

# Interactive Topology-aware Surface Reconstruction

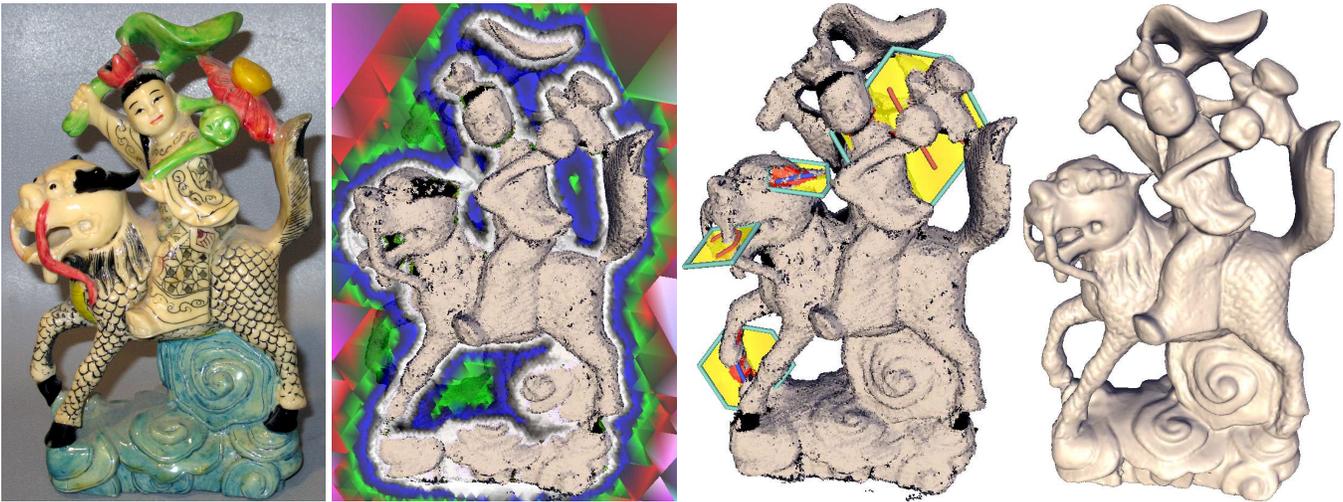
Andrei Sharf  
Tel Aviv University

Thomas Lewiner  
PUC-Rio de Janeiro

Gil Shklarski  
Tel Aviv University

Sivan Toledo  
Tel Aviv University

Daniel Cohen-Or  
Tel Aviv University



**Figure 1:** Interactive reconstruction of the riding monk (left). Our finite-element field formulation (center-left) incorporates user’s interactive scribbles at automatically-detected unstable regions (center-right) to obtain the expected shape (right).

## Abstract

The reconstruction of a complete watertight model from scan data is still a difficult process. In particular, since scanned data is often incomplete, the reconstruction of the expected shape is an ill-posed problem. Techniques that reconstruct poorly-sampled areas without any user intervention fail in many cases to faithfully reconstruct the topology of the model. The method that we introduce in this paper is topology-aware: it uses minimal user input to make correct decisions at regions where the topology of the model cannot be automatically induced with a reasonable degree of confidence. We first construct a continuous function over a three-dimensional domain. This function is constructed by minimizing a penalty function combining the data points, user constraints, and a regularization term. The optimization problem is formulated in a mesh-independent manner, and mapped onto a specific mesh using the finite-element method. The zero level-set of this function is a first approximation of the reconstructed surface. At complex under-sampled regions, the constraints might be insufficient. Hence, we analyze the local topological stability of the zero level-set to detect weak regions of the surface. These regions are suggested to the user for adding local inside/outside constraints by merely scribbling over a 2D tablet. Each new user constraint modifies the minimization problem, which is solved incrementally. The process is repeated, converging to a topology-stable reconstruction. Reconstructions of models acquired by a structured-light scanner with a small number of scribbles demonstrate the effectiveness of the method.

**CR Categories:** I.3.6 [Computer Graphics]: Computing Methodologies—Interaction techniques

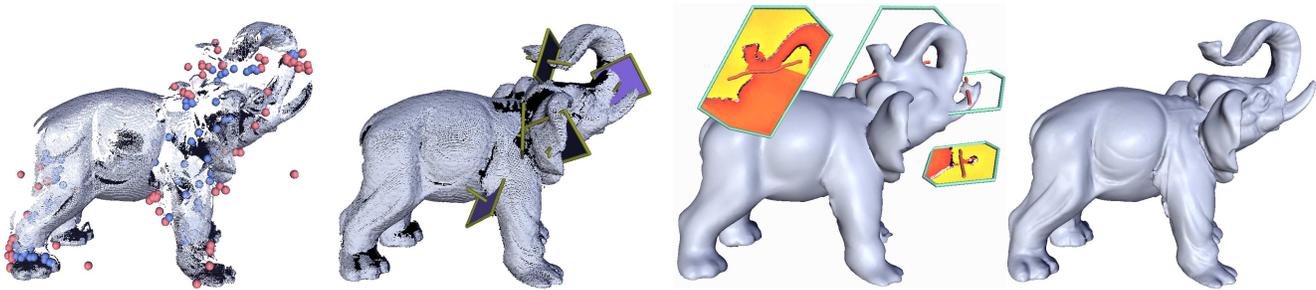
**Keywords:** Surface reconstruction, Interactive tools.

## 1 Introduction

As 3D scanners are becoming commonplace, surface reconstruction is becoming a common step in modeling workflows. Thus, there is an emerging need to allow users to create high-quality surfaces from scans, including imperfect scans. This need brought much attention to surface reconstruction techniques. Recent research efforts have led to significant progress in all aspects of the problem [Levoy et al. 2000]. However, the reconstruction of a complete watertight model that is faithful to the original physical object is still a difficult process.

One of the major difficulties is the coverage of the scanned model: As a result of physical inaccessibility, poor visibility and material properties, the coverage is often imperfect and significant portions of the surface are either under-sampled or completely missing. The problem is more acute for complex shapes with deep cavities and bifurcations (e.g., Figure 1). While it is reasonable to assume that scanning hardware will advance, future reconstruction systems will still need to employ algorithms that reconstruct under-sampled areas. Systems that reconstruct poorly-sampled areas merely based on priors, without any user intervention, fail in many cases to faithfully reconstruct the expected shape. At the other extreme, systems that rely only on explicit manual surface editing are too tedious.

Without prior assumptions and user constraints, the reconstruction problem is ill posed; an infinite number of surfaces pass through or near the data points. Smoothness and watertight constraints usually regularize the problem and remove the ill posedness. Nevertheless, even if the problem is successfully transformed into a well conditioned one, the reconstructed object is not necessarily the expected one. Our method is based on the observation that it is often possible to detect the ill conditioning and to ask the user for inside/outside constraints to locally resolve them and achieve the expected shape.



**Figure 2:** Reconstruction pipeline for a scanned elephant model. An initial coarse function is automatically computed from loose, automatic inside/outside (blue/red balls) constraints (left). The topological analysis of this function selects weak regions (center-left), where the user can make local decisions by scribbling over 2D tablets at a coarse resolution (center-right). Further iterations at finer resolutions lead to a complete reconstruction of the model (right).

In this paper we present a topology-aware reconstruction technique that requires minimal user input to make correct decisions at critical regions, where the topology of the shape cannot be induced automatically with a reasonable degree of confidence. Our method uses priors to reconstruct the surface, but it also allows the user to influence the prior distribution. Two aspects of the prior distribution are fixed: we assume that the surface is smooth almost everywhere, and that it should be watertight. Other aspects of the prior distribution are controlled by the user who specifies constraint points that should be inside or outside the surface.

To reconstruct a watertight surface given raw scans without normals, and possibly the user’s inside/outside constraints, we first construct a continuous function over a three-dimensional domain. The zero level-set of this function approximates the data points. We construct this function by minimizing a penalty that measures its non-smoothness, the deviation of its zero level-set from the data points, and its deviations from prescribed positive/negative values at the inside/outside constraints. Our function optimization problem is formulated in a mesh-independent manner, and mapped onto a specific mesh using the finite-element method. Computationally, the function is constructed by solving a large sparse linear system. However, at complex under-sampled regions these constraints might be insufficient. Therefore, we analyze the local topological stability of the zero level-set to detect *weak regions* of the surface. These regions are suggested to the user for adding local inside/outside constraints by merely scribbling over a 2D tablet corresponding to a cross section of the field (see Figure 2). The new user input augments the linear system with additional constraints, improving the reconstruction. The stability analysis is then repeated. If the surface is still topologically unstable, the user is prompted for additional constraints. This incremental process refines the surface until it is topologically stable.

## 2 Related Works

The problem of reconstructing a surface from scans has been researched extensively for almost two decades [Hoppe et al. 1992; Levoy et al. 2000]. Many different techniques have been developed, based on signed distance functions [Hoppe et al. 1992; Curless and Levoy 1996], Voronoï diagrams [Amenta et al. 1998; Boissonnat and Cazals 2002; Dey and Goswami 2004], radial basis functions and local implicit functions [Carr et al. 2001; Ohtake et al. 2003; Ohtake et al. 2004], moving least square approximation [Alexa et al. 2001; Amenta and Kil 2004], or wrapping techniques [Bernardini et al. 1999], to mention a few. Nevertheless, these techniques are concerned with a faithful reconstruction of the local structure of the surface, whereas we also focus on its global structure.

Some related works are concerned with the reconstruction of a surface from inhomogeneous sample density or missing data [Davis et al. 2002; Dey and Goswami 2004; Kolluri et al. 2004; Sharf et al. 2006; Hornung and Kobbelt 2006]. These techniques use some

heuristics to define the locus of the surface in under-sampled or noisy data. Our technique is similar to the works of [Kolluri et al. 2004; Hornung and Kobbelt 2006] in that we use a global optimization technique and that we guarantee a continuous watertight surface reconstruction. However, the method of Kolluri et al. [2004], requires filtering of the Voronoï diagram to obtain a correct pole graph. To compute a watertight surface the authors use global normalized cuts that smoothly complete large missing parts. The method of Hornung and Kobbelt [2006], requires the definition of a watertight voxel crust in which the unknown surface is supposed to lie. To define the crust, the authors use flood-fill and dilation operators. However, at problematic regions where the topology of the data is unclear, both algorithms cannot guarantee the generation of a topologically-correct crust or pole graph. Instead, we automatically identify these weak regions and allow the user to interactively correct them.

Our finite-element formulation of the reconstruction shares similarity with the work of [Carr et al. 2001; Ohtake et al. 2004; Kazhdan et al. 2006] in the definition of the surface as a solution of a global linear system. Carr et al. [2001] use radial basis functions (RBF) constrained with surface and off-surface values. To make the system sparser, Ohtake et al. [2004] use local RBF. Recently, Kazhdan et al. [2006] use a Poisson system to define a surface which agrees with the data points’ normals. These works make intensive use of points’ orientation (surface normals), which is often unreliable. In our work, we interactively reconstruct the surface using only the positions of raw scanned data, where the user defines the general in/out orientation.

Recently, the use of scribbles as an interactive tool has gained a lot of popularity in various computer graphics applications, such as image segmentation [Li et al. 2004], matting [Wang and Cohen 2005], colorization [Levin et al. 2004], or mesh editing [Nealen et al. 2005]. Scribbles are easy-to-use, loose, and do not require meticulous work. At the same time, they provide the algorithm minimal but necessary hints needed to solve a problem, which is otherwise extremely hard or even impossible to solve. Our work extends the effective power of scribbles to assist ill-posed problems in surface reconstruction. Nevertheless, it should be emphasized that, in this work, a careful analysis of the data leads the user to where assistance is needed. Such an automatic guidance is vital since, unlike previous work, the problem domain is volumetric, and the scribbles are not drawn in image space, but on a 2D tablet corresponding to an arbitrary planar cross section in 3D.

## 3 Our Approach

Our surface reconstruction method initially computes a smooth three-dimensional implicit function whose zero level-set is a coarse approximation of the data points. The method then explicitly extracts the reconstructed surface from this function as its zero level-set. We formulate this implicit function as the solution of a physi-

cal minimization problem, which we solve using the finite-element method (FEM). As Kazhdan et al. [2006] showed, defining implicit functions as a solution of a linear system guarantees a global *coherent* solution that is watertight. Moreover, it leads to a well conditioned sparse system whose size is proportional to the reconstructed surface. Nevertheless, our system is designed to be interactive and to incorporate the user’s constraints.

The main challenge, shared by any method which fits implicit functions [Carr et al. 2001], is to retrieve zero values only at the surface. Necessarily, these methods require additional information, like the points’ orientations (i.e., the normal) which is typically less precise than the points’ location. Furthermore in under-sampled regions, the orientation of the data points is not reliable. This may cause significant misinterpretation of the data, leading to an erroneous surface reconstruction.

Therefore, we argue that user assistance is necessary to correctly interpret the data, especially in under-sampled regions. The key point in our work is to analyze the implicit function and to identify topologically weak regions, in the sense that small perturbations of the field lead to different topological interpretations of the data. Our system indicates such weak regions, allowing the user to merely draw few scribbles over a corresponding 2D tablet to specify the correct local sign of the implicit function. As the user draws such scribbles, the solution in that region is correctly constrained and the local topological stability increases.

A common source of errors is the use of a *fixed* hierarchical data structure, since coarse resolution close to the surface would generate erroneous reconstructions. The volumetric nature and the required accuracy of the solution necessitate the use of an adaptive domain, which ideally is coarse away from the surface and fine close to it. However, since the locus of the surface is unknown a priori, the definition of a correct adaptive domain turns into a chicken/egg problem. Therefore, we use a *dynamically* adapted hierarchical structure, generated by triangulating the dual of an octree (see Figure 3 (right)). Our reconstruction process performs in a coarse-to-fine, incremental manner. At each iteration, the hierarchy is locally refined close to the zero level-set of the current implicit function. Next, the function is updated and analyzed, suggesting finer user interactions at weak regions.

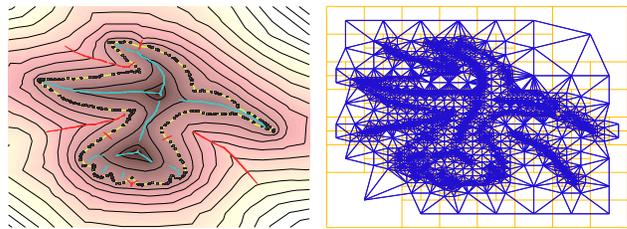
We illustrate the process in Figure 2. Initially, a set of automatic constraints is computed from the data points (Figure 2 (left)), defining approximate inside/outside constraints. Next, the FEM system is solved and the topology of the resulting implicit function is analyzed. At weak regions (Figure 2 (center-left)), the user is guided to add scribbles on 2D tablets for regularizing the local topology (Figure 2 (center-right)). Starting at this coarse resolution, the process is iteratively refined by locally increasing the resolution until achieving the expected shape (Figure 2 (right)).

Section 4 introduces our formulation of the reconstruction problem. In Section 5 we present the topological analysis of the implicit function, and the definition of weak regions. These techniques are integrated into an interactive framework described in Section 6. We conclude after showing reconstruction results of structured light scans in Section 7.

## 4 Constrained FEM Reconstruction

In this section, we describe the formulation of our implicit function optimization. Let  $P^s$  be the data points’ set, and  $P^{in}/P^{out}$  be sets of inside/outside constraint points. Our goal is to construct a smooth watertight surface  $Z$  that is close to  $P^s$ , and separates  $P^{in}$  and  $P^{out}$ .

The problem defines three criteria, one of which is binary (disconnectedness of  $P^{in}$  and  $P^{out}$ ) and two of which are continuous and high dimensional (smoothness of  $Z$ , closeness of  $P^s$  to  $Z$ ). We look for a solution that disconnects  $P^{in}$  and  $P^{out}$  while balancing the smoothness of  $Z$  and its closeness to  $P^s$ .



**Figure 3:** A 2D illustration of a hummingbird reconstruction. The reconstructed shape is extracted from the zero level-set (in yellow) of an implicit function, computed using a finite-element formulation (left). Our underlying hierarchy adapts to the data points and to the automatic constraints (right).

**Continuous optimization on implicit representation.** We first project the problem into a higher dimension; instead of searching for  $Z$  directly, we construct a continuous implicit function  $u(p)$ ,  $p \in \Omega$  defined over some domain  $\Omega \subset \mathbb{R}^3$ , and define  $Z = u^{-1}(\{0\})$  to be its zero level-set (see Figure 3 (left)). We translate the objectives defined with respect to  $Z$  onto the function  $u$ , by requiring it to be smooth (generically leading to a smooth  $Z$ ) and to satisfy :

$$u(p) \approx 0 \quad \text{for } p \in P^s \quad (1a)$$

$$u(p) > 0 \quad \text{for } p \in P^{in} \quad (1b)$$

$$u(p) < 0 \quad \text{for } p \in P^{out} \quad (1c)$$

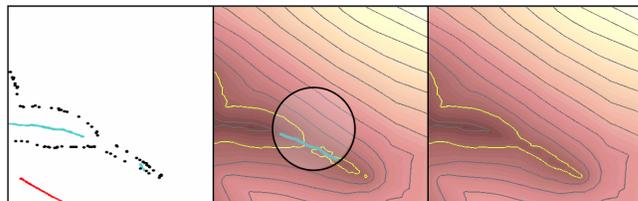
Unless the gradient of  $u$  near its zero level-set is small, the points of  $P^s$  are close to  $Z$ . In our interactive system, the user can easily add points to  $P^{in}$  and  $P^{out}$  to stabilize  $Z$  (see Figure 4). In the next section, we explain how to detect areas where more constraints are needed.

The method replaces the constraints (1b) and (1c) by specifying target values at  $P^{in}$  and  $P^{out}$  to remove the linear inequality constraints, and hence amenable to optimization methods. Thus, the three criteria on  $u$  can be combined into a single continuous score function  $\Psi$ , which assigns a penalty to each candidate. The score  $\Psi(u)$  is a sum of two scores, one that penalizes  $u$  for not being smooth and one that penalizes it for being far from zero, positive and negative target values at  $P^s$ ,  $P^{in}$  and  $P^{out}$ .

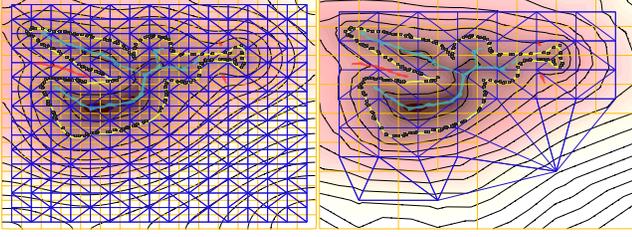
**Penalty functions.** We define the non-smoothness penalty for any function  $u$  that is differentiable almost everywhere in  $\Omega$  by:

$$\Psi_{\text{smoothness}}(u) = \frac{1}{2} \iiint_{\Omega} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial u}{\partial z} \right)^2 \right] dx dy dz. \quad (2)$$

This penalty has a physical interpretation [Strang 1986, Sec. 3.3]: it is the amount of work that is used to maintain state  $u$  in a uniform isotropic diffusive problem (heat conduction, electrostatics) with no forcing terms. If we add proper forcing terms on the boundary of  $\Omega$ , the steady-state solution of this physical problem minimizes this penalty function and satisfies the Laplace equation.



**Figure 4:** Adding scribbles to constrain weak regions achieves the expected topology in the Hummingbird head: from the data points and initial automatic constraints (left), a scribble is drawn over the initial FEM field (center) to correct the local topology (right).



**Figure 5:** A 2D example of the FEM solution on two different underlying grids: the smoothness penalty is independent of the mesh, adapting automatically to different element sizes and aspect ratio.

The penalty function  $\Psi_{\text{point constraints}}$  penalizes  $u$  for being far from the point constraints. The target values  $t(p)$  are defined by  $t(p) = 0$  for  $p \in P^s$  and heuristically setting  $t(p) = -d(p)$  (resp.  $t(p) = +d(p)$ ) for  $p \in P^{in}$  (resp.  $p \in P^{out}$ ), where  $d(p)$  is the distance to the nearest data point :

$$\Psi_{\text{point constraints}}(u) = \sum_{p \in P} (u(p) - t(p))^2,$$

where  $P = P^s \cup P^{in} \cup P^{out}$ .

**Finite-Element Method.** The set of admissible functions  $u$  is defined by interpolation using a finite element mesh  $M$  that partitions  $\Omega$  [Hughes 1987, Chap. III]. We denote the values of  $u$  at the vertices of  $M$  by the vector  $u_M$ .

We use the finite-element method to construct a matrix  $K$  such that:

$$\Psi_{\text{smoothness}}(u) = u_M^T K u_M. \quad (3)$$

Thus, in our method, the mesh defines only the family of admissible functions  $u$ . The non-smoothness penalty  $\Psi_{\text{smoothness}}(u)$  of a particular function is completely independent of the mesh (see Figure 5), since it is intrinsically defined using the mesh-free expression (2).

In this context, the finite-element method is essentially a recipe for constructing a matrix  $K$  given an integral expression such as (2) and a finite-dimensional set of admissible functions, such that  $K$  satisfies (3). The recipe is based on computing the integral (2) on a set of basis functions that span the space of admissible functions. Here, we use a tetrahedral mesh and define the admissible functions as piecewise tri-linear, interpolated from vertices' values on each tetrahedron of the mesh. The matrix  $K$  is sparse and easy to compute tetrahedron by tetrahedron (see Appendix).

The literature contains many methods, often referred to as Laplacian approximations, for constructing similar matrices from 2D meshes. The main advantage of our FEM-based approach is that the smoothness penalty formulation is independent of the mesh, since the left-hand side of Equation (3) is mesh free. Therefore it extends naturally to 3D meshes. Our method automatically adapts to functions interpolated on meshes with tetrahedrons of widely different size and aspect ratio, common in our data structure (see Figure 5).

We sum  $\Psi_{\text{smoothness}}$  and  $\Psi_{\text{point constraints}}$  to form a single least squares optimization problem. For each constraint point  $p \in P$ , we define  $c_p$  such that  $c_p u_M = u(p)$ . The row  $c_p$  represents a linear interpolation operator on  $M$ . It has at most four non-zero values for a tetrahedral mesh. The constraint for point  $p$  now writes  $c_p u_M = t(p)$ , and we weight this constraint with weight  $\omega_p$ . The smoothness constraint can be incorporated to the least squares formulation using any matrix  $E$  such that  $E^T E = K$  as follows:

$$\min_{u_M} \left\| \begin{bmatrix} E \\ \omega_1 c_1 \\ \omega_2 c_2 \\ \vdots \\ \omega_{|P|} c_{|P|} \end{bmatrix} u_M - \begin{bmatrix} 0 \\ \omega_1 t(p_1) \\ \omega_2 t(p_2) \\ \vdots \\ \omega_{|P|} t(p_{|P|}) \end{bmatrix} \right\|_2^2,$$

The matrix  $E$  is never computed explicitly. Instead, we solve the least-squares problem using its normal equation:

$$\left( K + \sum_{p \in P} \omega_p^2 c_p^T c_p \right) u_M = \sum_{p \in P} t(p) \omega_p^2 c_p^T. \quad (4)$$

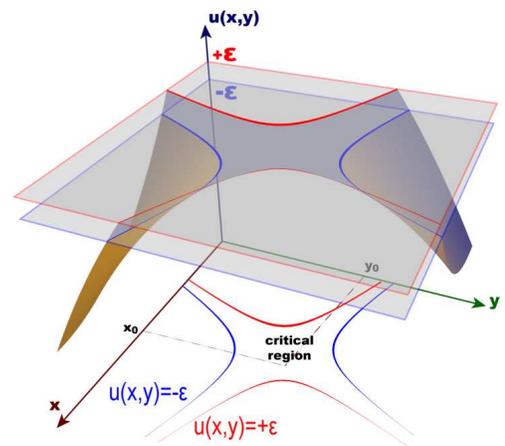
We construct the coefficient matrix of this linear system of equations by constructing  $K$  using the finite-element method, and then adding to it the sparse matrices  $c_i^T c_i$ . We solve using a fast sparse Cholesky factorization (CHOLMOD [Davis and Hager 2005]).

**Adding and removing constraints.** One aspect of this numerical approach allows real-time interaction in our surface reconstruction application. The structure of Equation (4) allows the method to incrementally update the linear system factorization when the user adds or removes inside/outside constraints. We compute the additional rows  $c_p$  and target values  $t(p)$  for each new point  $p$ , and add the new  $c_p^T c_p$  matrices and  $c_p^T t(p)$  vectors to left and right hand side of Equation (4). To update the sparse Cholesky factorization we use [Davis and Hager 2005]. Thus, the system factorizes the initial matrix only once. In most cases, updating this factorization is faster than factoring again.

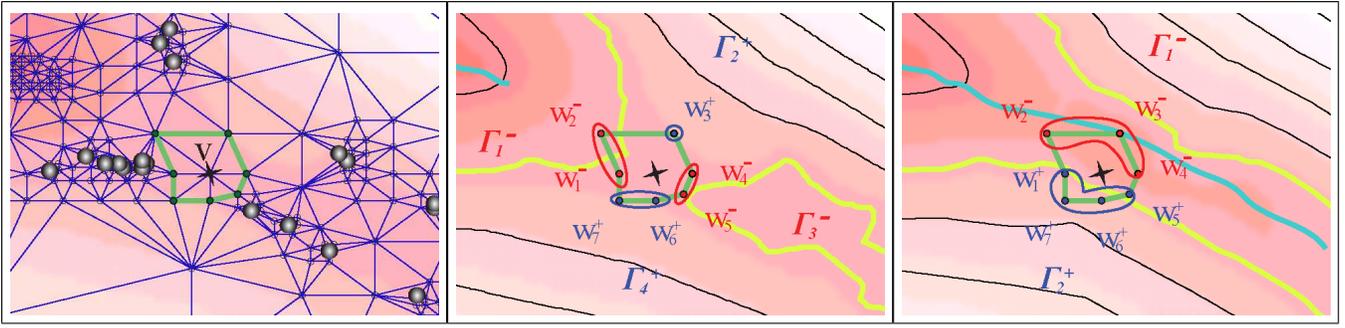
## 5 Detection of Topological Stability

The computation of the implicit function described above is designed to incorporate user information through inside/outside constraints. To avoid the laborious task of defining constraints everywhere, the system automatically detects weak regions of unstable topology. The definition of a weak region is quite intuitive. Its theoretical foundation and its computation are described next.

**Weak regions.** We define a weak region as part of the implicit function's domain  $\Omega$  where the local topology of the object is unstable. That is, little perturbations of the data lead to change in the local surface topology. This instability implies ambiguities that need to be solved by the user. These weak regions are generally



**Figure 6:** Weak region for a 2D field: the red (resp. blue) line is the  $+\epsilon$  (resp.  $-\epsilon$ ) level-set for  $u$ : this small level perturbation changes the level set connectivity, characterizing a critical point.



**Figure 7:** Weak regions computation, demonstrated on part of the hummingbird example from Figures 3 and 4. The link (in green) of vertex  $v$  in our adaptive mesh (left) is composed of two sets of vertices: negative  $\{w_1^-, w_2^-, w_4^-, w_5^-\}$ , and positive  $\{w_3^+, w_6^+, w_7^+\}$  (center), distributed in four connected groups  $\Gamma_1^-, \Gamma_3^-$  and  $\Gamma_2^+, \Gamma_4^+$  (blue and red ellipses). Counting a total of four ( $\geq 3$ )  $\Gamma_i^\pm$ 's, the vertex  $v$  is critical. Adding a constraint loosely along the critical line connecting  $\Gamma_1^-$  to  $\Gamma_3^-$  changes the values around  $v$  (right), making it a regular point (two  $\Gamma_i^\pm$ 's).

due to the low resolution of the underlying structure [Stander and Hart 1997] (like the hummingbird head on Figures 3, 4 and 7), to missing parts or to the intrinsic complexity of the shape (like in the head of the elephant on Figure 2).

To build a computable definition, we say that a point  $p$  is *critical* if, for an arbitrarily small  $\varepsilon$ , the  $u(p) - \varepsilon$  and  $u(p) + \varepsilon$  level-set surfaces have different topologies in a neighborhood of  $p$  (see Figure 6). The *weak regions* are the regions of these topological changes. For small values of  $|u(p)|$  they mark topological instabilities in the reconstructed surface (zero level-set of  $u$ ) induced by small level shifts. Moreover, this definition addresses the small gradient issue discussed in the Section 4.

A direct computation of weak regions by applying more complex random perturbations may generate critical regions everywhere, which would require stochastic simulations to select between them. Our approach uses a deterministic detection of the local stability by analyzing the topology under level shifts. This directly relates to strong mathematical notions derived from Morse theory [Milnor 1963]. Morse proved that for smooth, regular implicit functions  $u(p)$ , these critical regions are points  $p$  where the gradient vanishes:  $\vec{\nabla}u(p) = \vec{0}$ . This equivalence makes computable the abstract definition of weak regions, as described next.

**Discrete critical points.** The topological changes of smooth, regular implicit surfaces occur at critical points. Based on further structural properties of Morse's critical points, Banchoff [1967] introduced an equivalent definition for these critical points on polyhedral mesh, where the implicit function  $u$  is discrete, given only at the mesh vertices and linearly interpolated on the faces of the mesh. This definition is summarized as follows:

To determine whether a vertex  $v$  of a tetrahedral mesh is critical, we consider its *link graph*, whose vertices  $w_i$  are connected to  $v$  and whose edges connect adjacent  $w_i$ 's (see Figure 7 (left) for a 2D example). We partition the  $w_i$ 's into two sets, one of *positive* vertices, for which  $u(w_i) > u(v)$  and one of *negative* vertices, for which  $u(w_i) \leq u(v)$ . Each set, connected by the edges of the link graph, is composed of several connected groups  $\Gamma_i^\pm$  (see Figure 7 (center)). If the total number of positive and negative connected groups is 1 or above 3, the vertex is critical. On 2D meshes, a critical point has typically one or four connected groups. On 3D meshes, they have typically one or three connected groups.

The level-set at  $u(v) - \varepsilon$  (resp.  $u(v) + \varepsilon$ ) is contained in the negative (resp. positive) sets of the link of  $v$ . If these sets are made of several connected groups, the level-shift necessarily causes the level-set to "jump" between the connected groups  $\Gamma_i^\pm$ , inducing a topological change (see zero level-set (yellow) in Figure 7 (right)). Since we assume that the original object is connected, we ignore topological

changes that would create isolated components, i.e., critical vertices where the total number of  $\Gamma_i^\pm$  equals 1. We further discard the critical points  $v$  of high value  $|u(v)|$ , defining weak regions only close to the zero level-set of  $u$ , which will be the reconstructed surface.

**User interface.** After having related the weak regions to the formal definition of critical points, the remaining challenge is to provide, for each weak region, a 2D tablet over which the user can draw scribbles to correct or reinforce the topology of the zero level set. A correction either connects or disconnects two parts of the object (see Figure 7 (right)). In the critical points definition, connecting two parts of the implicit surface corresponds to merging two connected groups  $\Gamma_i^\pm$  of the same sign set. Similarly, disconnecting corresponds to connecting the complementary object. This requires drawing a constraint loosely across the *critical line*, i.e. the line joining the barycenter of each group.

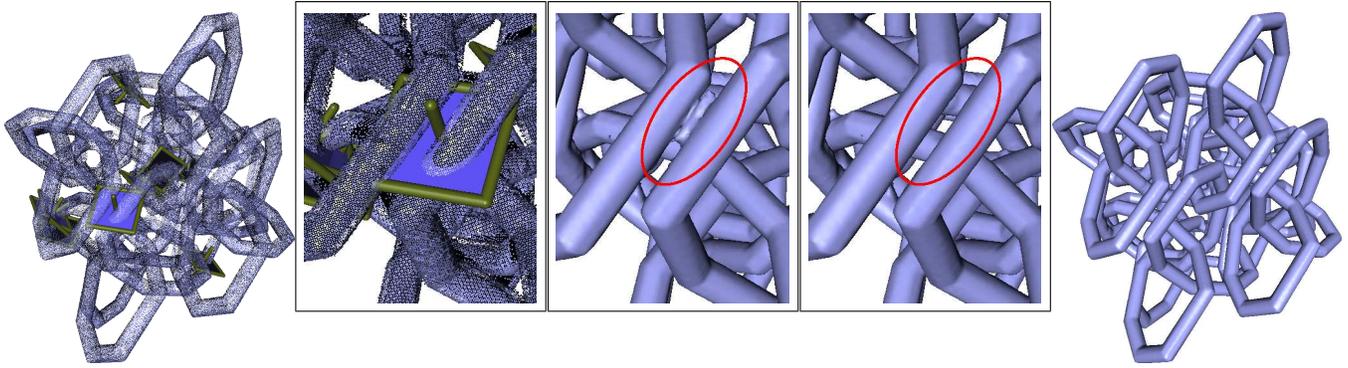
Consequently in the interface, tablets are located at critical points, oriented perpendicularly to the critical lines (see Figure 8). Note that with volumetric meshes, the common critical points have three  $\Gamma_i^\pm$ , defining a single critical line, and thus one tablet.

## 6 Technical Details

**The dual hierarchical graph.** The scanned data is given by a set of range images, where each data point is associated with the scanner position. The range images are registered [Levoy et al. 2000] to form a point cloud, which is structured by a hierarchical domain that tessellates the 3D space.

We use a tetrahedral mesh, which is generated from the dual of a dynamically adapted octree. This dual allows a better representation of continuous functions, as required by the FEM and the topological analysis. Each cell of the dual is triangulated with a static pattern of 6-tetrahedrons cube decomposition. To save memory, we keep only the edge graph of the tetrahedrons. A 2D illustration of this data structure is shown in Figure 3 (right). The initial octree is adapted just to separate the data points, restricting its maximum depth to five. Next, following user validation of weak regions, the octree is iteratively refined close to the zero level-set of the implicit function until reaching a visually-validated local feature size.

**Automatic initial reconstruction.** To initiate the reconstruction process, we use few, automatically generated, loose inside/outside constraints. The location and the sign of these initial automatic constraints can be computed heuristically. In this work we simply compute an unsigned distance transform from the data points by fast marching over the hierarchy. We use local maxima, detected by examining field gradients in the neighborhood of an octree cell, as



**Figure 8:** In complex shapes automatic detection of weak regions is necessary to guide the user: original scan with the weak regions (left), reconstruction without scribbles generates some spurious connections (center), which are removed after the user adds scribbles (right).

automatic constrains (in Figure 3(left) automatic constrains were extracted from the 2D medial-axis for illustration purposes). We automatically classify constrains as inside/outside applying a simplified space carving method [Curless and Levoy 1996]. Although this process is prone to errors, it serves only as an initial coarse guess for the function  $u$  as described in Section 4. This technique actually performs an initial reconstruction similar to automatic reconstruction based on global optimizations such as [Carr et al. 2001; Ohtake et al. 2003; Kazhdan et al. 2006] (see Figure 9), without using the normals of the input scan. In our experiments, this process could be easily skipped using only manual scribbles.

**User interaction.** The user can visualize arbitrary 2D cross sections of the implicit function using a pseudo-color map (see Figure 1 (center-left)). The user picks one of the cross sections which are displayed at the weak regions (see Section 5). This cross section of the field is reproduced in a separate window, which we call a *tablet*, over which the user draws the scribbles. The user is not required to precisely position the scribbles, but rather loosely define inside and outside relations locally at the weak regions. These in/out scribbles are inserted as constraints to the FEM system, with negative/positive sign and value according to the distance field.

With each additional constraint, the field and the weak regions are updated within less than a second. This interactivity relies on Equation (4), which allows a pre-factorization of the FEM matrix. As the user adds more scribbles, the topological stability of the implicit function locally increases. After the user’s validation, the octree is locally refined close to the zero level-set of the implicit function, and the FEM matrix is pre-factorized again. Our non-smoothness penalty constraint avoids spurious topological instabilities (as also observed in [Ni et al. 2004]), while incorporating the scribbles constraints. The whole process takes between a few seconds and a few minutes, depending on the octree depth and the shape complexity.

Note that except for the scribbles drawing, the user is not required for any parameter tuning. The critical points filtering is determined by the octree depth ( $|u(v)| < 8 \cdot 2^{-depth}$ ), and the relative weights  $\omega_p$  in Equation (4) are fixed to a low value (0.01) for the initial constraints, medium (1) for the data points and high (1000) for the user scribbles.

**Final surface reconstruction.** Once the implicit function achieves the expected topology, a final mesh is extracted from its zero level-set. Since the field is smooth everywhere (see Figure 10), we can use any isosurfacing method. We choose the dual marching cubes method [Schaefer and Warren 2004] since it guarantees the resulting topology, and since it works on the same data structure as our octree dual. We further improve mesh quality using standard mesh optimization techniques (edge flips/collapse and normal smoothing) that do not alter the shape topology.

## 7 Results

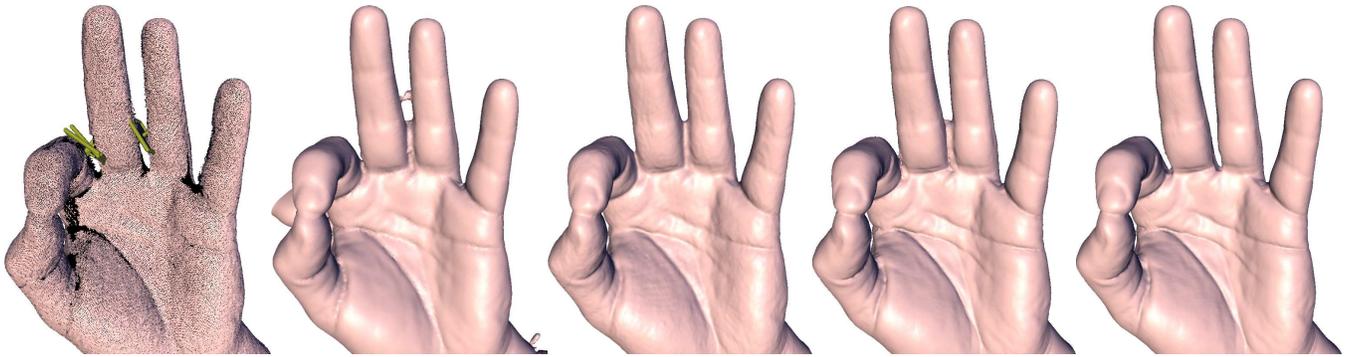
To demonstrate the effectiveness of the proposed method, we focused on complex objects, such as the riding monk (Figure 1), which we acquired with few structured light scans. This relatively inexpensive technology has the advantage of being fast: each model was scanned from 6 to 12 shots in less than five minutes. Our experiments exhibit the difficulties that are common to various scanning techniques: uncoverable areas as the elephant’s trunk in Figure 2 and the camel legs in Figure 13; highlights in the sitting woman in Figure 14; object’s material as the riding monk’s black areas in Figure 1; and shadows in the tiger’s head in Figure 12. For illustration purposes, we rendered the unoriented raw scans using normals computed from our reconstruction. For a demonstration of the interactivity of our system, we refer to the accompanying video.

To further analyze the specific characteristics of our technique, we consider feature-specific models, generated by an OpenGL-based virtual scanner. As we can see in the triple Möbius band (Figure 10), our FEM formulation yields a smooth field everywhere, even for thin shapes. Furthermore, our topological analysis detects all the weak regions to allow a complete reconstruction, even on models as complex as the knot (Figure 8). We also demonstrate the effectiveness of the interaction on virtual scans of real models such as the hand (Figure 9) and the hip (Figure 11).

Our method can handle input data with very large missing parts and outliers. The smoothness constraint of the FEM distinguishes gracefully between frontiers of under-sampled regions and outliers as in the tiger’s head (Figure 12). In such cases, the smoothness

	# points	# shots	Auto-matic	Inter-action	# scribbles
Riding Monk	469 k	10	130 s.	3 min	9
Elephant	217 k	6	88 s.	2 min	7
Knot	497 k	15	206 s.	4 min	10
Hand	259 k	8	92 s.	30 s.	2
Saddle	284 k	11	118 s.	0 min	0
Hip	222 k	9	109 s.	30 s.	4
Tiger	340 k	10	157 s.	6 min	12
Woman	333 k	9	116 s.	5 min	12
Camel	282 k	12	128 s.	3 min	6

**Table 1:** Reconstruction timings for our scanned models. From left to right: the number of data points, the number of structured light shots, the time of the automatic FEM reconstruction, the interaction time of the reconstruction session and the number of scribbles used.

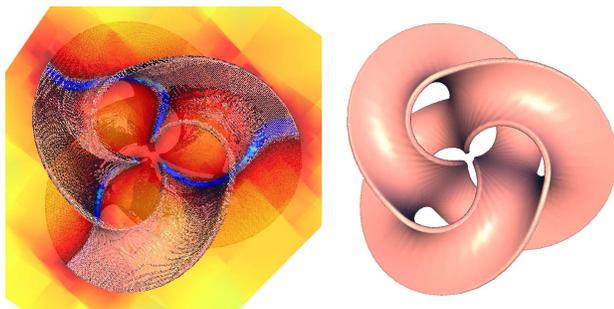


**Figure 9:** Comparing our automatic initial reconstruction with other implicit techniques: (from left to right) the original scan with our weak regions detection; reconstruction using MPU; using Poisson surface reconstruction; our FEM technique without any user interaction; our interactive reconstruction with two scribbles only.

criteria of the FEM, smoothly completes the missing parts and discards outliers, resulting in a watertight manifold surface. In similar noisy situations, the topological analysis accurately detects weak regions such as the arm-leg contacts of the woman in Figure 14. Furthermore, user’s scribble constraints can generate coherent geometries in under-sampled regions such as the camel’s legs (Figure 13), while guaranteeing watertightness and avoiding spurious connected components.

We compare the automatic part of our reconstruction technique with the implicit reconstruction techniques of [Ohtake et al. 2003] and [Kazhdan et al. 2006]. As shown in Figure 9 for a clean data set, the results are of comparable quality. The timings of the three methods are about the same, e.g., for this hand model, [Ohtake et al. 2003] performed in 141 seconds at octree depth 10, [Kazhdan et al. 2006] in 273 seconds for octree depth 10, and our initial reconstruction in 92 seconds at depth 8 (see Table 1). Note that our method does not require the data points’ normals. For the sake of this comparison, we provided the normals of our reconstructed surface to the other implicit techniques. Furthermore, the addition of only two scribbles recovers the correct model.

As a limitation, this work focuses only on the topological aspects of the reconstruction and not specifically on its geometry. The constraints we associate to the scribbles correct the local topology of the shape. Our scribbles merely indicate whether a region is inside or outside the shape. Therefore, this mechanism does not precisely define the geometry of the resulting shape. In particular, it completes smoothly large missing parts (such as the woman’s shoulder in Figure 14). Thus, the method does not deal with important issues such as the reconstruction of sharp features.



**Figure 10:** The smoothness penalty of the FEM formulation yields a field that is smooth everywhere, even for thin shapes.

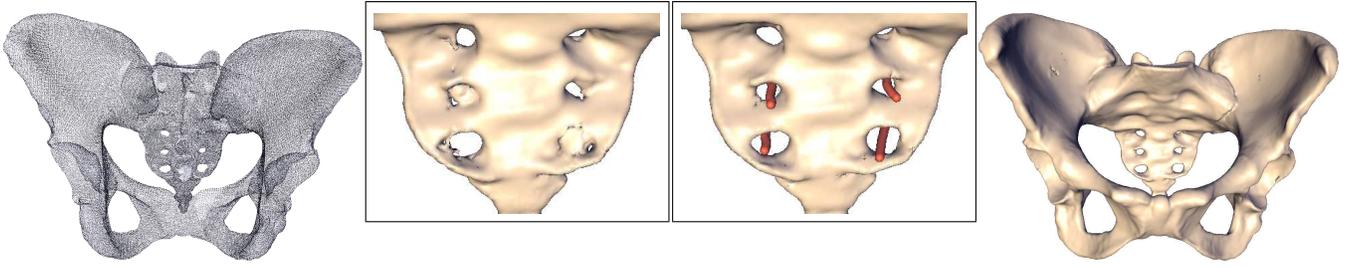
## 8 Conclusions

To conclude, we presented an interactive tool for surface reconstruction, where the user assists the interpretation of data at automatically detected topologically unstable regions. The use of inside/outside scribbles at these regions allows achieving the expected topology of the object. The reconstructed surface is faithful to the data points and guaranteed to be watertight even with noisy under-sampled data. Our experiments show that from only a dozen of structured light shots, we are able to reconstruct coherent models. In the future, we want to be able to guide the geometry of the reconstructed surface in the missing areas. This can be done in the same framework, by adding other semantics to scribbles, such as specifying additional data points, associating sharp features or smoothness.

**Acknowledgement.** We would like to thank Chaim Gotsman and Alexander Bogomjakov for the granted access to scan facilities. This work was partly supported by a grant from the Israeli Ministry of Science, and from the Brazilian Ministry of Science and Technology (MCT/CNPq 02/2006).

## References

- ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. 2001. Point set surfaces. In *Visualization*, IEEE, 21–28.
- AMENTA, N., AND KIL, Y. J. 2004. Defining point-set surfaces. In *Siggraph*, ACM, 264–270.
- AMENTA, N., BERN, M. W., AND KAMVYSSELIS, M. K. 1998. Crust: A new Voronoi-based surface reconstruction algorithm. In *Siggraph*, ACM, 415–422.
- BANCHOFF, T. 1967. Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry* 1, 257–268.
- BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. 1999. The Ball-Pivoting Algorithm for surface reconstruction. *Transactions on Visualization and Computer Graphics* 5, 4, 349–359.
- BOISSONNAT, J.-D., AND CAZALS, F. 2002. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry: Theory and Applications* 22, 1–3, 185–203.
- CARR, J., BEATSON, R., CHERRIE, J., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Siggraph*, ACM, 67–76.



**Figure 11:** The addition of scribbles in the weak regions allows a faithful reconstruction: the original scan (left), our FEM reconstruction without scribble (center-left) and with four scribbles to reinforce and correct the initial reconstruction (right).

- CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Siggraph*, ACM, 303–312.
- DAVIS, T. A., AND HAGER, W. W. 2005. Row modifications of a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications* 26, 3, 621–639.
- DAVIS, J., MARSCHNER, S., GARR, M., AND LEVOY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In *Symposium on 3D Data Processing, Visualization, and Transmission*, University of Padova.
- DEY, T. K., AND GOSWAMI, S. 2004. Provable surface reconstruction from noisy samples. In *Symposium on Computational Geometry*, ACM, 330–339.
- HOPPE, H., DE ROSE, T., DUCHAMP, T., McDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *Siggraph*, ACM, 71–78.
- HORNUNG, A., AND KOBBELT, L. 2006. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Symposium on Geometry Processing*, ACM/Eurographics, 41–50.
- HUGHES, T. J. R. 1987. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall.
- KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Symposium on Geometry Processing*, ACM/Eurographics, 61–70.
- KOLLURI, R., SHEWCHUK, J. R., AND O’BIEN, J. F. 2004. Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, ACM Press, 11–21.
- LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. In *Siggraph*, ACM, 689–694.
- LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital Michelangelo project: 3D scanning of large statues. In *Siggraph*, ACM, 131–144.
- LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. *Transaction on Graphics* 23, 3, 303–308.
- MILNOR, J. W. 1963. *Morse theory*. Princeton University Press.
- NEALEN, A., SORKINE, O., ALEXA, M., AND COHEN-OR, D. 2005. A sketch-based interface for detail-preserving mesh editing. *Transactions on Graphics* 24, 3, 1142–1147.
- NI, X., GARLAND, M., AND HART, J. C. 2004. Fair Morse functions for extracting the topological structure of a surface mesh. In *Siggraph*, ACM, 613–622.
- OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. 2003. Multi-level partition of unity implicits. *Transaction on Graphics* 22, 3, 463–470.
- OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. A multi-scale approach to 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Solid Modeling International*, IEEE, 31–39.
- SCHAEFER, S., AND WARREN, J. 2004. Dual marching cubes: Primal contouring of dual grids. In *Pacific Graphics*, IEEE, 70–76.
- SHARF, A., LEWINER, T., SHAMIR, A., KOBBELT, L., AND COHEN-OR, D. 2006. Competing fronts for coarse-to-fine surface reconstruction. In *Eurographics*, Eurographics, 389–398.
- STANDER, B. T., AND HART, J. C. 1997. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. In *Siggraph*, ACM, 279–286.
- STRANG, G. 1986. *Introduction to Applied Mathematics*. Wellesley-Cambridge.
- WANG, J., AND COHEN, M. F. 2005. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, IEEE, 936–943.

## Appendix: Construction of the matrix $K$

The construction of  $K$  is a standard finite-elements one, based on [Hughes 1987] but simplified to our particular case and made completely explicit.

Let  $\{(x_i, y_i, z_i)\}_{i=1}^n$  be the coordinates of the vertices of the mesh and let  $\{(n_j^{(1)}, n_j^{(2)}, n_j^{(3)}, n_j^{(4)})\}_{j=1}^m$  be the vertices’ indices of the tetrahedrons in the mesh. For each tetrahedron (linear tetrahedral element) we compute the 4-by-4 matrix:

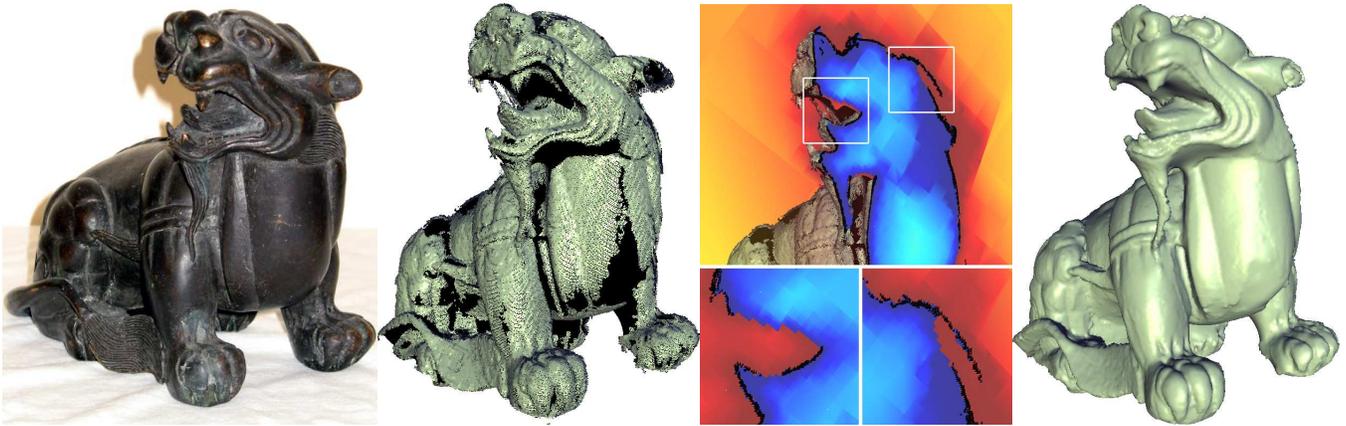
$$K_j = \frac{|\det(J_j)|}{6} E_j^* E_j, \quad \text{where}$$

$$J_j = \begin{bmatrix} x_{n_j^{(1)}} - x_{n_j^{(4)}} & x_{n_j^{(2)}} - x_{n_j^{(4)}} & x_{n_j^{(3)}} - x_{n_j^{(4)}} \\ y_{n_j^{(1)}} - y_{n_j^{(4)}} & y_{n_j^{(2)}} - y_{n_j^{(4)}} & y_{n_j^{(3)}} - y_{n_j^{(4)}} \\ z_{n_j^{(1)}} - z_{n_j^{(4)}} & z_{n_j^{(2)}} - z_{n_j^{(4)}} & z_{n_j^{(3)}} - z_{n_j^{(4)}} \end{bmatrix},$$

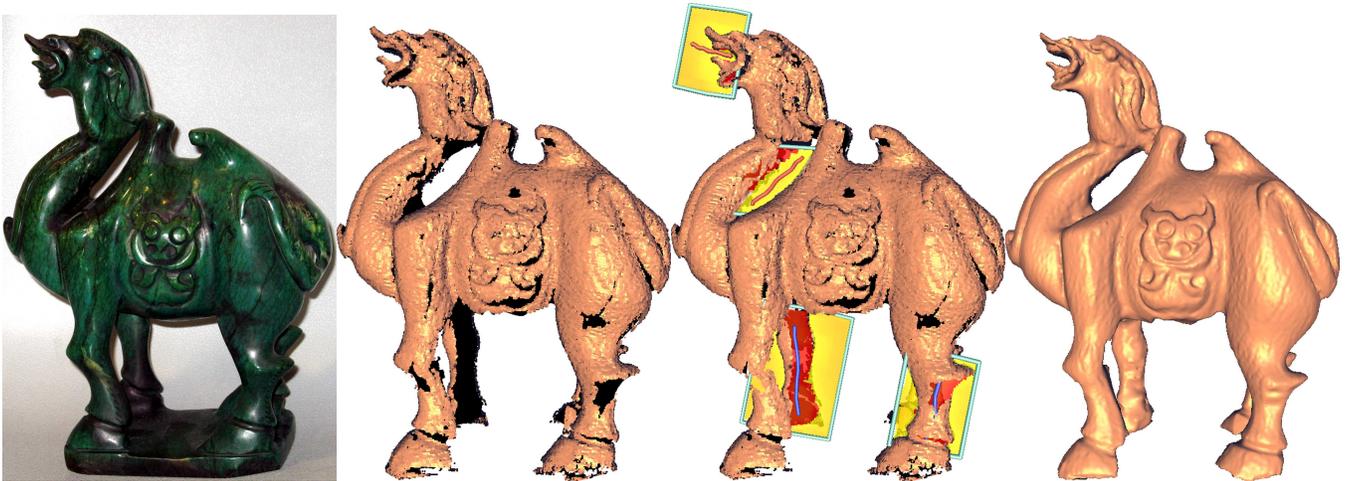
and  $E_j$  is the 3-by-4 matrix solution of the following linear system:

$$J_j^* E_j = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

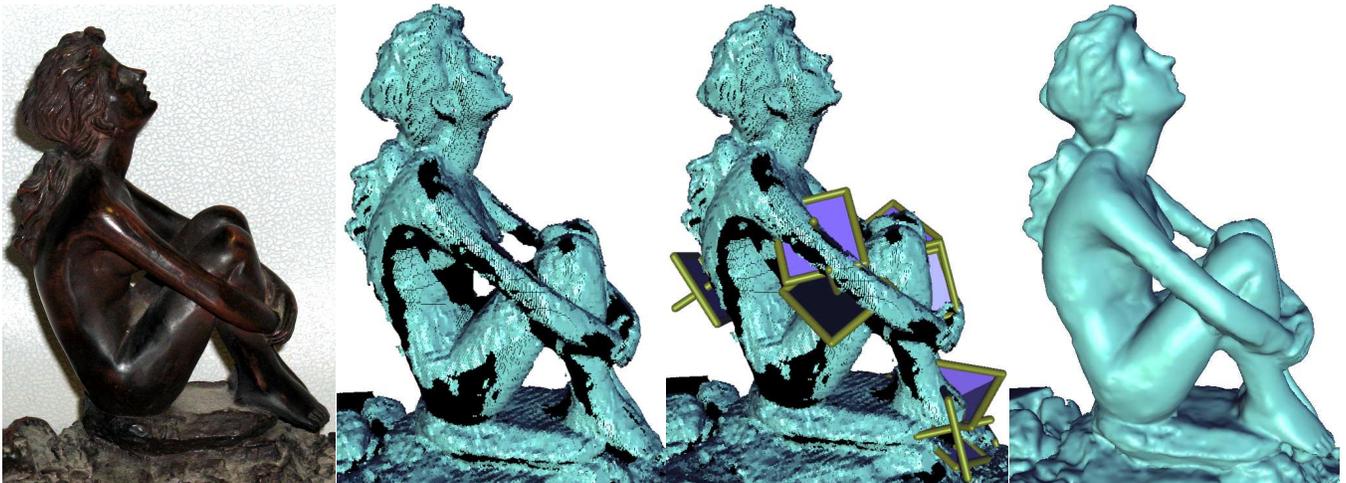
We define  $\hat{K}_j$  to be an  $n$ -by- $n$  symmetric matrix which is zero except for rows and columns  $n_j^{(1)}, n_j^{(2)}, n_j^{(3)}, n_j^{(4)}$ , where  $\hat{K}_j(n_j^{(a)}, n_j^{(b)}) = K_j(a, b)$ . Finally, we sum the  $\hat{K}_j$  to generate  $K = \sum_{j=1}^m \hat{K}_j$ .



**Figure 12:** The smoothness penalty discards outliers and completes smoothly missing parts, as can be seen in the tiger's head. From left to right: the original statue, the raw data points (the missing parts are in black), the FEM field with two close-ups on outliers and large missing parts in the head, and our interactive reconstruction of the model with 12 scribbles.



**Figure 13:** Due to the statue's base (left), the camel's legs cannot be covered (center-left). Scribble constraints drawn on automatically generated 2D tablets (center-right) can be used to generate a coherent geometry (right).



**Figure 14:** The topological analysis detects weak regions (center-right) in the presence of noise. Weak regions are typically located where two parts of the shape are close and the topology is ambiguous, such as the arm-leg contacts in the sitting woman. Missing parts with unambiguous topology, such as her right shoulder, are automatically smoothly completed.