Reconstruction with Voronoi Centered Radial Basis Functions

Marie Samozino, Marc Alexa, Pierre Alliez, Mariette Yvinec

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Reconstruction with Voronoi Centered Radial Basis Functions

Marie Samozino — Marc Alexa — Pierre Alliez — Mariette Yvinec

N° 6033

Novembre 2006

Thème SYM

ISSN 0249-6399 ISRN INRIA/RR--6033--FR+ENG

Reconstruction with Voronoi Centered Radial Basis Functions

Marie Samozino*, Marc Alexa†, Pierre Alliez* , Mariette Yvinec*

Thème SYM — Systèmes symboliques
Projet Geometrica

Rapport de recherche n° 6033 — Novembre 2006 — 6055 pages

Abstract: We consider the problem of reconstructing a surface from scattered points sampled on a physical shape. The sampled shape is approximated as the zero level set of a function. This function is defined as a linear combination of compactly supported radial basis functions. We depart from previous work by using as centers of basis functions a set of points located on an estimate of the medial axis, instead of the input data points. Those centers are selected among the vertices of the Voronoi diagram of the sample data points. Being a Voronoi vertex, each center is associated with a maximal empty ball. We use the radius of this ball to adapt the support of each radial basis function. Our method can fit a user-defined budget of centers: The selected subset of Voronoi vertices is filtered using the notion of lambda medial axis, then clustered to fit the allocated budget.

Key-words: Computer Graphics, Surface Reconstruction from Scattered Data, Radial Basis Functions, implicit function.

* INRIA Sophia-Antipolis, France
† TU Berlin

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Reconstruction à l'aide de fonction de bases radiales centrées sur des sommets de Voronoi

Résumé : Nous nous intéressons au problème de reconstruction de surface à partir de données éparses mesurées sur un object réel. La surface de la forme échantillonnée est approximée comme le niveau zéro d'une fonction. Cette fonction est définie comme une combinaison linéaire de fonctions radiales à support compact. Nous nous démarquons des travaux précédents en localisant les centres des fonctions radiales prés du médial axis au lieu de les placer sur les points de données. Nos centres sont choisit parmi un sous ensemble des sommets du diagramme de Voronoi des points de données. Chaque centre, étant un sommet de Voronoi, peut être associé au rayon de la sphere maximale centrée sur lui. Nous utilisons ce rayon pour adapter le support de chaque fonction de base. Notre méthode est paramétrée par le nombre de centres désiré : l'ensemble des sommets de Voronoi sélectionnés est filtré, en utilisant la notion de λ-medial axis, puis clusteré pour ne sélectionner, à la fin, que le nombre de centres définit par l'utilisateur.

Mots-clés : Reconstruction de surface, fonctions de base radiales, ensemble de points éparses, fonction implicite

1 Introduction

Recent improvements in automated shape acquisition has stimulated a profusion of surface reconstruction techniques over the past few years for computer graphics and reverse engineering applications. Data collected from scanning processes of physical objects are often provided as large point sets.

Reconstruction methods can be roughly classified as Voronoi-based and mesh-free. Voronoi based reconstruction algorithms compute the Delaunay triangulation of the sample points, the dual to the Voronoi diagram. A subcomplex interpolating the sampled surface is then extracted from the Delaunay triangulation [AGJ02, AB98, AS00, CSD04, DGH01, DG04, ACK01, KSO04]. Detailed surveys are presented in [CG04, Dey04]. In the mesh-free approaches, the surface is approximated or interpolated using explicit methods such as deformable models [DQ01, Set99], parametric methods such as NURBS, B-Spline [Far02] or implicit methods such as RBF or MLS (see [TO02] for a survey). Among the many techniques developed for surface reconstruction with implicit methods, the radial basis functions (RBF) approach has shown successful at reconstructing surfaces from point sets scattered on surfaces of arbitrary topology [Buh03, Duc77, Isk04, Wen04].

Radial Basis Functions (RBF) were introduced by Broomhead and Lowe in the neural network community [BL88]. Techniques based on radial basis functions are now common tools for geometric data analysis [FN80, LF99], pattern recognition [Kir01] and statistical learning [HTF01]. The radial basis functions approach is volumetric in the sense that it approximates the input surface as the zero level-set of a scalar 3D function. This function is expressed as a weighted sum of radial functions, whose centers commonly coincide with the input data points. The function is constrained to be zero on the input data points and to be non-zero on other points in order to avoid the trivial constant solution. Given a set of centers, a set of constraint points and a type of radial basis function, reconstructing the sampled surface amounts to finding the set of weights which minimize a least-squares error to fit the constraints.

Although Voronoi-based reconstruction has long been criticized for its computational burden, recent developments in the implementation of fast algorithms have alleviated this issue. As an example, computing the Delaunay of 50K points takes 1s using the CGAL library [FGK*00]. Such methods still depend on the quality of the sampling and on the differential and topological properties of the surface. In particular, sparsity, redundancy, noisiness of the sampling, or non-smoothness and boundaries of the surface makes the surface reconstruction a challenging problem. Besides, Voronoi-based reconstruction methods often fail to produce watertight surfaces.

Radial basis functions, on the other hand, still have issues with picking the right non-zero constraints (to avoid disconnected components), and with efficiently computing the weights. Functions with unbounded support give the best reconstruction results, but also lead to dense matrices. The only viable solution to this problem so far is the multipole expansion for polyharmonic functions developed by Carr et al. [CFB97]. Unfortunately this approach is notoriously intricate and difficult to reproduce. Compactly supported functions lead to sparse matrices [Wen95]. However, finding a proper support size for the functions in case of irregularly sampled surfaces is difficult. A recent trend is to perform a set of local reconstructions, which may be mixed with quadric or higher-order

jet fitting, and to blend them using the partition of unity [TI04, OBS04]. Although a great deal of effort has been put into the elaboration of multi-level techniques with local reconstructions to deal with large data sets, less effort has been spent at improving the compactness of the representation by center selection and optimization [CFB97, TI04, OBS04]. Besides, when the basis functions is compactly supported, the computed function is only defined in the vicinity of the input data points.

1.1 Contributions

Our approach combines both worlds and eliminates some of the aforementioned shortcomings. The sampled surface S is still reconstructed as the zero-level of a function f expressed as a linear combination of radial basis functions. The main advance in our method is to use radial basis functions centered at vertices of the Voronoi diagram of the data points. More precisely, centers of radial basis functions are chosen among a subset of those Voronoi vertices, which are called poles. Under certain sampling conditions, the poles are known to be closed to the medial axis of the sampled surface S [AB98]. Furthermore, each pole is the center of a Delaunay ball hereafter called polar ball. A polar ball is a maximal ball empty of sampled points. Such a ball is close to a maximal ball in $\mathbb{R}^3 \setminus S$. Considering that any smooth surface S can be viewed as the envelope of the maximal balls in $\mathbb{R}^3 \setminus S$, using poles as centers for radial basis functions is a rather natural idea. Furthermore, in our reconstruction process, we use the radius of each polar ball as a guidance for choosing the support of the corresponding basis functions. Hence, the support of each basis functions is locally adapted to the geometry and topology of the sampled shape. Also, because the radius of each polar ball is a good estimate of the distance between the pole and the sampled surface, we use this radius to set, as additional constraints, the value of the function at the poles. This leads to a reconstruction technique with the following features:

- The surface is represented as the zero-level set of a signed function, which is a good approximation of the signed distance field to the surface.
- The function is defined as a weighted combination of locally supported radial functions; The number of functions is independent from the number of input points and typically significantly smaller. The function can thus be evaluated faster than when using traditional (even compactly supported) RBF.
- While the computation of the weights potentially incorporates all data points as constraints, the size of the system matrix only depends on the number of centers, not on the number of constraints.
- A filtering of the poles based on the notion of λ-medial axis allows the surface to degrade gracefully with noise.

In comparison with Voronoi-based reconstruction, the most important advantages of our technique are the resilience to noise and the construction of a smooth watertight surface that approximates all data points. In comparison to the common compactly supported RBF, fewer centers are used for the same accuracy. This leads to faster computation of the weights and faster evaluation of

the functions. Using poles associated with their Voronoi ball radius as additional constraints leads to a better approximation of the distance field to the surface, and to fewer topological issues such as superfluous connected components away from the input points.

1.2 Overview

Our algorithm proceeds as follows: given a 3D point set scattered on a surface, we first compute its Delaunay triangulation and the dual Voronoi diagram. Our algorithm then repeatedly refines a subset of the Voronoi vertices. In the first stage, poles are extracted from the Voronoi vertices and are classified as inside or outside. In the second stage, we select a user-defined number of centers, m, among the set of poles. The selection is performed by filtering, then clustering the set of poles. Poles are filtered in order to adjust the level of detail to the budget of centers and clustered in order to achieve a center distribution nicely spread on the medial axis. We choose as radial basis function a Gaussian-like function with a compact support [Wen95], where the support size is locally adapted. As constraints, we impose the function f to be zero at the data points and to be non zero at the center points. A value set at a center point approximates the signed distance from this point to the sampled surface. The weights are obtained by computing the best least squares approximation of the function f with respect to the constraint points.

For completeness we list some key notions behind the radial basis functions in Section 2. Section 3 details the main steps of our algorithm. We show several experimental results in Section 4. Some work in progress and perspective directions are discussed in Section 5.

2 Background

Definition 1 *The approximation problem is formulated as follows. Given $\{p_i\}_{i=1\ldots n}$ a set of n points and n scalar numbers $F = \{f_i\}_{i=1\ldots n}$, find a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ satisfying the approximation condition:*

$$f^* = argmin_f E(f), \tag{1}$$

where E is the least squares error :

$$E(f) = \sum_{i=1}^{n} (f_i - f(p_i))^2 . \tag{2}$$

In the RBF approach, the function f is defined from a class of basis functions $\Phi_j : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, as a linear combination

$$f(x) = \sum_{j=1}^{m} \alpha_j \Phi_j(x, c_j), \tag{3}$$

where $\{c_j\}_{j=1\ldots m}$ denotes a set of m center points and $\{\alpha_j\}_{j=1\ldots m}$ denotes a set of unknown weights to be solved for.

The reconstruction problem boils down to determine the vector $\alpha = \{\alpha_1, \ldots, \alpha_n\}$, by solving a linear system of equations resulting from the minimization of E (Eq.2) :

$$\alpha = \left[G^t_{P,\Phi} G_{P,\Phi}\right]^{-1} G^t_{P,\Phi} F, \tag{4}$$

where matrix $G_{P,\Phi} = [\Phi(p_i, c_j)]_{i=1..n, j=1..m}$ and $F = [f_i]_{i=1..n}$.

In the following, the set of points, P, where the function value is specified a priori are called *constraints*. The set P includes the data points where the function f should vanish by definition, i.e. where all the f_i should be zero. To avoid the trivial solution $\alpha = \overrightarrow{0}$, during the minimization of E in (2), several interior and exterior constraints are added where the function f does not vanish. For each additional constraint point p_k, we assign to f a signed value f_k. This value is commonly the approximated signed distance between p_k and the sampled surface. The N constraints $\{p_i\}_{i=1..N}$ are now composed of the n input points and of the additional off-surface constraints where the function f is specified.

Most approaches locate centers both at the input data points and at the off-surface constraints, therefore the number of centers is such that $m = N$ and the minimization process (1) reduces to solving a $N \times N$ linear system which requires $O(N^3)$ machine operations and $0(N^2)$ bits for storage. Then, each evaluation of $f(x)$ requires $O(N)$ operations. This approach is therefore not suitable to a number of constraints greater than several thousands. To reduce the computational complexity, one first idea is to reduce the number of constraints. Notice that since most algorithms use the same points as constraints and as centers, this also leads to center reduction. This approach is commonly called *center reduction* in the literature.

Center reduction consists of optimizing the trade-off between fitting accuracy and number of centers. A greedy algorithm is proposed in [CBC*01]: centers are iteratively added at locations where the fitting error is maximum until a satisfactory accuracy is reached. Another idea to further reduce the number of centers while maintaining decent fitting accuracy is to relax the one-to-one correspondence between the centers and the constraints. This approach, which we follow in this paper, is called *Generalized Radial Basis Functions* (GRBF) in the neural networks community [PG89]. Let m be a user-defined number of centers, possibly located anywhere in space, and N the number of constraints, such that $m << N$. The size of the matrix to be inverted and stored is now $m \times m$, independently of the number of constraints. $O(m)$ operations are now required for a single point-wise evaluation. Each term of the matrix $G^t G$ being a sum of contributions arising from each constraint, the number of constraints conditions the cost for assembling the matrix. This paper investigates one of the most important degrees of freedom offered by the RBF method: the location of centers and constraints to obtained a satisfactory trade-off between number of centers and fitting accuracy.

3 Algorithm

The input data for our algorithm is a point set $P = \{p_i\}_{i=1..n} \subset \mathbb{R}^3$. All the input data points are supposed to lie on the surface so the function value f is set to zero a these points:

$$f_i = f(p_i) = 0, \qquad \forall i = 1 \ldots n. \tag{5}$$

We structure this section by the main components of the reconstruction algorithm, namely the choices made for the centers, for the constraints and for the radial basis functions.

3.1 Centers

Centers for RBFs are selected from the vertices of the Voronoi diagram of the input points. Selection is performed by refining a set of candidates in three steps.

Pole Extraction Let $\mathscr{O}$ be a shape with a closed continuous boundary $S = \partial\mathscr{O}$. A ball $\mathscr{B}$, included in $\mathbb{R}^3/S$, is said to be a maximal ball if there exists no other ball included in $\mathbb{R}^3/S$ and containing $\mathscr{B}$.

Definition 2 *Medial axis:*
The medial axis M of S is the topological closure of the set of points of $\mathbb{R}^3$ that have at least two nearest neighbors on S. Every point in M is the center of a maximal ball.

Definition 3 *Voronoi diagram:*
The Voronoi diagram of a point set P is a partition of the space in regions with the same closest point in P. Every Voronoi cell corresponds to exactly one point p_i and contains all points in the space that are closer to p_i than to any other points in P.

$$V(p_i, P) = \{x \in \Omega : \forall q \in P \quad \|x - p_i\| \leq \|x - q\|\}. \tag{6}$$

For the problem of reconstructing surfaces from point sets, we assume that all points are sampled on the surface. In 2D, it has been shown that if the sample is dense enough, all Voronoi vertices are closed to the medial axis. However, a similar result does not hold in 3D, where some Voronoi vertices may be located close to the surface and thus far from the medial axis, even when the sample density goes to infinity. The notion of a *pole* was previously introduced to handle this problem.

Definition 4 *Pole:*
A vertex of the Voronoi cell, $V(p_i, P)$, of a sample point $p_i \in P$ is called a pole if :

- *either it is the vertex v_i of $V(p_i)$ that is the farthest from p_i*
- *or it is the vertex w_i of $V(p_i, P)$ that is the farthest from p_i in the halfspace H_i^-, set of points x such that $(v_i - p_i) \cdot (x - p_i) \leq 0$.*

As a pole is a Voronoi vertex, there exists a maximal ball centered at each pole. This ball is called a *polar ball*. Amenta et al. [ACK01] and Boissonnat and Cazals [BC00] show that under some conditions the poles are close to the medial axis of the sampled shape. The conditions are that the surface is smooth and the sampling is dense enough. More precisely, the sample has to be an ε-sample. This means that for any point, x, on the surface, the distance from x to the sample is not greater than ε times the distance from x to the medial axis. The poles are shown to exhibit interesting properties:

- if v_i is a pole of the cell $V(p_i,P)$, the direction $\overline{v_i p_i}$ is a good approximation of the normal at p_i;
- the radius of the Delaunay ball centered at v_i is a good approximation of the distance from v_i to the sampled surface.

Let m be the user-defined budget of centers. Generally, the number of poles is greater than m, and we must select m relevant poles as centers. If m is small, there is no hope to reconstruct very small details and thus we need to remove the poles which correspond to the smallest details (which are not distinguishable from noise). This task is performed by filtering the poles based on the notion of the λ-medial axis. Notice that this filtering stage is different from the clustering stage, which is designed to distribute the final budget of centers on the λ-medial axis with a proper sampling density.

Pole Filtering A major problem arises when trying to approximate the medial axis of a sampled shape from the Voronoi vertices of the data points: The medial axis is known to be highly unstable with respect to small details of the shape. This means that even if two objects are very close with respect to the Hausdorff distance, they may have very different medial axis (Fig.1). Thus, the set of poles extracted from the Voronoi diagram of a sampled surface is very unstable with respect to noise as well. Several approaches have been proposed to tackle this problem [AM96, DZ03]. In this paper we follow the recent work of Chazal and Lieutier [CL05], which defines the notion of λ-medial axis.

For any point p, we denote by $\Gamma(p)$ the set points of the boundary $\partial\mathscr{O}$ that are closest to p.

$$\Gamma(p) = \{y \in \partial\mathscr{O}, d(x,y) = d(x,\partial\mathscr{O})\}. \tag{7}$$

The medial axis M of $\mathscr{O}$ can be viewed as the set of points $x \in \mathscr{O}$ such that $|\Gamma(x)| > 2$. For each point p, there is a smallest ball enclosing $\Gamma(p)$. We define the real-valued function $\gamma(p)$ as the radius of the smallest ball enclosing $\Gamma(p)$. The λ-medial axis M_λ is defined as :

$$M_\lambda = \{p \in \mathscr{O} | \gamma(p) > \lambda\}. \tag{8}$$

M_λ is a closed subset of the medial axis. Moreover, the medial axis is obtained for $\lambda = 0$. Chazal and Lieutier have shown that for any value for λ which is not a singular value of the map $\lambda \longmapsto M_\lambda$, the λ-medial axis of a surface is stable under small perturbations and can be estimated from a dense sampling. Roughly speaking, restricting the λ-medial axis with increasing value of λ, smooths out

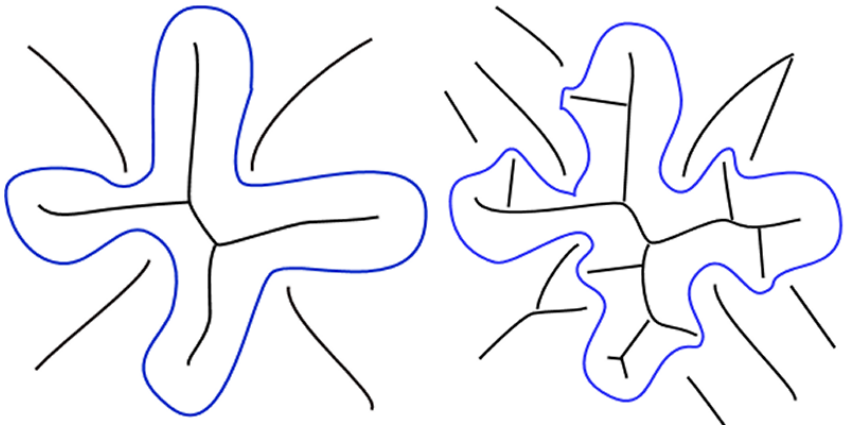

Figure 1: Instability of the medial axis. Left: a smooth shape and its medial axis (black). Right: the same shape with some bumps added and its (unstable) medial axis.

both small features and noise. We use this idea of medial axis filtering to smooth noise and adapt the level of detail of the reconstruction to the allocated budget of centers. More precisely, this means that we determine the value λ suitable to the sampled shape and to the budget of centers, and filter out the poles which are not close to the λ-medial axis. To estimate if a pole v is close to the λ-medial axis, we compute the radius $\gamma(v)$ of the smallest ball enclosing the set $\Gamma(v)$ of sample points closest to v. Poles with radius $\gamma(v)$ smaller than λ are discarded.

Pole Clustering The filtered set of poles now forms a set of *possible* centers, *PC*. Generally, the size of *PC* remains larger than m, the user-defined budget of centers. In order to select m centers from *PC*, we perform a k-means clustering over the set of possible centers [Mac67]. The goal is to obtain a sampling of the λ-medial axis with a local sizing field at a pole v_i proportional to the radius of the polar ball $r(v_i)$. Therefore, we compute the centroid, c, of a clustering cell $\mathscr{C}$ as

$$c = \sum_{v_i \in \mathscr{C}} \omega_i v_i, \tag{9}$$

using for each pole, v_i, a weight, ω_i

$$\omega_i = \frac{d(v_i)}{\rho(v_i)}, \tag{10}$$

where $d(v_i)$ denotes a quadrature term taking into account the actual pole density, and $\rho(v_i)$ denotes the desired local density. More precisely, and owing to the energy equi-distribution property [DFG99], we know that the density function $\rho(v_i)$ must be proportional to $\frac{1}{r(v_i)^{d+2}}$ to obtain a cluster density matching the field $r(v_i)$ in a underlying space of dimension d. In our case $d = 2$, because the filtered poles approximate the medial axis, which is a generically a two-dimensional manifold. As for the quadrature term $d(v_i)$, we take it proportional to $\frac{V(v_i)}{r(v_i)}$, where $V(v_i)$ is the vol-

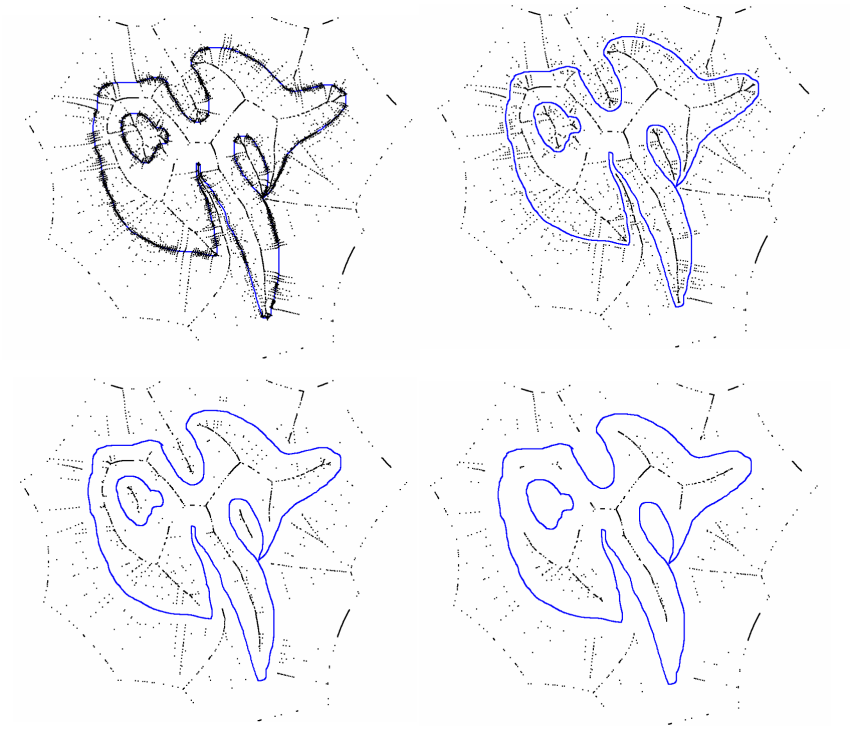

Figure 2: Medial axis filtering on a 2D shape (blue). The λ-medial axis is depicted in black. Top left: all extracted poles. Top right: pole filtering with parameter $\lambda = 0.01$. Bottom left: $\lambda = 0.03$. Bottom right: $\lambda = 0.05$. To get a better sense of these parameters: the diagonal length of the bounding box of the input point set is 1.4.

ume of the cell of v_i in the Voronoi diagram of the filtered poles, and $r(v_i)$ is the polar ball radius since each filtered pole v_i roughly represents the area $\frac{V(v_i)}{r(v_i)}$ of the λ-medial axis.

After convergence of the clustering procedure, the centroid of each cluster is replaced by the closest pole within its cluster, so that the final centers are guaranteed to be located near the medial axis of the sampled surface.

3.2 Constraints

We take as constraints both the input points where the function f is specified to be zero, and a set of additional constraints where f is specified to be non-zero. Recall that our goal is to consider as an approximation of the shape the zero-level set of f. Therefore, we wish to define a signed function f which is positive outside the shape, negative inside and with a non-zero gradient close to the sampled surface. A good candidate is a function approximating the signed distance function to the sampled shape where the distance is positive for points outside the shape and negative inside (Fig.4). At each

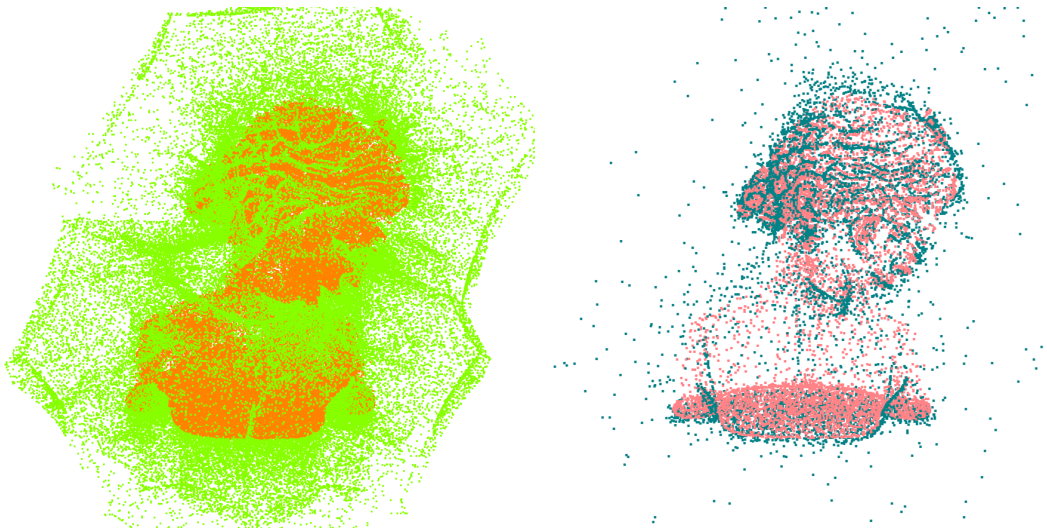

Figure 3: Pole clustering on the Bimba model (100K input data points). 200K poles have been extracted and clustered to 15K poles. Left: All poles (100K inside poles depicted in orange, 100K outside poles depicted in green). Right: After clustering to 15K poles (8K inside depicted in red, 7K outside depicted in green).

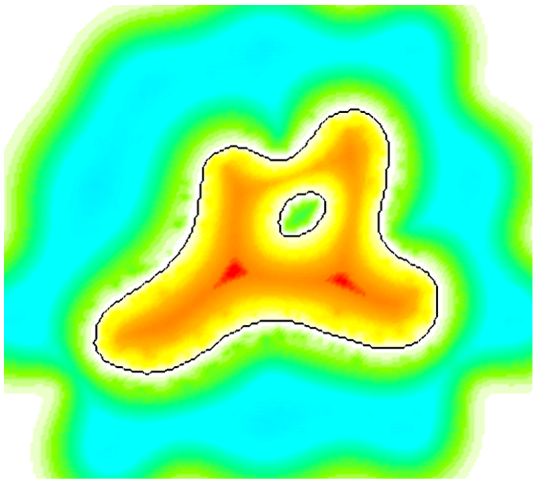

Figure 4: 2D shape (black) and the computed function. Colors range from cold color tones for positive distance values to hot color tones for negative distance values.

pole the radius of the polar ball corresponds to an approximation of its distance to the input point set. Thus, poles can be used as a constraints in order to approximate a distance function to the sampled surface. It remains however to determine the sign of this value, and therefore to *classify* the poles as inside or outside.

Pole Classification Pole classification is the process of labeling the poles as inside or outside the surface. Common approaches use an algorithm to propagate the pole labels through the graph built from adjacency relationships between the poles. In our implementation, we classify the poles using a variant of the algorithm proposed by Amenta [ACK01]. This variant, due to F.Cazals (internal communication), is more efficient and more robust against to noise. During the classification process, a location tag (inside, outside and undetermined) and a confidence value are attributed to each pole. If the confidence of a pole is lower than a certain threshold, the pole will not be taken into account as a constraint.

3.3 basis functions

The reconstructed surface is required to be independent of Euclidean transformation. The function Φ is thus restricted to the set of radial functions :

$$\Phi(x, c_i) = \phi(\|x - c_i\|) \tag{11}$$

where $\|.\|$ denotes the Euclidean distance and $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$.

When the ϕ function has a unbounded support, the corresponding constraint equations lead to a dense linear system. Recovering a solution is therefore tractable only for small data sets. In order to obtain a sparse interpolation matrix, compactly supported RBFs have been introduced by Wendland in [Wen95]. Other compactly supported RBFs (CSRBF) can be used for reconstruction as proposed in [Sch95, Wu95]. As centers are poles, each center c_i, has a corresponding to a scalar value, r_i, the radius of its polar ball. Our function of choice ϕ is compactly supported, and the support size s_i for the function centered on c_i is computed using to r_i. The ϕ function (11) centered on c_i is scaled according to the local support s_i:

$$\phi_i(\|x - c_i\|) = \phi(\frac{\|x - c_i\|}{s_i}) * s_i. \tag{12}$$

The basis functions chosen in our implementation is

$$\phi(r) = (1 - r)_+^4 (1 + 4r) \tag{13}$$

where the symbol + means $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise.

3.4 Solver

The centers are the set $\{c_j\}_{j=1...m}$ of m points in $\mathbb{R}^3$. The constraints are the set $\{p_i\}_{i=1...N}$ of N points where the value of f is known.

Let G be the matrix $[\phi_j(\|p_i - c_j\|)]_{i=1..N, j=1..m}$ and F be the vector $[f_i]_{i=1..N}$. The constraints points $\{p_i\}_{i=1..N}$ include both the n input points and the additional off surface points where we specify the function f value.

$$\mathbf{G} = \begin{pmatrix} \phi_1(\|p_1 - c_1\|) & \dots & \phi_m(\|p_1 - c_m\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\|p_N - c_1\|) & \dots & \phi_m(\|p_N - c_m\|) \end{pmatrix} \tag{14}$$

An approximation using the least squares method implies solving the system (4). With the new notations, the system is

$$G^t G \cdot \alpha = G^t F. \tag{15}$$

The size of the matrix is $m \times m$, where m is the number of centers. The use of compactly supported functions ϕ_i leads to a sparse matrix with about 90% zero elements.

Assembling of the matrix Each term a_{ij} of the matrix $G^t G$ is computed as a sum:

$$a_{i,j} = \sum_{k=1}^{N} \phi_i(\|p_k - c_i\|)\phi_j(\|p_k - c_j\|). \tag{16}$$

For each constraint p, we need to find the list l_p of centers which contain p in their support. To avoid searching exhaustively, we use a 3D Delaunay triangulation of the centers. The constraint p is first located in the triangulation, then our algorithm search outwards from p in the triangulation until all centers containing p in their support are found. For each pair of centers (c_i, c_j) contained in the list l_p, we add a term for p to a_{ij}.

4 Results

We have implemented our algorithm in C++. The Voronoi diagram and Delaunay triangulation are computed using the CGAL library [FGK*00]. The linear system is solved using the TAUCS library [Tol01]. We use an implementation of the *marching cube* algorithm [Blo94] to extract the zero-level set of the reconstructed function . To evaluate the fitting accuracy, we use the *Taubin distance* [Tau94] from the input points (17)

$$Err(f) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{f_i - f(p_i)}{\|\nabla f(p_i\|} \right)^2. \tag{17}$$

This distance is a first order approximation of the Euclidean distance between the input points and the zero level set of the function f. Since the gradient can vanish or go to infinity with compactly supported basis functions, we need to use a threshold S_1 such that :

$$Err_t(f) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{f_i - f(p_i)}{\Gamma(\|\nabla f(p_i\|)} \right)^2, \tag{18}$$

where :

$$\Gamma(\|\nabla f(p_i)\|) = \begin{cases} S_1 & if \;\; g < S_1 \\ \|\nabla f(p_i)\| & if \;\; S_1 < g \end{cases}$$

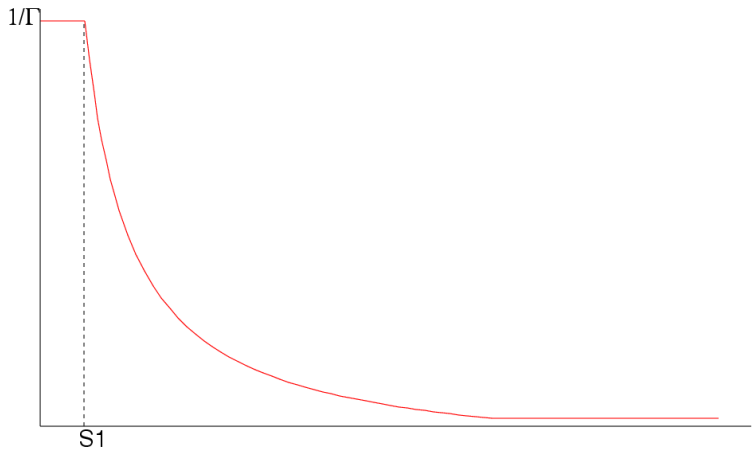

Figure 5: Error function. $1/\Gamma$ function.

Figure 6 summarizes all steps of our algorithm on a 2D shape.

As a typical example for our algorithm, we detail the timings of each reconstruction step for the *omotondo* model (80K points) (Fig.7).

1. point insertion in the Delaunay triangulation: 6.3s;
2. extraction of 16K poles: 2.75s;
3. classification (8K inside poles and 8K outside poles): 20s;
4. filtering and clustering to got 13k centers : 230s;
5. assembling the linear system: 674s;
6. solving the linear system: 78s.

In our current implementation, most of the time is spent assembling the linear system, specifically finding all pairs of centers whose supports intersect a constraint. Although the use of a 3D Delaunay triangulation avoids the naive exhaustive search, this part could be further optimized.

The importance of our choice for the centers is shown graphically by Figure 8. We plot the error against the number of centers for our method and for the common method where constraints and

centers coincide. In the common method, the set of data points is subsampled and the off constraints are taken along the normals estimated at the subsampled points.

Figure 9 illustrates several reconstructions of the Dinosaur with increasing number of centers.

As Fig. 10 depicts, our function is defined all over the space around the sampled shape. In contrast, when compactly supported radial basis functions are centered at the input data points, the function is only defined in a tubular neighborhood of the sampled surface.

The clustering step redistributes the centers among the set of poles as shown in figure 11.

The pole filtering step of our algorithm is useful to adapt the level of detail to the user-defined number of centers (Fig. 12), as well as to improve robustness against noise (Fig. 13). It also shows the effect of filtering when the allocated budget of centers is low.

Figure 13 illustrates an extreme example with a substantial amount of noise due to the misregistration of three range maps. Moreover, the sampling is highly non isotropic and non uniform due to the acquisition system. Figure 13 depicts the main stages of our algorithm applied to a noisy point set sampled on a hand. Although noise in the input data points leads to misclassified poles, the λ-medial axis is stable under such perturbations, and theses misclassified poles are filtered.

5 Conclusion

We have presented a new approach for reconstructing surfaces from scattered points, combining generalized radial basis functions and Voronoi-based surface reconstruction. In contrast to the Voronoi-based approaches, our method creates a smooth and watertight surface, similarly to the RBF approaches. The resulting function is an approximation of the signed distance to the sampled surface defined all around the sampled shape, instead of being defined only in a small neighborhood as in previous work. Our approach relies on a theoretically sound framework for pole extraction and λ-medial axis filtering. This framework provides us with reliable estimates of the normal at each data point, with an approximation of the distance to the sampled surface at each pole, as well as with a filtering method based on the stable λ-medial axis. As a result we can reduce the number of parameters for our algorithm to two: the number of centers, and λ, used to filter the medial axis.

In terms of efficiency, the only stage which impairs scalability is the assembling of the final matrix. We are expect to greatly improve this aspect by an optimized implementation or using new geometric data structure. In our study the medial axis filtering stage allows us to adapt the level of details to a user-defined budget of centers, the value for λ being fixed experimentally. In a future work, we will investigate how to automatically adjust this parameter to accommodate for the allocated budget of centers.

Acknowledgements Our sponsors include the EU Network of Excellence AIM@SHAPE (IST NoE No 506766). The dinosaur model is courtesy of Cyberware, other models being courtesy of the AIM@SHAPE shape repository.

References

[AB98] AMENTA N., BERN M.: Surface reconstruction by voronoi filtering. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry* (New York, NY, USA, 1998), ACM Press, pp. 39–48.

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications* (2001), ACM Press, pp. 249–266.

[AGJ02] ADAMY U., GIESEN J., JOHN M.: Surface reconstruction using umbrella filters. *Comput. Geom. Theory Appl. 21*, 1 (2002), 63–86.

[AM96] ATTALI D., MONTANVERT A.: Modeling noise for a better simplification of skeletons, 1996.

[AS00] ATTENE M., SPAGNUOLO M.: Automatic surface reconstruction from point sets in space. *Computer Graphics Forum 19*, 3 (2000), 457–465.

[BC00] BOISSONNAT J.-D., CAZALS F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry* (New York, NY, USA, 2000), ACM Press, pp. 223–232.

[BL88] BROOMHEAD D. S., LOWE D.: Multivariable functional interpolation and adaptive networks. *ComSys 2* (1988), 321–355.

[Blo94] BLOOMENTHAL J.: An implicit surface polygonizer. 324–349.

[Buh03] BUHMAN M.: *Radial basis functions : theory and implementations*, cambridge monographs on applied and computational mathematics ed., vol. 12. 2003.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH 2001, Computer Graphics Proceedings* (2001), Fiume E., (Ed.), ACM Press / ACM SIGGRAPH, pp. 67–76.

[CFB97] CARR J., FRIGHT W., BEATSON R.: Surface interpolation with radial basis functions for medical imaging, 1997.

[CG04] CAZALS F., GIESEN J.: *Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms*. Research Report RR-5393, INRIA, BP93, 06902 Sophia-Antipolis, France, 2004.

[CL05] CHAZAL F., LIEUTIER A.: The "λ-medial axis". *Graph. Models 67*, 4 (2005), 304–331.

[CSD04] COHEN-STEINER D., DA F.: A greedy delaunay-based surface reconstruction algorithm. *Vis. Comput. 20*, 1 (2004), 4–16.

[Dey04] DEY T. K.: Curve and surface reconstruction. In *Handbook of Discrete and Computational Geometry* (2004), Goodman and O' Rourke eds. , CRC press, 2nd edition.

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review 41*, 4 (1999), 637–676.

[DG04] DEY T. K., GOSWAMI S.: Provable surface reconstruction from noisy samples. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry* (New York, NY, USA, 2004), ACM Press, pp. 330–339.

[DGH01] DEY T. K., GIESEN J., HUDSON J.: Delaunay based shape reconstruction from large data. In *PVG '01: Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics* (Piscataway, NJ, USA, 2001), IEEE Press, pp. 19–27.

[DQ01] DUAN Y., QIN H.: Intelligent balloon: a subdivision-based deformable model for surface reconstruction of arbitrary topology. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications* (New York, NY, USA, 2001), ACM Press, pp. 47–58.

[Duc77] DUCHON J.: Spline minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables* (1977), Schempp W., Zeller K., (Eds.), vol. 571 of *Lecture Notes in Mathematics*, pp. 85–100.

[DZ03] DEY T. K., ZHAO W.: Approximating the medial axis from the voronoi diagram with a convergence guarantee. *Algorithmica 38*, 1 (2003), 179–200.

[Far02] FARIN G.: *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[FGK*00] FABRI A., GIEZEMAN G.-J., KETTNER L., SCHIRRA S., SCHÖNHERR S.: On the Design of CGAL, a Computational Geometry Algorithms Library. *Softw. – Pract. Exp. 30*, 11 (2000), 1167–1202.

[FN80] FRANKE R., NIELSON G.: Smooth interpolation of large sets of scattered data. *Internat. J. Numer. Methods Engrg. 15*, 11 (1980), 1691–1704.

[HTF01] HASTIE T., TIBSHIRANI R., FRIEDMAN J. H.: *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[Isk04] ISKE A.: *Multiresolution Methods in Scattered Data Modelling*, vol. 37. Springer-Verlag, 2004.

[Kir01] KIRBY M.: *Geometric Data Analysis*. John Wiley & Sons, 2001.

[KSO04] KOLLURI R., SHEWCHUK J. R., O'BRIEN J. F.: Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing* (July 2004), ACM Press, pp. 11–21.

[LF99] LODHA S. K., FRANKE R.: Scattered data techniques for surfaces. In *Dagstuhl '97, Scientific Visualization* (Washington, DC, USA, 1999), IEEE Computer Society, pp. 181–222.

[Mac67] MACQUEEN J. B.: Some methods for classification and analysis of multivariate observations. vol. 1, University of Clafifornia Press, Berkeley, Calif., pp. 281–297.

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: 3d scattered data approximation with adaptive compactly supported radial basis functions. In *SMI* (2004), pp. 31–39.

[PG89] POGGIO T., GIROSI F.: A theory of networks for approximation and learning, 1989.

[Sch95] SCHABACK R.: Creating surfaces from scattered data using radial basis functions, 1995.

[Set99] SETHIAN J. A.: *Level Set Methods and Fast Marching Methods*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.

[Tau94] TAUBIN G.: Distance approximations for rasterizing implicit curves. *ACM Trans. Graph. 13*, 1 (1994), 3–42.

[TI04] TOBOR I. REUTER P. S. C.: Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. vol. 12 of *Journal of WSCG 2004*, pp. 467–474.

[TO02] TURK G., O'BRIEN J. F.: Modelling with implicit surfaces that interpolate. *ACM Trans. Graph. 21*, 4 (2002), 855–873.

[Tol01] TOLEDO S.: *TAUCS Version 2.0*, November 2001.

[Wen95] WENDLAND H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics 4* (1995), 389–396.

[Wen04] WENDLAND H.: *Scattered Data Approximation*. Cambridge University Press, 2004.

[Wu95] WU Z.: Compactly supported positive definite radial functions. *Advances in Computational Mathematics 4* (1995), 283.292.

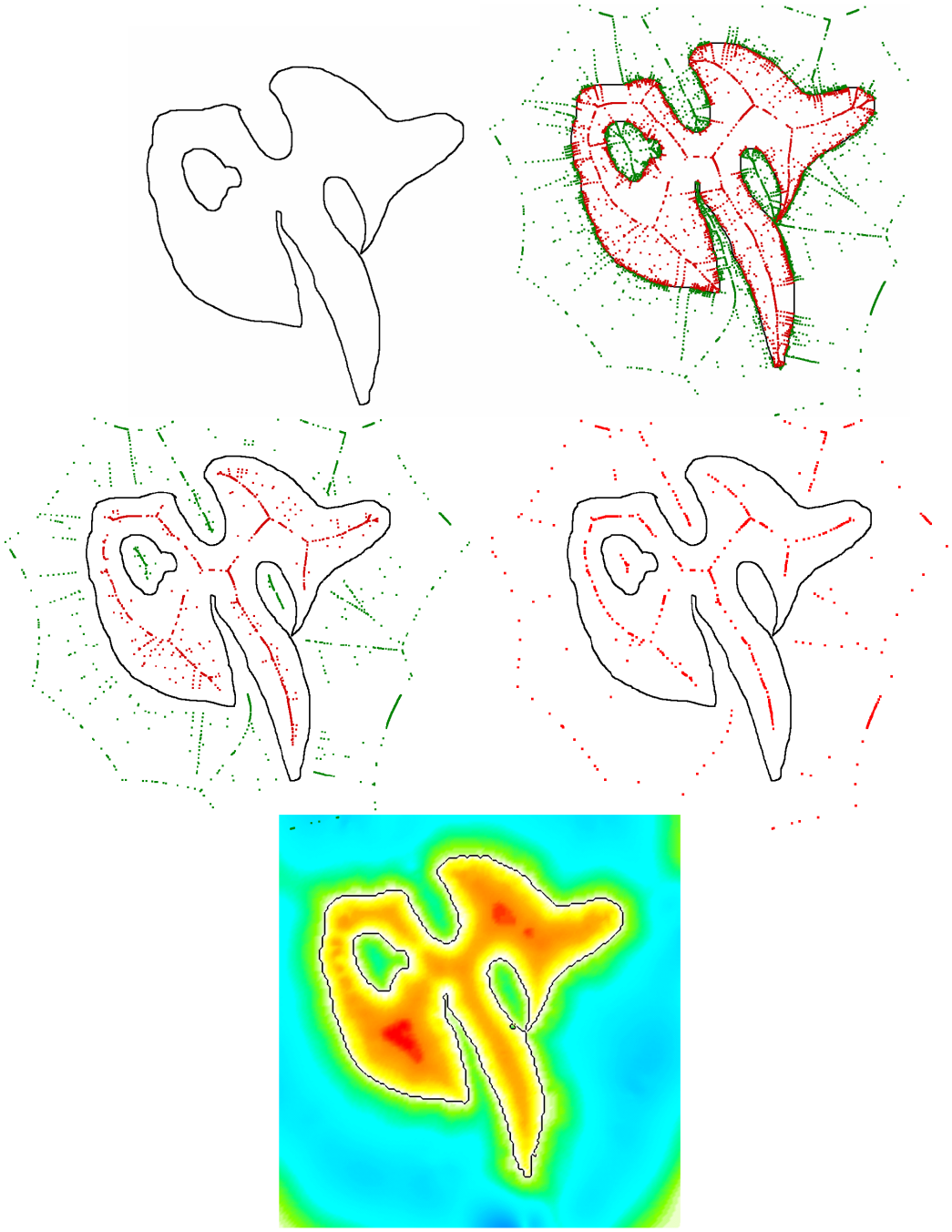

Figure 6: The main steps of our algorithm on a 2D shape. From left to right: input data points (black), all poles are extracted and classified from the Voronoi diagram (red inside, green outside), poles are filtered, poles are clustered into centers, the 2D scalar function is computed and the zero-level set is extracted (black).

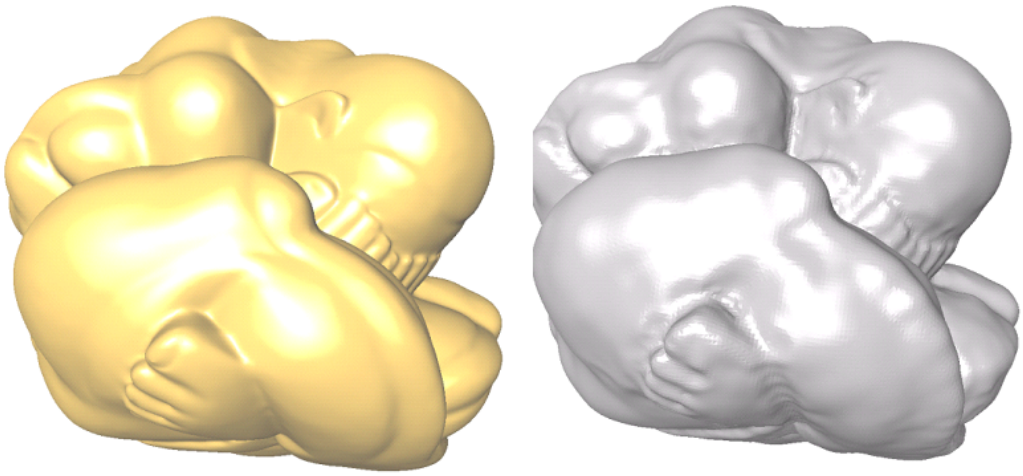

Figure 7: Reconstruction of the *Omotondo* model (80K points) with 13K centers. Fitting accuracy: 2.8×10^{-6}. Left : the original model (gold); Right : the reconstructed surface (silver).

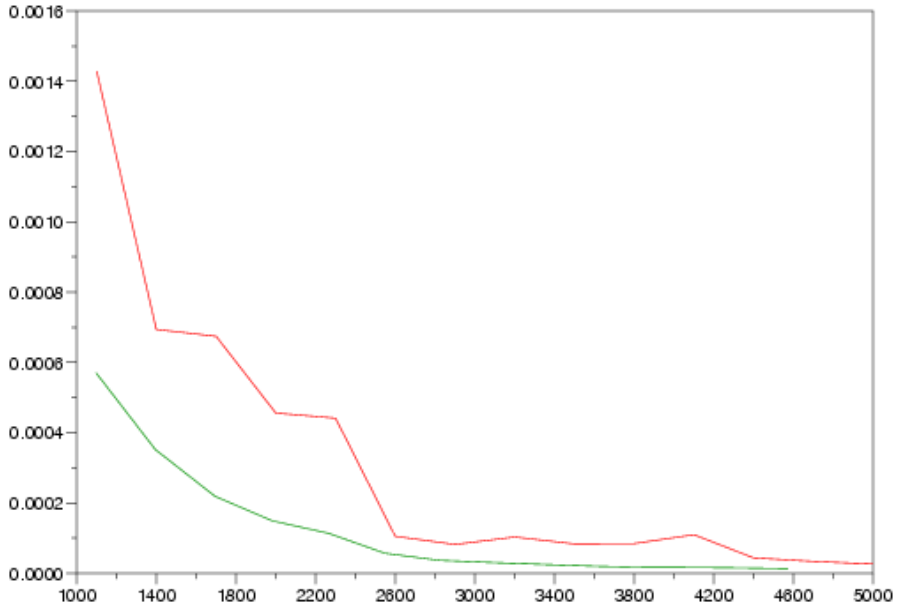

Figure 8: Plot of the error against the number of centers (from 1K and 5K). The red curve corresponds to the common method. The green curve corresponds to our method.

Figure 9: Reconstruction sequence of the Dinosaur with increasing number of centers. From left to right: original, then reconstruction with 1K, 3K, 4K and 5K centers.

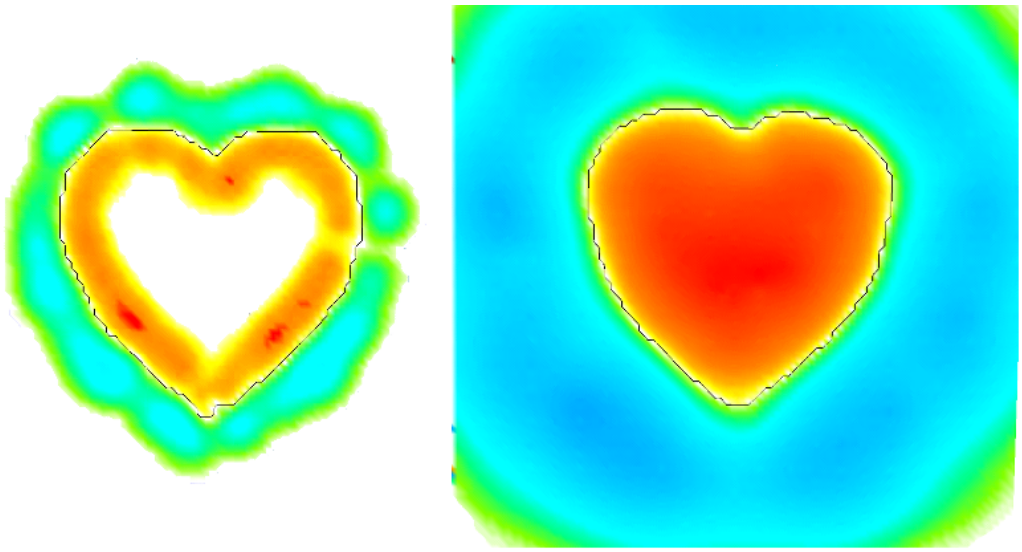

Figure 10: Reconstructed function. The colors represent the function values (cold tones for positive, hot tones for negative values and white for the zero values). Left: the reconstructed function for the common approach; The function does not vanish only in a tubular neighborhood of the point set. Right: the reconstructed function for our method.

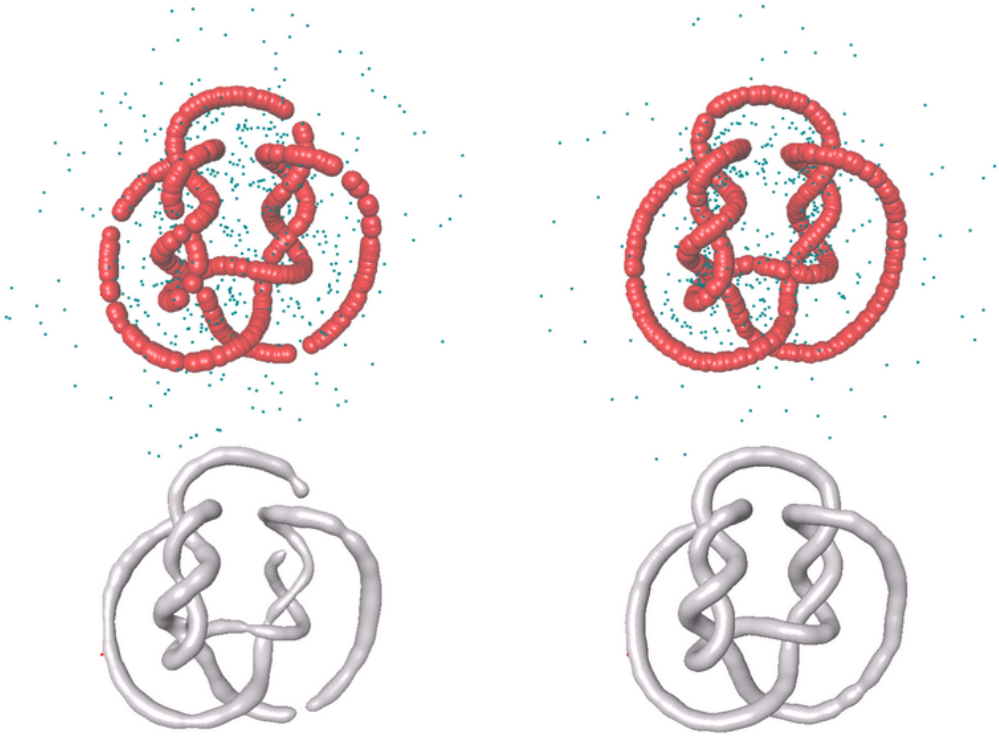

Figure 11: Knot model (6K input points). Left: centers and reconstruction without clustering (inside centers with their polar balls are depicted in red, outside centers are depicted in green). Right: centers and reconstruction with clustering (12K poles are clustered into 1K centers).

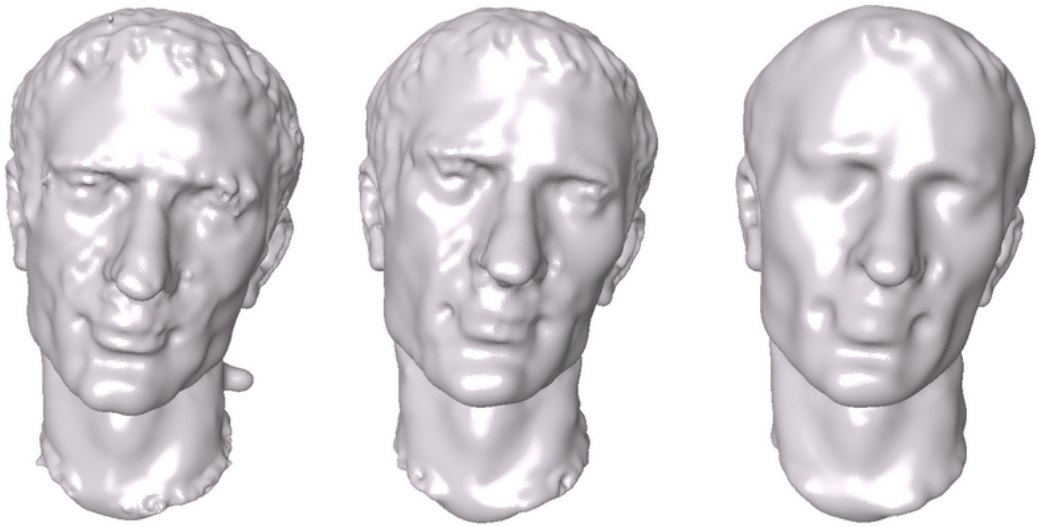

Figure 12: Effect of the filtering step on the Julius model (80K input points). The number of centers is $m = 5K$. Left: without filtering; Middle: poles filtered with $\lambda = 0.01$; Right: poles filtered with $\lambda = 0.02$ (to get a better sense of these parameters, the diagonal length of the bounding box of the input point set is 1.47).

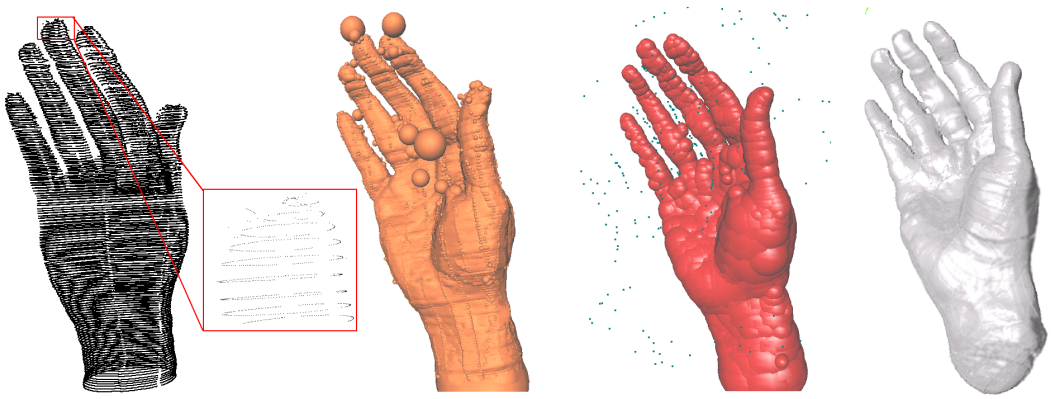

Figure 13: Noisy hand reconstruction. Left: noisy hand model (90K input points). The input points result from registering three range maps. Middle left: inside poles with their polar balls (88K poles, some of them being misclassified); Middle right: 2K centers after filtering and clustering (inside and outside centers with their polar balls (resp. red and green); Right: reconstructed hand.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399