3D Surface Reconstruction from Unorganized Sparse Cross Sections

Ojaswa Sharma* Nidhi Agarwal†

Indraprastha Institute of Information Technology Delhi, India

ABSTRACT

In this paper, we propose an algorithm for closed and smooth 3D surface reconstruction from unorganized planar cross sections. We address the problem in its full generality, and show its effectiveness on sparse set of cutting planes. Our algorithm is based on the construction of a globally consistent signed distance function over the cutting planes. It uses a split-and-merge approach utilising Hermite mean-value interpolation for triangular meshes. This work improvises on recent approaches by providing a simplified construction that avoids need for post-processing to smooth the reconstructed object boundary. We provide results of reconstruction and its comparison with other algorithms.

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations;Geometric algorithms, languages, and systems;

1 INTRODUCTION

Reconstructing an object from a finite set of planar cross sections is an interesting variant of the much studied problem of surface reconstruction from a point cloud [5]. The difference between the two problems is due to the nature of intersections. Planar cross sections provide dense and very localised object information only on the respective cutting planes. A sparse set of such cross sections provides little information about the global topology of the underlying object. As a consequence, multiple topological configurations are possible for a given set of cross sections, as discussed by Sidlesky et al. [20]. Amini et al. [1] call it *geometric tomography* and provide an in-depth study of theoretical guarantees on reconstruction and sampling conditions. The problem has recently received some attention for potential uses in medical reconstructions such as in ultrasound, where the acoustic beams from the probe form a set of planar cross sections penetrating the subject non-invasively. 3D reconstruction of organs are widely considered to be an important diagnostic aid in the medical world [14, 18]. Other application domains include underwater acoustic reconstructions in fisheries research and terrain modelling.

The specific case of reconstruction from parallel cross sections has been extensively studied in literature (see [3, 7, 9]), and can be considered as a solved problem. The problem of reconstruction from unorganized cross sections is relatively new and has been considered in both 2D and 3D settings [4, 6, 14, 15, 18, 20]. In most of the reconstruction algorithms, it is typically assumed that the provided input is already segmented into two or more regions that delimit the "inside" and "outside" regions of objects on each cross section. The goal is to create a compact manifold (curve or surface) that passes through all the intersections, while consistently preserving the inside and outside information.

The algorithm described here solves the problem in its most general setting, with no constraints on the objective. We follow a split-and-merge approach to solve the problem in 3D. Our reconstruction is continuous and smooth that results from a simple and robust algorithm.

*e-mail:ojaswa@iiitd.ac.in

†e-mail:nidhi1309@iiitd.ac.in

Graphics Interface Conference 2016
1-3 June, Victoria, British Columbia, Canada

2 PREVIOUS WORK

Sidlesky et al. [20] analyzed topological properties of solution to this reconstruction problem in the plane. The authors observe that a line not intersecting the object does not contribute to the reconstruction. Their algorithm enumerates all possible reconstructions that satisfy the interpolation and topological equivalence with the given input. Due to a large number of possible reconstructions, complexity of their algorithm is exponential in nature. There may be cases for which several reconstructions are topologically valid for a unique set of given cross sections.

Memari and Boissonnat [15] used the Delaunay triangulation for reconstruction. Input to the algorithm is a set of intersecting planes along with their intersections with the object. The authors consider a partitioning of space by all cutting planes in the space, and extract a closed triangular mesh within each partitioned cell. The mesh serves as an approximation of the object from its intersections with the boundary of cell. To complete the reconstruction, all the reconstructed segments within each cell are glued together.

Similar approaches based on the Voronoi diagram are suggested by Liu et al. [14], and Memari and Boissonnat [16]. Liu et al. construct medial axis of each partitioned cell (a convex polyhedron) and approximate the reconstruction by lifting the cross sections on the medial axis. Memari and Boissonnat, on the other hand, use the Voronoi diagram of the cross sections within each partitioned cell. The authors also provide rigorous proof of their reconstruction, where they propose a topological reconstruction method based on the Delaunay triangulation of the set of segments of intersecting lines. The authors claim an improvement over the method by Liu et al. by producing reconstructions that are not topologically affected by lines that do not intersect with the object under consideration. Their reconstruction boundary, however, is a piecewise linear approximation of the boundary of the original object and lacks smoothness. The most recent work on a topologically motivated reconstruction algorithm is by Zou et al. [22] where the authors provide topological control during surface reconstruction from a set of planar cross sections. The algorithm does so by performing a topological search using a divide-and-conquer optimization process to recover a surface that matches a user defined genus. The authors present the algorithm in context of biomedical image analysis and use the underlying 3D image for surface reconstruction along with the cross-sections. We distinguish the current line of work by limiting the input only to cross-sectional contours. When the underlying 3D image is available, a number of smooth surface extraction approaches can be effectively utilized (e.g. volume segmentation using level-set methods, and graph-cut based segmentation). A discussion of these is outside the scope of this paper.

The work of Sharma and Anton [18] suggests a different approach to reconstruction via continuous deformations. Generalizing on homotopy based reconstruction from organised cross sections, the authors perform reconstruction in an implicit setting by formulating homotopies in each partitioned cell. The authors define smooth functions (piecewise quadratic) along every cutting line. We note that this reconstruction algorithm suffers from two main problems:

- the piecewise quadratic functions associated with each cutting lines are a good choice locally, but are inconsistent globally. This choice results in multiple values of the function at points of intersection of two or more cutting lines.

- even though the resulting curve is very smooth, geometrically the curve is not simple and consists of very high curvature near the cutting lines.

The algorithm, though topologically motivated, does not result in a geometrically fair boundary.

The solutions described above have assumed that all input slices are complete, there are no missing data within the cross sections and the slices are segmented correctly. In practical cases, there may be some uncertain regions with incomplete information that are not reliably segmented. The algorithms suggested in [4, 6], attempt to solve the partial-slice situation where data may be missing in portions of the sections. Barequet and Vaxman [4] reconstruct the original object by interpolating simultaneously all the cross sections. Their algorithm attempts to minimize the surface area of the reconstruction by using an offset distance function in order to locally decide which contour features are to bind. Smoothing is performed as a post-processing operation to clean-up the resulting surface.

In this work, we present a simple and robust algorithm to generate a continuous and smooth object boundary from a set of planar cross section in 3D. The presented algorithm works on a sparse set of cross sections and does not perform any post processing. Our main contribution is in formulating a signed distance based globally consistent function on the cutting planes. Our approach is based on mean-value Hermite interpolation in polyhedral partitions.

3 Problem description

The problem of object reconstruction from unorganized planar cross sections can be formally described as follows. Given a set of cutting planes $\Pi = \{\pi_i, i \in [0, n-1]\}$ in $\mathbb{R}^3$, and intersections $\mathcal{S} = \{\mathcal{O} \cap \pi_i, i \in [0, n-1]\}$ of the planes with an object $\mathcal{O}$, compute a continuous and smooth reconstruction $\mathcal{R}$ similar to $\mathcal{O}$. The similarity of $\mathcal{R}$ to $\mathcal{O}$ implies that $\mathcal{R} \cap \Pi = \mathcal{S}$, and the boundary $\partial\mathcal{R}$ to be a smooth surface.

Intersections $\mathcal{S}$ on any plane π_i may have multiple disjoint components. Intersections from any two different cutting planes can intersect with each other. Figure 1 shows cross sections superimposed on the object.

As mentioned in [16], the reconstruction $\mathcal{R}$ is a manifold with boundary similar to $\mathcal{O}$ satisfying the constraint that $\mathcal{S} = \mathcal{R} \cap \Pi$. It is also desirable that $\mathcal{R}$ be topologically similar to $\mathcal{O}$.

4 Reconstruction algorithm

Starting with a given set of cross sections $\mathcal{S}$, we restrict reconstruction to domain $\Omega \subseteq \mathbb{R}^3$ that encloses all cross sections. We address the shortcomings of [18] (discussed in section 2) by constructing a globally consistent function using the signed distance function on the cutting planes. The use of signed distance also makes the algorithm robust while keeping the resulting reconstruction simple.

We follow a split-and-merge approach to reconstruction. From the arrangement of cutting planes, we perform a partitioning of Ω into a set of convex polyhedra $\mathcal{H}$. A distance function can be defined at any point in Ω considering the cross section boundaries as generators. This distance function serves as a basis to construct a globally consistent signed distance function (SDF) for any point on the cutting planes. This is possible since every cross section contains information about inside and outside regions within the respective cutting plane. Constrained Delaunay Triangulation (CDT) is computed on every face of polyhedra in $\mathcal{H}$. The SDF is evaluated at every vertex of the triangulations in $\mathcal{H}$; by construction, the SDF evaluates to zero at vertices of the cross-section edges. The reconstruction algorithm then interpolates the signed distance value inside each polyhedron in a Hermite fashion to ensure C^1 continuity across partitions. A reconstruction surface is then recovered as the zero level set of the computed field. Our reconstruction algorithm is summarised in Algorithm 1.

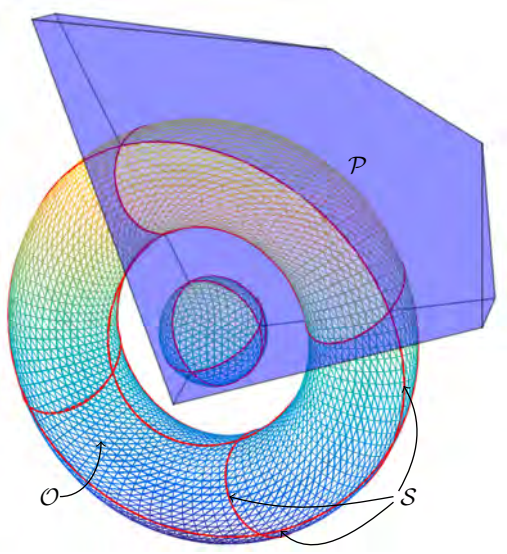


Figure 1: Planar cross sections of an object and a convex polyhedral partition induced by cutting planes.

Input: Intersections $\mathcal{S}$ on Π
Output: $\mathcal{R}$
Compute polyhedral partitioning $\mathcal{H}$ of Ω from Π
foreach polyhedron $\mathcal{P} \in \mathcal{H}$ **do**
  foreach polygonal face p of $\mathcal{P}$ **do**
    $\mathcal{S}' = \text{Clip}(\mathcal{S}, p)$
    Compute CDT (add constraint edges from $\mathcal{S}'$)
    Evaluate SDF at each vertex of CDT
  end
  foreach point $\mathbf{x} \in \mathcal{P}$ **do**
    Compute mean-value coordinates λ of $\mathbf{x}$
    Compute $\mathcal{F}(\mathbf{x})$ using Hermite interpolation
  end
end
$\mathcal{R} \leftarrow \ker \mathcal{F}$

Algorithm 1: The reconstruction algorithm.

4.1 Domain partitioning and boundary triangulation

We consider a bounding box around the domain enclosing the set of cross sections. The arrangement of cutting planes naturally partition the bounding box into a set $\mathcal{H}$ of convex polyhedra. Starting with a cuboid as a convex polyhedron, we compute the partitioning by successively dividing all polyhedra by each cutting plane and discarding degenerate polyhedra. One can think of this partitioning process as a binary tree where each node is a convex polyhedron and its two children (possibly degenerate or null) resulting from a split by a cutting plane. In fact, each level in this binary tree indicates a split by one of the cutting planes. Our computation does not become exponential since a majority of the nodes are not valid polyhedra.

For surface reconstruction within each polyhedron, we perform a Constrained Delaunay Triangulation of each planar face. The constraint edges belong to the boundary curves of cross sections of the respective cutting plane clipped by the face polygon (see Figure 2). By definition, these constraint edges sample the actual object surface.

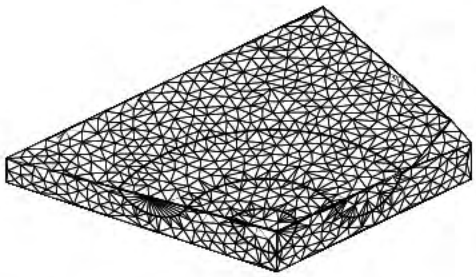


Figure 2: Constrained Delaunay triangulation on planar faces of a polyhedron.

4.2 Globally consistent function over polyhedron faces

The reconstruction algorithms suggested in [18, 6] work in an implicit setting. An implicit field is constructed by propagating function values from boundaries of polyhedra that partition the space. The surface of interest in such a setting is usually defined as the zero level set of the implicit function. Since the cross sections provide inside and outside information, Bermano et al. in [6] used the characteristic function to define an implicit function on any cutting plane π_i with cross-section $s_i \in \mathcal{S}$ as

$$f_\chi(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in s_i / \partial s_i \\ 0, & \mathbf{x} \in \partial s_i \\ -1, & \mathbf{x} \notin s_i. \end{cases} \quad (1)$$

According to this definition, $f_\chi(\mathbf{x})$ is discontinuous at the cross section boundaries; which will result in discontinuous implicit field at polyhedron faces. The resulting reconstruction thus suffers from ripples on the surface. To alleviate this problem, the authors in [6] suggest performing bi-Laplacian smoothing as post-processing. Sharma and Anton [18] define smooth C^1 functions along linear cross sections (in a 2D setting) using piecewise quadratic polynomials with zeros chosen to be the points in ∂s_i (here π_i is a cutting line). Even though these functions are continuous and smooth along their respective cutting lines, problems arise when two or more cutting lines intersect and the point of intersection has distinct function values from different quadratic functions defined on these cutting lines. This can lead to a discontinuous implicit field that is undefined at the points of intersections of all cross sections. Such a problem would also arise in a 3D setting when a suitable quadratic function is defined locally for a cutting plane.

In this work, we propose to use a globally consistent and continuous function defined over the arrangement of cutting planes. We build such a function using the distance field of cross section boundaries $\partial\mathcal{S}$. This function can then be evaluated at points on polyhedron faces.

4.2.1 Computing signed distance function

The set of points in $\partial\mathcal{S}$ refer to boundaries of the cross sections, i.e., points where the boundary $\partial\mathcal{O}$ of the original object intersects with the set of cutting planes Π. The distance function computes the distance from any point $\mathbf{x}$ to the closest point on the set $\partial\mathcal{S}$

$$\text{dist}_{\partial\mathcal{S}}(\mathbf{x}) = \inf_{\mathbf{p}\in\partial\mathcal{S}}(||\mathbf{p}-\mathbf{x}||).$$

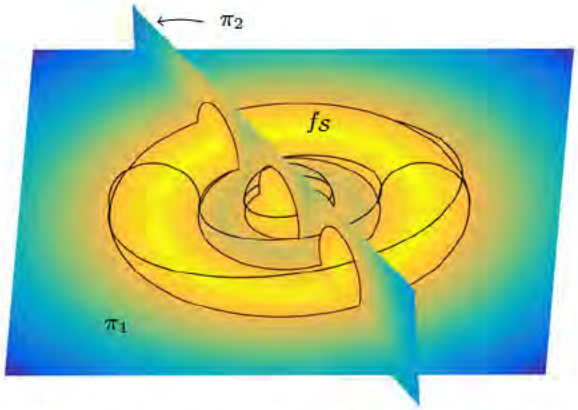


Figure 3: Globally consistent signed distance field over cutting planes.

A global distance field $\text{dist}_{\partial\mathcal{S}}$ can be computed at all points on planes in Π. At any plane π_i, $\text{dist}_{\partial\mathcal{S}}$ can be converted into a signed distance field $f_\mathcal{S}$ using the inside and outside information from s_i.

$$f_\mathcal{S}(\mathbf{x}) = \text{dist}_{\partial\mathcal{S}}(\mathbf{x}) f_\chi(\mathbf{x}). \quad (2)$$

This definition of $f_\mathcal{S}$ is consistent over the domain and is continuous everywhere since the signed distance function will agree at the intersection points of $\mathcal{S}$ (see Figure 3). We compute $f_\mathcal{S}$ at mesh vertices of all polyhedron faces. The global signed distance function computed at the boundary of a polyhedron is then propagated inside using Hermite mean-value interpolation.

4.3 Hermite interpolation using Mean-value Coordinates

Mean-value Coordinates (MVCs) [8] provide a simple and robust transfinite interpolation of a function defined on the boundary of a planar domain. At any point $\mathbf{x}$ inside an open, bounded and convex region Ψ, the mean-value boundary integral has a solution

$$g(\mathbf{x}) = \int_0^{2\pi} \frac{f(\mathbf{p}(\mathbf{x},\theta))}{\rho(\mathbf{x},\theta)} d\theta \Big/ \int_0^{2\pi} \frac{1}{\rho(\mathbf{x},\theta)} d\theta \,,$$

where f is a function defined on $\partial\Psi$, $\mathbf{p}(\mathbf{x},\theta)$ is the point of intersection of $\partial\Psi$ with a ray starting at point $\mathbf{x}$ and at an angle θ with the x-axis, and $\rho(\mathbf{x},\theta)$ is the distance $||\mathbf{p}(\mathbf{x},\theta)-\mathbf{x}||$. In a similar fashion, MVCs can be applied to non-convex domains. For the special case when Ψ is a triangular mesh, Ju et al. [11] generalize MVCs that are continuous everywhere and smooth inside the mesh. In our case, polyhedron faces are triangulated to obtain a mesh. In essence, we discretise the signed distance function $f_\mathcal{S}$ at polyhedron boundary for mean-value interpolation.

We observe that the function $f_\mathcal{S}$ interpolated with MVCs results in a smooth surface everywhere in Ω except at the cutting planes (i.e., at the boundary of any two polyhedra, as can be seen in Figure 4(b)). This is attributed to the fact that the gradient of the MVC's at points on a common face between any two polyhedra is not same (since the polyhedra may have arbitrary convex shapes).

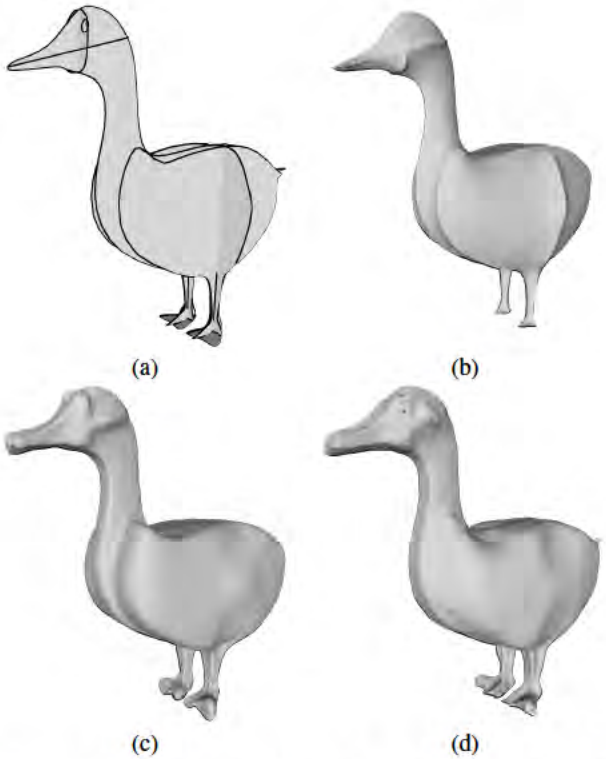


Figure 4: Reconstruction with MVC. (a) Set of cross sections (b) Reconstruction without Hermite interpolation (c) Reconstruction with Hermite interpolation using approximate normals (d) Reconstruction with hermite interpolation using actual normals.

Given normals of object's surface $\partial\mathcal{O}$ at cross-section points $\partial\mathcal{S}$, a Hermite interpolation will ensure that the gradient of $\mathcal{R}$ is consistent at polyhedron boundaries. Dyken and Floater in [8] suggest Hermite mean-value interpolation for a region with parametric boundary. We use the higher order barycentric coordinates of Langer and Seidel [12], where any barycentric coordinate can be lifted to a higher order and used for Hermite interpolation. In particular, using the first two terms of the Taylor series, the interpolation function becomes

$$\hat{f}(\mathbf{x}) = \sum_{i=0}^{k-1} \hat{\lambda}_i (f_i + \nabla f_i \cdot (\mathbf{x} - \mathbf{v}_i)), \tag{3}$$

where f_i, and ∇f_i are function values, and function gradients at the vertices respectively, and $\hat{\lambda}_i = g \circ \lambda_i$ are the higher order barycentric coordinates with $g : \mathbb{R} \mapsto \mathbb{R}$ being a smooth piecewise polynomial function as suggested in [12].

4.3.1 Computation of normals

In many situations, normals at the cross sections are not available. In our case, exact normals can be computed only at the intersection points of any two cross sections (given by cross product of the two tangents at the intersection point). We approximate normals at other points on the cross-section boundary by a linear interpolation of normals (followed by normalization) at successive pairs of such intersection points along the cross section curve as shown in Figure 5(a). In the case when such intersection points do not exist, we compute an in-plane approximation of the normals using finite differencing at points of cross section curves in a particular cutting plane (see Figure 5(b)). Figure 4(c) shows a Hermite reconstruction using approximate normals, while Figure 4(d) shows the same with exact normals. It must be noted that the staircase effect is no more seen in a reconstruction with exact normals. In our results below, we show reconstructions with approximate normals since exact normals may not be available in general.

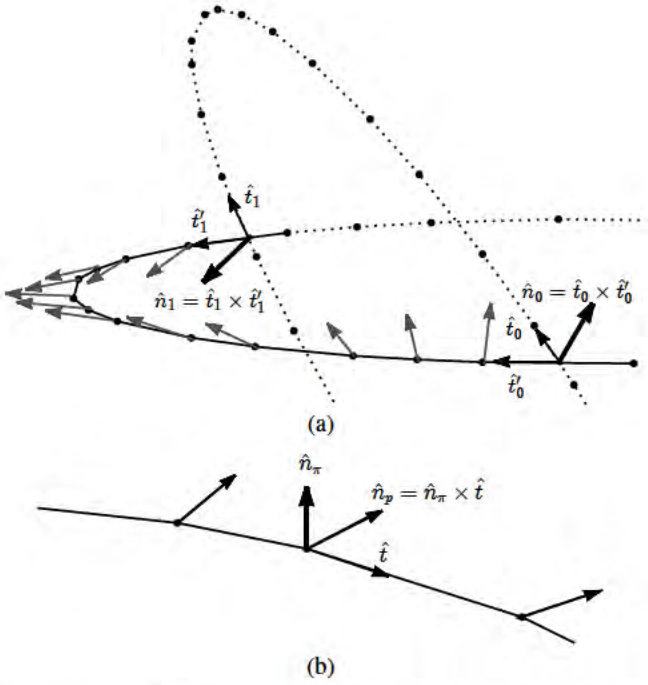


Figure 5: Normal computation. (a) Using interpolation of exact normals at points of intersection, (b) Using an in-plane approximation.

5 Results and Comparison

We compare results of our method (denoted by $\mathcal{R}_{SDT}$ in Table 2) with other reconstruction algorithms by Sharma and Anton [18], and Liu et al. [14]. For comparisons, we apply the homotopy based 2D algorithm described in [18] to shapes in 3D (denoted by $\mathcal{R}_{Hom}$ in Table 2) using the original code from the authors. The original 2D algorithm defines a piecewise quadratic function along any cutting line that is derived from the zeros along the cutting line. We defined a continuous function on any cutting plane to be the signed distance function generated from the respective cross section curves. Note that this function is different from our globally consistent function f_S and exhibits a local behaviour similar to that in [18]. Reconstruction algorithm of Liu et al. [14] is based on the medial axis of partitions of space by cutting planes. We evaluate this reconstruction algorithm (denoted by $\mathcal{R}_{MA}$ in Table 2) using the implementation provided by the authors, and compare the results with our method. We note here that the algorithm based on medial axis performs mesh smoothing after the projection step, while we do not perform any smoothing as a post-processing in our results.

The following results are generated on synthetic cross sections obtained by slicing a 3D mesh model with a set of cutting planes. While generating these cross sections, no particular arrangements of cross sections were considered, except that we tried to sample all prominent features of the 3D shapes. If this is not done, then it is not possible to faithfully recover the sliced object as a consequence of the Nyquist-Shannon sampling theorem applied to shapes. A de-

tailed analysis of reconstruction in geometric tomography is given by Amini et al. [1].

We use the Geom3D package [13] to partition the volume into a set of polyhedra. The faces of the polyhedra are triangulated using the Triangle library [19]. The 3D models used here are obtained from INRIA GAMMA 3D mesh research database [10], and Large geometric models archive [21]. Table 1 shows some of the important statistics about the input cross sections and their reconstructions.

Model	#Sections (#vertices)	#Polytopes	#Face triangles	#Voxels
Femur	7 (397)	33	2491	33000
Chess	9 (257)	10	23109	390600
Hand	16 (2216)	258	664837	227700
Dragon	13 (3217)	223	189624	308000

Table 1: Statistics about input data and reconstruction.

In order to compare performance of our algorithm with that of other methods, we compute three accuracy measures. Along with simple ratios of volume and surface areas, we use Housdorff distance based measure as discussed in [18]. The original 3D models are used as ground truth for evaluation. The ratio of volumes is computed as $\hat{\mathcal{V}} = \mathcal{V}_{\mathcal{R}}/\mathcal{V}_{\mathcal{O}}$ where $\mathcal{V}_{\mathcal{R}}$ and $\mathcal{V}_{\mathcal{O}}$ are the volumes of $\mathcal{R}$ and $\mathcal{O}$ respectively. In a similar fashion, ratio of surface areas is computed as $\hat{\mathcal{A}} = \mathcal{A}_{\mathcal{R}}/\mathcal{A}_{\mathcal{O}}$, where $\mathcal{A}_{\mathcal{R}}$ and $\mathcal{A}_{\mathcal{O}}$ are the surface areas of $\mathcal{R}$ and $\mathcal{O}$ respectively. A value of one for these ratios indicates a better reconstruction (although not necessarily always).

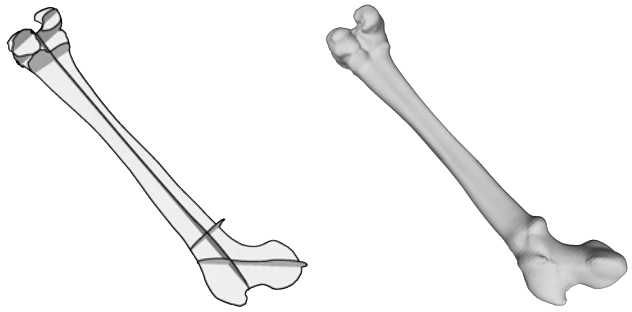
Another important measure is the Hausdorff distance [17, 2] that indicates how closely the points on boundaries of the two shapes match. The Housdorff distance between two shapes S and S' is given by

$$d_H(S, S') = \sup_{x \in S} \inf_{x' \in S'} d(x, x'),$$

where $d(\cdot, \cdot)$ is a distance metric. In our comparisons, we use the ratio $\hat{d}_H$ of the mean Hausdorff distance (with Euclidean distance metric) and the length of the bounding box diagonal. A lower value of $\hat{d}_H$ (close to zero) indicates a better reconstructed surface. Further, we also show per-point error of the reconstructed surface as distance measured from the corresponding point on the original surface mesh in Figure 10. The accompanying histograms show distribution of these errors, where the Y-axis represents error (or distance) values while the X-axis has the frequency.

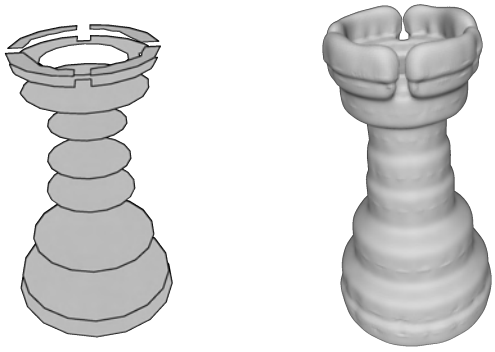
In reconstructions shown in 6, 7, 8, and Figures 9, it can be observed that the reconstructed surface is smooth everywhere and combines the cross sections correctly. The comparisons in Figure 10 and accuracy measures in Table 2 show that our reconstruction generates a fair surface matching closely with the original object, but results in slight bulges close to the cross section boundaries due to normal approximation. The homotopy based algorithm results in smooth surface patches with creases at intersection points due to inconsistencies in the function defined on the cutting planes. On the other hand, a reconstruction based on the medial axis is smooth everywhere, but usually results in very high deviations a a few places. Our method is more topologically stable, but with geometric error spread across on the surface.

Our first example of the Femur model shows reconstruction with as few as seven cross sections (see Figure 6). It can be seen that the reconstruction is free from noise and is smooth. In comparison with other reconstructions, we get better accuracy measures with our algorithm. Error histograms in Figure 10 for Femur shows higher errors for the homotopy based approach. While, medial axis based algorithm gives good results, it has a few regions of very high deviation from the actual surface. Our case also shows errors around such regions, but lower in magnitude.



(a) (b)

Figure 6: Reconstruction results with Femur. (a) A set of seven cross sections, (b) reconstructed surface.

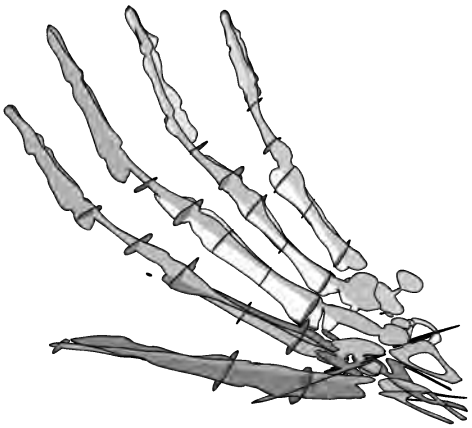


(a) (b)

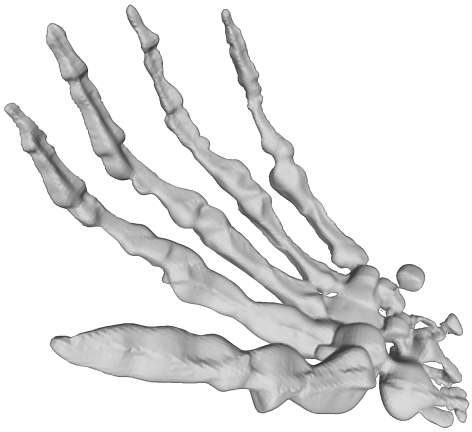
Figure 7: Reconstruction results with Rook. (a) A set of nine cross sections, (b) reconstructed surface.

For the Rook model we test our reconstruction algorithm on a set of parallel cross sections across the model. Figure 7 clearly shows that the choice of normals at cross section points leads to the reconstructed surface becoming orthogonal to the cutting planes. The volume and area ratios do not capture subtle variations in the surface. We notice that the reconstruction of Rook is low on these two ratios, while the Housdorff measure is able to accurately capture the variations in surface. The homotopy based reconstruction results in a degenerate surface in this case. Medial axis based reconstruction shows good results with a smooth surface overall. However, due to smoothing, the bottom of the reconstructed surface resembles a spherical section. The error histogram is mostly spread across with very few instances of very high error. The error histogram for our algorithm in case of Rook shows considerable high deviations from the base surface.

The next example of the Hand model (see Figure 8) shows reconstruction of a complex object with our algorithm. A geometric feature can only be reconstructed if it is sampled. Our algorithm also takes advantage from the absence of a signal (that indicates that there is no part of object present at that point). The homotopy based reconstruction shows defective surface for the mesh, which is

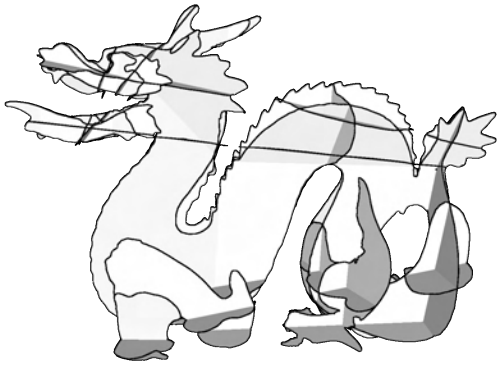


(a)

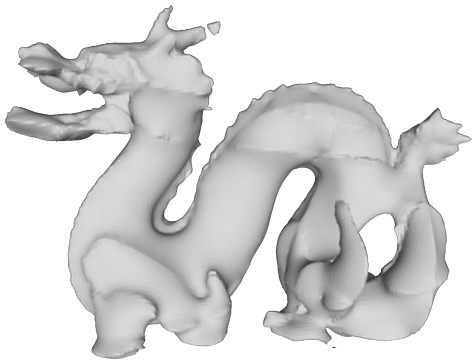


(b)

Figure 8: Reconstruction results with Hand. (a) A set of 16 cross sections, and (c) reconstructed surface.



(a)



(b)

Figure 9: Reconstruction results with Dragon. (a) A set of 13 cross sections, and (c) reconstructed surface.

also indicated by low scores in accuracy figures, and a flattened out error histogram. The medial axis based approach matches closely with ours in terms of the low frequency details of the surface. While this method results in higher deviations from the base surface, our results indicate more bulges at some places.

Finally, we show reconstruction of the Dragon model that has a complex topology (see Figure 9). The homotopy based algorithm shows comparable results in this case, but with prominent creases at the cross sections. The medial axis based reconstruction results in a good overall surface except that the topology of the resulting surface changed at some regions, for e.g. the hind leg is incorrectly joined with the tail. Also we observed very high deviations at the front leg. Our results seem to be topologically better but the error is more spread across in between cross sections.

Table 3 shows performance of our algorithm on the input cross sections. Cross sections of any model undergo various reconstruction stages enlisted in the table. The most time consuming stage of our reconstruction algorithm is Hermite MVC interpolation. Dense surface triangulation in our case increases the computational cost of mean-value coordinates. Further, we compute MVCs for every sampled voxel inside a polyhedron. Therefore, we perform this computation on the GPU using CUDA. A naive implementation of MVC interpolation would also have a high memory requirement of about $O(\sum_p v_p \cdot n_p)$ where p is the number of polyhedra, v_p is the number of voxels inside a polyhedron, and n_p is the number of vertices in the surface triangulation of the polyhedron. We optimised our MVC computation to have a very low memory footprint. Due to the nature of the computations, irregular reads from the GPU global memory are unavoidable. The reconstruction runtime depends on the size of the voxel grid; a high resolution grid will require more computations (see Table 1 for these numbers in our case). All of our computations use 64-bit floating-point precision (on both CPU and GPU) and are performed on a 2 GHz Intel Core i7 processor with 8 GB memory, and on an Nvidia Tesla K40 GPU.

The computational complexity of our algorithm depends on the number of cutting planes, triangulation density (for CDT), and grid resolution for MVC interpolation. We note that with increasing cutting planes, the number of polyhedra will increase, thus increasing the number of computations required. However, with increase in number of polyhedra, each one of those will cover fewer voxel grids for MVC interpolation. On average, we expect the computational cost of the algorithm to increase sub-exponentially, except for the MVC interpolation stage (which will increase much slower).

6 Conclusion

In this paper, we presented a simple and efficient algorithm for 3D object reconstruction from sparse set of unorganised planar cross sections. We illustrated a specialised construction of the signed distance function over the cutting planes that enables a consistent and smooth reconstruction. In its current form, the algorithm creates a smooth surface without performing mesh smoothing as an additional step but suffers from staircase effect (i.e, the generated

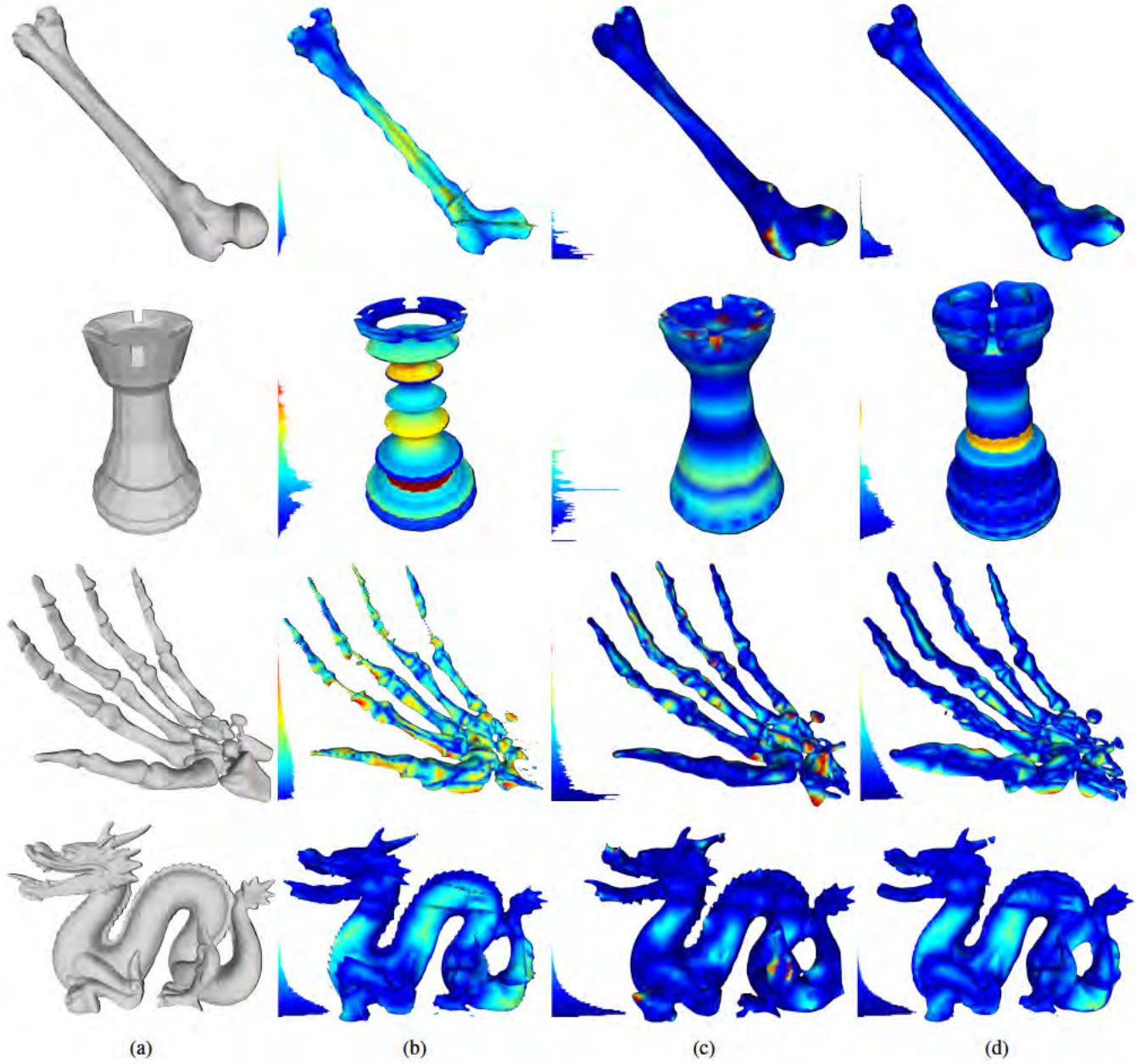


Figure 10: Comparison of results of reconstructions from different methods. (a) Original model, (b) homotopy based reconstruction [18], (c) medial axis based reconstruction [14], and (d) Signed Distance Field based reconstruction.

Model	$\mathcal{R}_{Hom}$ **Homotopy** [18]			$\mathcal{R}_{MA}$ **Liu** [14]			$\mathcal{R}_{SDT}$ **Signed Distance Field**		
	$\hat{\mathcal{V}}$	$\hat{\mathcal{A}}$	$\hat{d}_H$	$\hat{\mathcal{V}}$	$\hat{\mathcal{A}}$	$\hat{d}_H$	$\hat{\mathcal{V}}$	$\hat{\mathcal{A}}$	$\hat{d}_H$
Femur	0.3447	0.7501	0.0151	0.9134	0.9449	0.0033	0.9966	1.0116	0.0034
Rook	0.6189	1.0038	0.0192	1.1680	0.9893	0.0061	1.2243	1.1594	0.0064
Hand	0.6694	0.6980	0.0061	0.8961	0.8311	0.0032	1.0411	0.9802	0.0031
Dragon	0.8437	0.8686	0.0070	0.9999	0.9415	0.0039	0.8738	0.8181	0.0065

Table 2: Volume and surface area based accuracy measures for various reconstructions.

Model	Chain processing	Polyhedral partitioning	Section processing	Triangulation (2D CDT)	$\mathtt{dist}_{\partial S}$ (SDF)	Gradient	Hermite MVC (GPU)	Surface extraction	*Total*
Femur	0.114	0.310	0.992	1.684	0.212	0.152	0.010	0.045	*3.520*
Chess	0.068	0.129	0.058	0.559	0.589	0.063	2.455	0.103	*4.023*
Hand	0.634	2.922	20.032	15.062	24.871	1.012	7.034	0.065	*71.631*
Dragon	0.800	2.349	27.862	16.215	37.879	1.358	1.785	0.044	*88.293*

Table 3: Reconstruction runtimes in seconds.

surface being orthogonal to the cross sections) which can be further improved by a better choice of normals at points of cross sections. Such normals may be computed by an optimisation process. Another interesting problem in 3D is of reconstruction from linear cross sections (instead of planar), but the challenge with this is to consistently partition the space into a set of polyhedra. We plan to address these challenges as part of our future work.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] O. Amini, J. D. Boissonnat, and P. Memari. Geometric tomography with topological guarantees. *Discrete & Computational Geometry*, 50(4):821–856, 2013.

[2] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. MESH: Measuring errors between surfaces using the Hausdorff distance. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 705–708, 2002.

[3] C. L. Bajaj, E. J. Coyle, and K. N. Lin. Arbitrary topology shape reconstruction from planar cross sections. *Graphical models and image processing*, 58(6):524–543, 1996.

[4] G. Barequet and A. Vaxman. Reconstruction of multi-label domains from partial planar cross-sections. In *Computer Graphics Forum*, volume 28, pages 1327–1337. Wiley Online Library, 2009.

[5] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva. State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports*, volume 1, pages 161–185, 2014.

[6] A. Bermano, A. Vaxman, and C. Gotsman. Online reconstruction of 3D objects from arbitrary cross-sections. *ACM Transactions on Graphics (TOG)*, 30(5):113, 2011.

[7] J. D. Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 44(1):1–29, 1988.

[8] C. Dyken and M. S. Floater. Transfinite mean value interpolation. *Computer Aided Geometric Design*, 26(1):117–134, 2009.

[9] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from planar contours. *Communications of the ACM*, 20(10):693–702, 1977.

[10] P.-L. George. INRIA GAMMA 3D mesh research database. `https://www.rocq.inria.fr/gamma/download/`, 2013.

[11] T. Ju, S. Schaefer, and J. Warren. Mean value coordinates for closed triangular meshes. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 561–566. ACM, 2005.

[12] T. Langer and H. P. Seidel. Higher order barycentric coordinates. In *Computer Graphics Forum*, volume 27, pages 459–466. Wiley Online Library, 2008.

[13] D. Legland. Geom3D. `http://www.mathworks.com/matlabcentral/fileexchange/24484-geom3d`, 2012.

[14] L. Liu, C. Bajaj, J. O. Deasy, D. A. Low, and T. Ju. Surface reconstruction from non-parallel curve networks. In *Computer Graphics Forum*, volume 27, pages 155–163. Wiley Online Library, 2008.

[15] P. Memari and J. D. Boissonnat. Shape reconstruction from unorganized cross sections. In *5th Eurographics symposium on Geometry processing (SGP '07)*, pages 89–98. ACM, 2007.

[16] P. Memari and J. D. Boissonnat. Provably good 2D shape reconstruction from unorganized cross-sections. In *Computer Graphics Forum*, volume 27, pages 1403–1410. Wiley Online Library, 2008.

[17] J. Munkres. *Topology*. Prentice Hall, 1999.

[18] O. Sharma and F. Anton. Homotopic object reconstruction using natural neighbor barycentric coordinates. In *Transactions on Computational Science XIV*, pages 188–210. Springer, 2011.

[19] J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *Applied computational geometry towards geometric engineering*, pages 203–222. Springer, 1996.

[20] A. Sidlesky, G. Barequet, and C. Gotsman. Polygon reconstruction from line cross-sections. In *18th Canadian Conference on Computational Geometry (CCCG)*, Kingston, Ontario, 2006.

[21] G. Turk and B. Mullins. Large geometric models archive. `http://www.cc.gatech.edu/projects/large_models/`, 2003.

[22] M. Zou, M. Holloway, N. Carr, and T. Ju. Topology-constrained surface reconstruction from cross-sections. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.