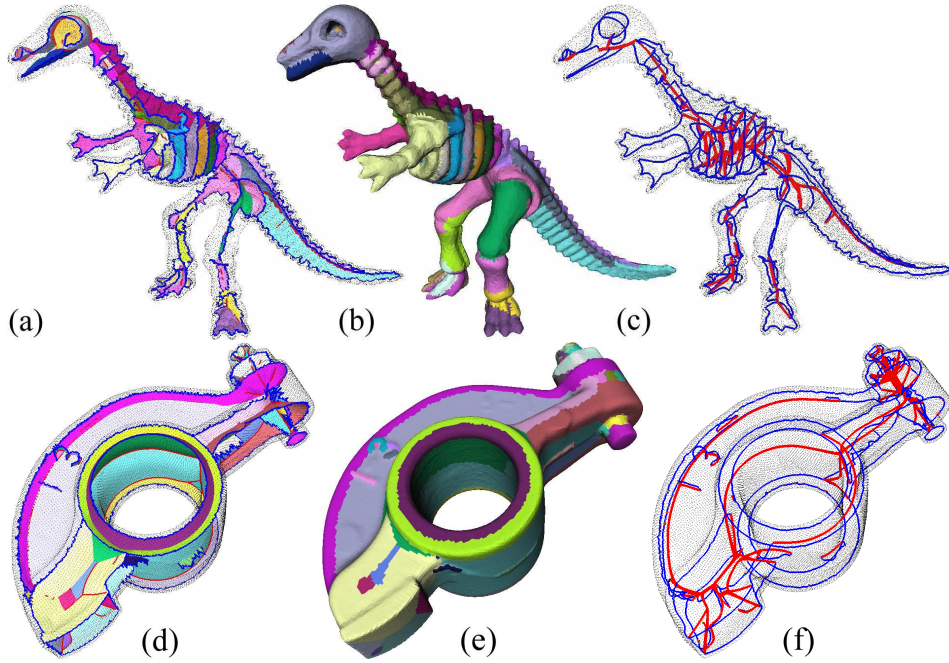


Fig. 22. The regularized \mathcal{MS} of (a-c) a toy dinosaur model (14,050 points) and (d-f) a rocker arm (40,177 points), data from Cyberware. Both \mathcal{MS} 's after regularization appear to be neat in structure and are close to the object surface to capture fine details. The tightly-coupled surface regions of the \mathcal{MS} 's (in corresponding colors) are suitable for further shape modelling and segmentation use.

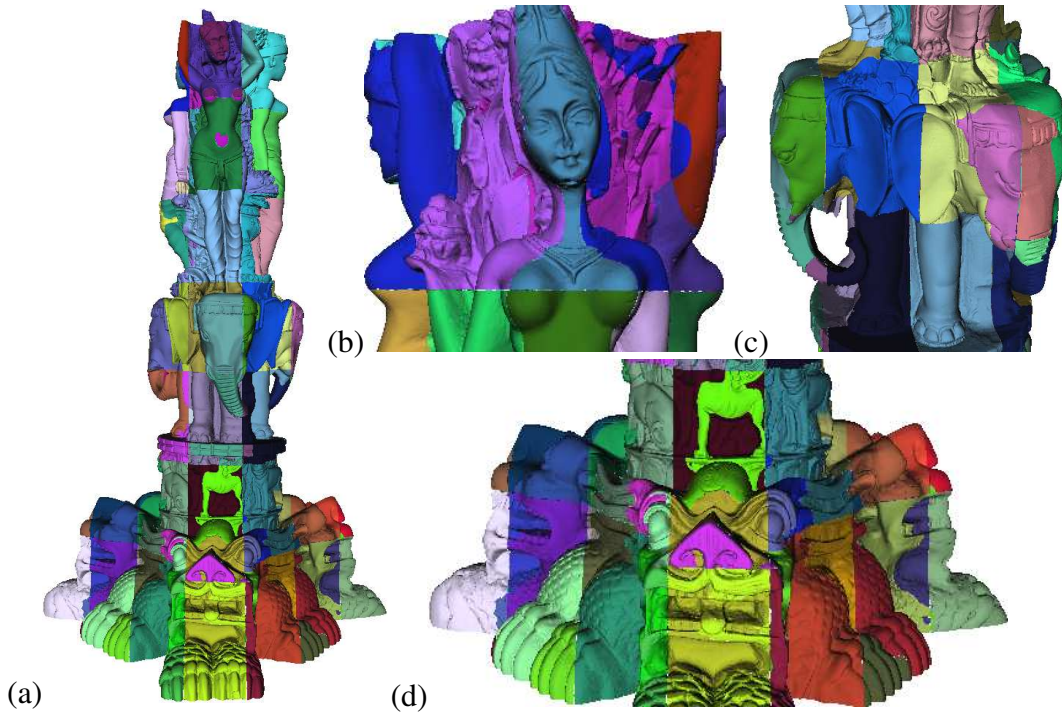


Fig. 23. Views of the individual reconstructed surface of the Stanford Thai Statue dataset ($\approx 5 \times 10^6$ points) inside the $7 \times 7 \times 7$ 3D buckets. Observe that the surface is fully meshed except at the boundary of the buckets.

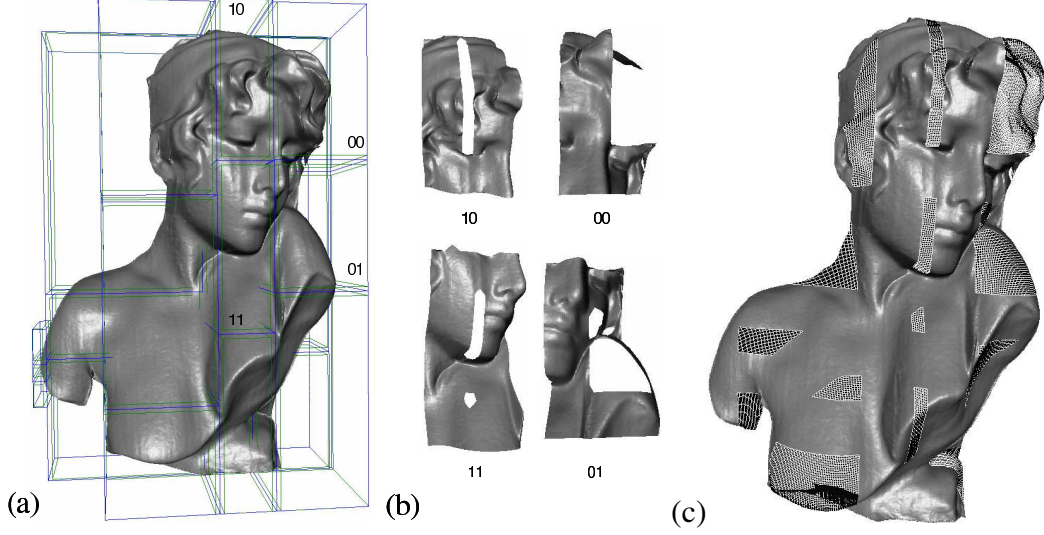


Fig. 24. This figure illustrates the bucketing/stitching approach on the Sapho dataset, where the bucket size is made small (10 K points each) to exaggerate the stitching regions. (a) The buckets (blue) and the final surface after stitching. (b) The *stitching meshes* of the 4 indicated buckets. (c) The final stitching surface is the union of all confident triangles in the stitching meshes.

cations. More \mathcal{MS} transforms (e.g., Fig.3) are involved to further regularize the remaining \mathcal{MS} into a simplified form, while in the process, the \mathcal{MS} is modified together with the tightly-coupled surface mesh [10].

Handling large datasets: An important point not raised thus far is that since we have not imposed any strong restriction on the surface topology, we can handle *very large datasets* by simply dividing the space into 3D buckets and meshing surface within each bucket, Fig.1g, 23, 24.¹⁵ We can then stitch the surface pieces together to get a final model using the same algorithm *again*. Prior to stitching, we exclude the un-reliable triangles near bucket boundaries (i.e., those which are built without support of nearby points from adjacent buckets). Then, the stitching of surfaces in adjacent buckets can be viewed as completing or repairing a partial mesh (by taking already meshed triangles near bucket boundaries as an initial solution).¹⁶ Fig.24

¹⁵ We make use of an efficient way of constructing 3D buckets in the course of computing the \mathcal{MS} as described by Leymarie & Kimia [30]. The two common ways to create buckets are as follows: (i) **grid-cell** bucketing: divide the space into $N \times N \times N$ buckets, where the number of sample points in buckets can be different, Fig.23. (ii) **adaptive** (variable-size) bucketing: each bucket contains roughly the same number of points and can be different in size, Fig.1f and 24.

¹⁶ The stitching algorithm is briefly described as follows. First, after the triangles inside each bucket are meshed, the un-reliable triangles near bucket boundary are excluded and all remaining triangles are considered *confident* and finalized. Second, the triangles to stitch each bucket near its boundary (in the *stitching sausage*) are meshed, by finding all finalized triangles in the enlarged vicinity of nearby buckets as initial triangles and run the meshing algorithm to seamlessly stitch them. Again the un-reliable triangles are excluded from

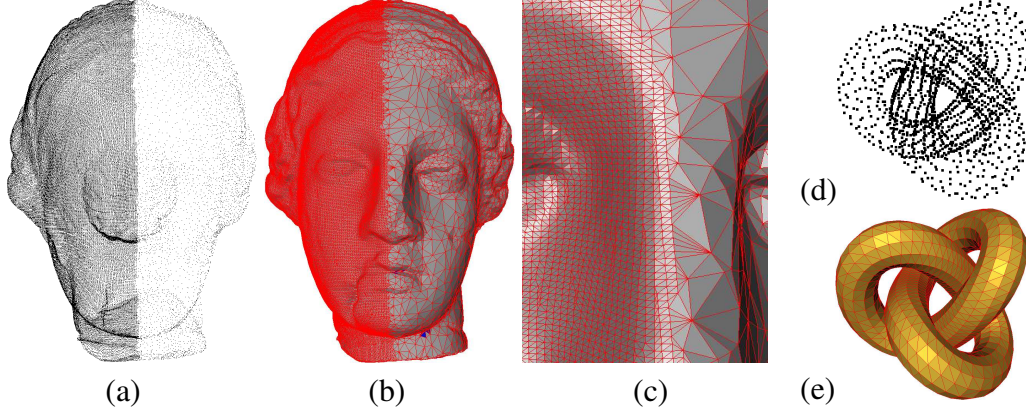


Fig. 25. (a) Igea dataset non-uniformly sampled with 75,545 points (from MPII). (b) Our result made of 75,545 vertices and 145,082 faces. (c) Observe how our method preserve fine details on the densely sampled surface. (d-e) The meshing of a low-sampling (1,440 points) knotting surface. Our algorithm recovers the surface by propagating from the (out-side) non-ambiguous regions which helps to solve the ambiguity in the tangled regions.

illustrates our implementation.

We have also tested our method for inputs made of *non-uniform* sampling as Fig.1d, 17, 19b-c, 22, and 25 demonstrate. The performance of our method degrades reasonably as *perturbation* increases, Fig.26. In addition, our method is capable to handle complex surfaces of relatively low (but more uniform) sampling, such as in Fig.1d, 19b, and 25e.

5 Conclusion

We have proposed a surface meshing method capable of dealing with a great variety of surfaces whether they are closed or not, orientable or not, uniformly sampled or not, with non-manifold intersections or not. The current implementation is roughly as fast (and with pseudo-linear complexity in the number of samples) as other recent popular methods (see Figure 27) and the potential to handle vary large datasets is also very promising. The coupled \mathcal{MA} computed together with the surface is useful in further shape modelling and matching application. We expect the capability of this approach to increase as additional \mathcal{MS} transforms (including the ones in [10]) are included.

Acknowledgments: This material is based upon work partly supported by the National Science Foundation under Grants NSF/ITR-0205477 and NSF/KDI-BCS-

this *stitching mesh* of each bucket. Finally, the union of all confident triangles (inside all buckets and all stitching meshes) gives the final surface, Fig.24. The operation inside each bucket/stitching sausage is *local*, so it is scalable to arbitrary large datasets. The only assumption is that the sampling density is limited to guarantee d_{max} , thus constraints the confident triangles. As an extension, a *meta-bucket* (multiple layers of buckets) can be exploited to handle huge datasets containing billions of points. Also, the meshing of surfaces inside each bucket can be run in parallel, *e.g.*, on multi-core architectures.

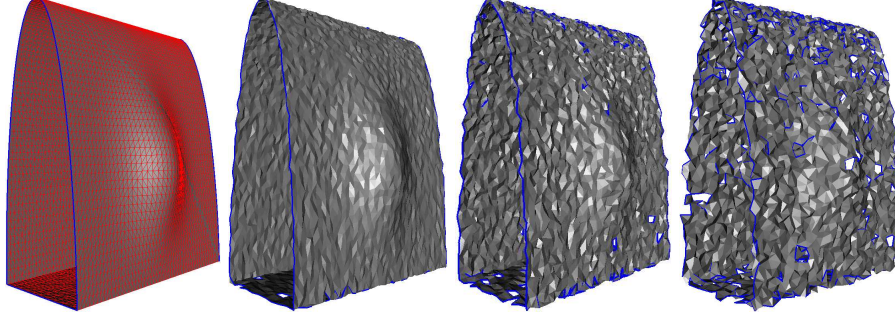


Fig. 26. This figure shows how our method performs on inputs (5,701 points) under perturbation *w.r.t.* d_{med} ; Max. displacement $d_n = n \times d_{med}$, where n is a percentage; each point coordinate, x, y, z , is perturbed by a factor $h = rand \times d_n$, where $rand$ is a random number between $[-0.5, 0.5]$. Left to Right: original dataset (8,014 faces); 50% of noise (11,196 faces); 100% (11,101 faces); 150% (11,018 faces). Topological quality is initially reasonable and then degrades as the noise extent increases.

Models	# of points	Time (sec)
sheep	22,619	6.22
Stanford bunny	35,947	10.97
bunny scan 1	40,256	10.297
igea (non-uniform)	75,545	26.56
sapho	120,965	33.6
alison (whole body)	121,723	40.954
igea	134,345	50.296
ball joint	137,062	52.734
john (whole body)	146,614	50.906
elephant	206,618	72.953
hip	265,081	105.906
boat	291,117	102.844
john	321,239	132.531

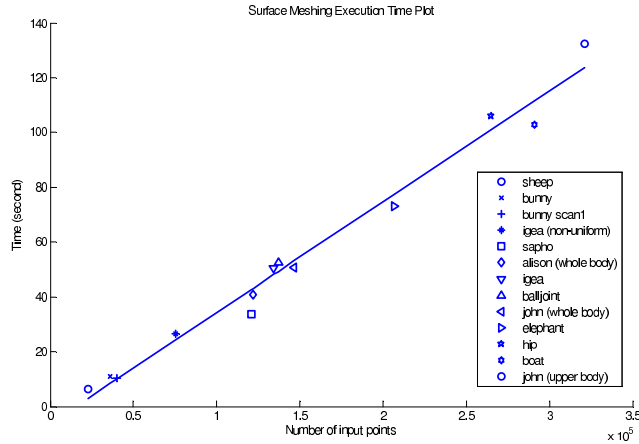


Fig. 27. Computation time plot of the proposed surface meshing implementation on a PC with a Pentium 4 3G MHz CPU and 2GB RAM.

9980091. The authors gratefully thank Gabriel Taubin for helpful early discussions. Thanks are also forwarded to the Stanford graphics lab, Cyberware, and the Max Planck Institute for Informatics (MPII), for providing test scan data and some of the 3D models.

References

- [1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete and Comp. Geometry*, 22:481–504, 1999.
- [2] N. Amenta, S. Choi, T. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and its Applications*, 12(1–2):125–1141, 2002.
- [3] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the

- medial axis transform. *International Journal of Computational Geometry and its Applications*, 19(2-3):127–153, 2001.
- [4] D. Attali. r -regular shape reconstruction from unorganized points. *Computational Geometry: Theory and Applications*, 10(4):239–247, 1998.
 - [5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. on Vis. and Comp. Graph.*, 5(4):349–359, 1999.
 - [6] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.
 - [7] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. on Graphics*, 3(4):266–286, 1984.
 - [8] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Symp. on Geometry Proc.*, pages 69–78, 2007.
 - [9] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell. Smooth surface reconstruction from noisy range data. In *Proc. ACM GRAPHITE*, pages 119–126, 2003.
 - [10] M.-C. Chang and B. B. Kimia. Regularizing 3D medial axis using medial scaffold transforms. In *Proc. IEEE CVPR, accepted*, 2008.
 - [11] M.-C. Chang, F. F. Leymarie, and B. B. Kimia. Surface reconstruction from point clouds by transforming the medial scaffold. In *Proc. 3-D Digital Imaging and Modeling (3DIM)*, pages 13–20, 2007.
 - [12] D. Cohen-Steiner and F. Da. A greedy Delaunay-based surface reconstruction algorithm. *Visual Computer*, 20:4–16, 2004.
 - [13] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM SIGGRAPH*, pages 303–312, August 1996.
 - [14] T. K. Dey, J. Giesen, and J. Hudson. Delaunay based shape reconstruction from large data. In *Symp. in Parallel and Large Data Vis. & Grap.*, pages 19–27. IEEE, 2001.
 - [15] T. K. Dey, J. Giesen, E. A. Ramos, and B. Sadri. Critical points of the distance to an epsilon-sampling of a surface and flow-complex-based surface reconstruction. In *Proc. Symp. on Comp. Geom.*, pages 218–227, 2005.
 - [16] T. K. Dey and S. Goswami. Tight cocone: A water-tight surface reconstructor. *Journal of Comp. and Info. Sci. in Engin.*, 3(4):302–307, 2003.
 - [17] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry*, 35:124–141, 2006.
 - [18] J. Esteve, P. Brunet, and A. Vinacua. Approximation of a cloud of points by shrinking a discrete membrane. *Computer Graphics Forum*, 24(4):791–807, 2005.
 - [19] P. J. Giblin and B. B. Kimia. A formal classification of 3D medial axis points and their local geometry. *PAMI*, 26(2):238–251, 2004.

- [20] P. J. Giblin and B. B. Kimia. Transitions of the 3D medial axis under a one-parameter family of deformations. *PAMI*, accepted, 2007.
- [21] M. Gopi and S. Krishnan. A fast and efficient projection based approach for surface reconstruction. In *SIBGRAPI*, pages 179–186, 2002.
- [22] H. Hiyoshi. Greedy beta-skeleton in three dimensions. In *Proc. Symp. Voronoi diagrams in Sci. and Engin.*, pages 101–109, 2007.
- [23] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH*, pages 71–78, August 1992.
- [24] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. *Symposium on Geometry Processing*, pages 41–50, June 2006.
- [25] J. Huang and C. H. Menq. Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology. *Computer-Aided Design*, 34(2):149–165, 2002.
- [26] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *Symposium on Geometry Processing*, pages 61–70, June 2006.
- [27] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3):189–224, 1995.
- [28] N. Kojekine, V. Savchenko, and I. Hagiwara. Surface reconstruction based on compactly supported radial basis functions. In *Geometric modeling: techniques, applications, systems and tools*, pages 218–231. Kluwer, 2004.
- [29] C.-C. Kuo and H.-T. Yau. A new combinatorial approach to surface reconstruction with sharp features. *IEEE Trans. on Vis. and Comp. Graph.*, 12(1):73–82, 2006.
- [30] F. F. Leymarie and B. B. Kimia. The medial scaffold of 3D unorganized point clouds. *IEEE Trans. on PAMI*, 29(2):313–330, 2007.
- [31] C. T. Lim, G. M. Turkiyyah, M. A. Ganter, and D. W. Storti. Implicit reconstruction of solids from cloud point sets. In *ACM Symp. on SMA*, pages 393–402, 1995.
- [32] H.-W. Lin, C.-L. Tai, and G.-J. Wang. A mesh reconstruction algorithm driven by intrinsic property of point cloud. *Computer-Aided Design*, 36(1):1–9, 2004.
- [33] B. Mederos, N. Amenta, L. Velho, and L. H. de Figueiredo. Surface reconstruction from noisy point clouds. *Symposium on Geometry Processing*, pages 53–62, 2005.
- [34] S. Muraki. Volumetric shape description of range data using “blobby model”. *SIGGRAPH*, 25(4):227–235, 1991.
- [35] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, July 2003.
- [36] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Multi-scale and adaptive CS-RBFs for shape reconstruction from cloud of points. In *Adv. in Multires. for Geom. Model.*, pages 143–154. Springer, 2005.

- [37] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graphical Models*, 68:15–24, 2006.
- [38] S. Petitjean and E. Boyer. Regular and non-regular point sets: Properties and reconstruction. *Comp. Geometry: Theory and App.*, 19(2-3):101–126, 2001.
- [39] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. Reconstruction with Voronoi centered radial basis functions. In *Symp. on Geom. Proc.*, pages 51–60, 2006.
- [40] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. on PAMI*, 26:551–571, May 2004.
- [41] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or. Competing fronts surface reconstruction. *Computer Graphics Forum*, 25(3):389–398, June 2006.
- [42] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. on PAMI*, 16(3):287–303, 1994.
- [43] H. Tek and B. B. Kimia. Boundary smoothing via symmetry transforms. *Journal of Math. Imaging and Vision*, 14(3):211–223, May 2001.
- [44] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE PAMI*, 13(7):703–714, July 1991.
- [45] I. Tobor, P. Reuter, and C. Schlick. Reconstructing multi-scale variational partition of unity implicit surfaces with attributes. *Graphical Models*, 68:25–41, January 2006.
- [46] G. Turk and J. F. O’Brien. Modelling with implicit surfaces that interpolate. *ACM Trans. on Graphics*, 21(4):855–873, October 2002.
- [47] J. Wang, M. M. Oliveira, and A. E. Kaufman. Reconstructing manifold and non-manifold surfaces from point clouds. In *Proc. IEEE Vis.*, pages 415–422, 2005.
- [48] J. Wang, M. M. Oliveira, H. Xie, and A. E. Kaufman. Surface reconstruction using oriented charges. In *Computer Graphics International 2005*, pages 122–128, 2005.
- [49] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *IEEE Workshop on VLSM*, pages 194–202, 2001.
- [50] M. Zilske, H. Lamecker, and S. Zachow. Adaptive remeshing of non-manifold surfaces. In *Prof. EUROGRAPHICS*, pages 393–402, 2008.