Competing Fronts for Coarse–to–Fine Surface Reconstruction

ANDREI SHARF[1], THOMAS LEWINER[1,2], ARIEL SHAMIR[3], LEIF KOBBELT[4] AND DANIEL COHEN–OR[1]

[1] School of Computer Science — Tel Aviv University — Israel
[2] Departament of Matematics — Pontifícia Universidade Católica — Rio de Janeiro — Brazil
[3] Efi Arazi School of Computer Science — The Interdisciplinary Center — Herzliya — Israel
[4] Computer Graphics Group — RWTH — Aachen — Germany
www.cs.tau.ac.il/~asharf. www.mat.puc--rio.br/~tomlew. www.faculty.idc.ac.il/arik.
www--i8.informatik.rwth--aachen.de. www.cs.tau.ac.il/~dcor.

Abstract. We present a deformable model to reconstruct a surface from a point cloud. The model is based on an explicit mesh representation composed of multiple competing evolving fronts. These fronts adapt to the local feature size of the target shape in a coarse–to–fine manner. Hence, they approach towards the finer (local) features of the target shape only after the reconstruction of the coarse (global) features has been completed. This conservative approach leads to a better control and interpretation of the reconstructed topology. The use of an explicit representation for the deformable model guarantees water-tightness and simple tracking of topological events. Furthermore, the coarse–to–fine nature of reconstruction enables adaptive handling of non-homogenous sample density, including robustness to missing data in defected areas.

Keywords: ***Surface Reconstruction. Deformable Models.***

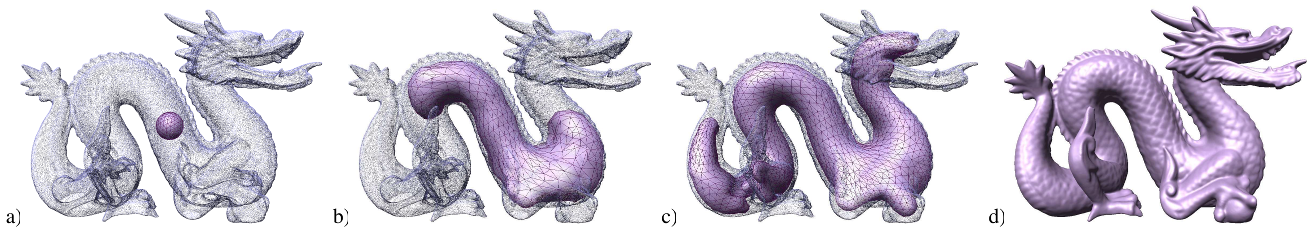

Figure 1: Growing a watertight, genus 0 mesh model inside the point cloud of the dragon (a-c). Attaching a handle between the body and the tail and projecting the model onto the point cloud (d).

1 Introduction

The generation of manifold polygonal representations of 3D shapes from point clouds is an elaborate task. Objects may have high genus and various scales of detail. Furthermore, many of today's model acquisition techniques generate imperfect results that cause difficulties in reconstruction. Scans of 3D objects are often very noisy, and may contain cracks and missing parts. However, in many cases the *a priori* assumption is that the reconstructed model should be a solid object with a certain topology and a watertight surface. These priors are *global* properties of the model.

In recent years *local* reconstruction techniques have been extensively used and proven to faithfully recover small features [1, 4, 16]. However, the local nature of these techniques may fail to recover the global structure of the shape, and cannot guarantee global properties such as watertightness especially for inhomogeneous sample density or missing data. In contrast, deformable models such as snakes [19, 23, 12] use a global structure to maintain watertightness and certain topology. However, previous deformable models adapt a global structure directly to the finest features size. This in turn, may lead to erroneous topological interpretation of the data.

In this paper we present a deformable model that uses an explicit evolving front technique for reconstruction of a 3D model. Our model uses a conservative approach while deforming, yielding a better interpretation of the topology of the reconstructed shape, and enabling better topology control to a certain extent. The model includes multiple competing evolving fronts at different locations that approach towards the finer (local) features of the target shape only after reconstructing the coarse (global) features. Hence, these

Preprint MAT. 04/06, communicated on February 6th, 2006 to the Department of Mathematics, Pontifícia Universidade Católica — Rio de Janeiro, Brazil. The corresponding work was published in the proceedings of the Eurographics 2006, Blackwell, 2006.

fronts adapt to the local feature size of the target shape in a *coarse–to–fine* manner. The fronts evolution is guided by a scalar-field representing the distance from the point set. The fronts always move in outward normal direction. Thus, they can move up-hill as well as down-hill, passing over local extrema of the guidance field and approaching sufficiently close the point-set.

The explicit representation of the deformable model is evolved in incremental steps which maintain some global invariants: it guarantees water-tightness and it allows simple tracking of topological events (such as handle attachment). While deforming, we apply mesh optimization operators which maintain a high mesh quality. Furthermore, the coarse–to–fine nature of the reconstruction enables dealing with non-homogenous sample density, including robustness to cracks and missing parts in defected areas of the input data (see for example Figures 1 and 2).

2 Related Work

There has been a substantial amount of work on surface reconstruction, in particular for the case of a reconstruction from point cloud. We briefly review the most related works with a focus on how they handle the topology and the scale of the details. It is out of the scope of this paper to cover this large research area. For a thorough review the reader is referred to [21].

Surface reconstruction can be formulated as a global minimization problem. For example, a surface approximating a point cloud can be defined as the zero–crossing of radial basis functions (RBF) [9, 25]. Such minimization techniques create implicit surfaces, which is then polygonized into a mesh. With such techniques it is difficult to control the topology. Another family of volumetric reconstruction techniques is based on Voronoï diagrams [3, 7, 13]. Under strong sampling conditions, these algorithms guarantee the validity of their results. However, these constraints are usually not met in practice and they fail to produce plausible results for sparse point clouds and non-uniform sampling. The method we present in this paper is explicitly designed to handle such delicate cases. In section 8 *Results* we compare our results with two representatives of the above mentioned methods.

To avoid expensive minimizations and volumetric structures, wrapping techniques [5, 26] iteratively grow a mesh surface which approximate the point cloud. The advantage of such techniques is the adaptation to the local features of the target shape. However, as opposed to our technique, they have difficulties to guarantee a watertight surface and lack awareness of shape topology. Local features can also be reconstructed by local projection mechanisms, such as the moving least squares (MLS) method [1]. In our work we use a simplified version of the MLS projection as means to reconstruct the finer details of the target shape.

A different approach to reconstruct the surface is to use deformable models. There are many types of deformable models, but they all derive from the well known notion of snake [19, 30, 18, 14]. Snakes are basically splines which minimize an external energy functional, while maintaining their smoothness by an internal tension term [19]. The minimum is formulated as the stable solution of a partial differential equation, and the spline is represented as a level set, which evolves by local integration steps [27, 6]. This evolution tends to be highly sensitive to the input data and to get stuck into local minima, although this last part of the problem is partially solved by globally smoothing the external energy snakes [30].

Extending snakes to 3D is a very delicate task mainly due to the difficulty to maintain surfaces as splines. To simplify the problem, the snake, or in general the deformable model, is typically discretized and represented by a set of voxels, embedded in discrete representation of an implicit function. Such implicit representation of the deformable model has been used for surface reconstruction [17, 15].

Unlike classical snakes, balloons evolve along the normal of their surface, with respect to an external scalar field. This formulation links them to mathematical morphology [2, 24]. For shape approximation, balloons were first used as a preprocessing step to get a good initial guess for snakes [11]. More recently, [14] proposed to adapt parametric balloons to reconstruct shapes of arbitrary topology. However, their balloon minimizes a single resolution energy functional similar to classical snake. Here, we use a multiscale approach, in a coarse to fine manner, where the model evolves with competing fronts. In addition, most previous work has dealt with regularly or well-sampled point clouds, while we focus on more challenging cases where the point cloud includes features at different scales and may contain holes and cracks (see Figure 2).

At large, the above surface reconstruction techniques can be classified into implicit/volumetric techniques vs. explicit/surface oriented techniques. Both come in hierarchical versions, but volumetric techniques use spatial hierarchies and surface techniques use geodesic scales. We combine both: the attraction field is computed with respect to a spatial hierarchy (adaptive octree) and the tension of our evolving fronts corresponds to a geodesic scale.

3 Overview

On an abstract level, the task of surface recovery by a deformable model is defined as follows: Given a target surface T and a deformable model D, evolve the shape of D such that the two–sided Hausdorff distance $H(T, D)$ is minimized. As observed in [10] it is sufficient to reduce the local Hausdorff distance below an ε fraction of the *local feature size* of T [3] because then the normal projection of D to T is injective. Hence, our algorithm brings the

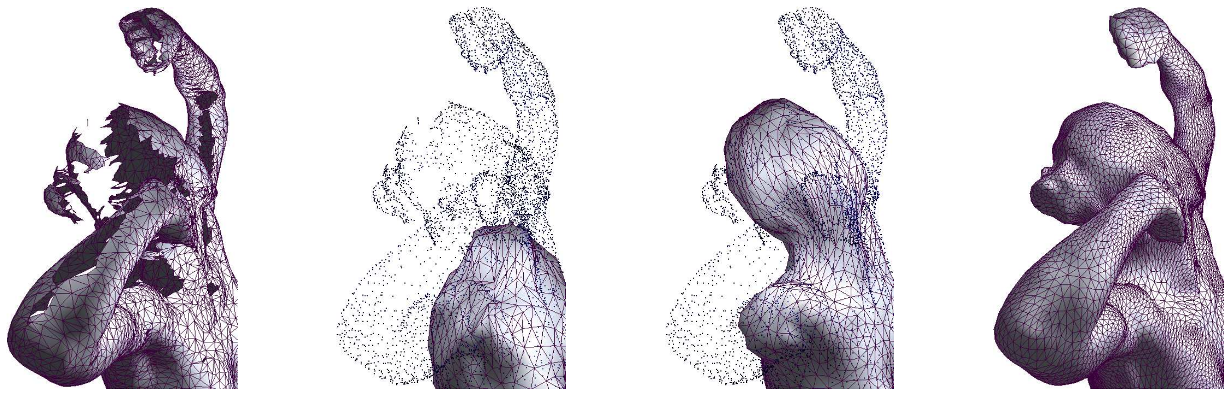

Figure 2: The Victoria model is sparse and contains many missing pieces. Our deformable model recovers the shape while completing missing part is a smooth manner.

deformable model D sufficiently close to the target surface T and then we apply a *moving least squares* projection [1], which eventually maps the vertices of D to T.

In practice, this approach faces several difficulties. First, the information we have regarding the target surface T, given by a set of sample points, is only partial. Hence, in sparsely sampled regions, the deformable model D should fill gaps, cracks and missing parts. Therefore, the one-sided distance from D to T has to be ignored there. On the other hand, D is usually implemented as a polygonal mesh with finite resolution. This implies that features of T which are smaller than the local edge length of D cannot be captured correctly. Such *unsatisfied* regions of T have to be detected by checking the one-sided distance from T to D and applying a local refinement to D, if necessary.

Our deformable model D is represented as an unstructured triangle mesh, initialized as a small sphere placed in the interior of the target point cloud T. Based on a simple volumetric distance map of T, the vertices of D move in outward normal direction towards T. A vertex stops moving when it reaches a local minimum of the distance function to T. This happens when the vertex is sufficiently close to a sample point of T or when D is about to pass through an undersampled region in the point cloud. By interleaving the mesh deformation with a mesh optimization scheme [8] we guarantee a high mesh quality in term of the triangle's aspect ratio.

As more and more vertices stop moving, the remaining regions of D at some point fall into several unconnected patches. These separate *fronts* denoted F continue to move independently. When D comes to a complete stop, it means that all vertices are sufficiently close to T or they belong to a part of D which spans a hole in the point cloud of T. However, what seems to be a hole in the point cloud could in fact be a small surface feature or a tunnel in T, which just cannot be resolved due to the current mesh resolution of D.

We can easily detect this by checking the one-sided distance from T to D. If there are *unsatisfied* samples in T, we re-activate a region in D which is closest to the most distant unsatisfied sample. To provide additional degrees of freedom, we locally refine the mesh D and let the new vertices continue to move in normal direction. This procedure is repeated until all samples of T are satisfied, i.e., their one-sided distance to D is below a certain threshold.

4 Algorithmic Features of the Fronts

Given a point cloud sampled from the boundary of a 3D object, the deformable model recovers the target shape with a watertight explicit mesh model. The evolution develops in an iterative manner by moving the vertices in outward normal direction (see Figure 3). The movement is guided by a volumetric attraction field imposed by the point set. Similar deformable models have been used in many previous works. However, defining an effective evolution of an explicit and discrete deformable model is usually a difficult task. This is especially challenging when the point cloud contains cracks and missing pieces and represents a shape which is not smooth and has a complex topology. The evolving model must penetrate delicate parts, preserve topological constraints, and fill missing data, while still maintaining high quality triangulation and preventing fold-overs.

Our model achieves this goal using the following elements: (i) working coarse–to–fine with competing fronts (ii) smoothing the fronts using least-square-meshes formulation, (iii) continuously using local mesh operators which preserve triangulation quality, (iv) allowing up-hill as well as down-hill movement of the fronts, and (v) applying local final fitting. Below we briefly describe these elements, and provide details in the following sections.

Competing fronts. A key element in our model is the coarse–to–fine evolvement of the fronts. Initially, all the

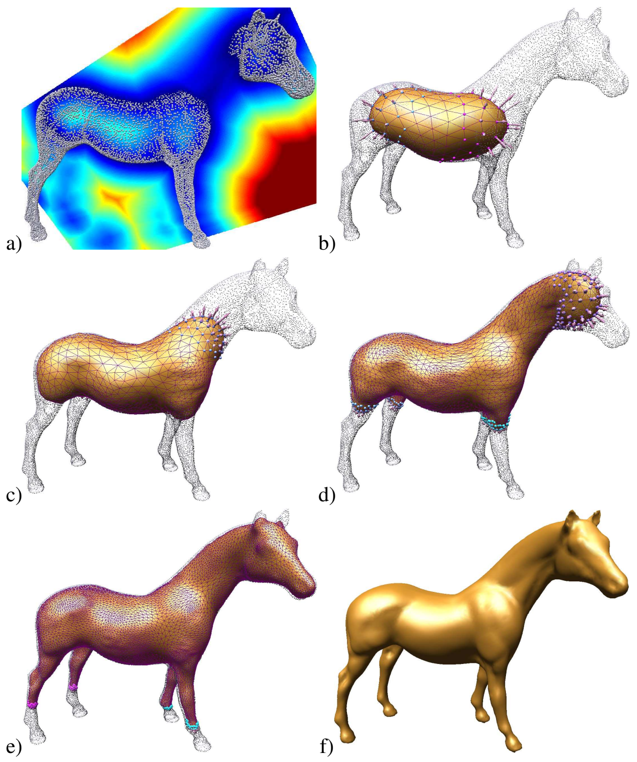

Figure 3: Some snapshots illustrating the evolving fronts: according to the attraction field (a), the vertices of the fronts (bold) incrementally move in outward normal direction (b-e). The final reconstruction (f) is then obtained by a reduced MLS projection to the point cloud.

vertices on the evolving mesh advance jointly as one front. As soon as parts of the front meet the point cloud, the front is split into several *competing* fronts. By using several independent fronts we are able to reconstruct the shape in growing level-of-details. Each front is assigned a tension factor, refraining or allowing greater adaptation. This tension factor is controlled in order to let the coarse fronts that are far from the points evolve faster than finer fronts. Competition continues until the model is ε away to all data points. This coarse–to–fine process leads to a better interpretation of the data, conservatively adapting to the local feature size of the target shape (see Figure 5).

Smooth fronts. The fronts are kept smooth to avoid fold-overs and self-intersections. This is accomplished by two means. First, at each iteration, driven by an implicit function that represents the distance to the input data, an offset vector for each vertex is set in its normal direction. The new positions for each vertex are then calculated using least-square meshes (LSM) formulation [28]. Second, since the fronts evolve faster near their center (where it is farthest away from the point cloud) than on the sides, they stay convexly bent.

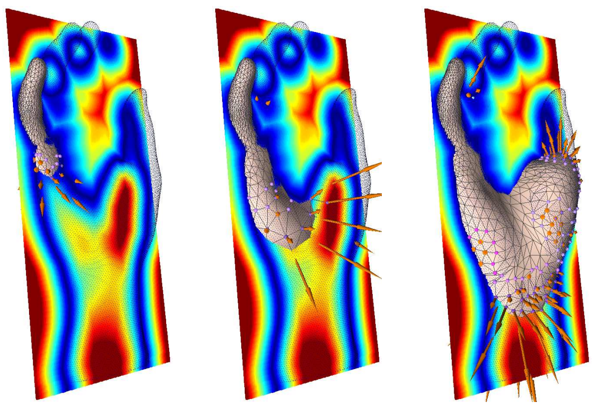

Figure 4: The fronts of our deformable model always move in outward normal direction with a speed defined by the *unsigned* distance value. Thus, they can move up-hill, from a blue region to a red one, as well as down-hill, passing over local extrema of the guidance field.

Outward Movement. A noteworthy property of the fronts is that they always move in outward normal direction. Thus, their advance does not depend on the gradients or the sign of the distance field. The movement speed is defined by the *unsigned* distance value. This allows the fronts to move up-hill as well as down-hill in the guidance field. In particular, a front can evolve from a narrow region into a larger one, against the gradient of the field (Figure 4).

Adaptive dynamic mesh. As the front evolves, local mesh operators [8] improve the connectivity of the mesh to maintain the quality of the output and the efficiency of the LSM solver. The mesh resolution adapts automatically to adjust to local characteristics such as curvature and surface details. This leads in the end to a high quality mesh representation of the reconstructed shape.

Final fitting. Once the deformable model has completed its evolution, it is close enough to the point cloud to perform a final fitting. In this work, we use a Moving Least–Square (MLS) projection to reconstruct the final shape. Areas, where the data is missing, are interpolated with a least-square-mesh using the MLS projected vertices as constraints. Since the model well approximates the target shape, the MLS procedure can be simplified and accelerated while achieving high-quality final results (see section 7 *Final Projection*).

5 Fronts Evolvement Details

The front evolving paradigm involves a simple inflation operator that is applied on an initial sphere-like mesh D iteratively. Therefore, D grows and expands towards the point cloud T using the metaphor of a balloon inflating inside a body. The balloon expands freely until it is blocked by the body walls. Nevertheless, in open regions it contin-

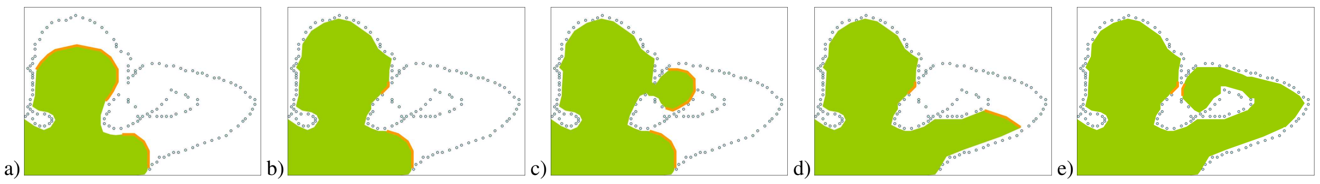

Figure 5: The coarse–to–fine approach avoids early penetrations of the fronts (in orange) into small tunnels. The deformable model first recover coarser regions (a), at (b) two fronts compete to penetrate through two different tunnels. A single level evolvement with no competition might yield the non-intuitive results in (c). In (d) and (e) the coarser front recovers the hand first, and completes the reconstruction.

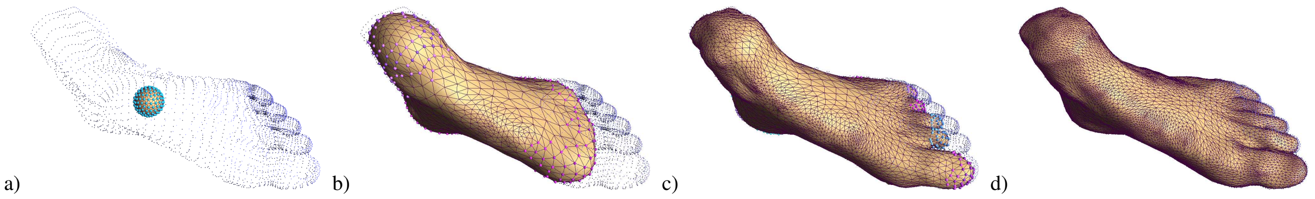

Figure 6: Surface tension guarantees a conservative reconstruction of surface and topology. First, coarse parts are reconstructed (a-b) followed by a finer details (c-d). Each fronts is marked by a set of colored dots.

ues to expand, intruding narrow corridors and expanding in wide chambers to match and recover the shape of T.

An adaptive 3D grid is used as an underlying structure. On that grid, we define a rough signed distance field from the shape. We use compactly supported RBF's [25] for (i) an initial coarse zero level–set approximation of the shape and (ii) to define the inside-outside relation on the points set. Next, we compute the distance to the zero level–set by fast marching [27]. This function serves as the guidance field to the fronts evolution (see Figure 3). To permit fronts intrude narrow corridors in the shape, the function is refined adaptively using finer levels of the octree.

The evolution process starts by placing a sphere-like mesh inside the shape. This step can be easily performed manually by the user or automatically using various heuristics. The algorithm proceeds by iteratively advancing the initial mesh towards the shape. In each iteration, new positions for all fronts' vertices are computed by solving in a least-square manner a constrained Laplacian system similar to [29]:

$$\arg\min_{\{v_i\}}\Big\{\sum_{v_i\in front}\left(w\cdot E_t^i+E_a^i\right)^2\Big\},$$

where:

$$E_t^i=\sum_{v_j\in N(v_i)} c_{ij}(v_j-v_i)\quad\text{and}\quad E_a^i=t_i-v_i.$$

E_t is the tension term (weighted by the tension factor w) which maintains the smoothness of the model using a Laplacian operator, with c_{ij} sum to one for each i and are proportional to $\cot\alpha_{ij}+\cot\beta_{ij}$, $\alpha_{ij}=\angle(v_i,v_{j-1},v_j)$ and $\beta_{ij}=\angle(v_i,v_{j+1},v_j)$ [22]. E_a is the attraction factor, which includes the constraints for new positions of all vertices v_i. In each step the new position t_i of a vertex is calculated as an offset from its current position in the direction of its outward normal. The offset is defined by the *unsigned* distance value. Notice that this implies that vertices can also move up-hill, apparently away from the target surface, e.g., to fill a chamber of the object after the front moved through a narrow tunnel (see Figure 4).

To avoid self-intersections and maintain the smoothness of the mesh, we control the surface tension terms on the evolving fronts with the tension factor w. In a more general context, we use this tension to achieve a coarse–to–fine reconstruction. Initially, w of each front is maximal ($w=w_{max}$), thus restricting the mesh to evolve smoothly, reconstructing only coarse details of the shape (Figure 6 left). The tension of a front is released ($w\leftarrow w-\delta w$) incrementally each time it is stuck. This happens when it does not move enough compared to the attraction guiding it. When the tension passes below a certain threshold, the mesh of the front is subdivided, allowing to reconstruct the fine details of the shape and reach narrow regions (Figure 6). A front is also subdivided when its triangle density is too sparse. In both cases, the tension of the front is reset to the minimum after subdivision ($w=w_{min}$). This gradual reconstruction guarantees that the mesh first reconstructs the coarse parts of the shape and then the finer parts, and hence interprets better the topology of the point cloud.

Vertices that are within ε distance close to the zero level–set become inactive. Effectively, such vertices partition the mesh surface into active and inactive mesh components where each active connected component becomes an inde-

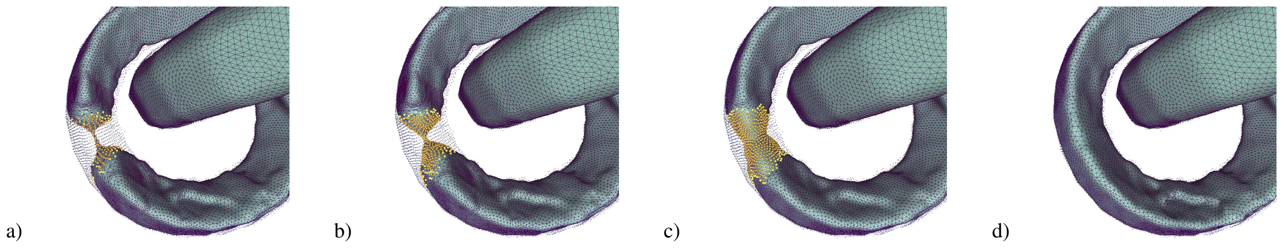

Figure 7: Fronts can be merged by stitching a prisma to their central triangles (a). This prisma forms a new front that evolves normally (b-c). This handle attachment is simple and does not perturb the reconstructed mesh (d).

pendent front. There are cases (e.g. coarse mesh triangulation, coarse grid) where the whole model D is inactive before all target points T being ε-close to D . In such cases we *wake-up* mesh components as fronts using the following procedure. We denote as *unsatisfied points* P, target points that are more than ε far from D. We compute a distance transform from P and re-activate the closest mesh components in D as fronts. The new fronts are first subdivided, their tension released and evolve based on a finer approximation of the distance transform of the unsatisfied points (see Figure 9 for pseudocode).

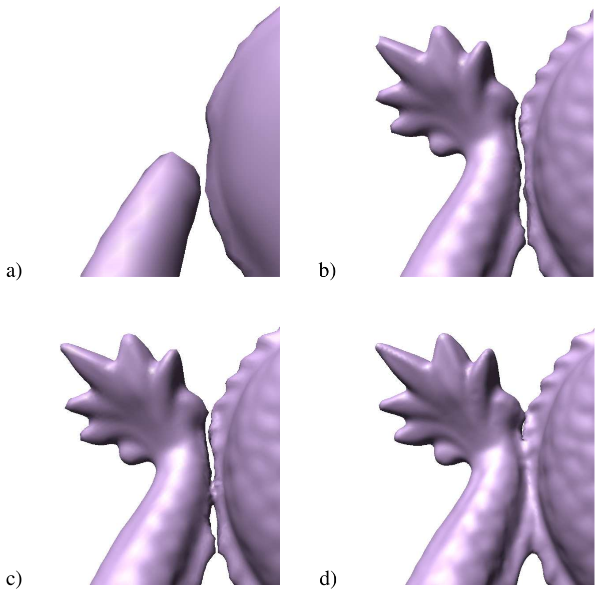

Figure 8: The model first grows into coarser regions and only then aims at growing into finer holes (a). Hereby, it interprets correctly the local topology and reconstruct a coherent genus 0 model (b). Then, a handle can be attached (c) and evolved normally to get a genus 1 result (d).

6 Topology Control

In a well sampled object, there are various techniques that can guarantee a topological correct reconstruction. However, this is seldom the case in real-world scanned models that may contain non-sufficient sampling, noise and outliers. Still, under reasonable assumptions, our framework is sensitive to the topology of the point cloud. Specifically, we target two topological ambiguities that may occur and should be handled by the model: hole filling and tunnel intrusion. Hole filling is the process of smoothly completing a region where no data is given and tunnel intrusion is the process of entering narrow regions to reach far-away parts of the object. Note that these may be difficult to distinguish since holes may contain noisy points and tunnels may contain no points.

Our coarse–to–fine approach for evolving fronts helps in solving these cases. At initial stages, the model reconstructs smooth and coarse approximations of the point cloud, ignoring the fine details. Thus, missing regions are completed smoothly by the mesh during these stages. However, this also means that some of the fronts will stop instead of intruding narrow regions. Nevertheless, due to the *wake-up* procedure, mesh regions which are close to unsatisfied points are re-activated. In these regions the mesh is subdivided and its tension released. This enables the front to enter into finer details and narrow tunnels of the shape and reconstruct it correctly (Figures 5 and 6).

We use a simple collision detection to control the genus of our model, which is composed of two tests: front/front collision and front/inactive parts collisions. The first detection is accelerated by updating the bounding boxes of each front. The second one simply tests cells of the grid for presence of inactive mesh triangles. Collisions are either prevented by deactivating the colliding parts, or induce a genus change by merging the colliding fronts. This allows controlling the maximal genus of the reconstruction (Figure 8). Fronts merging is performed by removing two triangles at each front center and carefully stitching their boundaries to avoid twist. This 6–triangles prisma forms a new front which evolves normally (Figure 7).

7 Final Projection

At the end of the evolution process we achieve a high quality watertight triangular mesh guaranteed that the

Table 1: Resulting meshes and topology. The NA indicates that the PowerCrust did not return any result in less than 12 hours, nv stands for the number of vertices, cnx for the number of connected components and gen for the genus. We generated only one component in all the presented examples.

	size	Mode	RBF		PowerCrust			Ours	
			cnx	gen	nv	cnx	gen	nv	gen
foot	4k	default	1	7	34k	1	8	9k	0
Victoria	12k	stiff	1	6	91k	1	5	25k	0
horse	20k	default	1	0	135k	1	0	14k	0
torus	20k	default	1	9	96k	1	9	5k	9
drill	33k	robust	15	6	164k	15	3	6k	0
hand	37k	default	1	3	256k	1	2	9k	0
CAD	83k	default	1	75	454k	1	99	53k	3
dragon	100k	default	1	5	511k	1	3	77k	0/1
chair	1669k	default	1	2	NA			75k	0

Table 2: Computation time on a Pentium IV 1GHz with 1Gb of memory. *Dist* and *March* stand for the signed distance transform, and fast marching from its zero level set (in minutes). *Evolve* include the LSM solver, *Collide* the collisions detection, *Remesh* the local remeshing and *Fitting* is the final MLS projection (in seconds).

	Dist.	March	Evolve	Collide	Remesh	Fitting	Total
foot	1.9	0.8	0.1	0.4	0.2	0.0	0.7
Victoria	15.7	0.6	0.4	2.6	1.9	0.4	5.3
horse	14.5	0.5	0.2	0.9	0.6	0.1	1.9
torus	21.9	0.4	0.1	0.6	0.3	0.0	1.0
drill	9.6	0.1	0.2	0.2	0.2	0.2	0.8
hand	2.6	0.5	0.1	0.3	0.1	0.1	0.5
CAD	34.8	1.1	0.2	1.2	0.5	0.1	2.0
dragon	32.3	1.0	0.9	4.3	3.4	4.6	13.2
chair	51.4	2.7	0.4	1.4	3.1	3.8	8.7

```
Procedure: Inflate $D$ to reconstruct $T$
$P$ denotes the set of unsatisfied points
and $F$ denotes the set of active fronts
Initialize $P = T$ and $F = D$
while ($P \neq \emptyset$) {
   while (evolve_fronts (F)) {
      remove inactive vertices from F
      update fronts in F
      remove satisfied points from P}
   if ($F = \emptyset$ and $P \neq \emptyset$){
      F = wake_up_fronts (D,P)
      reset_fronts_tension(F)}
   foreach front $f \in F${
      subdivide if small triangle density
      release front tension if too high}
}
```

Figure 9: The Competing Fronts pseudocode.

point-cloud is ε–close to it. This guarantee can be used for accurate reconstruction, for example using the abstract projection of [10]. In this final stage, we shift each vertex v of the mesh along its normal $\mathbf{n}_v$ towards the point cloud. This can be seen as a low-order moving least square projection (MLS) [1].

The main difficulty of the MLS approach resides in approximating the reference plane $\mathbf{T}_{MLS}$ of the MLS projection [4]. In our case this is gracefully solved by using the reconstructed mesh. Instead of solving a costly non-linear minimization for $\mathbf{T}_{MLS}$, we simply use the tangent to the mesh $\mathbf{T}_{MLS} \perp \mathbf{n}_v$ as the reference plane. The stability of this approach is grounded theoretically in the fact that ε–close surfaces have close normals [10]. In the case where the input data is sparse, the order of the MLS interpolating polynomial must be reduced, sometimes even to 0, in which case each mesh vertex v is simply projected onto the average distance along $\mathbf{n}_v$ of the point cloud in its neighborhood. Since the mesh is already a good approximation of the shape, this 0–order MLS was sufficient to produce the models in this paper (see Figure 15).

If data is too sparse or missing, mesh vertices in that region are too far from the point cloud (see Figure 13), hence can not be projected using MLS. Instead, they are interpolated again to generate a least–square mesh, by solving the same Laplacian equation used for evolving the fronts, using positions of the nearest MLS–projected vertices as the least–square constraints.

8 Results

We have experimented with our deformable model a number of input point clouds, focusing on point clouds which are usually difficult to reconstruct. The timings reported were taken on a Pentium IV 1GHz with 1Gb of memory. We compared our reconstruction to the PowerCrust [3] (version 1.2) (see Figure 10), and to the Radial Basis Functions of [25] (available from the author's web page) (see Figure 11). We evaluate RBF on the same grid as our finest adaptation level, and polygonize it using [20] to get the exact topology of the isosurface. As detailed in Table 1, unlike the other methods, we recover the correct topology in all cases, while maintaining a reasonable output size, and a high quality mesh (see Figure 10). Note in Table 2 that the remeshing time is negligible.

Also note that the final fitting is extremely fast. This is achieved using the local evolving model as a reference plane, which avoids the expensive optimization required in the classical MLS. Nevertheless, the final fitting still captures well highly detailed surfaces (see Figure 15).

One of the main advantages of using explicit deformable models is the ability to control the topology of the result. Even in delicate cases like the Figure 1, where the point

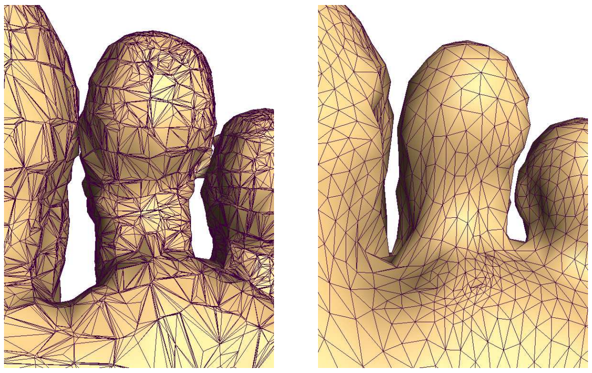

Figure 10: A side by side comparison with the PowerCrust reconstruction (left). Note that our evolved model (right) retains a high quality mesh with a simpler topology.

cloud is multiply connected, we can control our method and maintain a genus 0 surface, separating the tail from the body, or a genus 1 surface (see Figure 8).

We use a default setting for our deformable model that performs well on most scans. We refer to this mode as *default*: $w_{max} = 5, w_{min} = 3, \delta w = 0.5$ and the mesh is remeshed every 6 iterations. However, we need to use two other modes in other cases. To handle very noisy models, such as the raw scans of Figure 14, we use a *robust* mode which refrains the subdivision rate of the fronts to avoid interpreting outliers as features: $w_{max} = 6, w_{min} = 3, \delta w = 0.3$. On models with large missing parts, such as the Victoria of Figure 13, we use a *stiff* mode, since the deformable model must stay rigid to avoid leaking: $w_{max} = 40, w_{min} = 20, \delta w = 1$ and the mesh is remeshed every 18 iterations.

Our coarse–to–fine approach is particularly visible on Figure 6, where the toes of the foot are recovered only after the foot arch is. This feature avoids misinterpretation of the topology in locally, sparse regions; cases which are difficult for the PowerCrust (Figure 10 left).

Figure 11 illustrates the importance of having topology control. Other reconstruction methods such as the adaptive RBF may interpret the noise around the sharp features as many connected components. Our deformable model assures it is a single water-tight component with high quality mesh.

Our method can produce surfaces with a higher genus using a handle attachment as illustrated on the noisy scan of a CAD model Figure 7. We chose to reconstruct the whole model first with genus 0, and only then attach its handles. Thus, topology can be controlled and monitored by the user.

The limitation of our method is the use of a grid to represent the volumetric distance transform, especially for very thin parts in the model, for which the deformable model has not enough resolution to reconstruct (see Figure 12). Increasing both grid and deformable mesh resolutions will slow down the process considerably and can lead to numerical instabilities. For example, the reconstruction of the thin details in the cushion fabric in the Chair model (see Figure 15), significantly slows the deformable model evolution (see Table 2).

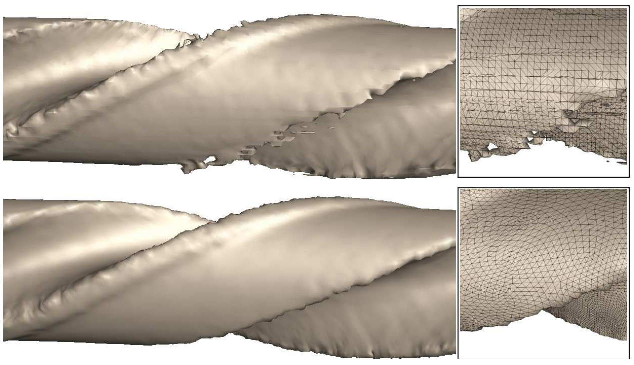

Figure 11: A reconstruction from a raw noisy scan with outliers. The adaptive RBF reconstruction (top) and our reconstruction (bottom). Note that the adaptive RBF creates 15 connected components, where the main one is of genus 6.

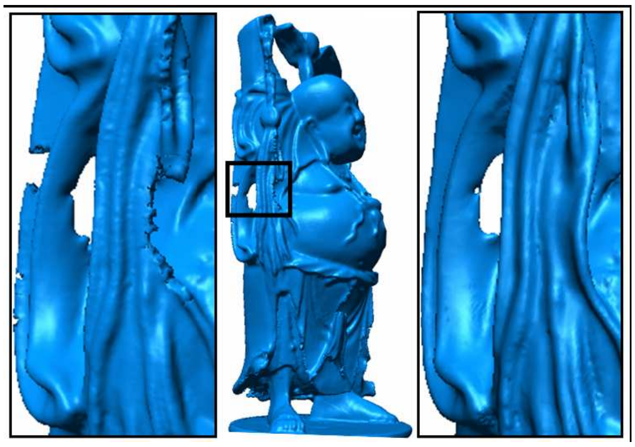

Figure 12: Very thin parts in the model hamper our reconstruction method. For example, the reconstruction of happy Budha (middle) is incomplete in the thin gown (left). For comparison the original part is shown on the right.

It is often the case that models cannot be fully scanned because of physical and viewpoint limitations. As shown in Figure 13 our method can reconstruct very difficult models such as the Victoria model, which includes very large missing parts (the head on Figure 2, and the legs, at the bottom of Figure 13), although a distance transform yields a wrong topology, in particular close to her ear.

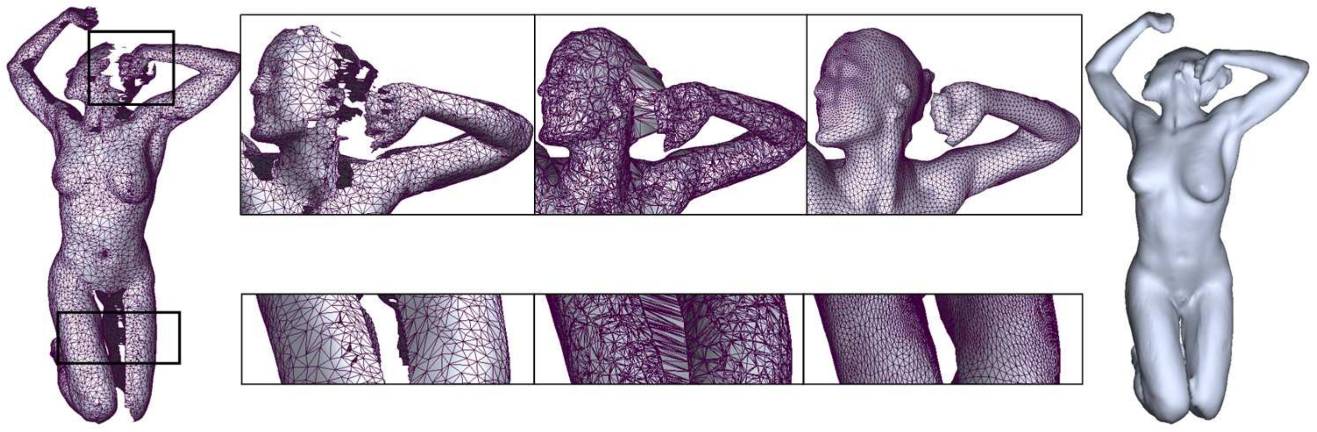

Figure 13: Our deformable model distinguishes between holes and tunnels in a coarse–to–fine manner. Note how it correctly distinguishes the two legs, using their separation at the hip level, and the topology of the arm close to the head. For comparison, we show in the middle the reconstruction of PowerCrust (version 1.2).

Conclusions

We presented a coarse–to–fine deformable model for reconstructing surfaces from point clouds. Our method has control and can monitor the topology and recovers details even on challenging scans with the presence of noise and with large missing parts. In the future we would like to extend this technique for mesh repairing and the consolidation of polygon soups. We also plan to continue this work to improve the final fitting to cope with sharp features, combining robust statistics and dedicated local mesh enhancements.

Acknowledgements

We would like to thank L. Gavish and O. Mohr for their work and contribution. We thank S. Azernikov for the drill model, Stanford repository for their models, Matmídia Lab for the 9-torus and A. Sheffer for Victoria model. This work was supported partly by a grant from Israeli Ministry of Science.

References

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

[2] L. Alvarez, F. Guichard, P.-L. Lions and J.-M. Morel. Axioms and fundamental equations of image processing. *Archives for Rational Mechanics*, 123(3):199–257, 1993.

[3] N. Amenta, M. Bern and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH*, pages 415–422. ACM, 1998.

[4] N. Amenta and Y. J. Kil. Defining point-set surfaces. *Transaction on Graphics*, 23(3):264–270, 2004.

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *Trans. on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[6] S. Bischoff and L. Kobbelt. Sub–voxel topology control for level–set surfaces. *Computer Graphics Forum*, 22(3):273–280, 2003.

[7] J.-D. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *Symp. on Computational Geometry*, pages 223–232. ACM, 2005.

[8] M. Botsch and L. Kobbelt. A remeshing approach to multiresolution modeling. In *Symp. on Geometry Processing*, pages 185–192, 2004.

[9] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH*, pages 67–76. ACM, 2001.

[10] F. Chazal, A. Lieutier and J. Rossignac. Orthomap: Homeomorphism-guaranteeing normal-projection map between surfaces. In *Symp. on Solid and Physical Modeling*, 2005.

[11] L. D. Cohen and I. Cohen. Finite element methods for active contour models and balloons from 2-D to 3-D. *Trans. on Pattern Analysis and Machine Intelligence*, 15(11), 1993.

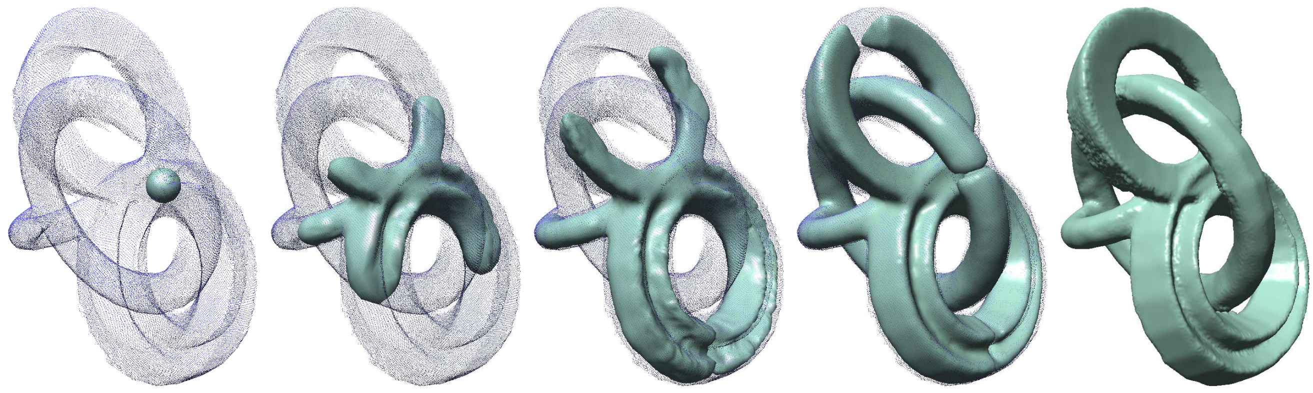

Figure 14: A reconstruction of a noisy scan with genus 3.

[12] T. F. Cootes, C. J. Taylor and D. H. Cooper. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[13] T. Dey and S. Goswami. Tight cocone: A water-tight surface reconstructor. In *Symp. on Solid Modeling and Applications*, pages 127–134. ACM, 2003.

[14] Y. Duan and H. Qin. A subdivision-based deformable model for surface reconstruction of unknown topology. *Graphical Models*, 66(4):181–202, 2004.

[15] J. Esteve, P. Brunet and Àlvar Vinacua. Approximation of a variable density cloud of points by shrinking a discrete membrane. *Computer Graphics Forum*, 24(2):791–807, 2005.

[16] S. Fleishman, C. Silva and D. Cohen-Or. Robust moving least–squares fitting with sharp features. *SIGGRAPH*, 24(3), 2005.

[17] C. H. Esteban and F. Schmitt. A snake approach for high quality image-based 3d object modeling. In *Variational, Geometric and Level Set Methods in Computer Vision*, pages 241–248. IEEE, 2003.

[18] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4(2):73–91, 2000.

[19] M. Kass, A. Witkin and D. Terzopoulos. Snakes: Active contour models. *Intl. J. of Computer Vision*, 1(4):321–331, 1988.

[20] T. Lewiner, H. Lopes, A. W. Vieira and G. Tavares. Efficient implementation of Marching Cubes' cases with topological guarantees. *J. of Graphics Tools*, 8(2):1–15, 2003.

[21] R. Mencl and H. Müller. Interpolation and approximation of surfaces from three–dimensional scattered data points. In *Scientific Visualization*, pages 223–232. IEEE, 1999.

[22] M. Meyer, M. Desbrun, P. Schröder and A. Barr. Discrete differential geometry operators for triangulated 2–manifolds. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer, 2002.

[23] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara and M. J. Wozny. Geometrically deformed models: a method for extracting closed geometric models form volume data. *SIGGRAPH*, 25(4):217–226, 1991.

[24] D. Mumford, C. Lu and Y. Cao. Surface evolution under curvature flows. *Visual Communication and Image Representation*, 13:65–81, 2002.

[25] Y. Ohtake, A. G. Belyaev and H.-P. Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Solid Modeling Intl.*, pages 31–39. IEEE, 2004.

[26] C. E. Scheidegger, S. Fleishman and C. T. Silva. Triangulating point set surfaces with bounded error. In *Symp. on Geometry Processing*, pages 63–72, 2003.

[27] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[28] O. Sorkine and D. Cohen-Or. Least-squares meshes. In *Shape Modeling Intl.*, pages 191–199. IEEE, 2004.

[29] O. Sorkine, D. Cohen-Or, D. Irony and S. Toledo. Geometry–aware bases for shape approximation. *Trans. on Visualization and Computer Graphics*, 11(2):171–180, 2005.

[30] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *Trans. on Image Processing*, 7(3):359–369, 1998.

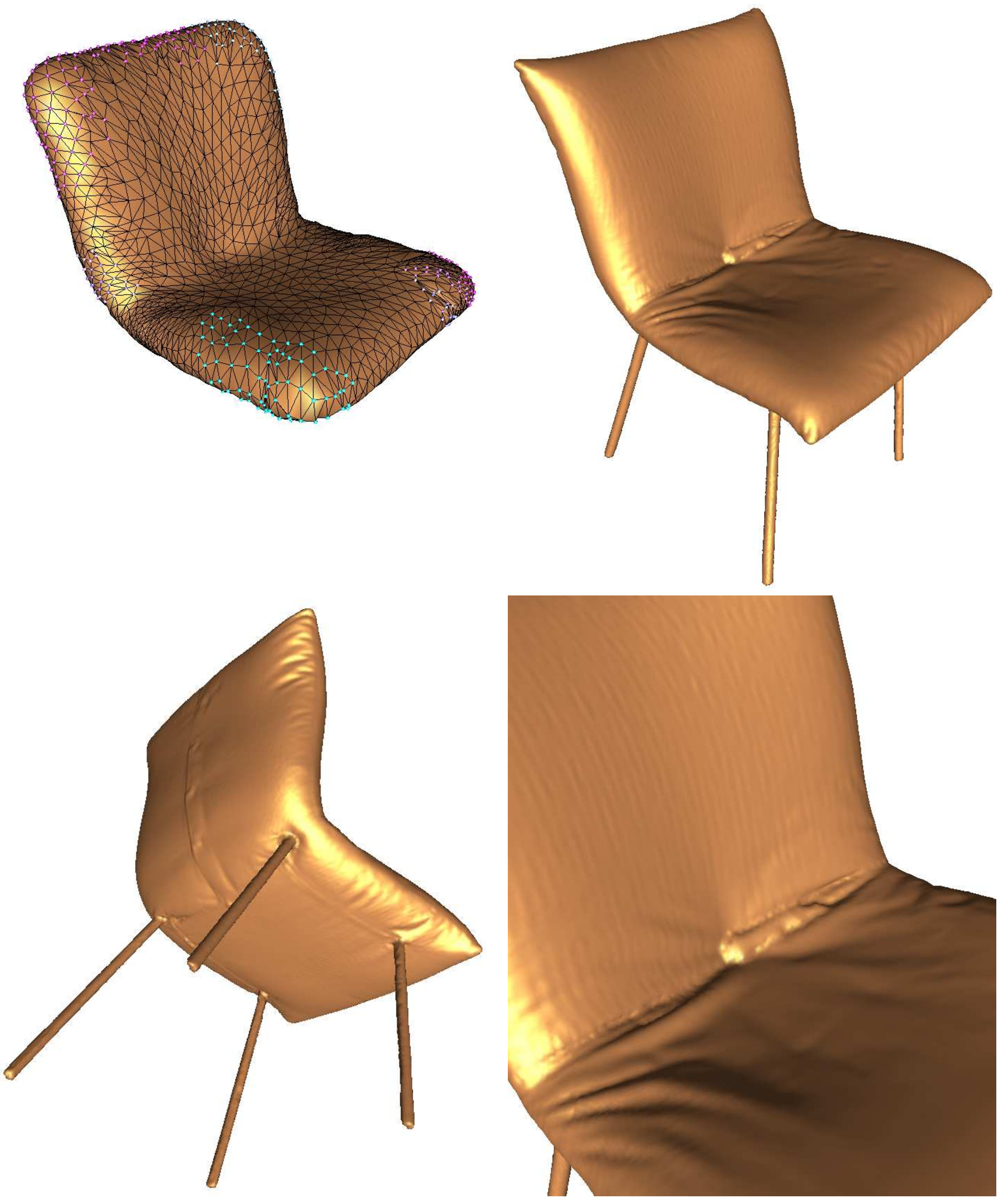

Figure 15: A highly detailed model. Our coarse–to–fine technique first recovers the body of the chair, and only then the four legs. Note the high quality of the reconstructed details.