

LSMAT

Least Squares Medial Axis Transform

Daniel Rebain^{1,2}, Baptiste Angles^{1,2}, Julien Valentin², Nicholas Vining², Jiju Peethambaran¹, Shahram Izadi², Andrea Tagliasacchi^{1,2,3}

¹Department of Computer Science, University of Victoria, Victoria, BC, Canada

²Google LLC, Mountain View, CA, USA

³David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

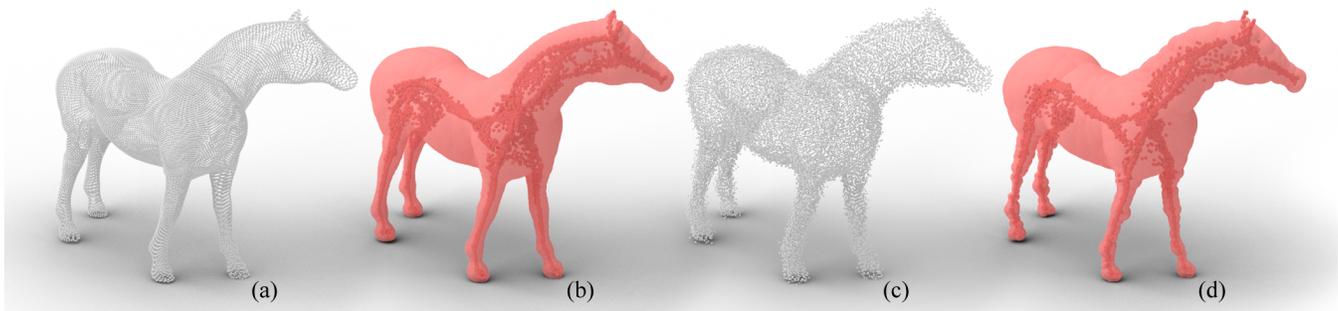


Figure 1: The LSMAT is a novel efficient least-squares formulation of the medial axis transform that operates on unorganized oriented point sets. (a, b) Regardless of input noise, the resulting medial representation is *stable*. (c, d) Its least-squares nature allows it to operate in the presence of heavy noise where most approaches would fail. We visualize the oriented point cloud with oriented splats, draw the union of medial spheres in light red, and their corresponding centers in dark red.

Abstract

The medial axis transform has applications in numerous fields including visualization, computer graphics, and computer vision. Unfortunately, traditional medial axis transformations are usually brittle in the presence of outliers, perturbations and/or noise along the boundary of objects. To overcome this limitation, we introduce a new formulation of the medial axis transform which is naturally robust in the presence of these artifacts. Unlike previous work which has approached the medial axis from a computational geometry angle, we consider it from a numerical optimization perspective. In this work, we follow the definition of the medial axis transform as “the set of maximally inscribed spheres”. We show how this definition can be formulated as a least squares relaxation where the transform is obtained by minimizing a continuous optimization problem. The proposed approach is inherently parallelizable by performing independent optimization of each sphere using Gauss-Newton, and its least-squares form allows it to be significantly more robust compared to traditional computational geometry approaches. Extensive experiments on 2D and 3D objects demonstrate that our method provides superior results to the state of the art on both synthetic and real-data.

Keywords: [medial axis transform] [optimization] [least squares]

CCS Concepts

•Computing methodologies → Point-based models; Volumetric models; Shape analysis;

1. Introduction

A medial representation of an object encodes its solid geometry as the union of a collection of spheres of different radii and origins; see Figure 2. While volumetric or surface mesh representations are more commonly used in computer graphics and computer vision,

since its introduction by Blum et al. [Blu67], medial representations have found applications in many 2D/3D geometric problems such as animation [BP07], fabrication [MHR*16], image processing [TD17], shape analysis [SSCO08], and real-time tracking [TPT16]. The process of converting an input model into a medial representation is

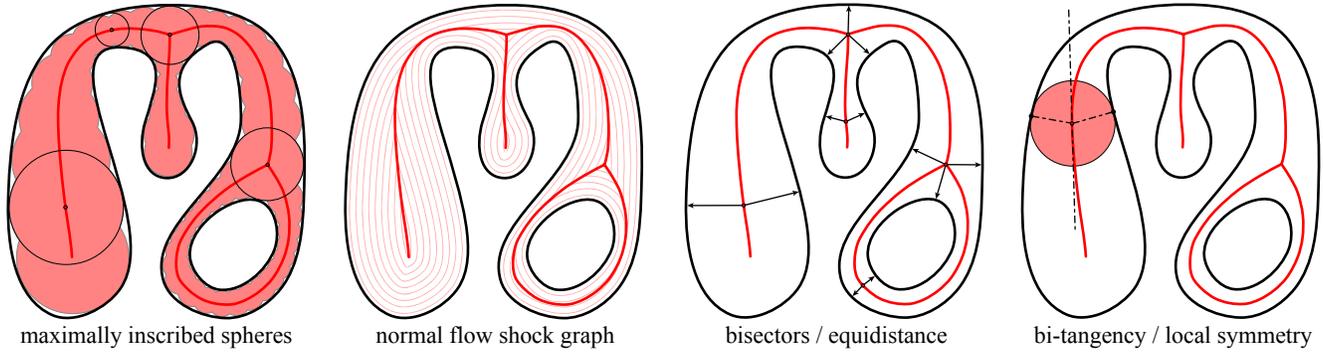


Figure 2: Visualization of four alternative definitions of medial axis; base image courtesy of [TDS*16].

generally referred to as the Medial Axis Transform (MAT), and the collection of origins of the spheres in a medial representation form a *medial skeleton*.

Stability. A common pitfall of medial axis methods is their sensitivity to noise: in a traditional medial representation, new branches of the medial skeleton may form at all negative local curvature extrema. If our shape is represented via a piecewise linear boundary, such as a polygonal mesh in 3D, spurious branches form when the surface exhibits even minor levels of noise; see [TDS*16, Fig.9]. These issues are typically resolved by resorting to postprocessing, in which sets of spheres are filtered out according to various engineered criteria. We present a new algorithm for computing the medial axis transform of an oriented point set which is naturally capable of handling noise, outliers, and other artifacts that create characteristic problems for traditional methods.

Definitions. As summarized in Figure 2 and [TDS*16], the medial axis transform may be defined in a few ways, with each definition producing equally valid and useful representations. In practice, each definition of the medial representation leads to a different way to compute the medial axis. For example, the *normal flow* variant leads to the commonly employed voxel thinning algorithms [SBdB16], while the *equidistance* definition leads to techniques leveraging Voronoi diagrams [BA92]. In this paper, we build over what is likely the most well known definition of the medial axis transform:

Definition 1.1. *The Medial Axis Transform $MAT(\mathcal{O})$ of \mathcal{O} is the set of centers \mathcal{M} and corresponding radii \mathcal{R} of all maximally inscribed circles/spheres in \mathcal{O} .*

While previous work such as Ma et al. [MBC12] has proposed methods to construct medial axis representations using this definition, it is interesting to note that they have treated the problem from a *combinatorial geometry* standpoint. Instead, we consider the medial axis transform from a *numerical geometry* perspective, where the medial axis is given by the solution of an optimization problem. We achieve this by expressing the concepts of *maximality* and *inscription* from Def. 1.1 in a least squares form. The robustness of the approach to imperfections arises from the fact that least squares optimization attempts to find an approximate, rather than exact, solution to the given problem. We are motivated in our approach by considering a least-squares problem as a maximum likelihood estimate of a

function in the presence of noise obeying a Gaussian probability distribution.

Method outline. Our method takes an *oriented* point cloud as input, and produces an unconnected medial point cloud as output by minimizing a combination of a maximality energy, and an inscription energy. Our key technical challenge is in formulating these energies correctly; our maximality energy is designed to have a constant magnitude, ensuring that the optimization energy of each medial sphere is constant regardless of its radius, and our inscription energy is based around a locally supported approximation of the signed distance function of the point cloud. In order to prevent spheres from sliding towards local maxima of local shape thickness, we introduce a pinning constraint as a quadratic barrier energy. The resulting optimization problem is quadratic, with differentiable but non-linear energy terms, and we solve it with an iterative Gauss-Newton solver.

Contributions and Evaluation. Our main contributions are a novel interpretation of a problem that is classically solved by standard computational geometry as a numerical geometry problem, and a novel algorithm for computing the medial axis transform of an oriented point set that inherently handles imperfections in the input. We present a number of qualitative results throughout the paper for both 2D and 3D oriented point clouds. Finally, we also present side-by-side quantitative evaluations of the proposed algorithm against several state-of-the-art methods.

2. Related Work

Many techniques exist to compute the medial axis transformation, as detailed in a recent survey [TDS*16]. We focus our discussion around two central aspects: methods that operate on *surface* 2D/3D representations (e.g. point clouds and triangular meshes), and methods that attempt to resolve the *instability* of medial axis via post processing.

2.1. Medial axis computation of sampled surfaces

Assuming the surface is *sampled* by a sufficiently dense set of points, the Voronoi diagram can be used to compute the medial axis transform with relative ease. For a closed boundary in 2D, any interior Voronoi vertex, as well as Voronoi edges connecting interior vertices, approximate the medial axis with a convergence guarantee [BA92].

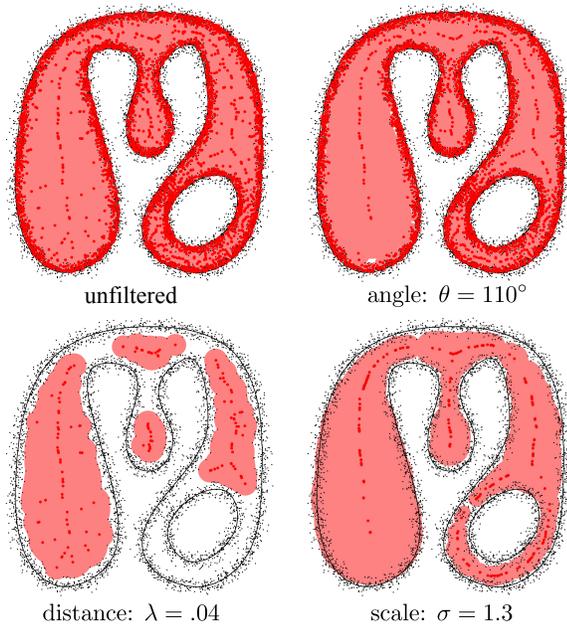


Figure 3: Effectiveness of standard filtering Voronoi methods with high level of noise. As expected, neither angle nor distance methods were capable of filtering out all noise for any selected threshold value. Leveraging global information, the scale axis is able to deal with noise, but with a significant computational burden. Further, note that we provide these technique the ground truth inside/outside labeling, which is *not available* in our context.

Unfortunately, Amenta et al. [AB99] has shown how this property does not hold in 3D due to *sliver tetrahedra*. Thankfully, as we increase the sampling rate, the Voronoi poles (see definition in [AB99]) do converge to the medial axis, allowing the use of 3D Voronoi diagrams for the task at hand. These methods suffer two fundamental shortcomings: ① they are *global* and optimize the entire set of centers at the same time, and ② the extracted axis is heavily susceptible to even minor levels of noise; see Figure 3.

Sphere-at-a-point methods. Leveraging an *oriented* sampling, Ma et al. [MBC12] proposed an alternative way to compute the transform by marrying the maximally inscribed definition of Figure 2(a) to the bi-tangency of Figure 2(d). Unlike Voronoi methods, which consider the entire point set at once, their method can compute a maximal sphere at a point in *isolation*, leading to extremely efficient GPU implementations [JKT13]. These methods represent the *most efficient* way of computing the medial axis, but, similarly to Voronoi methods, they suffer stability issues; see Figure 4. Another method for computing a local approximation to the axis was proposed by Shapira et al. [SSCO08] via casting rays in a cone oriented along the anti-normal. The distance between the point and the intersection with the surface is aggregated by a robust function (e.g. median) to estimate the *shape diameter function*. While this approximation has found widespread use as a shape descriptor, the radius estimate suffer of *bias*, and the algorithm does not generalize to point clouds.

Shape approximation methods. Recently, a new class of techniques has been proposed which attempt to *approximate* watertight surfaces via *Sphere Meshes* – linearly swept spherical primitives [TGB13, TGBE16]. This is achieved via local mesh decimation relying on iterative edge collapses, where *spherical quadrics* are employed in place of the traditional quadric metrics [GH97]. In many cases the produced model *resembles* a medial axis. When executed on 3D data the result can contain tetrahedra, however, the medial axis of a 3D shape is known to consist only of points, curves, and surfaces. Addressing these concerns, the *Medial Meshes* work by Sun et al. [SCYW16] extended sphere meshes to decimate a medial axis mesh, and discard unstable branches. Marrying sphere meshes to medial meshes, Li et al. [LWS*15] followed up this work and proposed QMAT, a more computationally efficient version based on spherical quadrics. While these techniques, in juxtaposition to our local method, are global, they can only cope with minor levels of noise: “with very noisy input, however, the simplified medial mesh is not a stable representation”; see [SCYW16, Fig.9]. The follow-up work by Li et al. [LWS*15] performs slightly better as it can optimize for sphere centers, but our algorithm can still cope with noise that is *one order of magnitude* larger; see [LWS*15, Fig.16].

2.2. Instability and filtering techniques

Techniques that do not attempt to produce an *approximation* suffer from instability when computing the medial axis. Filtering techniques attempt to *remove* portions of the medial axis that do not contribute significantly to the geometry *reconstructed* as the union of medial spheres. As shown in [MGP10, Fig.2], this works fairly well for inputs with little or no noise. Conversely, with large noise (and/or outliers), these methods become mostly inappropriate; see Figure 3.

Angle filtering – θ -medial axis. One way to identify the significance of a point is the largest angle formed by the center of the corresponding maximal sphere and two of its tangent points on the shape boundary. The θ -medial axis of [FLM03], filters out balls associated with low aperture angle. While this filtering can disconnect portions of the medial axis, see Figure 3(c), the issue can be avoided by homotopy-preserving pruning [AM97].

Distance filtering – λ -medial axis. Another metric for filtering discards a sphere whenever its tangent points lie on the surface below a certain distance [CL05, ACK01]. As Figure 3(d) illustrates, this results in a loss of features even before all noise has been removed. This shortcoming makes these solutions inappropriate whenever the input shape contains structures at different scales.

Scale filtering – σ -medial axis. The scale axis transform introduced by Giesen et al. [GMPW09] and extended to 3D by Miklos et al. [MGP10] are methods built over the *maximality* property of the medial axis. First medial spheres are scaled by a given $\sigma > 1$ value, and spheres that are no longer maximal (i.e. contained in another sphere) get discarded. The medial axis of the scaled union of balls is then computed, and its spheres unscaled by a factor $1/\sigma$. This method is *global* as it processes the whole point set at once, and computes multiple Voronoi diagrams in the process, resulting in reduced computational efficiency – e.g. ≈ 2 min. for a mesh with $\approx 100k$ vertices, see [MGP10, Tab.1].

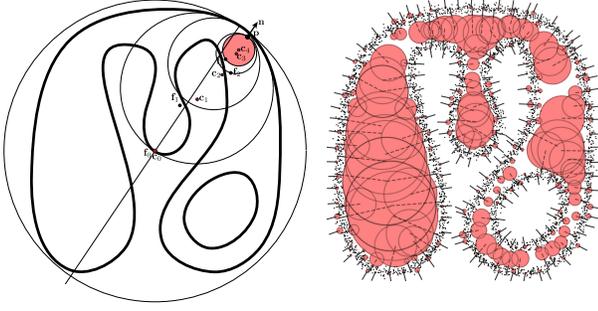


Figure 4: (left) The sphere-shrinking of Ma et al. [MBC12], and (right) its resulting maximal spheres on a noisy point set.

3. Technical details

In Section 3.1, we start by reviewing the highly relevant method of Ma et al. [MBC12], which computes the medial axis by searching for maximal spheres via *local* analysis. We then formulate our numerical optimization to compute medial spheres in Section 3.2.

Notation. We are given in input an *oriented* point cloud $\{(\mathbf{p}_n, \mathbf{n}_n)\}_{n=1\dots N}$ drawn from the solid object \mathcal{O} with watertight surface/boundary $\partial\mathcal{O}$. The cloud is affected by noise with standard deviation $\sigma_{\mathbf{p}}$. We give all numerical values of $\sigma_{\mathbf{p}}$ and other parameters which represent distance values as *percentages* relative to the diagonal of the bounding box of the input shape. With \mathcal{M} we refer to the *internal* medial axis of \mathcal{O} ; see [TDS*16, Fig.2]. With $\mathbf{s} = (\mathbf{c}, r)$ we indicate a sphere centered at \mathbf{c} of radius r , and with t the index of the solver iteration.

3.1. The sphere-shrinking algorithm – Figure 4

The algorithm proposed by Ma et al. [MBC12] is an iterative method that shrinks an initially large sphere until it satisfies the medial axis properties. Assuming \mathbf{p} is a point on the boundary, this process leverages two properties: ① that the sphere passing through \mathbf{p} should be empty; and ② that $\mathbf{p} - \mathbf{c}$ for a smooth input curve is parallel to the contact point normal \mathbf{n} . The sphere-shrinking algorithm is initialized with a large sphere, passing through \mathbf{p} , containing the entire point set, and whose center \mathbf{c} lies along the ray $(\mathbf{p}, -\mathbf{n})$. Whenever the distance to the closest point $\mathbf{f} \in \partial\mathcal{O}$ from \mathbf{c} is less than r (i.e. the sphere is not empty), the center \mathbf{c} is updated by computing the sphere tangent to (\mathbf{p}, \mathbf{n}) and passing through \mathbf{f} ; see Figure 4. Finally, the loop terminates when the sphere radius change across iterations is below numerical precision. This algorithm is highly efficient as each sample can be processed completely independently from the others, but as it treats all points as hard constraints in a combinatorial manner it does not cope well with noisy inputs. Our formulation borrows from this one, but generalizes it by ① reformulating it into a *continuous* optimization problem, and ② making it more *robust* to noise and outliers.

3.2. Optimizing for maximally inscribed spheres

We define a least-squares optimization capable of generating *maximally inscribed spheres* given an *oriented* point cloud \mathcal{P} . Our optimization energy for a sphere \mathbf{s} consists of the combination of two

terms:

$$E_{\text{medial}} = \omega_1 E_{\text{maximal}} + \omega_2 E_{\text{inscribed}} \quad (1)$$

We will now proceed to describe these terms in detail, and then provide comparison against alternative formulations in Section 4.2.

Maximality. The maximality energy creates a constant positive pressure term that tends to increase the sphere size at each iteration. By design, we define this energy to have a *constant magnitude*, that is, a contribution to the optimization energy that is constant regardless of the radius of the given medial sphere. This is achieved by considering the radius at the previous optimization iteration with a constant offset ε :

$$E_{\text{maximal}} = \|r - (r^{t-1} + \varepsilon)\|_2^2 \quad (2)$$

Inscription. Consider a signed distance function (SDF) $\Phi(\mathbf{x})$, where $\Phi(\mathbf{x}) < 0$ for an \mathbf{x} inside the shape. If the expression of this function were available to us, an inscription constraint could be easily expressed by the inequality $\Phi(\mathbf{c}) < -r$, which we can convert in our least squares formalism as:

$$E_{\text{inscribed}} = \mathcal{R}(r + \Phi(\mathbf{c}))^2 \quad (3)$$

where the ramp function $\mathcal{R}(x) = \max(x, 0)$ only penalizes a sphere when it *violates* an inscription constraint. Unfortunately, an SDF function for our oriented point set \mathcal{P} is not readily available without computing a full surface reconstruction of the point cloud [BTS*16]. However, the family of moving least square (MLS) methods are capable of building locally supported approximations in a neighborhood of $\partial\mathcal{O}$ – that is, as $\Phi \rightarrow 0$. For example, the MLS formulation by Kolluri [Kol08] approximates Φ as a weighted sum of locally supported point-to-plane functions:

$$\Phi(\mathbf{x}) = \frac{\sum_n \varphi_n(\mathbf{x}, h) \mathbf{n}_n \cdot (\mathbf{x} - \mathbf{p}_n)}{\sum_n \varphi_n(\mathbf{x}, h)} \quad (4)$$

where in our case $\varphi_n(\mathbf{x}, h) = \varphi(\|\mathbf{x} - \mathbf{p}_n\|_2, h)$ and $\varphi(x, h)$ is a smoothly decaying radial basis function [GG07] with compact support $[0, h]$:

$$\varphi(x, h) = b\left(\frac{x}{h}\right); \quad b(x) = \begin{cases} (1-x^2)^4 & x < 1 \\ 0 & x \geq 1 \end{cases} \quad (5)$$

Note that as Equation 3 evaluates the function Φ at the medial center, which could be potentially far from $\partial\mathcal{O}$, this representation is not immediately appropriate. However, in the context of registration, several works have shown how a quadratic approximation of Φ^2 can be built by appropriately blending *point-to-point* with *point-to-plane* distance functions [PH03, MGPG04]. In more detail, point-to-point distances are a good approximation of Φ^2 in the *far-field* (i.e. for $\Phi^2 \gg 0$), while point-to-plane distances are more suitable in the *near-field* (i.e. for $\Phi^2 \rightarrow 0$). Let us first define $\bar{\mathbf{c}}_n = \mathbf{c}^{t-1} - \mathbf{n}_n(\mathbf{c}^{t-1} - \mathbf{p}_n) \cdot \mathbf{n}_n$, the projection of the center on the hyperplane of point \mathbf{p}_n . We can use this to interpolate between the two metrics:

$$\Phi_{\text{blend}}(\mathbf{s})^2 = \text{mix}(\Phi_{\text{plane}}(\mathbf{s})^2, \Phi_{\text{point}}(\mathbf{s})^2, \varphi_n(\bar{\mathbf{c}}_n, h_{\text{blend}})) \quad (6)$$

where $\Phi_{\text{plane}}(\mathbf{s}) = \mathcal{R}(r - (\mathbf{p}_n - \mathbf{c}) \cdot \mathbf{n}_n)$ represents the distance from the sphere to a plane, $\Phi_{\text{point}}(\mathbf{s}) = \mathcal{R}(r - \|\mathbf{p}_n - \mathbf{c}\|_2)$ is the euclidean distance from the sphere to a point, and $\text{mix}(a, b, x) = xa + (1-x)b$ is the linear interpolation operator; see Figure 5. Based on the

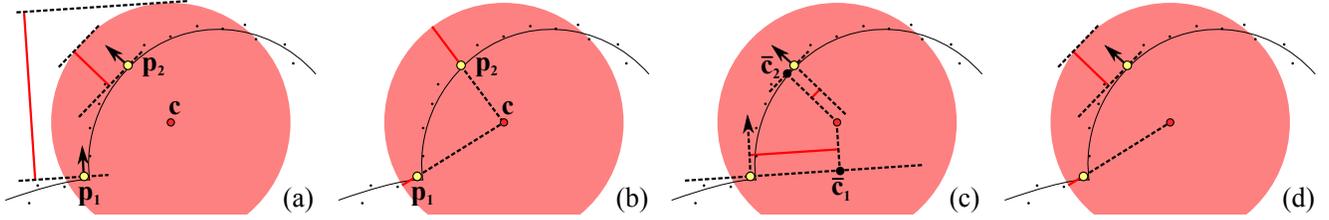


Figure 5: (a) The function $\Phi_{\text{halfplane}}$ uses point normals to give a signed distance value that penalizes spheres which move outside the shape, but may give nonsensical values for points belonging to parts of the surface that are not well approximated by the sphere. (b) The function Φ_{point} does not suffer from this issue but does not enforce that the sphere should respect the orientation of the surface. (c) We mix between the two distances independently for each point using the distance between the point and the projection of the sphere center onto the plane defined by the point. (d) This allows us to use the sided distance only for points where it makes sense, and use the unsigned point-to-plane function otherwise.

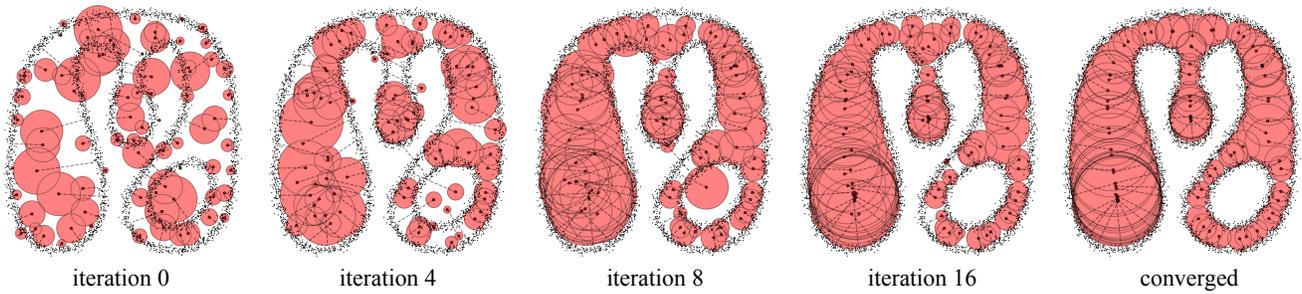


Figure 6: Iterations of LSMAT optimization on an input noisy point set. We randomly initialize sphere centers and radii, and demonstrate the excellent convergence properties of our optimization. Here the center-pin correspondence is marked by the dotted line. Notice how although some spheres are initially outside, the optimization pushes them into the interior of the point set.

geometry of Φ_{blend}^2 , we can then re-formulate our inscription with respect to each oriented point \mathbf{p}_n . Analogously to Equation 4, this energy is accumulated over all points in \mathcal{P} :

$$E_{\text{inscribed}} \approx \sum_n \underbrace{\varphi(\mathcal{R}(\|\mathbf{c}^{t-1} - \mathbf{p}_n\| - r^{t-1}), h_{\text{support}})}_{\neq 0 \text{ for a subset of the } N \text{ points}} \Phi_{\text{blend}}(\mathbf{s})^2 \quad (7)$$

The parameter h_{blend} defines the scale used for blending between distance types, and h_{support} limits how far outside of a sphere a point may be before its contribution falls to zero. As is typical in robust optimization (e.g. IRLS) the weights are computed with respect to the parameters \mathbf{s}^{t-1} at the previous iteration – inscription is evaluated only for points within, or in proximity of the sphere in its previous geometric configuration \mathbf{s}^{t-1} .

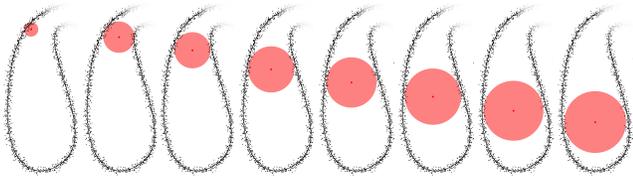


Figure 7: While the algorithm is initialized in a neighborhood, optimizing for larger spheres will cause the solver to have the sphere converge to areas of locally maximal radius.

Pinned spheres. The medial axis provides an estimate of the *local thickness* of the shape through its sphere local radius. However, as our variational formulation attempts to create *larger* spheres, nothing prevents a sphere from traveling along medial branches wherever we have a non-vanishing medial radius gradient on \mathcal{M} ; e.g. a sphere would slide from the tip of a cone to its base; see Figure 7. We can avoid this issue by “pinning” medial spheres. Generalizing the exact constraints of the sphere-shrinking algorithm by [MBC12], we subject the sphere associated with a given point \mathbf{p} to the *hard* constraint $\|\mathbf{c} - \mathbf{p}\| - r \leq d_{\text{pin}}$, which keeps the sphere in contact with a sphere of radius d_{pin} centered at \mathbf{p} . We include this constraint in our optimization via the penalty method [NW06], yielding a quadratic barrier energy:

$$E_{\text{pinning}} = \mathcal{R}(\|\mathbf{c} - \mathbf{p}\| - (r + d_{\text{pin}}))^2 \quad (8)$$

Optimization. Similarly to [LCOLTE07], our optimization induces the definition of medial sphere as the *fixed point solution* of an update equation:

$$\mathbf{s} = \mathcal{F}(\mathbf{s}) = \arg \min_{\mathbf{s}} E_{\text{medial}} + E_{\text{pinning}} \quad (9)$$

Our optimization problem is quadratic, but while its energy terms are differentiable, they are not linear. Hence, we iteratively compute a solution via Gauss-Newton. This requires the linearization of the arguments of the quadratic functions with respect to \mathbf{c} and r , which is straightforward in our setting.

Implementation. Due to the independent nature of the per-sphere optimization, our implementation is straightforward. At each step we compute Jacobian matrices and residual vectors separately for each sphere and thus are only required to solve a $D \times D$ linear system for each, where D is the number of degrees of freedom of a single sphere (3 or 4 in our experiments). This property eliminates the need for any advanced solvers or factorizations and enables the implementation to be completely parallel over the spheres, as required for effective GPU acceleration. Additional performance optimization could be achieved by collecting the points for each sphere whose contribution is non-zero using accelerating data structures such as kd-trees [FBF77].

4. Results and evaluation

We show medial representations generated by our method, in both 2D and 3D, throughout the paper and in Figure 16. As shown, our method produces valid medial representations for a wide variety of shapes and models, and in the presence of noise and outliers. Throughout the paper, we process all input with the same parameters whose values were set according to the analysis in Section 4.6. We evaluate our method in several ways. First, we consider variations of the inscription and maximality energy, and show how alternative formulations fail. Second, we evaluate the performance of our algorithm against ground truth on synthetic benchmarks contaminated by noise and outliers. Third, we compare our generated medial representations against the state-of-the-art in both 2D and 3D, again in the presence of noise. Finally, we provide an analysis of our algorithm parameters.

4.1. Variants of inscription energy – Figure 8

We consider two modified formulations of Equation 7 in which we either penalize the squared distance to points, or the squared distance to half-planes as opposed to our blended formulation. We investigate this behavior by randomly initializing the algorithm, and snapping $\varphi_n(\bar{c}_n)$ to either zero or one for all points. When $\varphi_n(\bar{c}_n) = 1$, point-to-point energy is used and the algorithm attempts to make *spheres empty* in a least square sense. However, nothing prevents the optimization from generating maximal spheres *outside* the shape in the ambient space. When $\varphi_n(\bar{c}_n) = 1$, point-to-plane energy is used, and

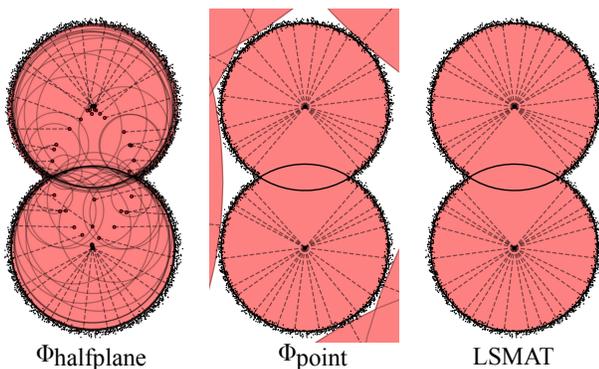


Figure 8: Qualitative evaluation of inscription energy variants.

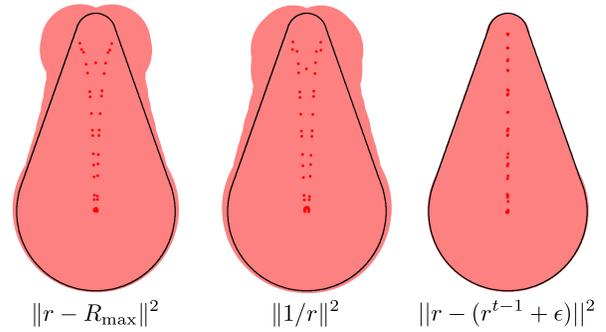


Figure 9: Qualitative evaluation of maximality energy variants.

the algorithm attempts to make *half-spaces empty*. As illustrated in Figure 8 and the **supplemental video**, this results in difficulties for the algorithm in dealing with sharp concavities. In the example above, we expect all centers to cluster to one of the two centers, but spheres in the neighborhood of the sharp concavity might read the normal of a point on the opposite side, with a halfplane requesting the radius of a sphere intersecting with it to be significantly smaller. This problem is caused by the fact that point-to-point and point-to-plane energies approximate the squared SDF of a point set respectively in the far and near field. Our LSMAT formulation respects this geometric property, and deals with both issues at once.

4.2. Variants of maximality energy – Figure 9

We consider two alternative formulations of the maximality condition in our optimization: penalizing the squared inverse of the sphere radius, expressed as $\|1/r\|^2$, or directly specifying R_{\max} , the maximum size of a medial sphere, and optimizing $\|r - R_{\max}\|^2$. While intuitively these are potentially feasible solutions, they incur a significant limitation: the amount of “pressure” a medial sphere will apply will be dependent on its size. That is, small spheres will be associated with a higher energy level. For example, using the first formulation, as $r \rightarrow 0$ its gradient will tend to $-\infty$; as illustrated in Figure 9, this can cause spheres in the neighborhood of small features to bulge out. Our solution, detailed in Equation 2, does not encounter this problem, as our energy is *constant* regardless of the sphere size.

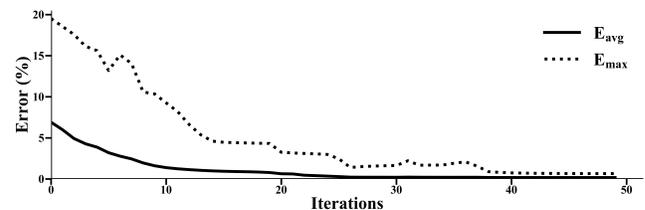


Figure 10: As the optimization of Equation 9 is executed, the average/max ground truth errors converge to the maximum precision. This plot illustrates the error for the iterations visualized in Figure 6.

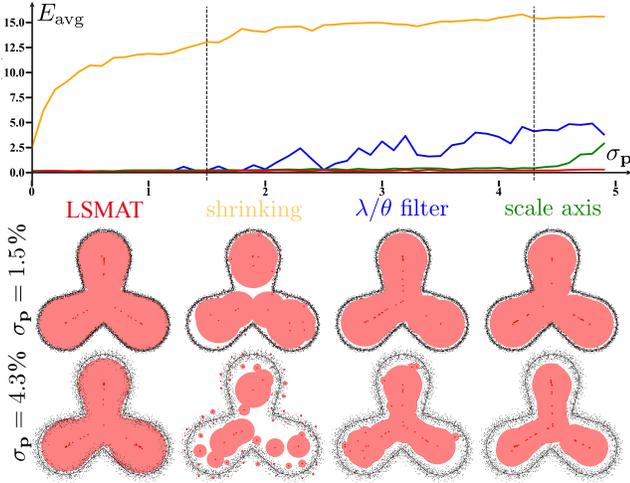


Figure 11: (top) Quantitative evaluation on state-of-the-art methods showing increasing error as we increase the noise level. (bottom) Qualitative results corresponding to the two dashed lines in the plot.

4.3. Quantitative evaluation metric – Figure 10

We consider ground truth geometry polluted by noise, and evaluate the quality of an extracted axis by computing the distance of each medial center to the closest point on the ground truth axis \tilde{M} :

$$E_{\text{avg}} = \frac{1}{N} \sum_n \arg \min_{\tilde{c}_n \in \tilde{\mathcal{M}}} \|\mathbf{c}_n - \tilde{c}_n\|_2 \quad (10)$$

$$E_{\text{max}} = \max_n \arg \min_{\tilde{c}_n \in \tilde{\mathcal{M}}} \|\mathbf{c}_n - \tilde{c}_n\|_2 \quad (11)$$

We compute \tilde{M} via the medial axis module of the python *skimage* package, derived from the ridges of the distance transform from the ground truth boundary $\partial\mathcal{O}$. As we discretize images with a bounding box at a 1024^2 resolution, the “numerical precision” of the metrics above is of one pixel. To obtain scale invariance, we report these errors in relation to the diagonal of the bounding box. In Figure 10, we visualize the convergence of our iterative optimization scheme.

4.4. State-of-the-art comparisons in 2D – Figure 11

Through the metric from Section 4.3, we quantitatively evaluate the performance of LSMAT against local filtering methods, as well as the global scale axis transform. Figure 3 shows how neither distance nor angle filtering is very effective; hence we employ a compound variant where we first filter by distance with $\lambda > \sigma_p$, and then by angle $\theta > 110^\circ$. For the scale axis, we set $\sigma = 1.3$, and for our method, we use our default parameters. We gradually increment the level of noise, and plot the corresponding error metric in Figure 11, as well as a few example frames from the plot. Note that LSMAT correctly captures the shape of the ground truth boundary, whereas all other local methods fail.

4.5. State-of-the-art Comparisons in 3D – Figure 12

We qualitatively evaluate the performance of LSMAT in three dimensions by comparing our results to those generated by the

SAT [MGP10], QMAT [LWS*15], and sphere-shrinking [MBC12] methods. To compare with methods that expect a mesh in input, we finely re-triangulate the surface, and apply normal displacement perturbation with $\sigma_p \in [0\%, 2\%]$ relative to the diagonal bounding box. On the *fertility* model, LSMAT produces a convincing medial axis representation even when the input oriented point cloud is affected by extreme noise – i.e. with a magnitude close to the one of the *local feature size*. For the *vase* model, LSMAT still produces a smoother and more faithful medial axis representation than the existing state-of-the-art. Even in the presence of minor amounts of noise (second row), LSMAT faithfully produces a medial axis representation with a smooth surface, and whose skeleton resembles that of the uncorrupted model. The results are particularly encouraging on the horse dataset, where we attempted to use a very small number of target primitives for QMAT (≈ 500) and a large value of scale for SAT (≈ 1.3). Our formulation is the only one capable of computing a relatively noiseless arrangement of medial spheres. The timings for our LSMAT and sphere shrinking results can be found in Table 1. All experiments were run on a machine with an Intel Xeon E5-1650 CPU and an Nvidia GTX 1080 GPU.

4.6. Algorithm Parameter Analysis – Figure 13

Our algorithm depends on five parameters: the kernel sizes h_{blend} and h_{support} , the relative weight ω_1/ω_2 , pinning distance d_{pin} , and radius expansion constant ϵ . The effect of varying the first four given different levels of noise is shown in Figure 13. Note we only consider the ratio between the ω_* , as the two energies balance each other. As h_{blend} increases, spheres near sharp concave corners begin to shrink as they eventually use the point-to-plane distance for all points in their neighborhood. When h_{support} grows significantly beyond the local feature size, the MLS formulation is no longer able to recover a meaningful surface. As ω_1/ω_2 increases and the relative importance of maximality versus inscription increases, spheres expand until they no longer form a faithful medial representation and ultimately escape the point cloud. Finally, as d_{pin} grows, the spheres are allowed to slide further from their starting positions towards areas of locally maximal radius. We assume that estimates of the input noise characteristics are available, and choose our default value for

| Model (σ_p) 10k Spheres | LSMAT | | Sphere Shrinking | |
|-------------------------------------|-------|------------|------------------|------------|
| | Time | Iterations | Time | Iterations |
| Fertility (0%) | 11.8s | 40 | 0.30s | 9 |
| Fertility (1%) | 11.6s | 40 | 0.35s | 11 |
| Fertility (2%) | 12.0s | 40 | 0.33s | 10 |
| Fertility (5%) | 16.2s | 40 | 0.34s | 10 |
| Vase (0%) | 11.2s | 70 | 0.16s | 9 |
| Vase (1%) | 12.1s | 70 | 0.17s | 10 |
| Vase (2%) | 11.8s | 70 | 0.16s | 9 |
| Vase (5%) | 6.1s | 70 | 0.16s | 9 |
| Horse (2%) | 6.8s | 60 | 0.16s | 9 |

Table 1: Run-times for LSMAT and Sphere Shrinking results presented in Figure 12. Both algorithms are GPU accelerated and run on the same hardware. Note that our efficiency claims are made with respect to QMAT and the scale axis transform, for which GPU acceleration is not available.

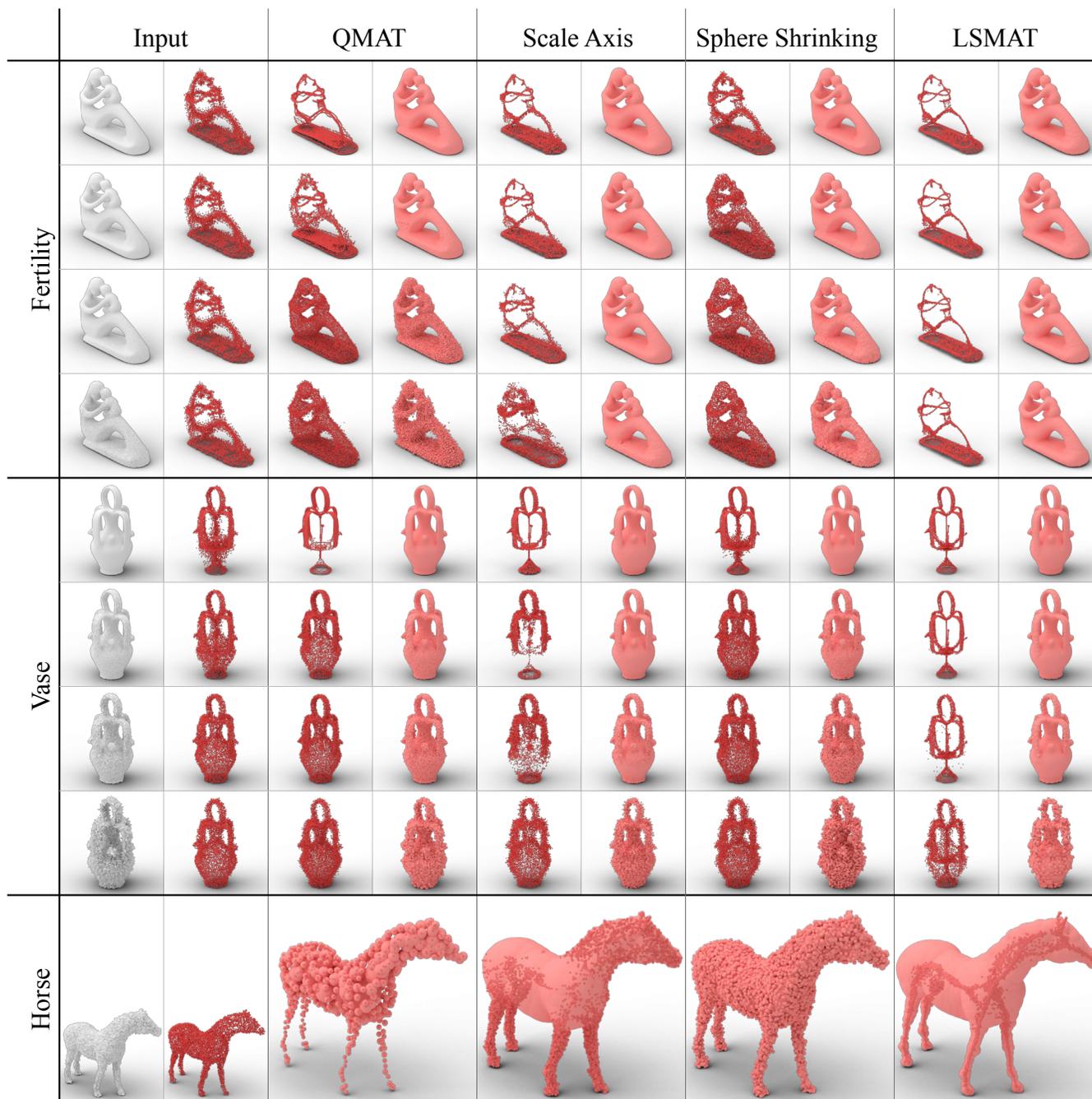


Figure 12: Qualitative evaluation on state-of-the-art methods as we increase the noise level. The QMAT [LWS*15] and SAT [MGP10] in the first two columns are global methods that assume a *watertight* surface, while the sphere-shrinking [MBC12] and the LSMAT proposed here are *local* methods.

each of these four parameters based on empirically derived linear functions of σ_p ; see Appendix B for details. Through experiments we observed that it is sufficient to choose a constant value for the fifth parameter ϵ , as different values of it merely change the optimal relation between ω_1/ω_2 and σ_p . For all our experiments we use $\epsilon = 100\%$. We believe improved formulations of LSMAT could

eventually coalesce some of these parameters hence improving the ease of use of our algorithm.



Figure 13: A qualitative analysis of parameters in our algorithm. Each quadrant shows the result of sweeping a parameter (horizontal) for different noise values (vertical). The highlighted images represent the "default" parameter choice as defined in Section 4.6. The right column in each quadrant shows an unreasonably high choice to demonstrate that there is an upper bound for each parameter.

5. Future works

Expressing the Medial Axis Transform as a non-linear least squares problem opens up several interesting avenues for future research.

Robustness to outliers – Figure 14. Least squares problems can be interpreted as a maximum-likelihood (ML) estimation given a *Gaussian* probability distribution of the noise variables. If the input is corrupted by other forms of noise, one could replace *Gaussian* with other error distributions, and derive the corresponding ML optimization scheme. For example, if we assume the probability distribution of the noise to be Laplacian, the least-squares problems would simply be transformed into an ℓ^1 (i.e. least norm) optimization. However, these type of problems can still be computed with Gauss-Newton type methods by using iteratively re-weighted least squares (IRLS) techniques [DDFG10]. As illustrated in Figure 14 this results in an IR-LSMAT algorithm that can cope with significant

amounts of outliers. While these results are promising, the convergence speed of the optimization is severely reduced, as a much smaller value for ϵ was needed to produce these results. The generalizability of LSMAT to ℓ^p robust norms [BTP13] is particularly interesting. More specifically, consider the optimization problem consisting only of the following energy:

$$\arg \min_{\mathbf{c}, r} \sum_n |||\mathbf{c} - \mathbf{p}_n|| - r|^p. \quad (12)$$

For $p = 2$ this simplified version of the sphere-fitting problem is convex, hence generating a single solution regardless of initialization. However, as $p \rightarrow 0$ the problem is non-convex and the local minima reached by optimization depends on the initialization. Our preliminary investigation revealed how these local minima correspond to spheres belonging to the *symmetry set*, a superset of the medial axis; see Figure 2(d) and [TDS*16, Sec. 2.1.4]. How to exploit (12) to

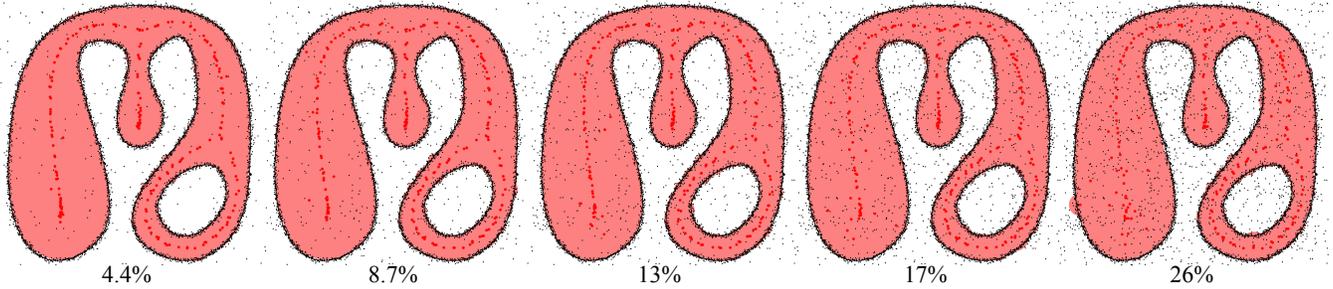


Figure 14: Iteratively Reweighted LSMAT and its ability to cope with an increasing number of outliers (as % of input point set).

efficiently compute the MAT of a point set is an interesting venue for future works.

Smoothness priors – Figure 15. Our pinning formulation is highly efficient, but its local nature can be also considered an intrinsic limitation. For example, in the noisy *maple leaf* example in Figure 16, at times the estimated medial spheres could be over and/or underestimated in size, resulting in medial centers that do not necessarily sample the underlying piecewise-smooth manifold of the MAT. However, if our input is a sampling of a smooth surface $\partial\mathcal{O}$, then we know that [SP08]: ① the medial centers \mathbf{c}_* should be lying on a piecewise smooth manifold, and ② the sphere radius function on this manifold should vary smoothly. Another desirable characteristic might be to have a uniform sampling of this manifold. We can convert these priors into least-squares energies, resulting in a

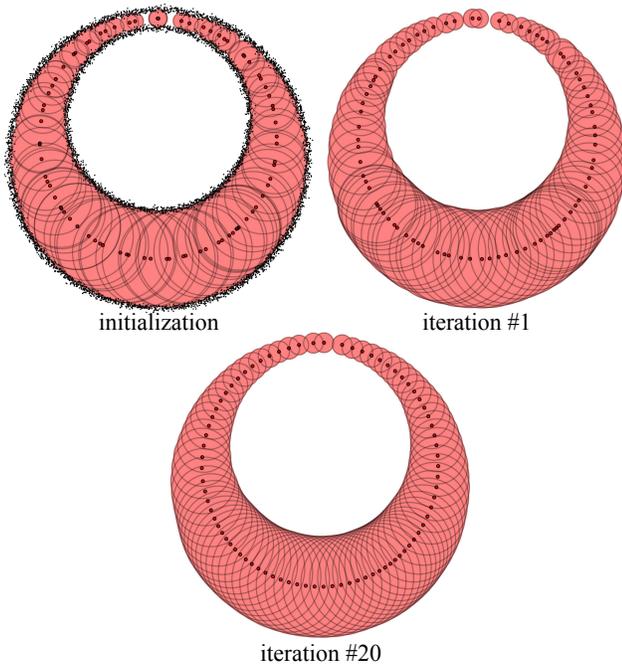


Figure 15: Optimizing LSMAT centers placement. To highlight the smoothness of the resulting shape, we only display the input point set overlaid to the initialization.

maximum a-posteriori optimization. In Figure 15 we illustrate a few iterations of this optimization on the *moon* shape, where the “pinning” constraints from Section 3.2 have been disabled. While the optimization behaves as expected, this suffers similar shortcomings to those illustrated in Figure 7, and would result in a single sphere if executed for $t \rightarrow \infty$. Modifying the variational LSMAT formulation to obtain a regularly sampled distribution of medial centers is an interesting venue for future works.

Optimization acceleration. In our experiments, we initialize the optimization with random sphere positions and radii. Nonetheless, given how a smooth object is composed of smooth piecewise manifolds \mathcal{M} , and a smooth radius function \mathcal{R} defined thereon, one could easily envision a locally bootstrapped version of the algorithm, where unsolved medial balls are initialized with the \mathbf{c}_*, r_* of their neighbors – which could also be re-interpreted as a *multi-scale* solver. Notice that techniques such as the sphere-shrinking algorithm from [MBC12] do not permit this type of acceleration.

Shape approximation vs. reconstruction. A number of methods leverage the medial axis for *interpolatory* reconstruction [Dey06], and our work is a stepping stone towards the creation of *approximating* reconstruction [BTS*16] algorithms based on the medial axis. Methods such as Medial Meshes [SCYW16] and QMAT [LWS*15] assume a reconstruction of the watertight surface is *already* available, and attempt to extract its *approximation* via swept-spheres. Conversely, our work could be extended to directly compute a *reconstruction* of the input point cloud by minimizing data fitting metrics based on Hausdorff distances [SCYW16] or spherical quadrics [LWS*15]. A “topology surgery” step could then be interweaved with our optimization to stitch medial spheres together and create a medial mesh with connectivity information.

Orienting a point set. Finally, while our formulation is based on an oriented point set and an approximation of its SDF in the near/far field, an interesting variant of our algorithm could consider an un-oriented point set, where the quantities to be optimized for would be the radii $[r_1, r_2]$, and contact plane $[\mathbf{k}, \mathbf{n}]$ of a pair of *twin spheres*. The contact point \mathbf{t} should then be optimized on the manifold $\partial\mathcal{O}$, while the complementary outside/inside label of each sphere be optimized to result in a *smooth signing* of the environment space. This approach would then provide a “medial axis” analogous to recent efforts in variational reconstruction of non-oriented point clouds [MDGD*10].

6. Conclusions

We have introduced the Least Squares Medial Axis Transform, or LSMAT, a continuous relaxation of the medial axis transform that is not only stable, but also robust to high levels of noise even though it is based solely on *local* optimization. While in most of the paper we visualized the generated maximal spheres covering more or less the entire shape, we would like to remind the reader that the algorithm operates on each sphere *independently*; the algorithm is therefore trivially parallelizable and particularly suitable for GPU implementations. Our method produces results on noisy inputs that state-of-the-art methods fail to handle, without a reliance on ad-hoc postprocessing. Our approach is efficient, parallelizable, and therefore suitable for real time applications where reliable medial representations are required, and where captured inputs are likely to be noisy.

Acknowledgements

We would like to express our gratitude to Mathieu Desbrun, Leonidas Guibas, Justin Solomon, Keenan Crane, and especially Sofien Bouaziz for their ideas and suggestions. In our experiments we use models provided courtesy of [CGF09] and the AIM@SHAPE Shape Repository.

References

- [AB99] AMENTA N., BERN M.: Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry* (1999). 3
- [ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proceedings of the ACM Symposium on Solid Modeling and Applications* (Ann Arbor, Michigan, USA, 2001). 3
- [AM97] ATTALI D., MONTANVERT A.: Computing and simplifying 2d and 3d continuous skeletons. *Computer vision and image understanding* (1997). 3
- [BA92] BRANDT J. W., ALGAZI V.: Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding* (1992). 2
- [Blu67] BLUM H.: *A transformation for extracting new descriptors of shape*. M.I.T. Press, 1967. 1
- [BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2007). 1
- [BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse iterative closest point. *Computer Graphics Forum (Proc. of SGP)* (2013). 9
- [BTS*16] BERGER M., TAGLIASACCHI A., SEVERSKY L., ALLIEZ P., GUENNEBAUD G., LEVINE J., SHARF A., SILVA C.: A survey of surface reconstruction from point clouds. *Proc. Eurographics (State of the Art Reports)* (2016). 4, 10
- [CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. In *Acm transactions on graphics (tog)* (New Orleans, Louisiana, USA, 2009). 11
- [CL05] CHAZAL F., LIEUTIER A.: The λ -medial axis. *Graphical Models* (2005). 3
- [DDFG10] DAUBECHIES I., DEVORE R., FORNASIER M., GÜNTÜRK C. S.: Iteratively reweighted least squares minimization for sparse recovery. *Communications on pure and applied mathematics* (2010). 9
- [Dey06] DEY T. K.: *Curve and surface reconstruction: algorithms with mathematical analysis*. Cambridge University Press, 2006. 10
- [FBF77] FRIEDMAN J. H., BENTLEY J. L., FINKEL R. A.: An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* (1977). 6
- [FLM03] FOSKEY M., LIN M. C., MANOCHA D.: Efficient computation of a simplified medial axis. In *Proceedings of the ACM Symposium on Solid Modeling and Applications* (Seattle, Washington, USA, 2003). 3
- [GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Transactions on Graphics (TOG)* (2007). 4
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proc. of ACM SIGGRAPH* (Los Angeles, California, USA, 1997). 3
- [GMPW09] GIESEN J., MIKLOS B., PAULY M., WORMSER C.: The scale axis transform. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry* (Aarhus, Denmark, 2009). 3
- [JKT13] JALBA A. C., KUSTRA J., TELEA A. C.: Surface and curve skeletonization of large 3d models on the gpu. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013). 3
- [Kol08] KOLLURI R.: Provably good moving least squares. *ACM Transactions on Algorithms (TALG)* (2008). 4
- [LCOLTE07] LIPMAN Y., COHEN-OR D., LEVIN D., TAL-EZER H.: Parameterization-free projection for geometry reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2007). 5
- [LWS*15] LI P., WANG B., SUN F., GUO X., ZHANG C., WANG W.: Q-mat: computing medial axis transform by quadric error minimization. *ACM Trans. Graph.* (2015). 3, 7, 8, 10
- [MBC12] MA J., BAE S. W., CHOI S.: 3d medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer* (2012). 2, 3, 4, 5, 7, 8, 10
- [MDGD*10] MULLEN P., DE GOES F., DESBRUN M., COHEN-STEINER D., ALLIEZ P.: Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum (Proc. of SGP)* (2010). 10
- [MGP10] MIKLOS B., GIESEN J., PAULY M.: Discrete scale axis representations for 3d geometry. *ACM Trans. Graph.* (2010). 3, 7, 8
- [MGPG04] MITRA N. J., GELFAND N., POTTMANN H., GUIBAS L.: Registration of point cloud data from a geometric optimization perspective. In *Proc. of SGP* (Nice, France, 2004). 4
- [MHR*16] MUSIALSKI P., HAFNER C., RIST F., BIRSAK M., WIMMER M., KOBBELT L.: Non-linear shape optimization using local subspace projections. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2016). 1
- [NW06] NOCEDAL J., WRIGHT S. J.: *Sequential quadratic programming*. Springer, 2006. 5
- [PH03] POTTMANN H., HOFER M.: Geometry of the squared distance function to curves and surfaces. In *Visualization and mathematics III*. Springer, 2003, pp. 221–242. 4
- [SBdB16] SAHA P. K., BORGEFORS G., DI BAJA G. S.: A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters* (2016). 2
- [SCYW16] SUN F., CHOI Y. K., YU Y., WANG W.: Medial meshes: a compact and accurate representation of medial axis transform. *IEEE Transactions on Visualization and Computer Graphics* (2016). 3, 10
- [SP08] SIDDIQI K., PIZER S.: *Medial representations: mathematics, algorithms and applications*. Springer, 2008. 10
- [SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* (2008). 1, 3
- [TD17] TSOVKAS S., DICKINSON S.: Amat: Medial axis transform for natural images. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, 2017). 1
- [TDS*16] TAGLIASACCHI A., DELAME T., SPAGNUOLO M., AMENTA N., TELEA A.: 3d skeletons: a state-of-the-art report. *Proc. Eurographics (State of the Art Reports)* (2016). 2, 4, 9
- [TGB13] THIERY J.-M., GUY E., BOUBEKEUR T.: Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Trans. Graph.* (2013). 3

- [TGBE16] THIERY J.-M., GUY E., BOUBEKEUR T., EISEMANN E.: Animated mesh approximation with sphere-meshes. *ACM Trans. Graph.* (2016). 3
- [TPT16] TKACH A., PAULY M., TAGLIASACCHI A.: Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* (2016). 1

Appendix A: Gradients of Energy Components

Gradients are written in the form $\nabla_s f(\mathbf{s}) = \left[\nabla_{\mathbf{c}} f(\mathbf{s}), \frac{\partial f(\mathbf{s})}{\partial r} \right]$.

Inscription Term. Given oriented surface point $(\mathbf{p}_n, \mathbf{n}_n)$:

$$\nabla_s \Phi_{\text{plane}}(\mathbf{s}) = \mathcal{H}(\Phi_{\text{plane}}(\mathbf{s}))[-\mathbf{n}_n, 1] \quad (13)$$

$$\nabla_s \Phi_{\text{point}}(\mathbf{s}) = \mathcal{H}(\Phi_{\text{point}}(\mathbf{s})) \left[\frac{\mathbf{p}_n - \mathbf{c}}{\|\mathbf{p}_n - \mathbf{c}\|_2}, 1 \right] \quad (14)$$

Maximality Term.

$$\mathbf{R}_{\text{maximal}} = r - (r^{t-1} + \epsilon) \quad (15)$$

$$\nabla_s \mathbf{R}_{\text{maximal}} = [0, 1] \quad (16)$$

Pinning Term. Given pin point \mathbf{p} :

$$\mathbf{R}_{\text{pinning}} = \mathcal{R}(\|\mathbf{c} - \mathbf{p}\| - (r + d_{\text{pin}})) \quad (17)$$

$$\nabla_s \mathbf{R}_{\text{pinning}} = \mathcal{H}(\mathbf{R}_{\text{pinning}}) \left[\frac{\mathbf{c} - \mathbf{p}}{\|\mathbf{c} - \mathbf{p}\|_2}, -1 \right] \quad (18)$$

Appendix B: Empirical Parameter Defaults

Default value of ω_1/ω_2 :

$$\omega_1/\omega_2 = 0.007\sigma_{\mathbf{p}} + 0.02 \quad (19)$$

Default value of h_{blend} and h_{support} (both the same):

$$h_{\text{blend}} = 0.74\sigma_{\mathbf{p}} + 0.49 \quad (20)$$

Default value of d_{pin} :

$$d_{\text{pin}} = 0.75\sigma_{\mathbf{p}} \quad (21)$$



Figure 16: A gallery of LSMAT results with varying levels of noise on shapes with complex topology and varying feature size. In the callout for the octopus, notice how the MLS kernel overlaps nearby surfaces, yet the algorithm can cope by producing erroneously located medial spheres with *zero* radius.

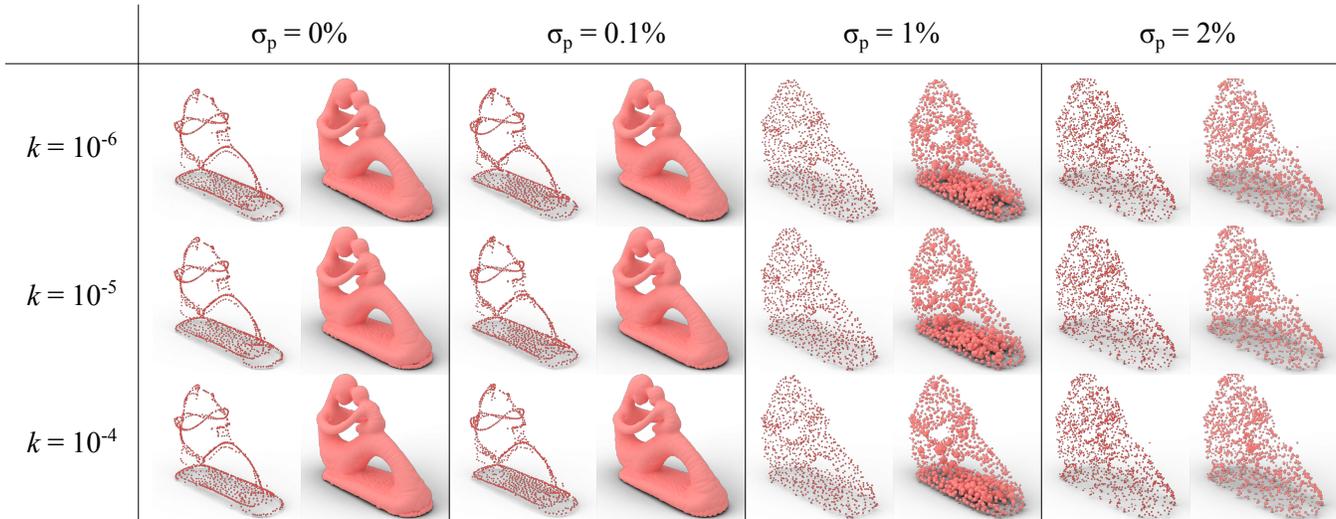


Figure 17: Parameter sweep for QMAT. The X axis represents noise, while the Y axis shows results for different values of the parameter k . We found that k had little effect on the noise tolerance of the algorithm. For the left two columns, the noise values are the minimum and maximum noise levels shown in the original publication, which yield good results. However, for the higher noise values that we test against QMAT fails to produce a useful medial representation. As noted in Section 2.1, this is a known and acknowledged limitation for this family of methods.

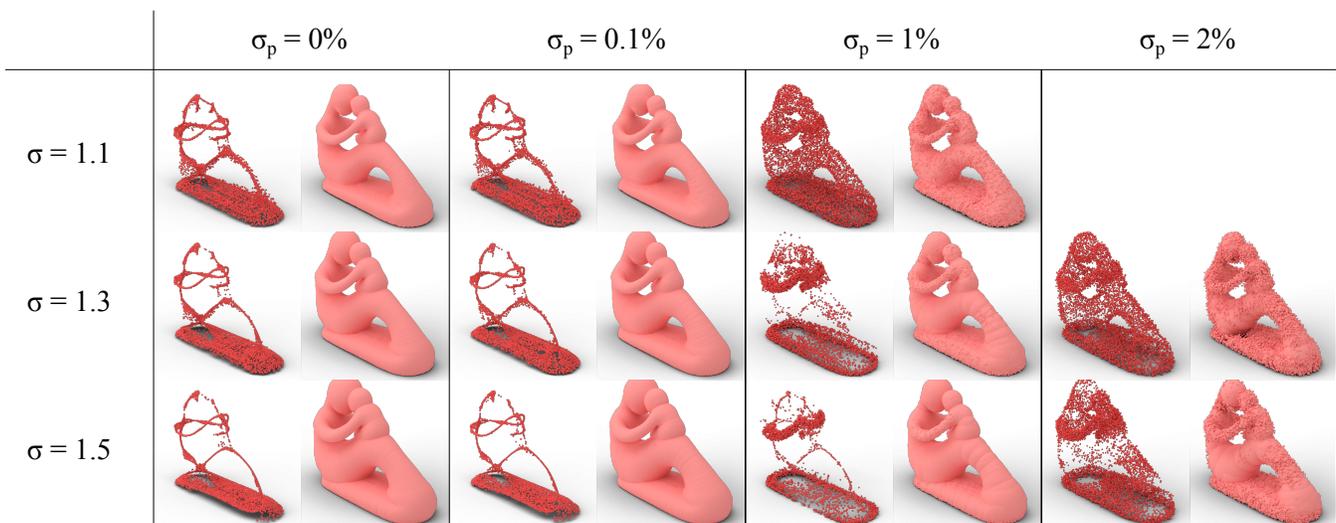


Figure 18: Parameter sweep for Scale Axis. The X axis shows the same noise levels used in Figure 17, while the Y axis shows results for different values of the scale parameter σ . As expected, this method performs well in areas where the local feature size is much larger than the noise. We show the intermediate up-scaled spheres to demonstrate this more clearly. The Scale Axis Transform is in some cases able to recover a useful medial representation even when the unfiltered MA is highly corrupted. The missing images are due to the implementation failing to complete within the allowed time for that combination of parameters.